

UNIVERZITA PALACKÉHO V OLOMOUCI
PEDAGOGICKÁ FAKULTA

Katedra technické a informační výchovy

Diplomová práce

Ing. Jan Mikulčík

**Návrh didaktického textu pro učitele SŠ k problematice
tvorby 3D počítačových her**

Prohlašuji, že jsem diplomovou práci na téma „Návrh didaktického textu pro učitele SŠ k problematice tvorby 3D počítačových her“ vypracoval samostatně a řádně jsem uvedl a citoval všechny použité prameny, které uvádím v seznamu bibliografických citací.

Olomouc 5. 4. 2022

Podpis autora práce:

Děkuji vedoucímu mé práce panu Mgr. Janu Kubrickému, Ph.D., za odborné vedení diplomové práce, poskytnutí konzultací, vstřícný přístup, poskytování rad a materiálových podkladů k práci. Děkuji také své rodině za podporu v průběhu studia.

Obsah

Úvodem	6
1 Didaktické texty.....	8
2 Koncipování metodické příručky	11
2.1 Orientace na výchovně vzdělávací cíle.....	11
2.1.1 Kognitivní cíle	13
2.1.2 Psychomototické cíle.....	14
2.1.3 Afektivní cíle	15
2.1.4 Možnosti výuky orientované na dosahování cílů	16
2.2 Didaktická analýza učiva	17
2.3 Vnitřní a vnější motivace k školní práci	18
2.4 Uplatnění problémového vyučování	23
2.4.1 Role učitele v problémovém vyučování	26
2.4.2 Jak probíhá příprava učitele na problémové vyučování	27
2.4.3 Uspořádání aktivit žákům	28
2.4.4 Role žáka v problémovém vyučování	28
2.4.5 Žádoucí myšlenková posloupnost činností žáka	28
2.5 Učitelova příprava na vyučování	30
2.5.1 Příprava na vyučování podle Slavomíra Rysa.....	32
2.6 Piktogramy užití v metodické příručce	33
3 Tvorba grafického obsahu	36
3.1 Modelování 3D objektů v Autodesk 3ds Max	40
3.1.1 Modifikátor Lathe.....	42
3.1.2 Modifikátor Loft	45
3.1.3 Skládání objektů pomocí ProBoolean	46
3.1.4 Modifikátor Displace.....	48
3.1.5 Modifikátor Edit Poly	50

3.2	Modelování objektů do hry „Fantasy golf“	54
3.2.1	Model míčku.....	54
3.2.2	Model terénu.....	57
3.3	Materiálové inženýrství	65
3.3.1	Export textur z WorldMachine	65
3.3.2	Tvorba textury terénu v Adobe Substance Painter	67
3.3.3	Materiálové sloty a dokončení v 3ds Max.....	69
3.4	Export objektů a materiálu do engine Babylon.js	73
3.5	Učební úkoly a shrnutí třetí kapitoly	75
4	Tvorba hry v Babylon.js	78
4.1	Základní projekt Babylon.js v HTML a JavaScriptu	82
4.2	Implementace herních objektů.....	85
4.2.1	Implementace základních 3D primitiv v Babylon.js	85
4.2.2	Import objektů do Babylon.js	86
4.3	Osvětlení	87
4.3.1	Metody nasvícení scény.....	88
4.3.2	Nasvícení scény, hry v Babylon.js	89
4.4	Kamera.....	90
4.5	Závěrečné poznámky	91
4.5.1	Možnosti uživatelského vstupu	92
4.5.2	Fyzika a kolize hry	92
	Závěr.....	97
	Užitá literatura	99
	Seznam obrázků.....	106
	Seznam programových kódů	107

Úvodem

Hraní počítačových her je dnes běžnou součástí moderní společnosti, napříč generacemi nebo pohlavím. Většina lidí vlastní jednu či více platforem určenou pro tento „jednoduchý“ zdroj pobavení. Nejedyn hráč pod vlivem libivé atmosféry z hraní, virtuální reality, zatouží po vytvoření si hry vlastní. Profesionální tvůrci her z Česka, zaznamenali v roce 2020 obrat z prodaných her v řádu několika miliard korun. O popularitě fenoménu jménem počítačová hra nemůže být sporu. Zde se rodí nápad na zpracování metodické příručky, pro učitele středních škol, která zpracovává atraktivní téma, s nímž se mohou snáze identifikovat žáci technických oborů. Domníváme se, že téma má díky své multimediální a interdisciplinární dimenzi synergetický potenciál, podněcovat vnitřní motivaci žáků a žáky aktivizovat v oblasti počítačové grafiky a programování, často až za hranicemi školního vyučování.

Důvodem výběru předmětného tématu, je také náš dlouhodobý zájem o danou problematiku a několikaletá zkušenost s vývojem výukového softwaru u společnosti TeraSoft a. s., která byla konfrontována vlastní praxí učitele ICT na střední škole. Hledali jsme pro žáky atraktivní téma – „průvodce“, s nímž se mohou snadno ztotožnit a potřebné znalosti z tematických celků algoritmizace a multimediální tvorby si tak snáze osvojit. Žák získá kompetence k programování multimediální aplikace a tvorbě 3D objektů, komplexních scén v počítačové hře. Učiteli text umožní postupovat s žáky problematikou jednotlivých kapitol více koncepčně a blíže postupům profesionálů.

Hlavním cílem diplomové práce tedy je - návrh metodické příručky, věnované tvorbě nenásilné 3D počítačové hře s pracovním názvem Fantasy golf. Práce je logicky členěna na teoretickou a praktickou část.

Teoretická kapitola první, vymezuje pojem didaktický text, jakožto didaktický prostředek materiální povahy a identifikuje postavení navrhované metodické příručky ve struktuře didaktických textů. První kapitola také čerpá z cenného výzkumu badatelky Jany Staré (2021), který identifikuje preferované vlastnosti metodických příruček českými učiteli.

V druhé, rovněž teoretické kapitole, identifikujeme další vlastnosti účinné metodické příručky. Klíčovým je nám třetí typ přípravy učitele na vyučování podle Slavomíra Rysa (1979). Odpovídáme na problémové otázky, kterak je možno kvalitní metodickou příručku na pozadí relevantní pedagogické teorie a naší pedagogické zkušenosti vypracovat. Kapitola ale současně řeší otázku, jak obecně realizovat výuku s dopomocí navrhované metodické příručky. Věnujeme se tak postupně problematice stanovení výchovně vzdělávacích cílů,

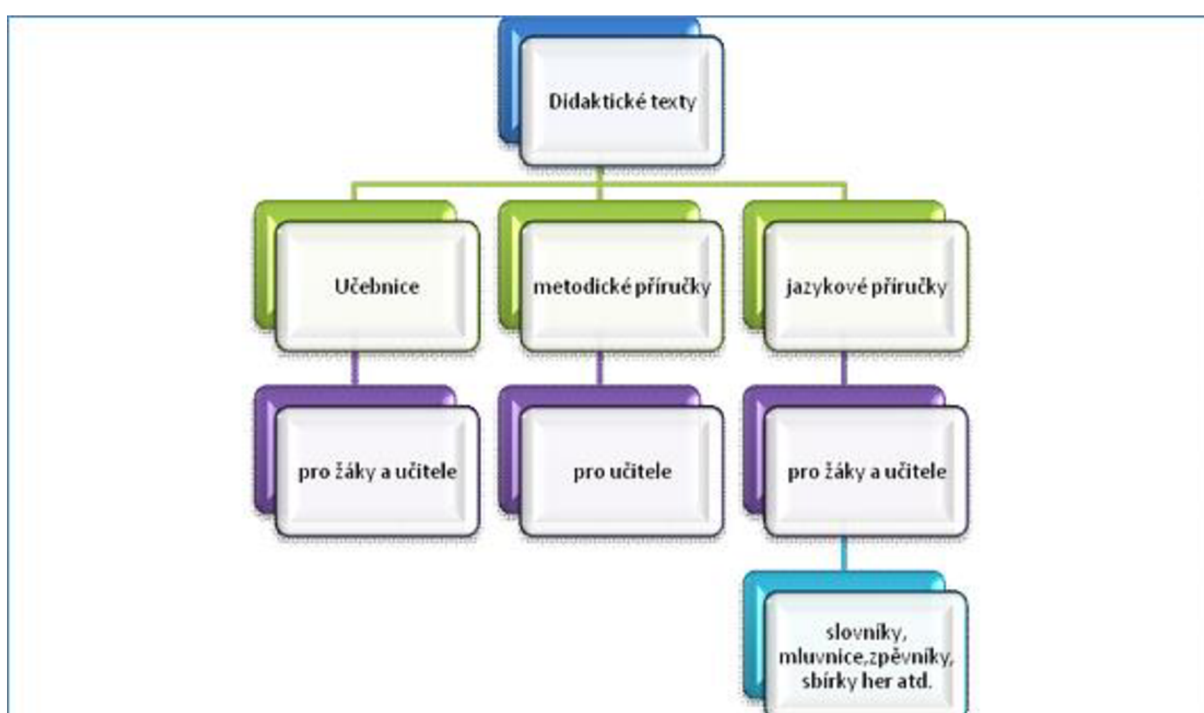
didaktické analýze učiva nebo zásadám správné motivaci žáků pro školní práci. Podrobněji se věnujeme problémové vyučovací metodě, která zaujímá ve výuce technických předmětů výjimečné postavení a nejinak je tomu v kontextu s představovanou metodickou příručkou.

Kapitola třetí a čtvrtá je samotnou metodickou příručkou. Tyto kapitoly jsou rozděleny na teoretickou část, která odpovídá na výzkumnou otázku, jak konkrétně a účinně vytvářet interaktivní 3D obsah s žáky a co nejlíže profesionálnímu pracovnímu postupu. Praktickým výstupem těchto kapitol jsou metodické, procesní pokyny k řízení výuky, řešené učební úlohy s rostoucí náročností a konečně spustitelná počítačová hra Fantasy golf, včetně komentovaného zdrojového kódu. Přiložené zdrojové kódy a další soubory, jsou důležitou součástí naší metodické příručky.

1 Didaktické texty

„Didaktický text je jakýkoli text, který je zkonstruován tak, aby byl nosičem didaktické informace.“ (Průcha, Walterová, Mareš, 2003).

Školní didaktické texty, řadíme mezi didaktické prostředky materiální povahy. Jsou to materiály s verbalizovaným působením, jež byly vytvořeny za účelem podpory výchovně vzdělávacího procesu našich žáků. Jejich podstatou je jistě obsahová hodnota, ovšem také samotný proces, tedy jak si tento zamýšlený soubor vědomostí, dovedností, schopností, postojů a hodnot žáci osvojí. Základní dělení didaktických textů, jakožto textových materiálů, ilustruje obrázek č. 1, jenž didaktické texty vhodně kategorizuje.



Obrázek 1: *Didaktické texty* (Převzato z: Stejskalová 2009)

Výčtem níže, však uvedeme také klasifikaci didaktických textů dle Průchy (2003), pro jeho detailnější pojetí.

- učebnice,
- cvičebnice (pracovní sešity a listy),
- slabikáře,
- čítanky,
- sborníky,
- didaktické příručky (přehledy učiva),
- sbírky (jazykových her, matematických úloh, diktátů aj.),

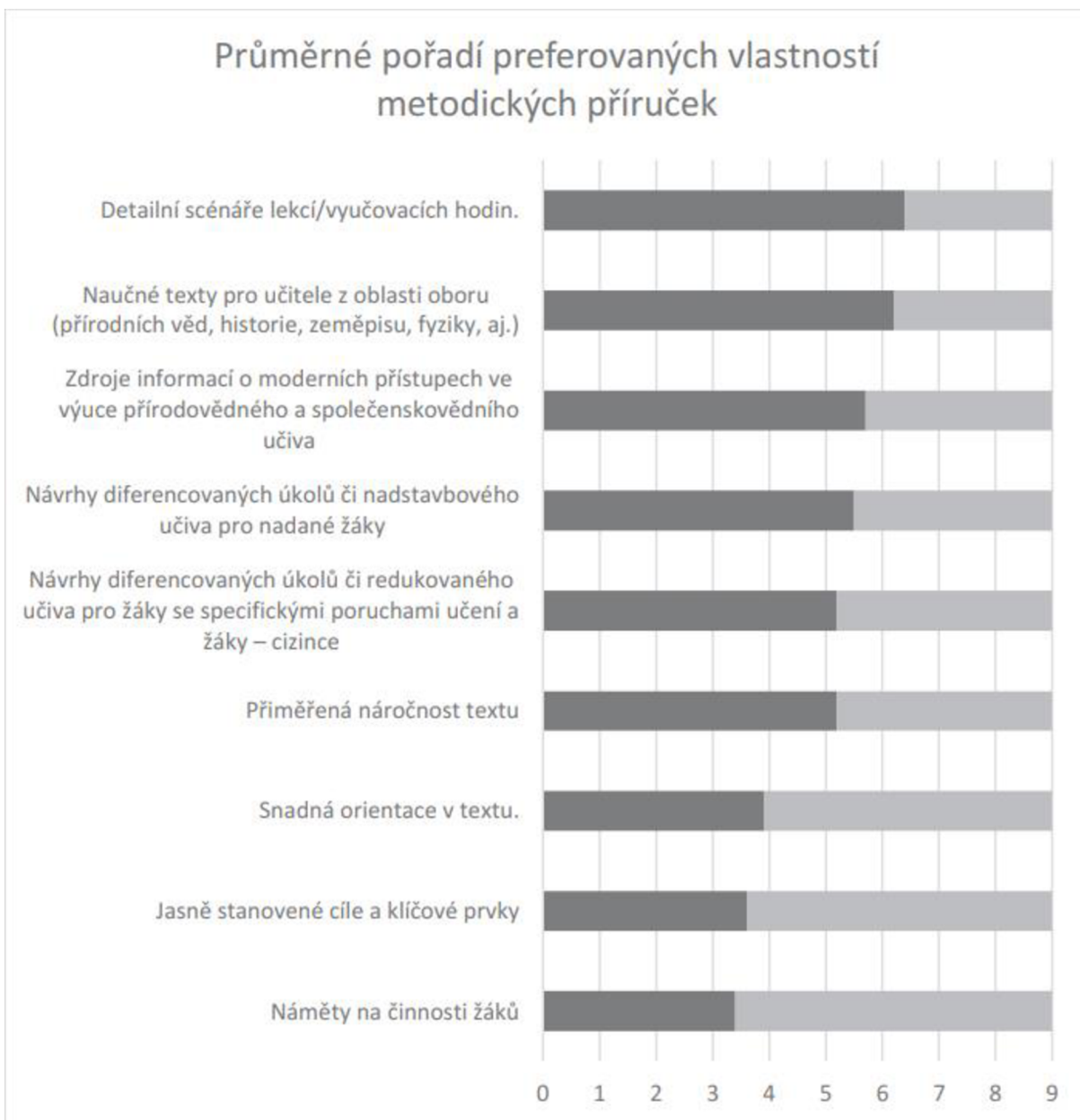
- *stručnou mluvnici českou,*
- *slovníky (terminologické, výkladové, překladové),*
- *zpěvníky,*
- *atlasy a mapy (geografické, demografické, historické aj.),*
- *odborné tabulky (matematické, chemické aj.),*
- *testy a testové manuály.*

Zatímco učebnice je určena pro žáky a učitele, kteří mnohdy pracují s učebnicí, jakožto s výhradním zdrojem přípravy na výuku, metodická příručka, která je představena ve třetí a čtvrté kapitole, tedy praktické části této práce, je materiálem určeným k rukám pedagogů, rovněž k usnadnění přípravy na výuku takových témat, která jsou z nějakého důvodu obtížnější uchopit po metodické stránce. Obsahují obvykle doporučení výukových postupů, metod a technik.

Dobrý důvod, proč by se měl učitel vyvarovat výhradní práci s učebnicí, nabízí také badatel Robert Čapek. Podle Čapka (2019) nejsou učebnice většiny českých nakladatelů kvalitní, respektive neobsahují dostatečnou podporu pro pestrou výuku. Jsou plné poznatků na nejnižší myšlenkové úrovni.

Podle Jany Staré (2021) si metodické příručky zaslouží větší pozornost. Tyto totiž doposud stojí na okraji zájmu českých výzkumníků (a bohužel také učitelů). Díky této skutečnosti v podstatě neexistuje ucelená teorie k jejich tvorbě. Totéž plyne z námi provedené rešerše. Této skutečnosti si tak všímá nezanedbatelný počet autorů klasifikačních prací, ale také oponentních posudků těchto klasifikačních prací, jež jsou zaměřeny právě na tvorbu metodických příruček. Tyto posudky připomínají nouzi teoretické báze k jejich tvorbě, ale také malý počet autoritativních příkladů, kterak vlastní strukturou metodickou příručku vůbec koncipovat. Badatelka však publikuje hodnotné výsledky vlastního výzkumu, které odpovídají na otázku, jaké vlastnosti metodických příruček učitelé preferují a jsou tedy pro učitele důležité.

Výsledky výzkumu ilustruje obrázek č. 2. Je patrné, že učitelé tedy v metodických příručkách nejvíce preferují detailní scénáře vyučovacích hodin, následují kvalitní naučné texty z předmětného oboru, zdroje informací o moderních přístupech ve výuce nebo návrhy diferencovaných učebních úloh. Prezentovaný výzkum badatelky Staré využijeme v praktické části této práce.



Obrázek 2: *Preferované vlastnosti metodických příruček (Převzato z: Jana Stará 2021)*

2 Koncipování metodické příručky

V následujících podkapitolách, formulujeme teoretické báze ze školní a oborové didaktiky ICT a technických předmětů, které nám umožnily metodickou příručku pro učitele, krok za krokem smysluplně vypracovat. Stejnou měrou jsme zde vyvinuli maximální úsilí, uplatnit naši dvanáctiletou praxi učitele ICT na SŠ a praxi vývojáře výukového softwaru. Postupně budeme identifikovat typy metodických pokynů, které v metodické příručce, tedy ve třetí a čtvrté kapitole systematicky uplatňujeme tak, abychom maximalizovali účinek edukativního působení učitele, počínaje fází plánování expozice nového učiva a konče ověřováním nabytých vědomostí a dovedností, schopností, postojů a hodnot žáků.

2.1 Orientace na výchovně vzdělávací cíle

Význam tvorby cílů ve vzdělávání, byl publikován již v první polovině minulého století, pedagogem, filozofem a expertem na evaluaci Ralphem Tylerem (1949).

Pro Tylera (1949) by měl správně definovaný cíl, obsahovat zcela jasný odkaz na konkrétní typy výstupních znalostí a dovedností, ovšem také odkazy na projevující se doprovodné chování, jenž o nabyté znalosti svědčí. Před Tylerem tomu tak obvykle nebylo. Učivo bylo obecněji vymezeno oborovou oblastí nebo tématem. Mělo širší tematické oblasti, například „3D modelování“ nebo „druhá světová válka“. Tylerovo dílo tak jasně ukázalo, že učitelé, musejí definovat konkrétní cíle, aby byl výchovně vzdělávací proces více efektivním.

Kendall (2007) dělí cíle do trojice kategorií. Globální cíle jsou nejobecnější a široké. Neodrážejí pouze aktuální stav poznání vědecké disciplíny, vyučovacího předmětu, ale také společenské požadavky výchovné. Instrukční cíle jsou nejspecifičtější ze tří typů. Výborně je rozpracoval badatel Robert Mager a věnujeme se jim blíže v následující podkapitole. Uprostřed diskutované „triády“ jsou vzdělávací cíle. Formulují konkrétní oblasti lidského poznání, neidentifikují však výkonnostní podmínky nebo kritéria úspěchu.

Reiser (2007) vysvětluje, že důvodem koncipování jasných cílů, jsou aktivity realizované ve školní třídě beze směru a dobrá výuka začíná stanovením relevantních cílů.

Dnes je učitel veden k cílování výuky skrze závazné kurikulárními dokumenty. Uvažuje nad cíli, tematickými celky ve třech základních dimenzích – kognitivní, afektivní a psychomotorické. Výukovým trojdimenziálním cílem tedy rozumíme představu o pozitivních proměnách stavu poznání žáka, jenž má být dosaženo ve stanoveném čase v procesu výuky.

Stanovené cíle musí ve všech třech dimenzích tvořit konzistentní strukturu. Nižší a obvykle snadnější cíle, například na pamětní reprodukci, jsou podřízeny cílům vyšším, zaměřeným na tvůrčí vývoj multimediální aplikace. Vyšší cíle jsou ovšem dosahovány v závislosti na dosažených cílech nižších. Z hlediska pedagogické praxe je účelným si uvědomit, že zatímco fixní cíle vyšší – programové, projektové, globální.. - deklarované například v RVP a ŠVP můžeme modifikovat obtížněji, potom k precizování účinnosti v běžné praxi pedagoga, dochází v hledání a stanovování cílů nižších, které součinně s cíli fixními z RVP, vytvoří účinnou pyramidu cílů výchovně vzdělávacího procesu. Správná formulace požadovaných cílových výstupů, je charakteristická také možností výsledky žáků analyzovat, činnost žáka zhodnotit, včetně podmínek jeho výkonu. Klasifikace cílů, co do jejich obtížnosti dosažení, a tedy vhodnosti jejich nasazení v tom kterém okamžiku výchovně vzdělávacího procesu, usnadňují jednotlivé taxonomie výukových cílů¹, pro dimenzi kognitivní ale také psychomotorickou a afektivní.

Řada badatelů se zamýšlela nad definováním dalších „kvalit“, které by cíle, necht' už jakkoli obecné nebo konkrétní musely mít, aby byly co nejvíce platné v reálném výchovně vzdělávacím procesu. Pro naše účely považujeme za příléhavé, stručně níže ozřejmit, mnemotechnickou pomůcku zvanou „SMART“, badatele George Dorana (1981). Tento učitele také nabádá, aby cíle a s tím spojené požadavky kladené na žáky, obsahovaly „pozitivní obrazy“. V takovém případě existuje větší šance, že se stanovené cíle promění v realitu u většího počtu našich žáků.

S (specific) – cíle musejí být jasně formulované a konkrétní, rovněž ale stručné, bez možnosti jejich dezinterpretace.

M (measurable) – dosažení stanoveného cíle by mělo být reálně měřitelné.

A (achievable) – cíl musí být pro žáka dosažitelným, a to jak absolutně, tak relativně s ohledem na další povinnosti studenta a s respektem k jeho volnému času.

R (realistic) – cíl musí být také proveditelný. Neměl by být tak snadná, aby jeho dosažení nebylo výzvou. Neměl by ale být tak náročný, aby žákům dosažení cíle způsobilo frustraci a pocit selhání.

¹ Taxonomií výukových cílů v pedagogice rozumíme uspořádání výukových cílů podle jejich vzrůstající náročnosti, kladené na duševní procesy žáka.

T (timed) – je také důležité, aby si žák sám nebo po poradě s pracovní skupinou, případně učitelem, vypracoval časový plán, pokud možno také graficky a strukturovaně. Vyvaruje se tak snáze selhání, při postupném plnění učebních úkolů k cíli vedoucím.

Dle Sheldricka (2013) je výhodné, doplnit mnemotechnickou pomůcku „SMART“ a pracovat s dalšími dvěma kroky. Vzniká tak aktualizovaná pomůcka – „SMARTER“.

E (evaluate) – je důležité, naučit žáka hodnotit svůj pokrok v procesu dosahování stanoveného cíle. Jestliže byly nastavené cíle podhodnoceny nebo naopak nadhodnoceny, je potřeba provést jejich korekci (tím se nově stanou realistickými, dosažitelnými = přiměřenými).

R (re-evaluate) – žáci i učitelé si musejí uvědomit, že cíle by neměly být statické. Mohou se změnit a někdy také musí. V našem životě, jak osobním, profesním nebo specificky školním, je tolik faktorů, které ovlivňují naše krátkodobé i dlouhodobé cíle, že je musíme stále přehodnocovat tak, aby byly aktuální a nebyly zdrojem pocitů selhání. Takové pocity nepřispívají dobré práci našich žáků.

Všechny tři výše představené dimenze cílů by měly být učitelem v učivu včas rozpoznány a následně důmyslně plánovány a implementovány do výuky. Je zapotřebí si uvědomit, že posláním základní nebo střední školy je nejenom vychovat zručnou kadeřnici nebo elektrikáře, ale také povšechně vyzrálou a na osobní i pracovní život připravenou harmonickou osobnost.

2.1.1 Kognitivní cíle

Nejslavnější taxonomií kognitivních cílů je jednoznačně taxonomie Benjamina Blooma, formulovaná v roce 1956 v USA. Z této taxonomie primárně vychází také ČŠI. Jednotlivé kategorie jsou vzestupně řazeny podle náročnosti kognitivních procesů, kterými musí žák disponovat, aby cíl výuky zvládl. Kategoriemi taxonomie tedy jsou: zapamatování – porozumění – aplikace – analýza – hodnocení – tvoření.

Skrze zapamatování informací, pojmů či definic a jejich reprodukci, se může rodit porozumění, evidentní v schopnosti generalizovat. Žák dokáže znalosti vysvětlit, interpretovat, vyjádřit vlastními slovy a podobně. Aplikací rozumíme kompetenci žáka použít znalosti v praktickém životě. Této dovednostní kategorie by měl v rámci využití naší metodické příručky, dosáhnout každý z našich žáků. Hlubší znalosti a pochopení objektu je charakterizováno kategorií analýza. Žák dokáže objekt výuky fragmentovat na logické dílčí části. Takto prokazuje porozumění vztahům v tomto objektu. Žák dokáže odlišit fakta

od hypotéz nebo odlišit důležité od méně důležitého. Kategorii hodnotit žák prokazuje schopnost vyhledávat logické celky z různých relevantních zdrojů v jeden smysluplný celek a vytvořit tak novou kvalitu s novou strukturou nebo pojmovým aparátem. Bloomova taxonomie je tedy pro pedagogickou praxi mimořádně přínosná.

Pasch (1998) i přes řečené upozorňuje na tenkou hranici mezi jednotlivými kategoriemi a ty nejsou vždy zcela jednoznačné. Jestliže nám Bloomova taxonomie usnadní formulovat cíle podle náročnosti kladené na zapojení kognitivních funkcí, potom Magerova teorie formulování cílů nám umožní zamyslet se nad cílem tak, aby jeho vznikající znění bylo jednoznačně vyjádřeno, a to jak obsahem, tak očekávaným průběhem jeho plnění.

Podle Roberta Magera (1962) nejprve specifikujeme požadovaný výkon žáka, tedy to, co bude žák po našem pedagogickém působení schopen dokázat. Tento cíl někdy popisuje produkt nebo výslede nějaké činnosti. Dále Mager zmiňuje podmínky výkonu, za nichž má dojít ku splnění stanoveného cíle (jestliže existují). Cíl můžeme dále precizovat, jestliže je to možné a vhodné, normováním nebo kritériem výkonu, tedy například počtem správných úloh žáka, tolerancí přesnosti nebo časovým limitem vymezeným pro splnění cíle.

2.1.2 Psychomototické cíle

Dle Blahutkové (2005), jsou psychomotorické učební cíle, odlišné od cílů s převahou kognice v tom, že jejich realizaci evidujeme méně „poražených“ žáků. Tyto cíle umožňují projevit individualitu, tvořivost, fantazii v celkovém projevu žáka.

Psychomotorické cílování podle našeho názoru nejlépe přiblížila taxonomie H. Davea, která vznikla v roce 1968. Dave rozlišuje pět kategorií (imitace, manipulace, zpřesňování, koordinace, automatizace), ovšem také subkategorie. Pro účel naší metodické příručky si vystačíme s hlavními kategoriemi. Žák bude nejprve podle ukázky učitele či instruktáže napodobovat manipulaci s jednotlivými elementy polygonálního modelu a zpřesňovat pohyby myši nebo Space mouse a dalších zařízení. Postup v praktických cvičení, přináší stále zlepšování účinnosti v kvalitě i rychlosti provedení pohybu žákem. Žák pracuje více samostatně. Výrazným mezníkem v naší činnosti je kategorie zpřesňování. Žák je schopen realizovat koordinovaný sled různých činností, které tvoří sekvenci. Maximum výkonu při minimální námaze k cíli vedoucí, je charakteristická pro poslední kategorii automatizace. Žák dokáže rychle a správně realizovat zadaný úkol. Profesionální uplatnění v oblasti 3D modelování žákem, předpokládá dosažení schopností v intencích čtvrté nebo páté kategorie.

Psychomotorické cíle v naší metodické příručce, mohou být spojeny s potřebou náčrtků, sloužících jako předloha nebo šablona pro následné 3D modelování. Mimořádně účelným je fotografování textur² žákem přímo v terénu a podobně.

2.1.3 Afektivní cíle

Podle Nakonečného (2020) - právě vztah k hodnotám, tvoří obsah postojů. Tyto vztahy - afektivní cíle u žáků učitel buduje postupným zvnitřňováním hodnot interiorizací. Jako užitečnou a nestárnoucí studnicí inspirace, se prokázala taxonomie o pěti kategoriích. Jejím autorem je badatel Kratwohl a jeho spolupracovníků (1965). Jedná se o kategorie přijímání, reagování, oceňování hodnoty, integrování hodnot a internalizace hodnot žáka v charakteru. Žák by měl nejdříve ochotně přijímat, naslouchat a nevyhýbat se podnětům, které mohou postojovou orientaci ovlivnit. Reagování je charakteristické větší mírou zainteresovanosti k předkládanému podnětu. Podněty by měly mít citový náboj, který urychlí možný vstup do další kategorie. V třetí kategorii žák již oceňuje předkládané podněty jako hodnoty žádoucí a užitečné pro něj nebo fungování společnosti. Čtvrtou kategorií žák organizuje dříve akceptované hodnoty do soustav hodnot, které tvoří vzájemnou relativní harmonii. S nově integrovanou hodnotou, mohou být zdánlivě nesouvisějící životní situace, řešeny žákem pod vlivem této nové hodnoty. Poslední název kategorie „internalizace hodnot v charakteru“ vystihuje plně její obsah. Budování zdravých postojů našich žáků, nelze odtrhnout od faktu, že musejí být v souladu s uspokojováním jejich potřeb. Brilantně vystihuje řečené následující citace.

„Hodnoty jsou vlastnosti, které člověk přisuzuje jevům, objektům, činnostem souvisejícím s uspokojováním jeho potřeb. Odráží se v nich to, co má pro něj význam, čeho si váží, co považuje za žádoucí. Hodnoty významně ovlivňují chování člověka. Vytvářejí vnitřní provázaný systém – tzv. hodnotovou orientaci, v níž bývá zpravidla jedna hodnota ústřední.“
(Pauknerová, 2006).

Podle Obsta, Kalhouse a kol. (2009), pro dosažení afektivních cílů je vhodné při výuce vyvolat polemiku o určitém problému, který se týká probíraného učiva. V naší metodické příručce bychom si měli všimnout především dodržování autorského zákona, respektování licenčních podmínek tvůrců software, případně využívání svobodného software. Postoje k životnímu prostředí můžeme mobilizovat vysláním žáků do míst, kde mohou pořídit kvalitní

² Textura je rastrový nebo procedurálně generovaný grafický zdroj, tvořící součást výsledného vzhledu objektu (jeho materiálu).

textury. Z hlediska ekologie se však může jednat o místa rozporuplná. Pro žáky však tyto lokace musí být stále zcela bezpečné. Podobně poslouží společná exkurze. Učitel by měl znát dobře ŠVP (ale také svůj obor), aby afektivní cíle dokázal přesněji určit a plánovat.

Problém s afektivními cíli, spočívá podle našeho názoru v obtížnosti ověření jejich interiorizace žákem. Žák může reálný obraz vlastních hodnot skrývat. Z naší praxe se ukázalo alespoň částečně užitečné, jestliže učitel realizuje pro své žáky současně zájmové kroužky nebo doučování. Vládne v nich obvykle uvolněnější atmosféra - pod dojmem, že nejsou žáci klasifikováni. Z takových projevů žáka lze snáze postoje analyzovat a vlastní pedagogickou úspěšnost vyhodnotit. Podobně poslouží konzultace s kolegou, který tyto kroužky administruje.

2.1.4 Možnosti výuky orientované na dosahování cílů

V rámci pozorování vyučovací jednotky se může zdát, že mnoho prvků třídy, organizovaných podle systému výukových cílů a z toho odvozených, již konkrétních učebních cílů, je svou povahou docela tradiční. Učitel například představí své zkušenosti s učivem, motivuje k výkonu nebo explanuje učivo, které studentům pomůže získat odborné kompetence. Učitel také poskytne žákům příležitosti k ověření dříve probraného učiva, procvičení dovedností a postupů, které byly řešeny dříve. Stručně řečeno, pozorovatel třídy, která je organizována kolem systému cílů učení, může vidět mnoho věcí, které vypadají stejně jako třída, která není organizována kolem takového systému. Jestliže by však tento pozorovatel zůstal ve třídě delší dobu, spatřil by věci, které se ve třídách bez důsledného akcentu na dosahování cílů obvykle nenacházejí. Žáci získávají především více kompetencí a v jasně specifikované výstupní kvalitě.

Jednou z charakteristických vlastností, možností třídy organizované podle učebních cílů je, že se žáci zlepšují v určité kompetenci napříč vyučovací jednotkou a během roku, pro každý vzdělávací cíl. To je dost dobře možné, neboť učitel poskytl stanovením cílů jejich jasný výčet a měřítko pro každý vzdělávací cíl, který je vhodný ku sledování rozvoje kompetencí.

Ve třídě organizované podle učebních cílů, představuje dosažení určité úrovně, například v hierarchii Bloomovy taxonomie, stav kompetencí žáka pouze v jednom časovém okamžiku. Pokud žák později prokáže zvýšenou kompetenci, jeho skóre se změní, aby představovalo jeho nový status. Žáci do určité míry přebírají odpovědnost za prokázání požadovaných kompetencí.

Marzano (2009) uvádí zase jiný netradiční aspekt třídy uspořádané podle učebních cílů. Žáci si mohou navrhnout vlastní způsoby, kterak prokázat pokroky ve studijních cílech. V realitě škol v Česku by učitel na začátku školního roku patrně předložil žákům učební úlohy příslušné náročnosti k výběru. V případě cílů kognitivních by bylo patrně využito tvorby učebních úkolů podle taxonomie badatelky Dany Tollingerové. Nyní příkladem. Žák Petr může mít zvládnuty učební úlohy úrovně č. 3. Dokáže konkrétně použít modifikátor Edit Poly a provádět jednoduché transformace segmentů jejich změnou polohy v prostoru. Jestliže je ale cílem žáka nebo učitele získání úrovně č. 5, potom musí žák kombinovat plnou nabídku desítek příkazů tohoto modifikátoru a příslušných parametrů, aby byl schopen je syntetizovat v jeden smysluplný celek – objekt. Po prozkoumání škály úkolů žákem, může náš žák navrhnout učiteli prokázání kompetencí, například modelováním hradní věže. Zvládnutí tohoto učebního úkolu předpokládá osvojení jednotlivých příkazů, vhodně kombinovaných tak, že tvoří smysluplný celek. Další žák může zvolit zcela jiné učební úkoly. Ty však musí respektovat hranice vymezené výukovým cílem a příslušným stupněm náročnosti zvolené taxonomie.

Podle našeho názoru žák, který si alespoň částečně volí vlastní cestu k realizaci výukových cílů, více či méně vědomě přebírá také větší odpovědnost za jeho zdárnou realizaci. Respektujeme a podporujeme více jeho zájmy a potřeby.

System klasifikace učebních úloh doporučujeme žákům objasnit vždy. Pochopení systému volby učebních úloh nedělá obvykle žákům střední školy zásadní potíže. Některým však může dělat potíže dostatečně vymežit pojmy aplikuj, analyzuj a podobně.

2.2 Didaktická analýza učiva

Skalková (2007), přibližuje didaktickou analýzu učiva, jako hlubokou myšlenkovou činnost, která umožňuje učitelům proniknout do probíraného učiva a pedagogicky ho posoudit. Učitel provádí rozbor relevantních činností, zvažuje zejména jejich užitečnost a přiměřenost vzhledem k stanoveným cílům výuky. V rámci výuky informatiky a techniky, je občas díky častým proměnám v těchto oborech, potřeba realizovat celou cestu didaktické transformace, protože vhodné texty nejsou dosud k dispozici, nemají celistvou podobu a podobně. Ve většině případů však zpracování obsahu učiva, probíhá v rámci pojmové a operační analýzy. Dochází k vytváření logické struktury učiva: základního – prohlubujícího – rozšiřujícího. Zvoleny jsou relevantní pojmy – vztahy - generalizace. Následuje zvážení logiky, kterak dojde k osvojení učiva. Aktualizujeme – aktivizujeme dříve osvojené učivo, provádíme osvojení nových poznatků, které upevňujeme, prohlubujeme, vytváříme potřebné znalosti dovednosti, postoje

a kompetence. Kontrolujeme stanovené cíle, kognitivní, afektivní či psychomotorické, zda bude moci být cílů v požadovaném rozsahu dosaženo. V této fázi je inspirujícím konceptem „FOCUS“ badatele Pettyho.

Podle Pettyho (2013), je přínosným zohlednit v učivu dostatek fantazie, příležitost k ocenění žáků, orientace na cíle. Učivem a zvolenými příklady zvolit adekvátní náročnost, která umožní prožití úspěchu co největšímu počtu žáků. Konečně osnování učiva musí dávat celkový smysl, zejména pro praxi, osobní život nebo navazující studia. Učení volenému učivu, musí dávat smysl učitelům i žákům.

Samotné zpracování učiva lze provést podle teoretické báze badatele J. S. Brunera. Bruner (1967) kladl důraz na osvojení logických struktur poznatků. Každé vyučovací téma, má svoji strukturu, která se skládá z pojmů, generalizací a faktů. Podstatné je hluboké porozumění generalizacím. Badatel připodobnil svoji představu analogií se stromem. Kmen a větve stromu jsou generalizace a hlavní pojmy, podle nichž je téma uspořádáno. Listy jsou nespočetem faktů. Žáci musí porozumět struktuře učiva, aby se učivo stalo pochopitelným a poznání trvalým vlastnictvím žáků. Operační analýza konkrétních kompetencí žáka, vyrůstá z pochopení vztahů pojmové analýzy. Nezbytným je také promyšlení mezipředmětových vazeb a s tím spojených konsekvencí. Na Brunerovo strukturování učiva navázalo množství následovatelů. Badatel Vávra, je jedním z nich a rovněž zdůrazňoval význam generalizace.

„Fakta jako listy stromu se během ročního cyklu nejvíce obměňují, v menší míře se obměňují větve (pojmy) a relativně stálým, a navíc posilujícím prvkem je kmen (generalizace).“ (Vávra, 2006, str. 24).

Praxí se osvědčilo strukturování učiva pomocí pojmových a mentálních map. Mentální mapy se dle Buzana (2007) dají přirovnat k plánu města. Střed města symbolizuje nejdůležitější představu učiva. Hlavní třídy vedoucí do města symbolizují hlavní myšlenky našeho myšlenkového procesu. Vedlejší ulice mohou být našimi vedlejšími myšlenkami (nebo rozšiřujícím učivem).

2.3 Vnitřní a vnější motivace k školní práci

„Zkušenosti i začínající učitelé pokládají motivaci za předpoklad úspěšného učení a pro mnoho z nich je největším úkolem přimět své žáky k tomu, aby se učili. Jestliže se žáci učit nechťejí, může jejich učení být natolik neefektivní, že se případně nenaučí vůbec nic.“ (G. Petty, 2013, s. 40).

Titulní citace, je výzvou celé kapitoly věnované motivaci. Budeme se v ní snažit najít odpověď na otázku, jak motivovat žáky k učení, cestami přijatelnými pro žáky i učitele a samozřejmě závazné kurikulární dokumenty.

Podle Sheldricka (2013), motivace zahrnuje tři vzájemně interagující komponenty – iniciaci ku kýženému chování, nastavení správného směru, postupu k dosažení cíle a úroveň intenzity úsilí, kterou je ochoten jednatel vložít pro dosažení stanoveného cíle. Pochopení první komponenty, tedy iniciace je zásadní. Zdroje iniciace k činnosti obvykle členíme na vnitřní a vnější.

Vnější motivací rozumíme tu, která přichází zvenčí. Typicky když žák vykoná to, co od něj očekává učitel, rodiče a žák získá pěknou známku, odměnou může být privilegium sedět s kamarádem, získá samolepky, pamlsky a podobně. Tyto vnější faktory způsobí (pokud si jich žák váží), že bude motivován k dokončení úkolu. Výsledné uspokojení žáka však není z práce na úkolu nebo výsledku snažení, ale spíše z odměny, kterou dostane.

Podle MacConvilleho (2012), je na vnější motivaci příjemné, že takto lze přimět žáka, aby udělal jinak nepříjemný úkol.

Naopak Kohn (2018), v jeho příznačně nazvaném díle „Punished by Rewards“ před vnější motivací varuje. Když totiž žáci pracují „pro odměny“, učí se méně rozvíjet vlastní motivaci, která se dlouhodobě externalizuje. Přirozená dispozice žáků učit se, jako v dětství se deformuje směrem od učení se k získání odměny. Čím déle to trvá, tím více naši žáci ztrácejí radost z učení a smysl pro překonávání výzev.

Vnitřní motivace je zcela odlišná. Je to vnitřní síla – snaha něco udělat, protože žák prožije potěšení nebo pocit úspěchu z určité činnosti. Žák, který sám sebe vyzve dokončit co nejvíce otázek z pracovního listu, protože chce porazit své osobní maximum to udělá jistě s větší radostí než žák, který to dělá pro samolepku. Vnitřně motivovaný žák se může samolepce jistě také těšit, ale splnění výzvy, kterou si stanovil, je pro něj totiž „obohacující“. Motivovaný žák snadno udrží pracovní koncentraci a odvede na něm vynikající práci, protože samotný úkol si to takto zasluhuje.

Sheldrick (2013) uvádí v našem kontextu zajímavý příběh s mladšími dětmi. Když děti poprvé přijdou do mateřské školy, většina pracuje ze své vnitřní motivační základny, která se jim osvědčila po většinu jejich života. Naučili se samostatně mluvit, chodit a řešit mnoho

dalších problémů. Tyto pozitivní dispozice k učení vzešlé z reálného světa by měly být ve školním prostředí zachovány.

Co deaktivuje žákův přirozený vnitřní zápal? Jmenujeme příkladem špatné klima ve školní třídě, nepraktičnost učiva pro žákovy zájmy a budoucí profesi, naše nepodporující reakce na práci žáka, podcenění šikany ve školní třídě a bohužel mnoho dalšího. Můžeme si klást otázky (a někdy to dělají také žáci – „odvážlivci“), je pro naše žáky toto učivo důležité? A je vůbec učitel nadšen učivem, které je prý tolik důležité? Je uvěřitelným?

Šumavská (2012) si všímá podobného. Vyzdvihuje možnost žáků využít své poznatky v oblastech jejich zájmu a mít prostor své nápady rozvíjet. Mnohdy by stačilo, kdyby byly žákům dány příklady dalšího využití učiva v praxi, díky nimž by jim poznatky nepřipadaly tolik zbytečné. Tataž autorka uvádí, že účinnou motivační výukou je ta, která zároveň žáky podporuje k seberealizaci a odpovědnosti, například vypracování samostatné práce a její prezentování před třídou.

Mají tedy naši žáci dostatečnou míru autonomie, při řešení učebních úloh? Tuto otázku považujeme za důležitou pro správnou motivaci žáků, bohužel je učiteli opomíjená.

Podle Kohna (2018) platí, že čím více autonomie budeme k práci na učební úloze či projektu žákům poskytovat, tím produktivnější žáci budou, stejně tak budou mít větší potěšení z práce. Potěšení z práce má tedy dva podstatné rozměry. Dodržujeme důležitý humanistický ráz výchovně vzdělávacího procesu a současně vytváříme podhoubí pro rozvoj zájmů žáka a jeho učení ve volném čase. V dalším textu o motivaci, uvedeme stručně myšlenky, jak zajistit takové učební podmínky, které budou podněcovat vnitřní motivaci žáka, která mu posléze umožní povšechně uspět.

Proč nejsou žáci vnitřně motivováni? Možnou odpověď by nám mohla poskytnout odpověď na otázku – kdy vlastně pro děti končí zábava z učení a začíná dřina z učení? Závěrečné zkoušky a známkování vůbec? Pro někoho jsou zkoušky traumatickým zážitkem, pro někoho méně. Žák, který však ve škole stále selhává, napříč předměty ve vzdělávacím systému, nebude jistě do pokračování ve studiu motivovaný. S dostatečně negativní zpětnou vazbou to nakonec vzdá každý. Tito žáci se najdou snad v každé třídě. Pro mnoho takových žáků je povinná školní docházka obdobím stráveným ve „vězení“. Mohli bychom se více zamýšlet nad tím, co je na školní docházce špatně. Místo toho však bude lépe, zaměřit na to, co můžeme jako učitelé udělat, abychom zlepšili zkušenosti žáků v našich třídách.

„Bud' změnou, kterou chceš ve světě vidět.“ (Ghandi)

Podle Sheldricka (2009) je důvodem snížené vnitřní motivace žáků zpětná vazba o své školní práci. Jestliže budou žáci dostávat po mnoho let pouze negativní zpětnou vazbu o své školní práci, začnou tomu věřit a stane se skutečně jejich „standardem“ vždy. Přestože je stanovení a dosahování cílů ve škole i osobním životě velmi důležité, učitel by se měl snažit v práci svých žáků nalézat také pozitivní „obrazy“. Tentýž autor zmiňuje podle našeho názoru další důležitý aspekt vnitřní motivace – rozvoj sebekoncepce žáka a jeho sebeúčinnosti. Nízká úroveň víry ve vlastní účinnost v jednom nebo více předmětech, způsobená několikerými selháními, může být problematická jak pro žáka, tak učitele. Především pak tehdy, když mají tito žáci entitní pohled na inteligenci – ta je podle nich fixní a oni nad ní nemají žádnou kontrolu. Takové žáky bychom však měli povzbudit, aby přijali přírůstkový pohled na inteligenci člověka, tedy víru, že inteligence může být navýšena zvýšeným úsilím. Selhání tak není vnímáno jako absolutní odraz schopností žáka ale mnohem spíše, jako příležitost k učení nebo přestávka mezi minulými a budoucími úspěchy. Takový pohled žáka na sebe sama, povede jistě k příznivějšímu sebepojetí.

William Glasser (2010) nám poskytuje další možnou odpověď na otázku, která vnitřní motivaci žáků rozvíjet. Tuto odpověď lze pro přehlednost rozdělit do trojice komponent.

1. Ukazujme raději, než vyprávějme. Spíše než přednášku nebo demonstraci, coby hlavní způsob výuky - Glasser doporučuje, zvážít větší zapojení učitelů směrem k jednotlivcům a malým skupinkám žáků, kde učitel pomůže žákům překonat větší nebo menší nástrahy v učivu.
2. Posilujme spíše, než přemáhejme. Jakmile budou žákům objasněny cíle vyučovací jednotky nebo tematického celku učiva, žáci by měli dostat příležitost k diskuzi o způsobech, kterak stanovených cílů dosáhnout.
3. Zdůrazněme výhody kooperativní práce pro dosažení společných cílů. Sdílené cíle třídy by měly být akcentovány. Sociální prostředí podporuje vnitřní motivaci tím, že uspokojuje potřebu žáka v oblasti autonomie, snadnějším získáváním kompetencí a sounáležitostí se spolužáky.

K podobným závěrům o podstatných faktorech vnitřní motivace dospěl Carole Ames. Podle Amese (1992) je důležité, aby si učitel připravil vyučovací hodiny tak, aby pomohl žákům vytvořit si nový obraz o sobě, jako o žácích. K tomuto účelu vypracoval mnemotechnickou pomůcku – TARGET

T (tasks) – učební úlohy jsou vybírány tak, aby poskytovaly optimální úroveň výzev a zdůrazňovaly činnosti, které jsou pro žáky zajímavé a vnitřně poutavé, jsou tedy odrazem jejich přirozených zájmů.

A (autonomy) – motivace žáka se zvyšuje, jestliže žák pocítuje určitou míru autonomie. Tak posilujeme pocit odpovědnosti za vypracování učebního úkolu.

R (recognition) – učitelova kompozice vyučování by měla umožňovat alespoň občasné uznání všem žákům, přestože to nemusí být snadné. Uznání by mělo náležet také žákům, kteří dosáhli výrazného pokroku, vzhledem k sobě samému.

G (grouping) – učitel řídí vyučovací jednotky tak, aby podporoval kooperativní model učení, mezilidské vztahy, potřebu socializace, kooperaci a odpovědnost za společnou práci.

E (evaluation) – hodnocení je tvořeno pomocí více kritérií, která umožňují individualizované hodnocení práce žáka, jeho pokroku více než hodnocení absolutní, tedy například porovnání se třídou.

T (time) – hledáme kreativní způsoby, například takové hodnotné učební úlohy, abychom využili čas vymezený pro vyučovací jednotku co nejefektivněji.

Glasser (2010) zkoumal také motivační činitele, které žáci potřebují ve školním kontextu a předpokládá, že vzdělání žáků roste a vzkvétá, jestliže jsou učitel a žák společně zapojeni do zdolávání požadavků učebních osnov, prostřednictvím myšlení - v problémech. Žáci s učitelem kooperují na řešení problémových učebních úloh, nejlépe v rámci problémového vyučování.

Doposud jsme si všímali vývoje a obecnějších aspektů motivace. Závěrem přiblížíme teorii čtveřice motivačních činitelů - emocí (láska, moc, zábava, svoboda), podle Glassera. Tyto determinují žakovu míru motivace ve vyučování.

1. Žák potřebuje pocítovat příslušnost, patřit do určité skupiny. Rodina je první takovou skupinou, kterou obvykle zažije. Třída a školní komunita druhou. Střední škola je často místem, kde se rozvíjí fakticky nezávislost žáka na rodině a žák buduje svoji identitu více odvislou od školy a přátel.
2. Druhou potřebou žáka ve školním prostředí je moc. Nikoliv ve smyslu nadřazenosti jednoho člověka nad druhým, nýbrž moc ve smyslu schopnosti získávat znalosti k tomu, co chce žák umět a tomu by měl být učitel nápomocen.

3. Další potřebou je zábava. Škola by měla být zdrojem nejenom plnění výchovně vzdělávacích cílů ale také zdrojem zábavy. Dosažení nějakého výukového cíle by mělo být zdrojem skvělé zábavy, protože místo, kde se bavíte, je také místem, kde chcete trávit svůj čas. Každý žák by měl alespoň někdy zažít pocit úspěchu či uznání za odvedenou práci.
4. Konečně poslední potřebou je Svoboda. Jeden z největších demotivátorů našich žáků je přesvědčení, že nemají žádnou svobodu. Učitel by měl své žáky vest k svobodné volbě všude tam, kde je to jenom možné. Na druhé straně by je měl také vest k odpovědnosti za důsledky své práce nebo činů.

2.4 Uplatnění problémového vyučování

Na tomto místě se soustředíme pouze na nezbytnou teorii k problémovému vyučování. K této metodě uvedeme podrobněji také její aplikační možnosti. Budeme zde klást akcent na efektivní využití této metody, jejichž ovládnutí je potřebné pro náležité využití naší metodické příručky. Výběr této metody není náhodný a pro výuku nejen ICT mimořádně vhodný. Žáci budou řešit mnoho problémových úkolů. Některé budou více či méně explicitně dány zadáním učebních úkolů, jiné vyplynou z aktivní práce samotné. Vždyť komu se například „nepoštětilo deaktivovat“ často používané menu programu s nimž právě pracujeme?

Pojetí problémového vyučování se u různých badatelů liší, prolíná, avšak v hlavních bodech se shoduje. Výrazní badatelé problémového vyučování, patří již k starší generaci autorů, jejich dílo však podle našeho názoru nebylo v mnohém doposud překonáno.

Americký progresivismus první poloviny 20. století odmítal formalismus, striktní disciplínu nebo pasivní dril v učení. V tomto duchu badatel Dewey, při své tvrdé kritice tradiční školy uvádí “...žáčkové jen sbírají vědomosti jako teoretičtí diváci, jako duchové osvojují si vědění přímou činností rozumu. Samo slovo žák nabylo již takřka významu bytosti, která se zaměstnává tím, že má vědomosti přímo jen hltati, nikoliv však míti užitečné zkušenosti ...“ (Dewey 1932, s. 191). Je patrné, že Dewey ve své kritice tehdejšího školství, obnažuje hned několik jeho rovin. Z humanistického pohledu upozorňuje na nedostatečné vnímání žáka, jako lidské bytosti, která by neměla být pouze výhradním příjemcem transmisivního pojetí vyučování. Druhou rovinu můžeme spatřovat v oprávněné potřebě procvičovat znalosti a dovednosti - **na praktických učebních úlohách** a vytvářet tak v žácích předpoklad uplatnění v profesním životě. Implicitně lze vyvodit poslední rovinu, tedy absenci praktického prožitku, skrze adekvátní učební úlohy. Právě prostřednictvím tohoto prožitku („identifikací

s problémem“), dochází k vnitřní motivovanosti žáka a skrze emoční zápal, probíhá umocnění výkonu kognitivních funkcí, zejména pozornosti a koncentrace, rychlosti zpracování informací a v řadě neposlední paměti.

Pro kritiku klasického školství však nemusíme chodit daleko - až „za velkou louží“, neboť naše školství za první republiky, mělo také své kritiky „*všechny přirozené schopnosti, mají své vztažné předměty a rostou právě tehdy, vycházejí-li z těchto předmětů popudy k činnosti.*“ (Uher 1930, s.66).

V našem prostředí se ozývají podobné snahy, korigovat nebo dokonce přetvořit vyučovací proces větší aktivitou dětí a jejich samočinnost.

„*Učte děti věcem, jež mají být konány tím, že jim je dáváte dělat*“ (Uher 1930, s. 77).

Definujme nyní problémové vyučování následovně: „*Problémovým vyučováním rozumíme soubor takových činností, jako organizování problémových situací, formulování problémů (postupně si tuto činnost osvojují žáci sami), poskytování nezbytné pomoci žákům při řešení problémů a při ověřování těchto řešení a konečně řízení procesu systematizace a upevňování takto získaných poznatků.*“ (Okoň, s.79).

Problémové vyučování v kontextu naší didaktické příručky, vychází z učiva pro realizaci problémového vyučování příhodného. Zásadním je také učitelem systematicky sestavená baterie problémových učebních úloh, které mají potenciál žáka dovést až k pochopení požadovanému učivu. Problémovou učební úlohou, chápeme takovou učební úlohu, na kterou žák není schopen podat okamžitou odpověď. Na základě aktuálního stupně svých znalostí a dovedností tak řešení musí postupně, systematicky nalézat. Nejčastěji s pomocí relevantní odborné literatury, kterou může učitel doporučit.

Problémové vyučování, je v odborné literatuře zmíněno často v kontextu s jeho velkou efektivností, tedy schopností „naučit“ a relativně trvale. Jednu z dalších odpovědí na otázku, proč tomu tak je, poskytuje badatel Robert Čapek (2019), který komparuje efektivitu v učení v závislosti na žákově činnosti. Podle Čapka (2019) v nás zanechá stopy, osvojíme si 20 % z toho co slyšíme a současně vidíme. Jestliže ale totéž prožijeme nebo použijeme, efektivita vzroste na 80 %. Konstatované ponecháme bez dalšího zbytečného komentáře.

Jak tedy může být problémové vyučování učiteli technických předmětů a v naší metodické příručce nápomocno? Problémové vyučování je tedy efektivním přístupem k učení, zaměřený na žáky. Zahrnuje jednotlivce nebo skupiny žáků, pracujících na řešení problému

reálného světa, zcela odlišně od metody přímé výuky učitele, který prezentuje fakta a koncepty konkrétního předmětu ve třídě žáků. Prostřednictvím praktického vyučování, žáci nejen posilují své týmové - osobnostní, sociální, komunikační a výzkumné dovednosti, ale také si zdokonalují své schopnosti kritického myšlení, jakožto nezbytné komponenty lidské moudrosti. Žák získává kompetenci k řešení problémů, nezbytných pro celoživotní učení. Právě tato kompetence, je v dnešní turbulentní a informační společnosti téměř krucióální. Problémové vyučování nestojí v pedagogickém vědění osamocně, naopak je často odrazovým můstkem k realizaci projektové výuky. Opomenutí problémového vyučování nebo projektově orientované výuky, je především v technickém vzdělávání pro základní a střední školy téměř nepředstavitelné.

Povšechné proměny společnosti a školství v trojici rovin, dříve naznačených se promítají také do závazných kurikulárních dokumentů, zejména RVP a ŠVP. Odtud pro učitele, ředitele - ožívá potřeba efektivního plnění závazného učiva, ale také klíčových kompetencí, průřezových témat nebo mezipředmětových vazeb. Vrcholným cílem učitelova snažení, je v souladu s pozitivistickým pojetím výchovně-vzdělávacího procesu, pomoci svým žákům, citlivě konstruovat jejich poznání světa, krok - za krokem, berouce v patrnosti stav jejich prekonceptů. Dopomoci tak vychovat z žáků všeobecně vyzrálou osobnost.

Ve zbývající části kapitoly, si klademe za cíl, vytvořit stručnou metodiku rolí aktérů problémového vyučování, především tedy žáka a – učitele, použitelnou v rámci výuky technických předmětů, a tedy také v naší metodické příručce. Využíváme zde ponejvíce vlastní zkušenosti s výukou ICT na SŠ. Jednotlivé role aktérů problémového vyučování, jsou uvedeny pro přehlednost v odrážkách.

Tabulka 1: Porovnání procesu tradičního a problémového vyučování

Tradiční pojetí vyučování:		
▶ Učitel řekne , co potřebuje žák znát	▶ Žák si termíny, vztahy, generalizace.. zapamatuje, pomocí kognice	▶ Učitel problém přiřadí k ilustraci, příkladem, jak znalosti použít
Problémové vyučování:		
▶ Učitel iniciuje a motivuje vhodnou prezentací problém	▶ Žák si postupně vytváří představu, jak problém vyřešit, zvažuje i jiné alternativy řešení (nejčastěji formou písemnou, např. vývojový diagram..)	▶ Žák problém řeší vlastními znalostmi a hledáním dalších relevantních zdrojů informací. Uspokojení z řešení = předpoklad další činnosti

Při implementaci problémového vyučování (dále jenom PV) se role ve výuce přesouvá od tradičtějšího modelu, který sleduje lineární postupný vzorec, kdy učitel předkládá relevantní materiál, informuje třídu o tom, co je třeba udělat, a poskytuje studentům podrobnosti a informace, aby mohli své znalosti aplikovat na daný učební příklad. U PV působí učitel jako zprostředkovatel; učení je řízeno žákem s cílem vyřešit daný problém (poznámka: problém je nastolen na počátku učení oproti tomu, aby byl v tradičním modelu prezentován jako poslední). Úkoly se také liší délkou - od relativně krátkých, po úkoly trvající celé pololetí a ještě ve formátu pro skupinovou práci. Rozdílnost pojetí, ilustruje tabulka č. 1.

2.4.1 Role učitele v problémovém vyučování

Jak již bylo zmíněno dříve, učitel určí problém, **který je pro studenty zajímavý, relevantní** a nový, praktický, s nímž se žák může vnitřně ztotožnit a jeho řešení - vyřešení bude doprovázeno kladným prožitkem, motivující do další práce. Rovněž musí být dostatečně mnohostranný, aby zapojil všechny žáky do výzkumu a umožnil hledání různých řešení. Problémy musí korespondovat s požadavky kurikulárních dokumentů a odrážet tak možné využití v budoucích pracovních situacích na základních, středních školách a profesním životě.

- Určete problém sladěný s kurikulem. Problém musí být natolik náročný, že ho žáci pravděpodobně sami nevyřeší. Musí se také naučit novým dovednostem. Když sdělíte problém s žáky, uveďte jej ve vyprávění, doplněném o příslušné základní informace, tj. bez nadměrných informací. Umožněte žákům zjistit více podrobností při práci na problému.
- Umístěte žáky do heterogenních skupin, dobře promíchaných v rozmanitosti a úrovních dovedností, aby se skupiny posílily. Pomozte žákům úspěšně pracovat s minimem korekcí. Jedním ze způsobů je nechat žáky převzít různé role ve skupinovém procesu poté, co sami vyhodnotí své silné a slabé stránky (někdo vyhledává informace v knihovně, jiný na internetu, další formuluje závěry...).
- Podporujte žáky s porozuměním obsahu na hlubší úrovni a způsoby, jak nejlépe zorganizovat různé fáze procesu řešení problémů.

2.4.2 Jak probíhá příprava učitele na problémové vyučování

Na základě souhrnu teoretických studií v této problematice a naší pedagogické praxe, vyslovujeme body přípravy.

- Stanovte cíle učení (tj. co chcete, aby se vaši žáci skutečně učili a byli to schopni dělat po dokončení problémového učení).
- Najděte problém ze skutečného světa, který je relevantní pro žáky; často se jedná o problémy, s nimiž se žáci mohou setkat ve svém vlastním životě nebo budoucí kariéře, které mají s největší pravděpodobností několik odpovědí, které divergují.
- Diskutujte o příslušných pravidlech pro práci jednotlivců či ve skupinách, abyste maximalizovali úspěch ve vzdělávání. Při výběru nepochybně zvažte očekávané zájmy žáků, aby učení se řešením problému, mělo emociálně kladný potenciál.
- Procvičujte skupinové procesy: naslouchání, zapojení ostatních, hodnocení jejich práce nebo kolegů z týmu (aktivizační metody, hraní rolí, brainstorming..).
- Vezměte v potaz vyučovací téma, fragmentujte je na logické dílčí jednotky – „sub-problémy“. Rozhodněte o typech problémů, které vaši žáci vyřeší. Právě toto, budou vaše cíle.
- Uspořádejte vhodné zdroje dostupné žákům (předložený materiál, školní knihovna, mobilní telefon s internetem – k tomuto účelu žákům povolený).
- Prozkoumejte možnosti různých rolí žáků, aby dokončili práci, kterou je třeba udělat, nebo abyste viděli problém z různých perspektiv v závislosti na problému (např.

u problému černé skládky, mohou žáci přebírat rozličné role: starosta, vlastník postižené firmy, pravděpodobný pachatel, rodič, dítě, sousedé, vládní úředníci atd.).

- Rozhodněte, jak budou žáci své výstupy prezentovat (PowerPoint, blogy, Fcb..) a zohledněte metodiku hodnocení, tedy míru účasti žáků ve skupině (např. procentem, možné vzájemné hodnocení ve skupině atd., slovní..).

2.4.3 Uspořádání aktivit žákům

- Vysvětlete žákům PV: jeho odůvodnění, místo ve výuce, očekávání od třídy, metodika jejich hodnocení.
- Poslužte jako model a zdroj pro proces PV; učitel může například pracovat společně v úvodním problému, který žákům procesně usnadní „náběh“.
- Pomozte žákům v případě potřeby, zajistit různé zdroje informací.
- Poskytněte dostatek času třídě pro skupinovou spolupráci.
- Poskytněte zpětnou vazbu každé skupině poté, co sdílí postup prostřednictvím stanoveného formátu; kritizujte řešení co do kvality i důkladnosti přípravy. Pomozte žákům pochopit, že kromě řešení je důležitý i předchozí koncepční proces myšlení.

2.4.4 Role žáka v problémovém vyučování

- Při práci v rámci PV, se budou žáci zapojovat do otevřených situací, které asimilují praktický svět práce, techniky a výchovy.
- Účastnit se diskuse ve skupinách (ideálně, je-li zvolena taková forma výuky).
Ve skupině snáze identifikujeme známé a doposud neznámé, analyzují se metody, kterak vyhledávat relevantní informace, mající potenciál problém vyřešit.
- Řešit problém, prostřednictvím kritického myšlení, případně vytvoří další jedinečná řešení problému. V kontextu s požadavky současného kurikula (RVP, ŠVP), na požadavky mezipředmětových vztahů, průřezových témat a klíčových kompetencí, žák zohlední například také: ekologickou udržitelnost, ekonomickou realizovatelnost řešení.

2.4.5 Žádoucí myšlenková posloupnost činností žáka

- Žák analyzuje problém, který je představen. Rozdělí problém na různé části. Pokračuje ve čtení studijních materiálů, diskutuje a přemýšlí o problému.

- Vytvořte seznam toho, co je o problému známo. Co o problému vědí vaši spolužáci? Mají nějaké zkušenosti spojené s problémem? Diskutujte o příspěvcích očekávaných od členů týmu. Jaké jsou jejich silné a slabé stránky? Postupujte například podle pravidel metody brainstormingu (tj. přijímejte všechny odpovědi bez úsudku), abyste vytvořili možná řešení problému.
- Vytvořte prohlášení o problému vlastními slovy a vezměte v úvahu znalosti a zkušenosti týmu, jak již bylo dříve diskutováno, a také to, co je ještě třeba znát, abyste problém vyřešili. Postupujte podle následujících kroků:
 - Získejte souhlas členů týmu ohledně prohlášení o problému.
 - Vytvořte prohlášení o problému v písemné formě.
 - Vyžádejte si zpětnou vazbu od učitele.
 - Buďte otevřeni změně písemného prohlášení na základě jakékoli relevantní informace nebo z poskytnuté zpětné vazby.
- Vytvořte seznam možných řešení. Zahrňte relevantní myšlenky, nápady a poučené odhady, stejně jako příčiny a možné způsoby řešení. Poté seřaďte řešení a vyberte to řešení, které vaše skupina s největší pravděpodobností bude vnímat jako nejlepší z obhájitelných důvodů.
- Zvažte, jaké informace může učitel ještě poskytnout.
 - Jaké zdroje jsou k dispozici pro pomoc (např. učebnice, primární / sekundární zdroje, internet).
 - Určete výzkumné úkoly na členy týmu.
 - Stanovte termíny realizace.
- A konečně, stanovte časovou osu s konkrétními akcemi.
 - Uveďte, co je třeba vědět a udělat pro vyřešení zjištěných problémů.
 - Uveďte všechny kroky plánovaného procesu řešení - posloupnost.
 - Zvažte, jak kroky ovlivňují všechna možná řešení.
 - Podívejte se, zda skupina souhlasí s tvořenou časovou osou (deadline); pokud ne, rozhodněte se, jak dosáhnout dohody.

- Uspořádejte poznámky a napište zprávu týmu (návrh / konečnou verzi) řešení problému. Nezapomeňte přidat podpůrné dokumenty, jako zdroje.. Postupujte podle pokynů učitele, pokud jde o formát, očekávaných náležitostí zprávy.
 - Určete, jak vaše skupina představí řešení problému. V PV obvykle každá skupina prezentuje svá řešení prostřednictvím týmové prezentace buď třídě ostatních žáků, nebo také všem, kteří s problémem souvisejí.
 - Je třeba zahrnout proces i výsledky učební činnosti. Zahrňte následující: prohlášení o problému, otázky, shromážděná data, analýza dat, důvody řešení nebo doporučení odrážející analýzu dat.

- Je důležité si uvědomit, že cílem PV je představit závěry, na kterých tým pracoval. Je proto důležité následující:
 - Dobře stanovený problém a závěr.
 - Proces prováděný skupinou při řešení problému, různé diskutované možnosti a použité zdroje.
 - Podpůrné dokumenty, hosté, rozhovory a jejich účel, aby vaše řešení bylo přesvědčivé pro vaše publikum.
 - Kromě toho buďte připraveni na jakékoli komentáře a dotazy publika. Určete, kdo odpoví, a pokud váš tým nezná odpověď, připusťte to a buďte připraveni se na otázku podívat později.

- Reflektování vlastního přemyšlení a přenositelnost nabytých znalostí, jsou důležitými složkami PV. To vše pomáhá žákům, lépe si uvědomit své vlastní učení a učit žáka, jak klást vhodné otázky, vedoucí k řešení problémů. Určete, co jste se naučili a jak se zlepšit do budoucna. Žák si je závěrem vědom, kterak získané poznatky aplikovat na jinou situaci, pro svůj vlastní život.

2.5 Učitelova příprava na vyučování

„Pečlivá příprava na každou vyučovací jednotku (učební den) je nezbytnou podmínkou úspěšné vzdělávací činnosti každého učitele a zárukou, že z výuky bude odstraněna improvizace a náhodnost“ (Stejskalová, 2013, s. 75).

Každý učitel se jistě chce připravovat na výuku svědomitě tak, aby svým žákům zprostředkoval, pokud možno co nejvíce ze svých odborných dovedností a zkušeností. Základním předpokladem splnění tohoto cíle je toliko nasnadě. Učitel by měl být odborně způsobilým k vyučování svého předmětu, ale měl by rovněž neustrnout v rozvoji svých znalostí tak, aby v dnešní informační společnosti, byl schopen stále aktualizovat své odborné poznání a následně také s tím související metodický aparát pro realizovaný výchovně vzdělávací proces.

Rys (1979) k odbornosti uvádí, že čím více a hlouběji učitel rozumí svému předmětu, tím jasněji a srozumitelněji dokáže vysvětlovat složitější jevy a procesy. Tento závěr je poměrně logický, přesto hodný připomenutí. Jestliže učitel rozumí svému oboru, zná bezpečně strukturu terminologického aparátu, významy, vzájemné vazby, je schopen nabídnout žákům přílehavější příklady nebo generalizace. Odborně připravený učitel dokáže vybrat všestranně vhodné obsahy, učivo, kterým zabezpečuje aktivitu žáků. K tomu, aby si žák učivo osvojil, je důležitá jeho aktivní činnost s učivem. Učivo je páteří veškeré pedagogické činnosti učitele, jeho výběr má však svá úskalí.

Podle Chlupa (1966), který se zabýval problematikou přiměřenosti učiva a přetěžováním žáků učivem, je nutné vymezit základní učivo, jako vědomosti, dovednosti i návyky, které ve svém souhrnu představují základy věd, umění, techniky a mají zásadní význam pro další vzdělávání a pro praktický život. Základní učivo má mít všestrannou hodnotu výchovnou. Má si je, přestože v rozdílné kvalitě, osvojit většina žáků příslušných věkových stupňů. Konečně má být probráno, procvičeno a upevněno ve škole, bez přetěžování žáků. Vedle učiva základního, je potřeba reflektovat potřeby zvláště nadaných žáků ve výuce a obohatit učivo partií určenou právě pro ně.

Vašutová (2007) uvádí podobně, že především u začínajících učitelů, je možno se setkat s „naddimenzováním“ výuky. Tento jev vede u našich žáků k problematickému strukturování poznatků, dovedností nebo kompetencí.

Dle Rysa (1979) je jedním z největších problémů příprav učitelů na vyučování, že se spoléhají příliš na učebnice, jejich výběr příkladů, cvičení a úkolů. Naše metodická příručka však aplikuje efektivní didaktické postupy, způsoby účinné motivace, stanovení cílů. Nabízí nové cesty a především inspiraci kolegům. Naproti tomu učitel při zmíněné činnosti, výhradně s učebnicí, přečte obvykle tematicky odpovídající text a rozhodne se, jak velkou část probere v plánované vyučovací jednotce. Učivo doplní příklady z učebnice k procvičení učiva a tím je

orientace na obsah vlastně skončena. Jedná se tak o klasické podtrhávání „důležitého“ z učebnice, na které navazuje obvykle úvaha, jakým způsobem žákům učivo zprostředkovat.

2.5.1 Příprava na vyučování podle Slavomíra Rysa

Přestože se přípravě učitele na vyučování věnovala řada současnějších badatelů, jmenujme například badatelku Skalkovou (2007) nebo badatele Osborna (2007)., přípravu na vyučování podle Rysa (1979), respektive jeho třetí typ přípravy, považujeme pro účely tvorby metodické příručky zdaleka za nejvhodnější. Níže představíme zmíněné tři základní typy, které se od sebe odlišují baterií otázek, které si učitel klade, odpovídá na ně a jsou mu podkladem pro vypracování vlastní přípravy.

Typ č. 1, můžeme nazvat přípravou – „bleskovkou“. Učitel hledá odpovědi na dvě otázky.

1. Co?
2. Jak?

Vymezí tak konkrétní obsahy učiva, a volí metody a prostředky, jimiž učivo žákům zprostředkuje. Učitel nevěnuje pozornost dalším okolnostem výchovně vzdělávacího procesu.

Typ č. 2, reflektuje už čtveřici otázek.

1. Co již bylo?
2. Čeho chci dosáhnout?
3. Jak a čím toho dosáhnout?
4. Jaké bude pokračování hodiny?

V tomto typu přípravy již učitel sleduje kontext pedagogických a didaktických cílů, logickou návaznost učiva. Ani tato příprava však neposkytuje odpověď na otázku, za jakých podmínek bude hodina realizována.

Typ přípravy č. 1 nebo č. 2, je pro naše účely poměrně nezajímavý. Typ č. 3 je naproti tomu zajímavý mimořádně. Vytváří nám ucelený přehled o tom, co vše musí učitel postupně zohlednit při své přípravě. Chceme-li totiž vytvořit metodickou příručku, která bude kolegům nápomocna, nemůžeme pominout analýzu myšlenkových pochodů, které kvalitní přípravu na vyučování charakterizují. Rys (1979) připouští vysokou časovou náročnost této třetí přípravy, pro kterou není bohužel hojněji realizována. To je výzva pro tvůrce této metodické příručky, usnadnit učitelům přípravu všude tam, kde je to možné. Klasická učebnice

totiž všechny prvky z této přípravy zdaleka neobsahuje a ani by to u učebnice nebylo příliš žádoucí.

Typ č. 3 učitelské přípravy tedy zohledňuje následující otázky nebo možné problémy.

1. Stanovení cílů hodiny.
2. Volbu učiva a metodiky. Z metodiky zejména volbu vhodných vyučovacích metod, didaktických pomůcek, techniky a metodický postup.
3. Zvláštní didaktická hlediska, například co bude pro žáky nejobtížnější? Jak žáky aktivizovat? Jak zajistíme diferencovaný a individuální přístup k žákům? Vypracován by měl být také systém učebních úloh.
4. Jaké výchovné možnosti učivo nabízí - a jak je mohu v průběhu vyučování použít?
5. Organizace vyučovací hodiny není možné podcenit. Které pracovní podmínky je nutno zabezpečit? Jaké pomůcky budou potřeba, jak a kdy ověřit jejich funkčnost, jak organizovat prostor atp.
6. Časový projekt hodiny, odpovídá na otázku, kolik času mohu věnovat jednotlivým fázím. Je potřeba domácí příprava žáka a je účelná?
7. K realizaci zvážíme, jak budeme zajišťovat pracovní součinnost žáků a jak ověřit pracovní výsledky žáků.

2.6 Piktogramy užití v metodické příručce

V první kapitole jsme odpověděli na klíčovou otázku, co učitel od metodické příručky očekává, aby pro něj byla přínosná. To plyne především z výzkumu badatelky Jany Staré (2021). Pečlivé analyzování hledisek přípravy učitele na vyučování, nám zase umožnilo odpovědět na otázky, které by učitel měl zohlednit, aby byl výchovně vzdělávací proces úspěšným. Zde mohou autoři metodické příručky opět mnoho inspirativního načerpat, neboť je zřetelněji patrné, čím vším je možno metodickou příručku mimo bazálního učiva obohatit, aby příprava na vyučování mohla být pro učitele snadnější.

Tyto teorie nám tak nepřímo poskytují také vodítko k stanovení struktury metodické příručky. Tato je částečně zachycena v tabulce č. 2, která reprezentuje jednotlivé piktogramy, tedy metodická „zastavení“, která se věnují určitému specifiku, který je potřeba ve vyučovacím procesu zohlednit. Struktura metodické příručky je ale z větší míry determinována zpracovaným učivem, počínající kapitolou třetí a končí kapitolou čtvrtou.

Podkapitoly druhé kapitoly, jsou věnované také osnování cílů výuky, didaktické analýze učiva, motivaci žáků ku školní práci a metodě problémového vyučování. Tato teoretické podhoubí nám poskytlo možnost, vytvořit obsahově metodickou příručku podle moderní pedagogiky naší ale také inspirovanou badateli z USA. Sebelépe vytvořená metodická příručka však nemůže ignorovat fakt, že vyučování je do značné míry dynamickým procesem, v jehož průběhu bude nutné, více či méně často, realizovanou přípravu učitele na vyučování upravit.

Tabulka 2: Seznam užitých piktogramů v textu metodické příručky

	Motivátor. Motivační nebo aktivizační návrh k realizaci, který si zasluhuje zvláštního zřetele.
	Teorie. Báze základních pojmů a jejich vztahů, také výklady, které jsou základem pro porozumění učivu a předpokladem k dosažení aplikační a vyšší kognitivní úrovně u žáků.
	Učební úkoly. Prověřují, do jaké míry studující naplnil cíle výuky a zda je nutno výklad doplnit.
	Řešený učební úkol. Nabízí žákům krok za krokem, praktický vhled do problematiky a pochopení vztahů z teoretické části.
	Poznámky z praxe. Přinášejí náměty k zamyšlení rozšiřující úrovně. Učitel je zařadí dle uvážení. Jejich opomenutí však často omezí zdokonalení do úrovně, obvykle vyžadované v praxi.
	Metodika. Procedurální poznámka.
	Nastavení obtížnosti. Návrh variant obtížnosti s ohledem na žáky zvláště nadané a rozdílnou časovou dotaci předmětu.
	Náročné učivo. Upozornění na obtížnou část učiva s metodickým návrhem řešení.
	Shrnutí. Připomenutí klíčového učiva kapitoly.
	Literatura. Osvědčená, proto doporučená literatura, pro doplnění poznatků, procvičení dovedností nebo kompetencí žáků.

3 Tvorba grafického obsahu

Cíle kapitoly

Žák bude:

- tvořit samostatně 3D modely, pomocí modifikátorů Lathe, Loft, ProBoolean, EditPoly, Displace a především jejich kombinací v aplikaci 3ds Max.
- tvořit fraktální modely terénu pomocí aplikace WorldMachine.
- dokáže zhotovit materiály pro učební úkoly metodické příručky.
- fotografovat v přírodě materiály, použitelné pro tvorbu textur.
- souhlasit s nezbytností ochrany autorských práv tvůrců software.

Dílčí cíle podkapitol

Kapitola 2.1 směřuje k porozumění, filozofii nebo principům tvorby 3D modelů. Kapitoly 2.1.1 – 2.1.4 žáky provádí látkou s cílem naučit žáky aplikovat, použít modelovací techniky. Dalším procvičováním žáci lépe nastavují parametricky užití modifikátory, a především je dokáží lépe kombinovat. Dochází u žáků k schopnosti výstupy analyzovat a hodnotit. Kapitola 2.1.5 je dovršením modelovacích možností a umožňuje žákům, po náležitém procvičení tvůrčí práci, tedy realizovat vlastní tvůrčí představu v ose návrh, náčrtek – modelování – materiál - výsledek. Kapitola 1. 4, věnovaná materiállovému inženýrství, je svým osnováním totožná. Učivo je mimořádně náročné na prostorovou představivost a na dobu po kterou žák může procvičovat. Musíme se smířit s faktem, že část žáků dospěje pouze do aplikační úrovně.

Motivace: Úvodem lze motivovat nabytím výsledných kompetencí v ose: vizualizace – multimedia – návrh výrobků. Také vhodným obrazovým materiálem, inspirující snad každého (například publikace knižní série Digital Art Masters). Průběžná motivace by měla být generována z dílčích úspěchů žáků - z užití jednotlivých modifikátorů, využití nabývaných dovedností ve školní praxi nebo soutěžích. Nutno však respektovat obecné principy motivovanosti žáka k práci - viz kapitola věnovaná motivaci. Respektován musí být obor studia. Identifikujeme však také vlastní preference, potřeby, zájmy žáků a tomuto zjištění plán motivace organizujeme.

Metody: Problémové vyučování viz blíže procedurálně v kapitole 2. 4, dále brainstorming, výklad, názorná ukázka, prezentace, procvičování pod dohledem učitele, skupinová práce. Metodickou příručku lze ale také využít jako inspiraci k projektovému vyučování.

Metodický postup: metodická příručka je koncipovaná tak, že obsahuje bazální učivo, obsahující nejdůležitější termíny, jejich vztahy a příklady využití v praxi. Některé části

obsahují zdrojové soubory s modely a příslušným nastavením, které usnadňují pochopení náročnějšího učiva. Jednotlivé kapitoly jsou řazeny postupně tak, jak by měly být žákům explanovány. Učivo je mimořádně náročné na představivost, zejména prostorovou. Žákům je potřeba nabídnout dostatek příkladů a prostoru k procvičování ve škole a nejlépe také doma.

Program realizace výuky

Kapitola 3.1 | Učitel žákům připomene probrané učivo, zejména podobnost 2D vektorové grafiky (Bézierova křivka) s 3D počítačovou grafikou (nová dimenze). Uplatňujeme zásadu od známého k neznámému [*frontální výuka*]. Jestliže žák nemá předchozí zkušenost s prací ve 3D modelovacím programu, demonstruje učitel zásady navigace v pracovních výřezech aplikace 3ds Max názornou ukázkou (demonstruje nástroje: pan, orbit a transformační gizmo). Uplatní projektor [*instruktáž, názorná ukázka, rozhovor*].

Učitel přímo v aplikaci 3ds Max demonstruje možnosti tzv. Command panelu (neobjasňuje však GUI programu 3ds Max jako celku, přestože je toto obecně častý postup. Program je natolik rozsáhlý, že by žáci měli problém, potřebný objem informací hodnotně osvojit). Objasní v panelu možnosti tvorby základních 2D a 3D primitiv a proces modelování složitějších objektů (postupně v ose: a) 2D, 3D primitiva, b) modifikátor globální a c) případně lokální modifikátor – tj. Edit Poly). Učitel dále demonstruje základní možnosti modifikátorů, rozsahem viz obrázek č. 7. Využije projektor [*instruktáž, rozhovor*].

Následuje burza nápadů na téma „jmenujte typické objekty, které by bylo možno pomocí sady diskutovaných modifikátorů modelovat“ [*brainstorming*]. Z nápadů napsaných na tabuli učitel vybere reálně proveditelné nápady (objasní proč) a po jednom je přidělí jednotlivým žákům k realizaci [*procvičování pod dohledem učitele*].

Zde končí část kapitoly věnované modelování.

Kapitola 3.11 - 3.15 a celá kapitola 3.2 | Postup realizace výuky, je pro každou z těchto kapitolách totožný. Učitel nejprve explanuje použití modifikátoru, jeho typické případy využití a jeho základní parametrické nastavení. Využije projektor. [*frontální výuka*].

Následuje burza nápadů na téma „jmenujte další typické objekty, které by bylo možno pomocí tohoto modifikátoru (Lathe, Loft, ProBoolean, Displace, Edit Poly) modelovat“ [*brainstorming*].

Následně učitel promítne řešený učební úkol, který mají žáci následovat, realizovat. Pro nemalý počet žáků bude přesto řešení náročné a bude představovat problémový učební úkol. Bude potřeba žáky například informovat o tom, jak uvést GUI programu 3ds Max

do výchozího nastavení nebo je necháme vyhledat potřebné v nápovědě k programu. Učitel prochází žáky a spoluřeší obtížnější individuální problémy v práci žáků *[řešení problémů, výuka podporovaná počítačem, procvičování pod dohledem učitele]*. Závěrem učitel dovolí žákům, otevřít si vzorový soubor s modelovaným objektem. Žák má možnost modely porovnat, analyzovat a syntetizovat použité parametry a jejich závislost na kvalitě vzniklé topologie sítě modelu. Svůj model může podle výsledných zjištění upravit. Žák v této fázi pracuje v režimu problémového vyučování. Parametry upravuje a porovnává výsledky, například metodou „pokus - omyl“. Libovolně pracuje s internetem, vyhledává, třídí, vybírá relevantní informace. Využívá nápovědu k aplikaci 3ds max *[řešení problémů, výuka podporovaná počítačem]*.

Kapitola 3.3.1 | Učitel názornou ukázkou explaruje základní pojmy materiálového inženýrství (význam termínů: textura – mapa - materiál). Dále učitel připomene a naváže na hodinu v níž žáci modelovali model terénu a doplní, jak k výslednému modelu generovat texturu (v programu WorldMachine) a exportovat ji do rastrového obrazu. Uplatní projektor. *[instruktáž, názorná ukázka, rozhovor]*.

Kapitole 3.3.2 | Učitel vysvětlí účel použití programu Adobe Substance Painter (kreslení a kompozice textur) a stručně zmíní GUI programu (zejména panel „Layers“). Výhodnou je, jestliže žáci absolvovali výuku v rastrovém editoru (typu Adobe Photoshop, GIMP). Uplatní projektor *[instruktáž, názorná ukázka, rozhovor]*.

Následně učitel promítne řešený učební úkol, který mají žáci následovat, realizovat. Pro nemalý počet žáků bude přesto řešení náročné a bude představovat problémový učební úkol. Bude potřeba žáky například informovat o tom, jak uvést GUI programu Adobe Substance Painter do výchozího nastavení nebo je necháme vyhledat v nápovědě k programu. Učitel prochází mezi žáky a spoluřeší obtížnější individuální problémy v práci žáků *[řešení problémů, výuka podporovaná počítačem, procvičování pod dohledem učitele]*. Závěrem učitel dovolí žákům, otevřít si vzorový soubor, obsahující sadu vrstev, textur k objektu terén. Žák má možnost jednotlivé vrstvy porovnat, analyzovat a syntetizovat použité parametry (pořadí vrstev, nastavení blend modes, typologii materiálů) a jejich závislost na kvalitě, vlastnostech vzniklého materiálu. Výsledný materiál může žák podle vlastních zjištění upravit. Žák v této fázi pracuje v režimu problémového vyučování. Parametry nastavení upravuje a porovnává vzniklé výsledky, například metodou „pokus - omyl“. Libovolně pracuje s internetem, vyhledává, třídí, vybírá relevantní informace z nápovědy k programu.

Pracuje také s dalšími online zdroji (například cgsociety.com) *[řešení problémů, výuka podporovaná počítačem]*.

Kapitola 3.3.3 | Učitel explanuje zásady práce a možnosti nastavení v materiálovém editoru 3ds Max. Především to, jak vkládat vytvořené textury do nastavení materiálů *[instruktáž, rozhovor]*. Učitel předloží žákům (promítne) řešený úkol, kterak vkládat textury do materiálových slotů. Žák zadání reprodukuje a v interakci s nápovědou nastavení vylepšuje *[řešení problémů, výuka podporovaná počítačem]*.

Kapitola 3.4 | Učitel žákům předloží k řešení problémový učební úkol. Jak realizovat export objektu a materiálu pomocí programu 3ds max do formátu Babylon.js? *[řešení problémů, výuka podporovaná počítačem]*. Řešení úkolu je pro kontrolu uvedeno v kapitole 3.4.

Kapitola 3.5 | Učitel promítne zadání jedenácti učebních úloh. Jejich řešení opakuje ale také prohlubuje osvojené znalosti a dovednosti. Učební úlohy 2 – 11, jsou tvořeny pro režim problémového vyučování *[řešení problémů, výuka podporovaná počítačem]*.

Závěrečné zadání učebního úkolu pro kapitulu č. 3 | Zvolte si některý z pomocných objektů hry (míčec a terén již máme), například objekt: trávy, balvanu, stromu, golfová atrakce. Zvolený objekt vytvořte. Doporučujeme realizovat ve skupině žáků (nejlépe 3 – 4 žáci v jedné skupině). Žákům přidělíme role a požadované výstupy, za něž odpovídají (například, tvorba modelu, úprava modelu, tvorba textury..). *[skupinová výuka, řešení problémů, kritické myšlení, výuka podporovaná počítačem]*. Žák smí a aktivně využívá všech relevantních informačních kanálů informační společnosti.

Časová dotace: na každou z kapitol připadá jedna dvojhodinovka. Procvičování předmětných dovedností však může představovat násobky naznačené časové náročnosti.

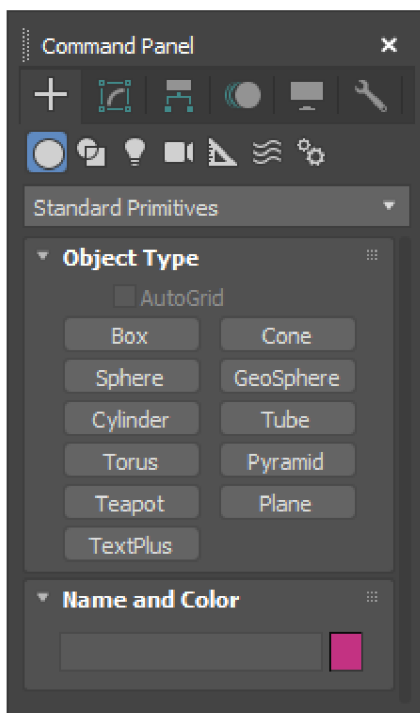
Vstupní požadavky: Žák porozuměl grafickému uživatelskému rozhraní aplikace 3ds Max. Dokáže použít navigační ovládací prvky k prohlížení scény. Uspořádá pracovní výřezy, použije stavové informace a základní transformace objektu - změnu pozice objektu, velikost a rotaci.

Didaktické prostředky ICT: projektor, PC s OS Windows 10 a internetovým připojením pro každého žáka. Minimální konfigurace PC stanice, determinovaná HW požadavky aplikace 3ds Max 2021. Další aplikace: Adobe Substance Painter (alternativně Gimp, Adobe Photoshop) a WorldMachine.

3.1 Modelování 3D objektů v Autodesk 3ds Max



Chceme-li vytvořit 3D model, můžeme začít se základní řadou 3D geometrických primitiv. Tyto základní objekty, stejně jako i další modelovací možnosti nalezneme v pravém postranním panelu, tzv. „Command panelu“, který ilustruje obrázek č. 3. Command panel se skládá ze šesti vnořených záložek grafického



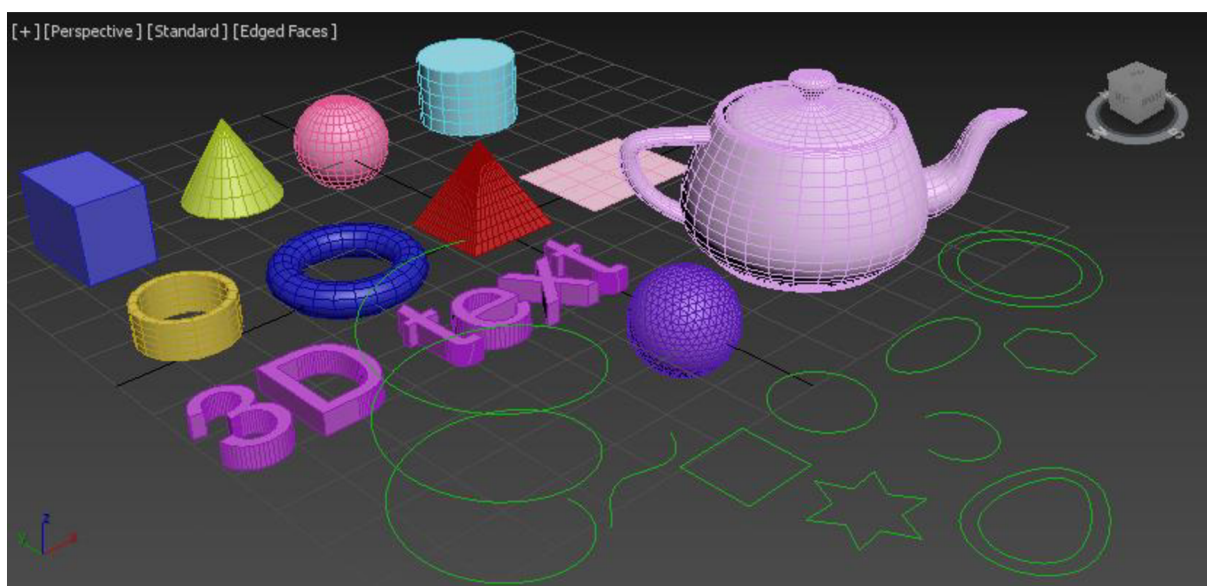
Obrázek 3: *Command panel*

uživatelského rozhraní, které nám umožňují přístup k většině modelovacích funkcí aplikace 3ds Max, stejně jako k některým funkcím animace, možnostem zobrazení a různým nástrojům. Najednou je viditelný pouze jeden panel. Chceme-li zobrazit jiný panel, klepněte na jeho kartu, příslušnou ikonu v horní části příkazového panelu. Podržení kurzoru myši na jednotlivých ikonách, zobrazí 3ds max nápovědu v podobě názvu toho, kterého panelu.

Prvním krokem vzrušující cesty, za vytvořením i těch nejsložitějších objektů, je kupodivu vytvoření jednoduché 3d primitivy. K výběru typu objektu se dostaneme postupně takto:

Command panel > Create > Geometry

Seznam základních objektů vidíme na obrázku č. 4. Jedná se o kvádr, kužel, koule, válec, trubka, torus, jehlan, rovinu a text. Poněkud zvláštní postavení zaujímá objekt „Teapot“ – konev. Jedná se o příklad složitějšího typu objektu, synonymum pro počátky 3D grafiky a současně šikovný objekt pro testovací účely. Inspirativně si lze jak 2D, tak 3D primitiva prohlédnout na obrázku č. 4. Jsou vykresleny v pracovním výřezu 3ds max. Tyto základní objekty vytváříme snadno, a sice kliknutím na tlačítko s popisem objektu (například box). Následným kliknutím a tažením kurzoru myši v jednom z pracovních výřezů objekt vytvoříme na obrazovce. Po uvolnění tlačítka myši, nám 3ds max umožní v oblasti Command panelu,



Obrázek 4: 2D a 3D primitiva 3ds Max

provádět parametrická nastavení, tj. změny vytvořeného objektu. Nejčastěji se jedná o hodnoty výšky a šířky nebo poloměr objektu.



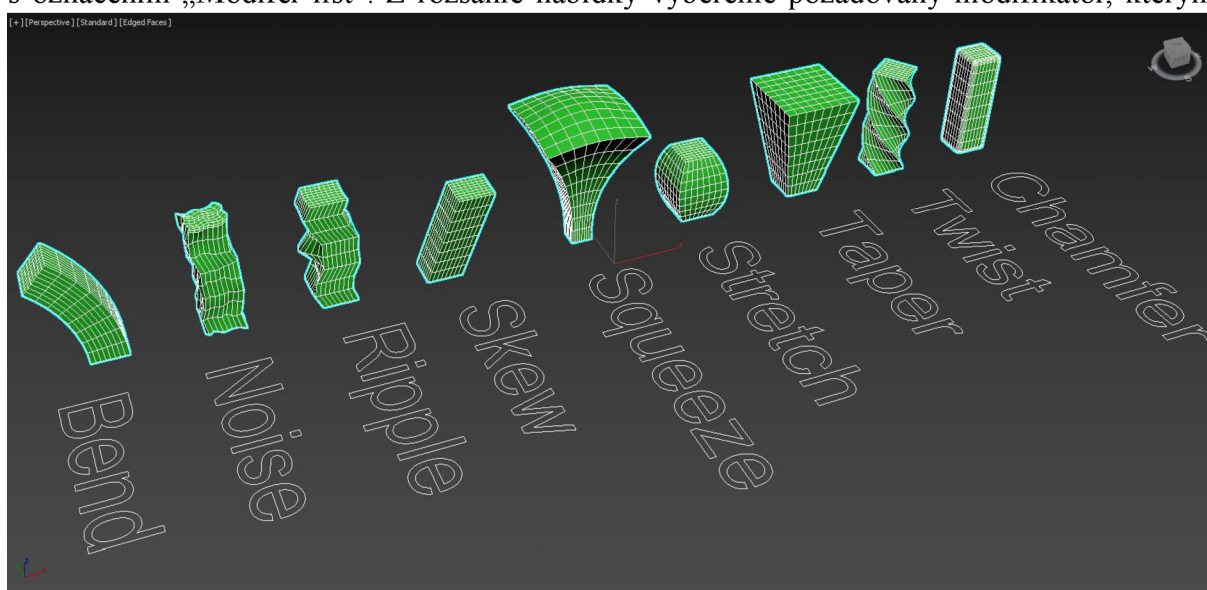
Začátky s 3D grafikou, jsou pro žáky poměrně dosti náročné na představivost. Žáci musí začít chápat také složité objekty, jako logickou soustavu základních 2D nebo 3D primitiv, které postupnými modifikacemi - upravíme do výsledného stavu, tj. složitějšího objektu. Z několika kružnic různého průměru, můžeme vytvořit pohlednou vázu. Z několika kvádrů, můžeme vytvořit věž budoucího modelu hradu. Například metoda brainstormingu na téma využití základních objektů, pro objekty složitější, je pro žáky obvykle přínosná. Především talentovanější žáci, přinášejí všem zajímavé nápady a inspiraci pro vlastní tvůrčí proces. Žáci se učí „myslet“ ve 3D.



Výborně, žák se naučil správně zvolit výchozí objekty, pro tvorbu složitějších objektů. Jak ale tyto jednoduché objekty upravit do podoby těch úžasně vyhlížejících objektů, které žák zná z televizní obrazovky nebo počítačových

her? 3ds Max obsahuje širokou škálu takzvaných modifikátorů. Modifikátory, jak název napovídá, upravují geometrickou strukturu – tvary výchozího objektu a nějakým způsobem jej deformují. Když například použijeme modifikátor zúžení na konec válce, vrcholy poblíž konce se posune blíže k sobě. Modifikátory provádějí změny v geometrii, které zůstávají v platnosti, dokud modifikátor neupravíme nebo neodstraníme. Pro dosažení požadovaného výsledku často aplikujeme hned několik různých nebo stejných modifikátorů, ovšem s jiným parametrickým nastavením.

Vhodný modifikátor přiřadíme objektu tak, že jej v pracovním výřezu vybereme tlačítkem myši. V Command panelu, zvolíme záložku Modify. Rozklikneme rozbalovací menu s označením „Modifer list“. Z rozsáhlé nabídky vybereme požadovaný modifikátor, kterým



Obrázek 5: Základní modifikátory 3ds Max a jejich účinky

budeme měnit geometrii objektu. Mezi základní parametry k nastavení jsou v závislosti na modifikátoru, například úhel deformace, směr v osách x, y či z. Obrázek č. 5 ilustruje možnosti základních modifikátorů lépe než dlouhý výklad.

3.1.1 Modifikátor Lathe



Modifikátor lathe (soustruh), je pro začínající tvůrce mimořádně vděčným modifikátorem. Během několika snadných kroků, žák získá potřebný objekt.

Modifikátor volíme především tehdy, jestliže vystižením siluety objektu, například pomocí Bézierovy křivky a její rotací v některé z os, vytvoříme celý nebo alespoň část zamýšleného objektu. Mezi takové objekty často patří technické objekty vytvořené lidským úsilím. Může se jednat o části nábytku, nejrůznější druhy lahví, svítilna, baterie a mnoho dalších

objektů. Výstupem mohou být objekty podobné těm z klasického soustružení. Soustružení v 3ds Max má však tu výhodu, že je lhostejno z jakého materiálu soustružíme.

Žák může snadno vytvořit objekt nejen pro účely multimediálních projektů ale například také pro 3D tisk, který je ve školním prostředí v posledních letech - z pochopitelných důvodů vděčným námětem k výuce. Obrázek č. 6 ilustruje typické využití tohoto modifikátoru v našem případě se jedná o objekt dětské láhve. Obrázek si zaslouží podrobnější popis, postupně zprava doleva. Zelená křivka představuje polovinu siluety objektu. Rotací křivky v ose Y, vytvoří 3ds max z křivky celistvou síť polygonů³ objektu. Obrázek uprostřed je obrázkem referenčním, vloženým přímo do jednoho z našich pracovních výřezů. Díky tomuto přednímu pohledu na zdrojový originál láhve, můžeme snadno křivku definující tvar objektu vytvořit. V posledním kroku vidíme precizní model dětské láhve s přiřazeným metalickým materiálem.



Pracovní, metalický materiál, je na objekty s požadavkem na vysokou úroveň detailu ve scéně aplikován často. V pracovním výřezu tak můžeme snáze zjistit, zda má objekt kvalitně tvořenou geometrii. Některé složitější objekty



Obrázek 6: Příklad užití modifikátoru Lathe

³ Polygonová síť představuje soustavu logicky uspořádaných vrcholů, hran a ploch, které definují 3D objekty.

jsou tvořeny celé hodiny. Je na ně během této doby použito mnoho modifikátorů a desítky nebo stovky příkazů. Snadno se může stát, že se některé polygony dostanou do pozic vzájemného překrývání a síť polygonů modelu není „dokonalá“. Pravidelnost odlesků metalického materiálu však prozradí sebemenší nepřesnost nebo překryv. Tento metalický materiál nemusíme tvořit, je totiž součástí základní sady materiálů programu.

V následujícím textu si žák vyzkouší tři kroky, ilustrované na obrázku č. 6, vedoucí k výslednému modelu dětské láhve. Rastrový obrázek láhve z předního pohledu nám poslouží jako referenční obrázek, který „obkreslíme“ pomocí 2D křivky do úrovně přesné půle siluety modelu. Posledním krokem je, přiřazení modifikátoru Lathe s nastavením správných parametrů, docílíme výsledku.

1. Příprava referenční šablony



Create panel > (Geometry) > Standard Primitives > Object Type > tlačítko Plane.

Tlačítkem myši klikneme a táhneme objekt Plane do požadované velikosti, nejlépe objekt vytvoříme v předním pracovním výřezu. V Command panelu upravíme velikost plane objektu podle velikosti, poměru stran našeho referenčního obrázku. Referenční obrázek nalezneme v Průzkumníku Windows a přetáhneme jej myší na objekt, který jsme právě vytvořili.

2. Vytvoření křivky siluety objektu

Create panel > Shapes > tlačítko Line.

Vybereme požadovaný typ křivky, tj. corner, smooth nebo bézier. Kurzor myši umístíme do pracovního výřezu a táhneme vznikající křivku do požadovaných pozic. Nejsnáze křivku vytvoříme v předním pracovním výřezu. V Command panelu můžeme dodatečně označit jednotlivé body definující křivku a upravit ji do finální podoby.

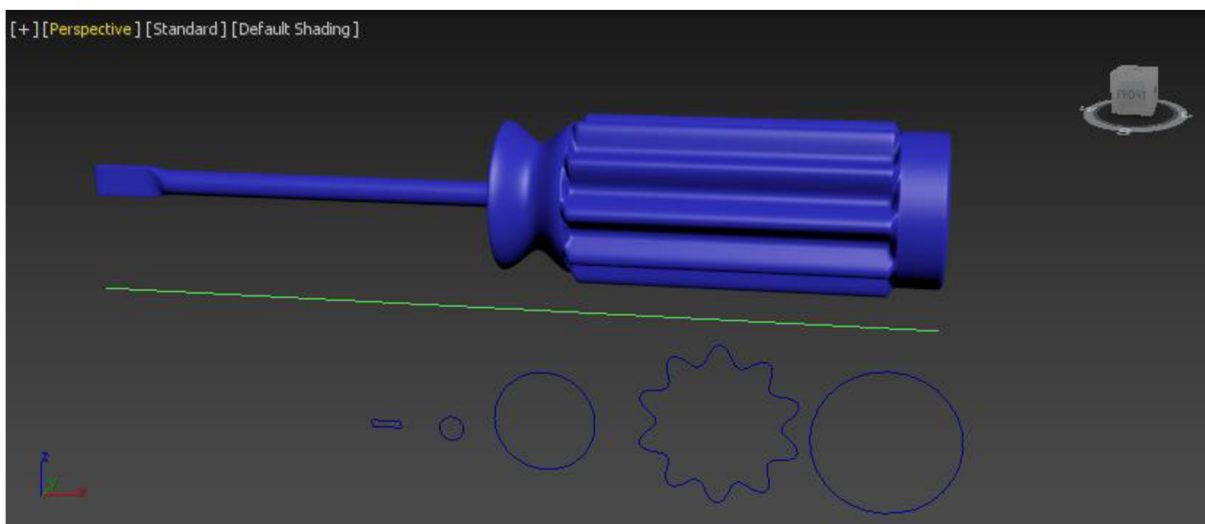
3. Přiřazení modifikátoru

Označíme křivku, která bude objektu „profilem“ > Modify panel > Modifier List > Lathe. Parametr Direction nastavíme na osu Y. Objekt bude vytvořen rotací kolem zvolené osy. Rozkliknutím trojúhelníku u názvu modifikátoru, můžeme zvolit položku „Axis“. Polohu osy gizma je možno takto také upravit. Tuto možnost by měl žák pečlivě vyzkoušet. Pomocí zdrojového souboru **lathe.max**, může žák vlastní postup zkontrolovat nebo si jej zjednodušit.

3.1.2 Modifikátor Loft



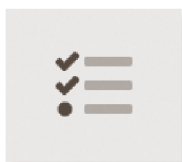
Proces loftingu nejprve vyžaduje, abychom vytvořili objekt křivky, který bude sloužit jako cesta, pro další křivky. Ty budou tvořit objem objektu, jeho průřezy. Termín lofting pochází z ranného období stavění lodí, kdy byla postavena velká konstrukce zvaná „loft“, toto držela trup lodí, když byla sestavena. Proces



Obrázek 7: Pracovní výřez s aplikovaným modifikátorem Loft

zvedání žeber (průřezů) trupu lodí, vešel v povědomí pod termínem „lofting“. Tradiční metodou tvorby trojrozměrných modelů moderního designu je právě kreslení průřezů v řadě klíčových bodů, tedy v místech, které objemový tvar modelovaného objektu definují. Tyto průřezy tvoří dvourozměrné šablony, které jsou umístěny na dříve vytvořené, zmíněné cestě. Tvůrci modelu tímto modifikátorem následně vyplní prostor mezi šablonami, aby vygenerovali povrch modelu. Nejprve tedy vytvoříme dva nebo více křivek. Jedním z těchto křivek tzv. cesta. Zbytek křivek jsou shapes (průřezy) našeho budoucího objektu. Když uspořádáme své průřezy podél cesty, 3ds Max vytvoří povrch mezi tvary. Jako praktický příklad uvedeme objekt šroubovák. Tvorba tohoto objektu dobře vystihuje možnosti tohoto modifikátoru.

Modelování šroubováku



1. Vytvoříme postupně všechny křivky, jako na obrázku č. 7.

Create panel > Shapes > Splines

2. Označíme křivku, která bude „cestou“ a dále postupujeme > Create panel > (Geometry) > Compound objects > Loft

V nastavení modifikátoru, vybíráme postupně další a další průřezy, pomocí tlačítka Get Shape. Každému výběru nového průřezu šroubováku předchází jedinečné nastavení modifikátoru:

- path: 0 > Get Shape > vybereme velký kruh
- path: 5 > Get Shape > vybereme velký kruh
- path: 5 > Get Shape > vybereme velký kruh
- path: 6 > Get Shape > ozubené kolo
- path: 40 > Get Shape > ozubené kolo
- path: 41 > Get Shape > velký kruh
- path: 49 > Get Shape > velký kruh
- path: 50 > Get Shape > nejmenší kruh
- path: 90 > Get Shape > nejmenší kruh
- path: 93 > Get Shape > obdélník
- path: 100 > Get Shape > obdélník



Obrázek 8: *Vizualizace šroubováku*

Obrázek č. 8 ilustruje jednoduchou vizualizaci právě vytvořeného modelu šroubováku. Pomocí zdrojového souboru **loft.max**, může žák vlastní postup zkontrolovat nebo si jej zjednodušit.

3.1.3 Skládání objektů pomocí ProBoolean

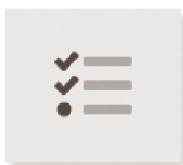


Modelování objektů pomocí Booleanovských operací, spočívá v kombinaci dvou nebo více objektů s nimiž jsou vzájemně provedeny logické operace.

ProBoolean nabízí řadu funkcí, jako je schopnost kombinovat více objektů najednou, z nichž každý používá jinou Booleovskou operaci. V prvních verzích 3ds Max byl tento způsob modelování poměrně nedokonalý, docházelo často k neočekávanému zhroucení sítě objektu a výsledky operace nebylo možno využít. ProBoolean je však nyní velmi účinným

modelovacím nástrojem. Jaké logické funkce tedy ProBoolean podporuje a jak si jejich účinky v konečném výstupu představit? Jedná se o sjednocení, průnik, odčítání, sloučení, připojení a vložení. První tři operace fungují podobně jako v logické algebře. Operace sloučení se protíná a kombinuje dvě sítě bez odebrání původních mnohoúhelníků. To může být užitečné v případech, kdy je třeba selektivně odstranit části sítě. Operace připojení kombinuje více objektů do jednoho, aniž by ovlivnila jejich topologii⁴; zůstávají v podstatě samostatnými prvky složeného objektu. Operace vložení odečte ohraničující tvar operandu⁵ A (objektu A) od operandu B (objektu B) a poté nahradí vyřezávku operandem A. Podporovány jsou také dvě varianty logických operací: Imprint a Cookie Cutter. Možnost Imprint vloží (otisky) hrany průsečíku mezi operandy a původní sítě bez odstranění nebo přidání ploch. Otisk rozdělí pouze plochy a přidá nové hrany do sítě základního objektu (původní vybraný objekt). Cookie Cutter provede zadanou Booleanovskou operaci, ale nepřidá plochy z operandů do původní sítě. Může být použit k vyříznutí díry v síti nebo získání části sítě z jiného objektu.

Modelování pomocí ProBoolean



1. Vytvoříme 2D objekt hvězdy: Create panel > Shapes > Star.

2. S označeným objektem hvězdy: Modify panel > Modifier List > Extrude.

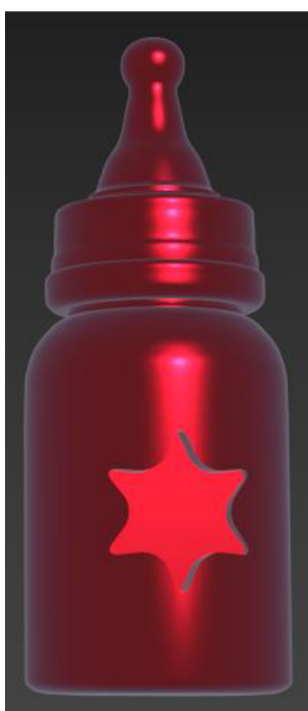
Nastavíme parametr Amount například na hodnotu 50. Takto získáme již 3D objekt hvězdice. Pomocí gizma přesuneme objekt do požadované pozice tak, aby se objekt hvězdice a dětské lahve vzájemně protínaly.

3. S označeným objektem dětské láhve klikneme na tlačítko: Create panel > Compound object > ProBoolean.

⁴ Topologie sítě je povrchová charakteristika počítačového 3D objektu. Jedná se o jeho konkrétní rozložení vrcholů, hran a polygonů v prostoru.

⁵ Operand je vstupním objektem modifikátoru.

4. V paletce Parametrs/Operation zkontrolujeme, zda je parametr pro následující operand nastaven na „Substraction“ a jestliže je tomu tak, klikneme na tlačítko „Start Picking“ a v pracovním výřezu vybereme objekt hvězdy. Výsledek patrně nejtypičtější operace tohoto způsobu modelování, kdy z jednoho objektu odečteme část druhého, ilustruje obrázek č. 9. Pomocí zdrojového souboru **boolean.max**, může žák vlastní postup zkontrolovat nebo si jej zjednodušit.



Obrázek 9: Příklad použití ProBoolean

3.1.4 Modifikátor Displace



Modifikátor Displace funguje jako silové pole, které vytlačí a přetváří geometrii modifikovaného objektu. Jeho proměnnou sílu aplikujeme především z rastrového obrázku, díky jeho odstínům šedi. Každý z odstínů šedé, představuje jinou intenzitu vysunutí nově vznikající 3D sítě.

Na požadovaný objekt aplikovaný modifikátor Displace lze parametricky nastavit, tímto základním způsobem. Efekt míry vysunutí geometrie, ovlivníme přímo nastavením hodnot parametřů Strength (síly) a Decay (rozpad). K vygenerování vysunutí, použijeme obrázek ve stupních šedi. Ten umístíme například přetažením myši do tlačítka s popisem „NONE“. Světlejší barvy ve 2D obrazu, tlačí geometrii směrem ven, tmavší barvy opačně, tedy dovnitř objektu, což má za následek 3D vysunutí geometrie a faktický vznik 3D objektu z rastrového obrazu.

Displace modifikátor obecně vytlačuje geometrii ve směru osy Z, ovšem s přihlédnutím k způsobu a poloze distribuce síly, prostřednictvím trojice různých metod mapování⁶ vytlačované bitmapy na objekt. Jedná se o tyto nejčastěji používané metody mapování: Planar (planární), Cylindrical (válcový), Spherical (sférický). Bez zbytečných podrobností uvedeme, že správná volba tohoto parametru, závisí na tvaru objektu, na který bude Displace modifikátor působit. Planární použijeme na rovinný objekt, například na tvorbu reliéfu terénu (naš praktický projekt) nebo vrásní nanukového dortu. Cylindrický parametr zvolíme při práci s válcovitými objekty, nejčastěji láhve, sloupové stavební prvky a podobně. Sférický parametr použijeme všude tam, kde má cílový objekt složitou kulovitě charakteristickou formu, tj. míče, stavební prvky, zeměkoule a podobně. Zmíněná trojice parametrů se také používá jako mapovací souřadnice pro použití mapování bitmap (textury) na objekt. Spherical a Shrink Wrap mají při modelování stejný účinek, liší se ale způsobem mapování. Rozdíl je pro naše účely zanedbatelný. Ve výchozím nastavení jsou tzv. gizma, trojice popisovaných parametrů výše (Planar, Cylindrical, Spherical a Shrink Wrap) soustředěny na objekt tak, aby jej, co nejvíce kopírovaly. Tím je zajištěno kvalitní promítání bitmapy na objekt. Jedině tak získáme kvalitní reliéf, tedy požadovanou deformaci geometrie. Kterýkoli z těchto pomocných gizmo tvarů, však můžeme transformovat (posun, rotace, změna měřítko) a lepe tak bitmapový tvar na objekt situovat.

Displace je univerzálně použitelný modifikátor s mnoha možnými aplikacemi, včetně možnosti nastavitelné parametry animovat. Přestože se může zdát použití tohoto modifikátoru mocné a bezproblémové, své nevýhody samozřejmě má. Objekt nebo jeho část, na který má být modifikátor přiřazen, musí mít síť polygonů dostatečně hustou, aby mohla potřebný detail z bitmapy zpodobnit. Větší hustotu sítě polygonů objektu, dosáhneme například pomocí modifikátoru MeshSmooth nebo Tessellate. Někdy je potřeba zahuštění tak vysoká, že mnoho milionů polygonů nedokáže systém zpracovat, podobně tak cílový engine, umožňující vykreslování 3D grafiky v reálném čase. Použití musí předcházet komplexní úvaha o popsaném. Objasnění základů Displace modifikátoru v prostředí 3ds Max je výhodné s ohledem na fakt, že hlavní objekt do naší hry – terén, je vytvářen na stejném principu. Obrázek

⁶ Metody mapování představují způsoby, kterým je materiál promítán na objekt.

č. 10 ilustruje detail modelování pomocí Displace, na známé dětské lahvi. V pravé části obrázku je černobílý obrázek, který byl zdrojem vysunutí pro modifikátor Displace.

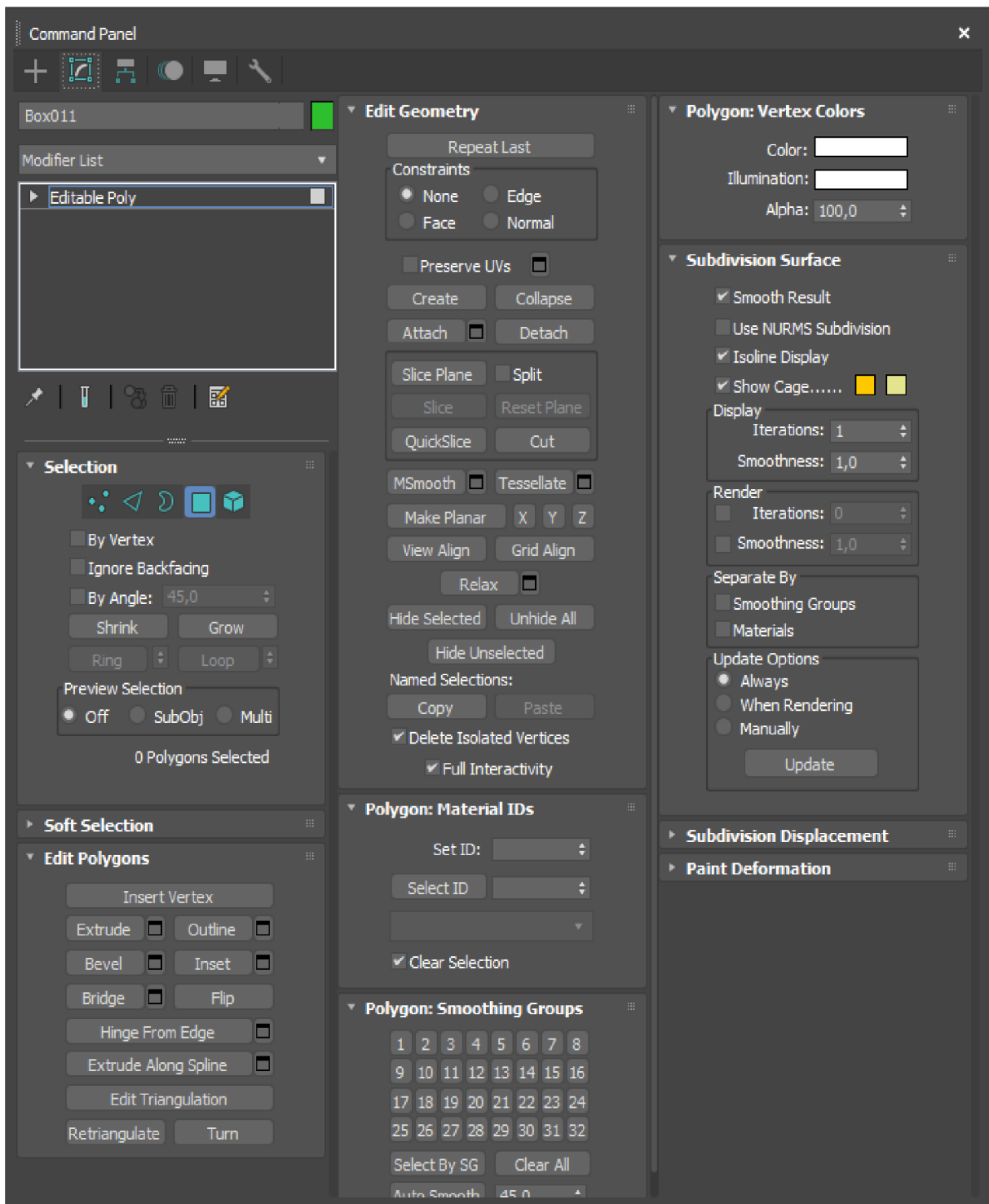


Obrázek 10: Příklad použití modifikátoru Displace

3.1.5 Modifikátor Edit Poly

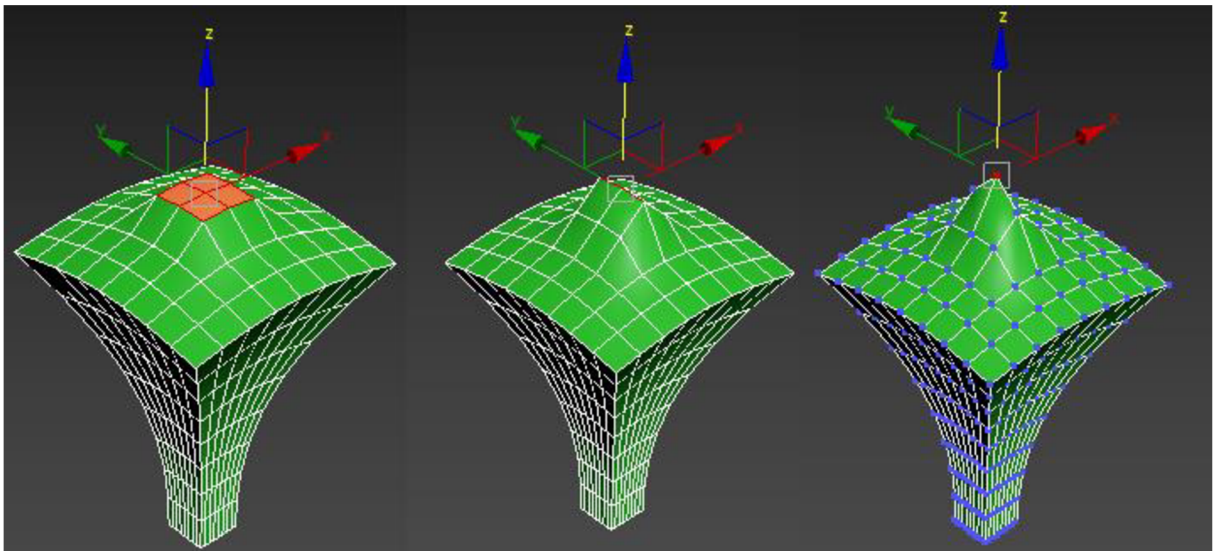


Modifikátorem, který zasluhuje naši zvýšenou pozornost je Edit Poly. Jestliže žáci doposud pracovali, modifikovali objekt jako celek, tento modifikátor poskytuje explicitní nástroje pro úpravy různé úrovně dílčích objektů vybraného objektu, důrazně zmíníme: **vertex** (vrchol), **edge** (hranu) a **polygon** (mnohoúhelník). Žák musí důsledně pochopit, že geometrie objektu je tvořena sítí mnohoúhelníků (nejčastěji tří nebo čtyřúhelníků). Mnohoúhelníky se dále sestávají z hran a vrcholů a tyto sub-objekty můžeme označit a pracovat s nimi. Jestliže tedy tento modifikátor na objekt aplikujeme, obvykle nejprve v rozbalovacím menu „Selection“, vybereme typ sub-objektu s nímž budeme pracovat. V našem případě to bude polygon. V závislosti na této volbě 3ds max zobrazí další možnosti nastavení, které se týkají právě našeho výběru. Dostupné funkce a parametry se budou tedy v závislosti na našem výběru lišit. Výsledek naší volby ilustruje obrázek č. 11. Jestliže čtenář z uvedeného obrázku tuší stoupající obtížnost na pochopení funkcionality právě tohoto rozsáhlého modifikátoru, musíme konstatovat, že je tomu tak. Postupně však i základní taje tohoto modifikátoru odhalíme.



Obrázek 11: Pohled na základní funkce modifikátoru Edit Poly

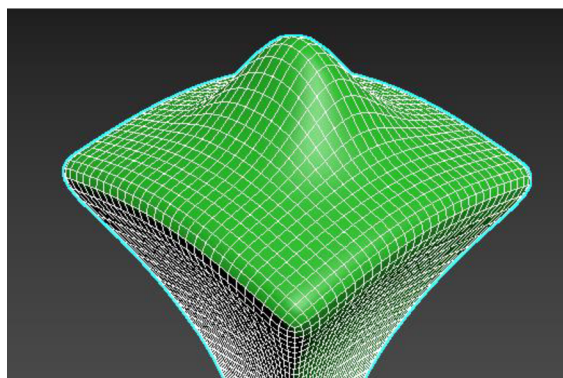
S aplikovaným (přiřazeným) modifikátorem k požadovanému objektu a s označenou volbou „polygon“, můžeme pomocí myši vybrat požadovaný polygon nebo skupinu polygonů (s podrženou klávesou ctrl), přímo v pracovním výřezu. Správný výběr čtveřice polygonů,



Obrázek 12: *Sub-objekty modelu*

dvojice hran a jednoho vertexu, jejich základní transformaci v prostoru posunem v ose z, ilustruje postupně zleva doprava obrázek č. 12.

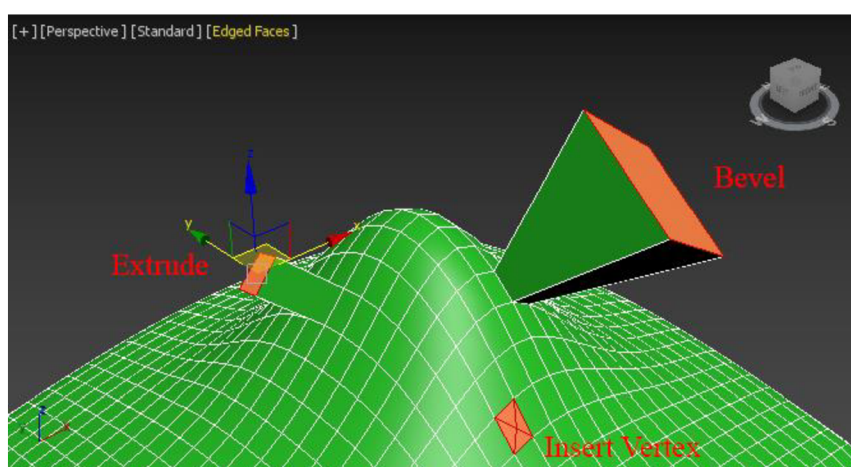
Samotný posun v ose z, byl proveden pomocí pomocného objektu gizmo. Jedná se o pomocný objekt, v tomto případě určený pro posun selektovaného objektu nebo objektů v prostoru. Žák musí ihned z počátku pochopit, že gizmo se objeví automaticky po našem výběru segmentů. S vybranými segmenty můžeme provádět také další typy základní transformace, tedy rotaci a změnu velikosti. Přepínání mezi jednotlivými transformacemi, je účelné provádět pomocí klávesových zkratk: W(posun), E (rotace), R (změna velikosti). Jestliže obrázek č. 12 vyvolal v čtenáři dojem, že za pomoci tohoto modifikátoru, můžeme vytvářet pouze nevyhlazené povrchy objektů, obrázek č. 13 ilustruje aplikovaný příkaz MSmooth (Mesh Smooth), který naše obavy mírní. Příkaz MSmooth (také samostatný modifikátor MeshSmooth), vyhlazuje geometrii v naší scéně pomocí několika různých metod. Umožňuje nám však především rozdělit geometrii interpolací úhlů nových ploch v rozích a na okrajích a aplikovat jednu vyhlazovací metodu na všechny plochy objektu nebo pouze



Obrázek 13: *Aplikovaný MeshSmooth*

na vybranou část. Pozorovatelným výsledkem MeshSmooth, je zaoblení rohů a hran, jako by byly hladce hoblovány. Pomocí parametrů MSmooth můžeme kontrolovat velikost a počet nových ploch a jejich vliv na povrch objektu.

Mezi další, pro žáky důležité příkazy sub-objektu polygon, jmenujme nejpoužívanější - „Insert Vertex“. Zvoleným příkazem a následným výběrem polygonu, vložíme uvnitř polygonu nový vertex a tím polygon rozdělíme na čtyři polygony. Výsledkem je možnost pracovat na síti s větším detailem. Příkaz Extrude, vysune selektované polygony ve směru normálového vektoru. Podobného účinku docílíme příkazem Bevel, pouze s tím rozdílem, že u vysunovaného



Obrázek 14: Základní funkce modifikátoru Edit Poly

polygonu, budeme mít kontrolu nejenom nad hloubkou vysunutí ale také rozměrem polygonu samotného. Účinky jmenovaných příkazů ilustruje obrázek č. 14.

V režimu hran, můžeme použít také originální příkazy, například Bridge, který vytvoří „most“ z polygonů mezi dvěma zvolenými (otevřenými) hranami. Příkaz weld svaří více hran do jediné. Příkaz Cut, vytvoří novou hranu mezi dvojicí vybraných hran. Nejčastějším příkazem režimu vertex je zřejmě weld, který spojí dva a více vertexů v jeden.



Některé příkazy jsou shodné napříč dvěma nebo více typy segmentů, jejich efekt je ale pochopitelně odlišný a je nezbytné, aby žák všechny varianty vyzkoušel a vytvořil si účinnou strukturu představ o použití jmenovaných, základních příkazů. Modifikátor Edit Poly je mimořádně silným tvůrčím nástrojem, který nám dovoluje vytvářet i ty nejsložitější objekty, ať již přírodního nebo technického původu, které obvykle modelujeme pomocí technologie NURBS. Edit Poly nabízí natolik širokou nabídku příkazů, jejich parametrů a jejich vzájemné interakce, že jeho dobré ovládnutí je opravdu

nesnadným úkolem. V kontextu práce s ním je hodina nebo den naprosto bezvýznamná jednotka. Žáci s nedostatkem trpělivosti mohou mít problém.



Obvyklý postup, kterak si možnosti modifikátoru osvojit, můžeme vystihnout v několika (obvykle ve třech) etapách s narůstající obtížností. Zkoušení jednotlivých příkazů a jejich účinků na jednoduchém 3D objektu. Dále vytváření jednoduchých objektů a v poslední etapě žák vytváří složité objekty, často s využitím blueprintů⁷ nebo dokonce bez jejich pomoci. Tvůrčí mistrovství potom spočívá v schopnosti kombinovat možnosti celé nabídky modifikátorů aplikace 3ds Max a jejich parametrů. Výsledkem je dokonalý model. Požadavky na žáka kladené, musí respektovat požadavky oboru a žáka nepřetěžovat. V rámci modifikátoru Edit Poly je to velmi snadné.

3.2 Modelování objektů do hry „Fantasy golf“

3.2.1 Model míčku



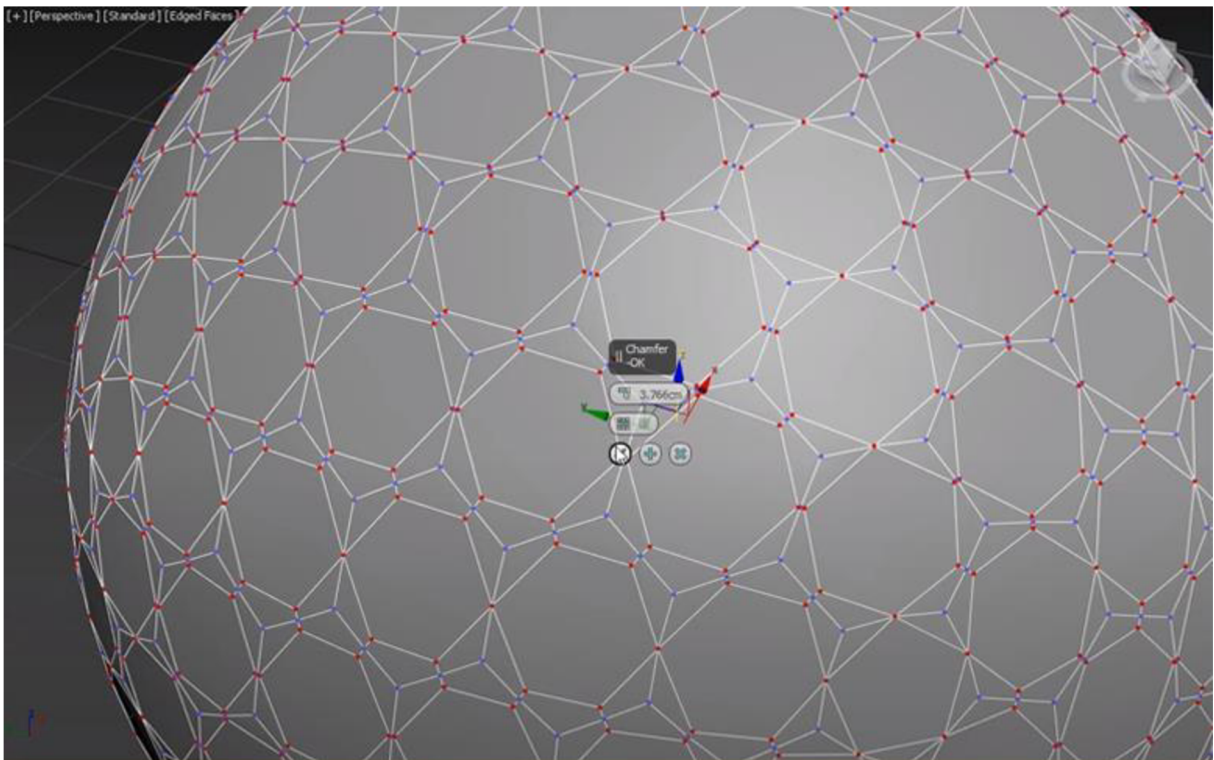
Golfový míček je ústřední neživou „postavou“ naší hry. Kamera na tento objekt bude v podstatě stále směřovat svůj objektiv, a ještě z bezprostřední vzdálenosti. LOD⁸ musí být tedy vysoký a na počtu polygonů objekt ztvárňující šetřit právě nemusíme a nebudeme. Přestože se jedná o objekt na první pohled poměrně jednoduchý, jeho vytvoření nás ale něco přemýšlení stát bude. Klíčovým je výběr základního objektu, nejlépe svým tvarem míčku. Tentokrát ale také zohledníme topologii sítě výchozího objektu. Bez dalších okolků se pustíme do podrobného postupu procesu tvorby.

1. Create panel > Geometry > Standard Primitives > GeoSphere. Po vytvoření GeoSphere, změníme následující parametry: Radius: 40, Segments: 6.
2. S označeným objektem: Modify panel > Modifier List > TurboSmooth.
3. S označeným objektem: Modify panel > Modifier List > Edit Poly. V nastavení modifikátor přepneme výběr segmentů typu vertex v paletce Selection. Vybereme dva vertexy, které jsou pozicí uvnitř zřetelného šestiúhelníku. Výběr všech vertexů uvnitř všech šestiúhelníků můžeme automatizovat a práci si usnadnit v RibbonMenu (velké vodorovné menu). Zde vybereme v rozbalovacím menu „Modify Selection“ možnost Similar/Edge Count. Takto vybereme všechny požadované vertexy.

⁷ Blueprint je podrobný plán nějakého modelu. Obvykle pořízený z více pohledů, například z předního a horního.

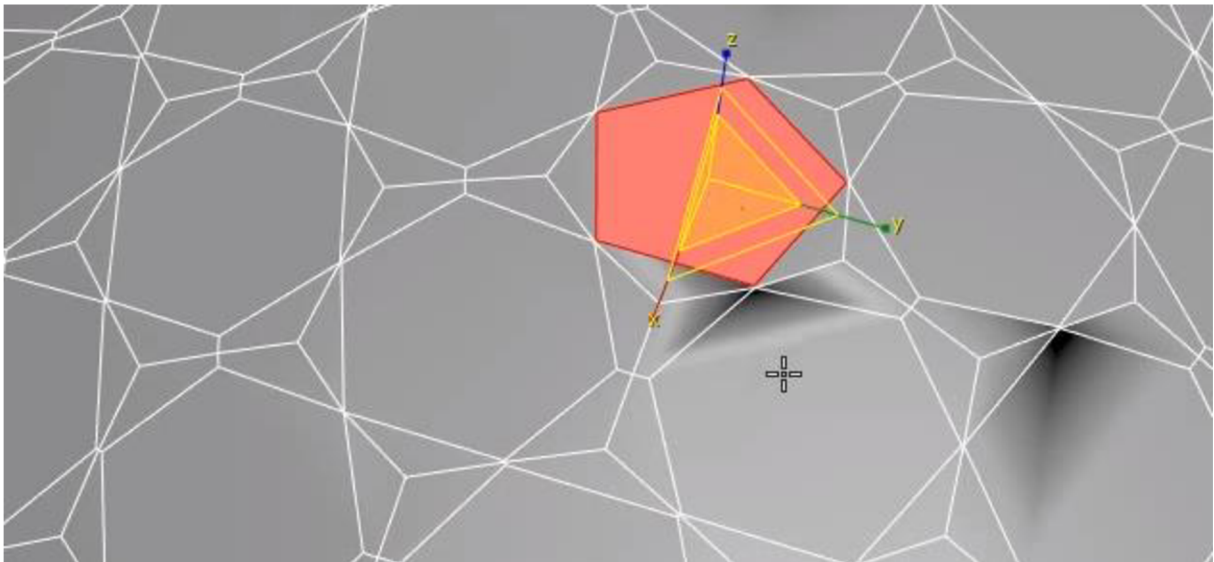
⁸ Level of Detail (LOD), udává úroveň složitosti, tedy hustoty sítě 3D modelu. Čím více je objekt v pozadí, tím méně detailu obvykle musí mít, při zachování požadované realismy výstupu.

4. V nastavení modifikátoru Edit Poly, vybereme příkaz chamfer a pomocí transformačního gizma nově vzniklé skupiny polygon, umístíme jako na obrázku č. 15.



Obrázek 15: *Nastavení funkce Chamfer*

5. Některé z mnohoúhelníků, jsou vykreslovány v pracovním výřezu deformovaně, černým stínováním. Velikost těchto mnohoúhelníků je potřeba individuálně upravit. V paletce „Selection“ vybereme možnost “polygon” a tyto “zlobivé” mnohoúhelníky postupně vybereme. Každý polygon nepatrně zmenšíme, pomocí transformačního gizma. Pamatujete si ještě na klávesovou zkratku „r”? Situaci ilustruje obrázek č. 16.

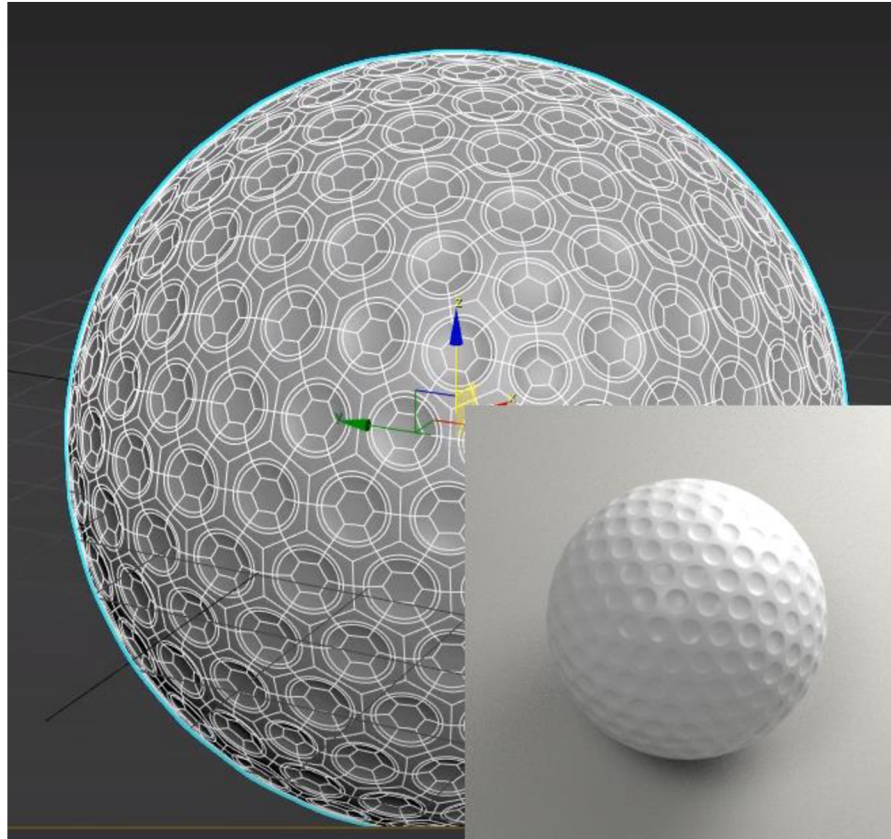


Obrázek 16: *Redukce velikosti polygonu*

6. Nyní se pokusíme vybrat všechny mnohoúhelníky, tolik charakteristické pro golfový míček tak, abychom na ně mohli aplikovat modifikátor pro vysunutí a míček dokončili. Vybereme jeden z polygonů. V Ribbon menu klikneme na rozbalovací nabídku „Modify Selection“ a u příkazu Similar, zatrhneme nejprve parametry „Topology“ a „Edge Count“ a příkaz potvrdíme.

7. S vybranými mnohoúhelníky aplikujeme v rámci modifikátoru Edit Poly, postupně tyto dva příkazy s odpovídajícími parametry:

Inset (Amount = 0.13), Bevel (Height = -0.45; Outline = -1.5). Takto vytvoříme typickou strukturu povrchu míčku. Její jemnější variantu zobrazíme přiřazením posledního modifikátoru TurboSmooth (parametricky přímočařejší varianta modifikátoru MeshSmooth) s parametrem Iterations = 2. Výsledný vzhled v pracovním výřezu by měl být podobný obrázku



Obrázek 17: Na míček aplikovaný *MeshSmooth*, vpravo vizualizován

č. 17. V pravém dolním rohu obrázku vidíme také jednoduchou vizualizaci objektu. Pomocí zdrojového souboru **golf_ball.max**, může žák výsledek porovnat.

3.2.2 Model terénu



Tvorba virtuálních světů, terénů, planet nebo pobřeží, má v počítačové grafice svoji historii. Nalézá své uplatnění v oblastech dopravních simulátorů a trenažerů, počítačových her a obecně v oblasti multimediálních projektů nebo virtuální realitě.

Dle Žáry (2004), tvorbu těchto přírodních a do značné míry náhodně vyhlížejících útvarů, členíme do dvou fází. Nejdříve generujeme hrubý tvar krajiny. Zde může jít o data reálná, například z družic nebo častěji získaná fraktální metodou. V druhé fázi, úpravou zdrojů z prvního kroku, dochází k aplikaci erozních algoritmů. Tento postup budeme následovat rovněž. Využijeme fraktálního základu s kombinací algoritmů eroze. Fraktální a erozní algoritmy, převedené do rastrového obrazu, budeme interpretovat do dvojrozměrné matice, v níž každý prvek reprezentuje odstín šedi, který odpovídá výškovému vysunutí v ose „Z“ (vzhledem k definované výšce nula). Pomocí naznačeného postupu, vytvoříme výsledný objekt terénu určený k dalším úpravám.

V našem případě musíme totiž doplnit ještě krok třetí. Fraktální metodou, do značné míry náhodně členěný povrch terénu, musíme v našem případě editovat v aplikaci 3ds Max a do takového terénu doplnit jamku pro míček a pozměnit cíleně lokální výškové charakteristiky terénu, aby byl terén nejen esteticky akceptovatelný, ale hra byla současně hratelná, atmosférická nebo přiměřeně obtížná. Terén musí proto být uvedeným způsobem upravován, a nakonec akceptován během testování hratelnosti hry.

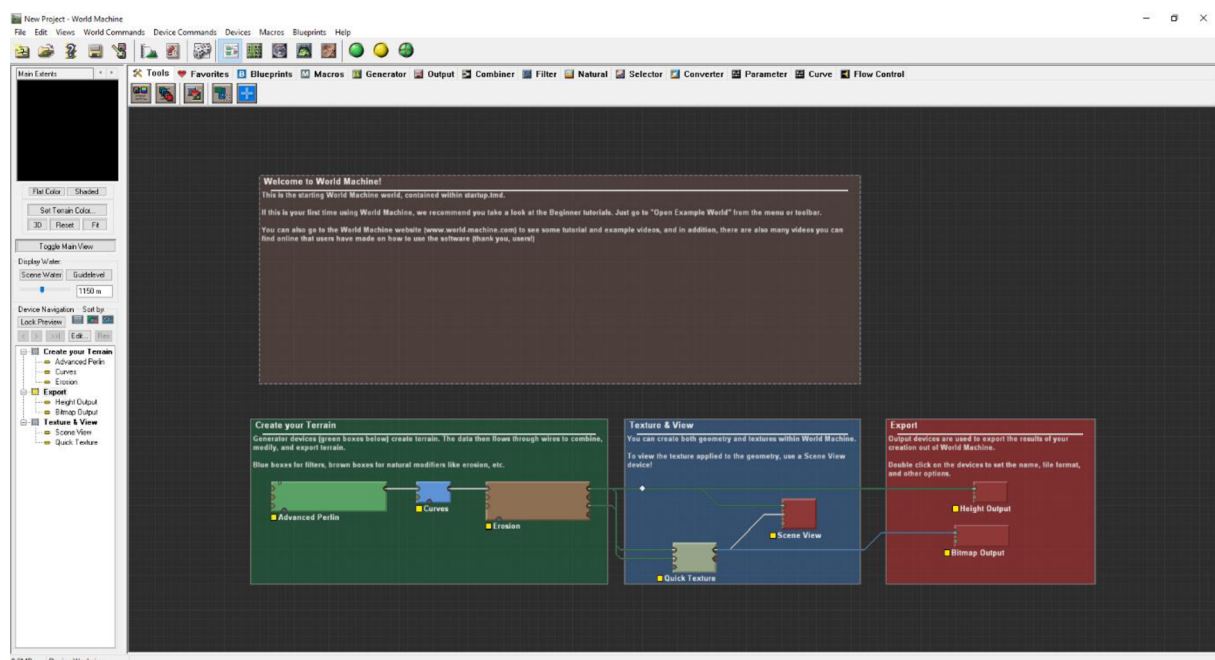
Vytvoření realistického, tedy přirozeně vyhlížejícího terénu, není právě snadným úkolem. Nebudeme vycházet z klasického terénu typického pro golf a můžeme si to dovolit, pracujeme přeci na fantastickém golfu. Klasický golfový terén a trávníček, vyměníme za terén hornatý a s údolím. Pro modelování geometrie terénu, budeme využívat postupu podobného z výkladu o Displace modifikátoru v aplikaci 3ds max. Modelování ale budeme realizovat v programu WorldMachine. Aplikace 3ds Max sice disponuje možností generování fraktálních, procedurálně generovaných map, trefně nazvaných smoke, spackle, swirl a podobně, jejichž kombinací lze skutečně terén vytvořit. Nedisponuje však erozním algoritmem a vzhledem k tomu, že naším cílem je vytvořit herní objekty, pokud možno co nejvíce věrné realitě, bude výklad věnovaný tvorbě vůbec nejsložitějšího objektu naší hry, realizován pro aplikaci WorldMachine.

WorldMachine, respektive jeho nynější verze číslo dvě, je nástupcem populárního profesionálního nástroje pro úpravu terénu WorldMachine. WorldMachine funguje na procedurální úrovni. Nevytváříme síť polygonů v rámci obvyklého pracovního postupu - polygon za polygonem, což je typický postup například pro 3ds max ale vybíráme procedurální algoritmy, které umě kombinujeme a ty vytvářejí finální terén. Výsledkem našeho snažení, může tedy být vytoužený polygonální model terénu. Nepracujeme však na úrovni polygonů ale s procedurálními mechanismy aplikace. Tento postup je zdrojem síly programu ale také uživatelské složitosti.

Při práci ve WorldMachine, je naší rolí určit celkový vzhled a vlastnosti terénu a poté umožnit generátorům a přidavným efektům, vytvořit a podrobně popsat vzhled našeho světa. Tento způsob práce a myšlení na vyšší úrovni vyžaduje určité přivyknutí, ale odměny jsou obrovské: můžeme vytvořit působivé a velmi rozsáhlé terény s relativně malým množstvím práce. Jakmile totiž vytvoříme základní terén, který se nám líbí, můžeme použít stejný soubor algoritmů a jejich parametrů na další a další podobné terény nebo celé rozsáhlé světy.

WorldMachine pracuje s termínem svět. Svět je prostě soubor kroků použitých k vytvoření terénu. Terén se generuje v algoritmických obměnách „do nekonečna“, navazuje na sebe a tvoří tak právě „svět“. Uživatel poté v náhledu snadno vybere určitou část světa, kterou poté může exportovat jako terén pro své účely. Kroky, které vedou k vytvoření takového světa, z něhož si vybereme ten náš oblíbený kousek terénu, vedou přes použití několika Devices (zařízení). Každé z těchto „zařízení“, provede nějakou operaci na světě, jehož jsme tvůrci. Pracovní náhled v levém horním rohu GUI je místem, kde můžete vidět proměny světa v závislosti na použitých zařízeních a jejich čtených parametrech.

Když poprvé spustíme WorldMachine, zobrazí se GUI, ilustrující obrázek č. 18.



Obrázek 18: GUI programu WorldMachine

Nejdůležitějším je, aby žák pochopil, jak každý ze základních konceptů souvisí s tím, co můžeme vidět na obrazovce.

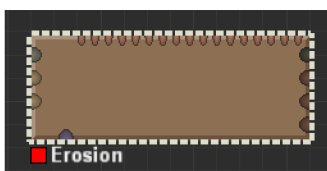
- Svět je tvořen zařízeními, která je možno vkládat na pracovní plochu černé barvy s mřížkou. Naše ilustrační obrazovka výše, obsahuje příklad zařízení, umístěných na pracovní ploše - zelené, modré, hnědé, šedé a červené barvy (mající tvar obdélníků).
- Skupiny zařízení je možno pro přehlednost vkládat do kontejnerů v podobě barevných, větších obdélníků. Kontejner má vždy přiléhavý popisek (v levém horním rohu), například „Export“. Na obrázku výše tyto bloky vidíme. Konkrétně tři - zelený, modrý a červený. V případě většího počtu zařízení nám umožní použití kontejneru větší přehlednost a také jedním

kontejnerem je možno po jeho označení pohybovat po pracovní ploše. Najednou tak měníme pozice všem zařízením, která jsou v kontejneru umístěna.

- Barevné jemné čáry mezi zařízením, jsou „dráty“, které ukazují pořadí, ve kterém jsou akce na terénu prováděny, tedy jejich návaznost a kombinace.
- WorldMachine zobrazuje v levém horním rohu náhled účinku aktuálně vybraného zařízení.
- Chceme-li generovat svět v lepším, tj. v detailnějším a podstatně větším náhledu, musíme v horním panelu nástrojů, kliknout na možnost „Build“ (sestavit). Při provádění výpočtů, program zobrazí dialogové okno se zprávou o průběhu zpracování jednotlivých zařízení. Jakmile je výpočet světa dokončen, můžeme si terén prohlédnout ve slíbeném větším detailu klávesou „F8“ nebo můžeme terén exportovat jako model, což je také posledním krokem našeho snažení ve WorldMachine. V levém postranním panelu nástrojů, můžeme také vybrat pouze jedno dílčí zařízení a prohlédnout si jeho účinek na terén.

Nyní se podíváme detailněji na pravidla správného pracovního postupu se zařízeními a jak tato zařízení vlastně fungují. Zařízení představuje akci. Ve WorldMachine jsou zařízení vizualizována jako malá barevná políčka na pracovní ploše - na pozadí s mřížkou, kam je přetahujeme z horního menu, které je dále strukturováno do tematických záložek.

Zařízení má porty. Port je prostě místo, kam data z jiných zařízení vstupují a dále interagují nebo vystupují a slouží zařízení navazujícímu, které tato data dále zpracovává. Porty lze tedy rozdělit do dvou hlavních kategorií: vstupy a výstupy. Vstupy jsou na levé straně zařízení a přenášejí data do zařízení, s nímž toto zařízení bude dále pracovat. Poté, co dojde k akci zařízení, jsou data výstupem na pravé straně připraveny k napojení do zařízení navazujícího. A konečně, některá zařízení mají vstupní port masky, což nám umožňuje ovládat oblast působení zařízení. Toto maskování funguje velmi podobně, jako žákům dobře známé maskování z rastrových editorů. Ve WorldMachine je ještě jeden typ portu, který se nazývá port parametrů. Ty jsou umístěny podél horní hrany zařízení a umožňují



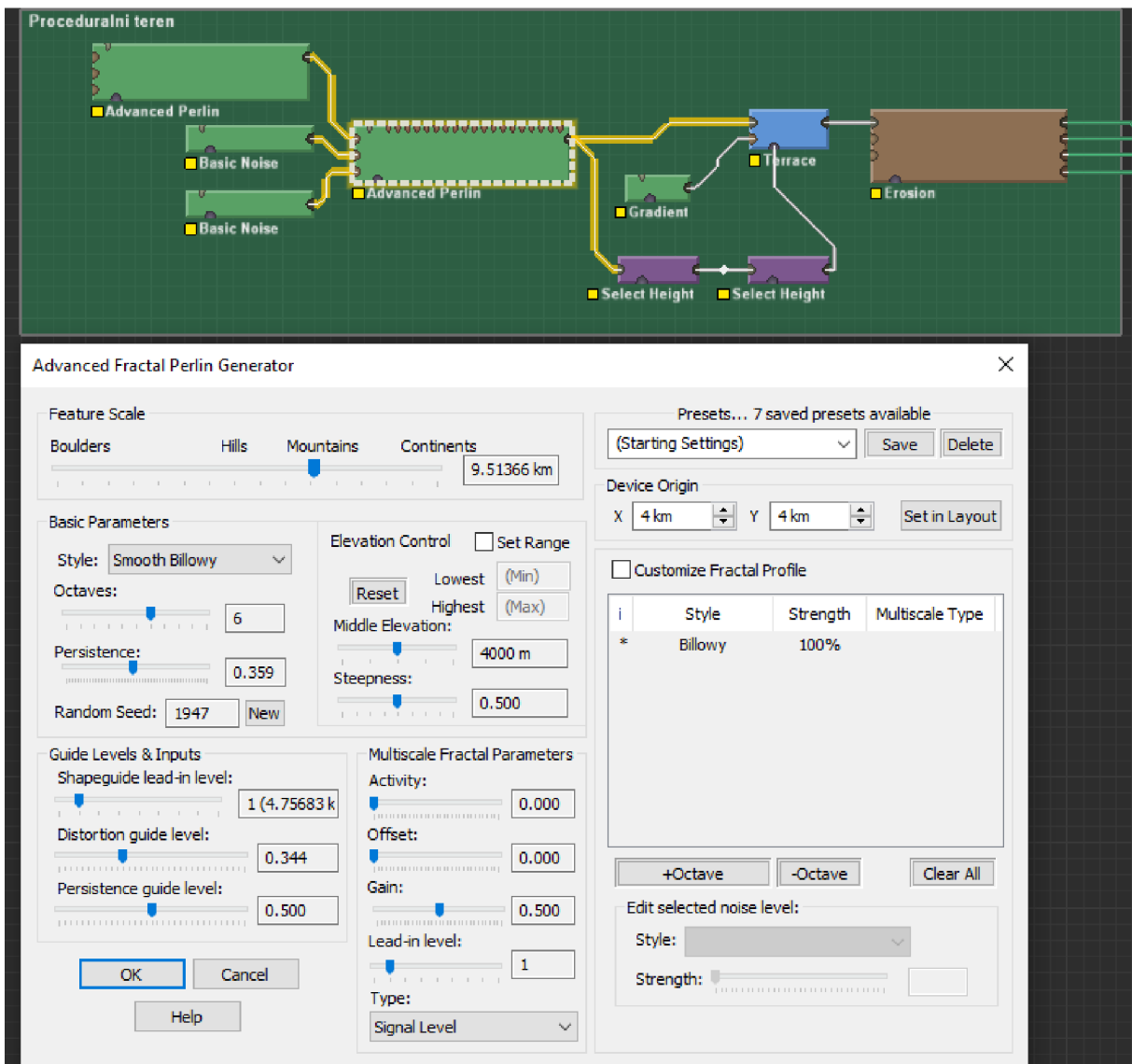
Obrázek 19: Příklad zařízení

pokročilým uživatelům upravit parametrická nastavení, například funkčním předpisem. Obrázek č. 19, ilustruje porty všech popisovaných typů.

Každý typ zařízení má jinou odpovědnost za výsledek tvorby. Popíšme žákům stručně nejdůležitější kategorie zařízení, kdy každá kategorie je charakteristická, přiřazenou barvou k zařízení. V menu „Generator“ nalezneme skupinu zelených zařízení. Pomocí nich vytváříme základní procedurální mapy. Ty jsou odvozeny od Perlinova šumu⁹ (Basic noise, Advanced Perlin). Můžeme však do systému importovat náš vlastní terén, jako rastrový obraz z jakýchkoli jiných zdrojů (v odstínech šedi). Generovat můžeme také bohatou nabídkou přechodů (Gradient). Z Menu „Filter“, můžeme na pracovní plochu přetáhnout zařízení modrá, která často navazují na zelená zařízení. Nejpoužívanějším zástupcem této kategorie je patrně zařízení „Terrace“, které vytváří stupňovité členění terénu v ose Z. Pomocí tmavě modré kategorie v menu „Combiner“, můžeme snadno kombinovat více terénů nebo generátorů zvoleným způsobem. Menu „Natural“ a jeho hnědá zařízení, konkretizují účinek Perlinova šumu v konkrétnější zpodobnění terénu, na který jsme zvyklí v reálném světě. Zařízení „Erosion“ v této kategorii je pro naši tvorbu nejdůležitější, vytváří reálně vyhlížející vrásnění terénu. Fialová zařízení v menu „Selector“, slouží, jak jinak než k selekci charakteristických prvků terénu. Toto selektování nám výrazně dopomůže k vytvoření hezkého materiálu pro náš terén. Selekcí obvykle provádíme pomocí úhlů v terénu, interval výšky terénu a podobně. Na konci řetězce tvorby, bývají červená zařízení z menu „Output“. Připojením zařízení „Scene View“, můžeme generovat detailní náhled terénu přímo v prostředí WorldMachine. Zařízení „Mesh Output“ vygeneruje výsledky našeho snažení, coby 3D polygonální model, pro další použití v jiných aplikacích. Export terénu můžeme provést také do rastrového obrazu v odstínech šedi, který můžeme použít například v modifikátoru „Displace“, programu 3ds Max.

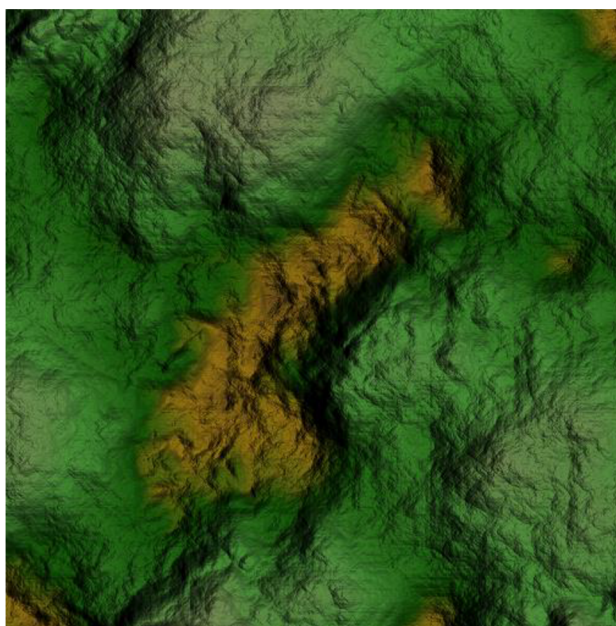
Nyní by měli mít žáci dostatek informací, potřebných pro tvorbu jednoduchého terénu a experimentování s jeho parametrickým nastavením. Obrázek č. 20 ilustruje konkrétní příklad, který jsme vytvořili pro účel tvorby naší hry a stručně jej popíšeme. Zobrazena je pouze jedna polovina pracovní plochy, která zabezpečuje nastavení procedurálního modelování terénu. Zdrojový soubor je přílohou tohoto textu (**terrain.tmd**), a proto nebudeme komentovat každý ze stovek parametrů k výsledku vedoucí. Budeme charakterizovat pouze ty parametry a postupy, které obecně uplatňujeme téměř ve všech typech terénu. Dvojice zařízení „Advanced perlin“ a „Basic Noise“, vytvářejí procedurální základnu pro náš výsledný terén.

⁹ Perlinův šum je metodou generování náhodně vyhlížejícího grafického šumu, kterou roku 1985 publikoval Ken Perlin.



Obrázek 20: Zařízení generující herní terén

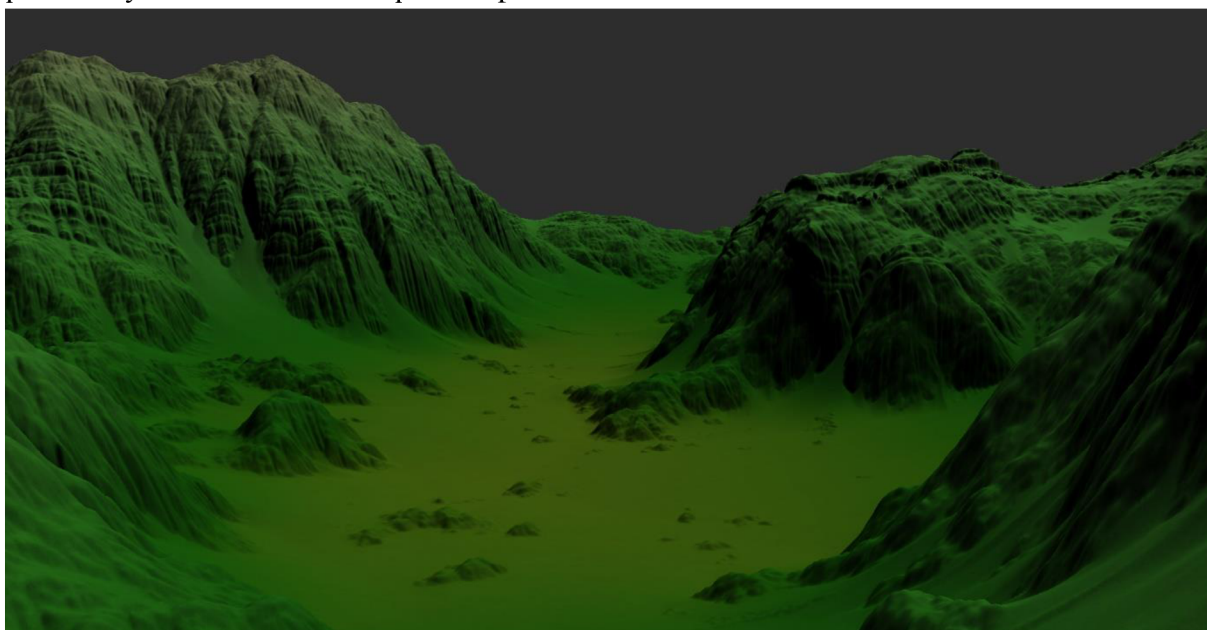
Podle Žáry (2004) by měla optimální šumová funkce poskytovat generování bílého šumu v našem prostředí WorldMachine na černém pozadí. Praktické využití této funkce ale začíná až jejím skládáním, které dokáže vytvářet náhodně vyhlížející procedurální mapy – zárodky „světů z šumu zrozemých“. Obrázek č. 21, ukazuje výstup skládaného Perlinova šumu. Podle stupně generované šedi, tento šum již obsahuje materiál s barevným přechodem. Kliknutím na naše zařízení „Advanced perlin“, dojde k zvýraznění zeleného obdélníku představující toto zařízení a současně zobrazíme okno s nastavitelnými parametry tohoto zařízení.



Obrázek 21: *Perlimiv šum, kolorovaný*

Stručně parametry popíšeme, abychom žákům usnadnili začátky s programem. Parametr „Feature scale“, určuje celkový charakter generovaného světa. Jeho měřítko, a tedy vhodné použití. Pohybujeme se v intervalu kopců až po celé kontinenty. V části „Basic parameters“, nastavujeme styl generovaného fraktálního šumu. Dochází k tvorbě „konzistence“ terénu. Od hrudkovitého až po hřebenovité ztvárnění. Další nastavení ovlivňují výsledek zařízení mnohem méně významně, doporučujeme přesto, nechat žákům prostor pro jejich vyzkoušení. Druhým krokem je použití zařízení „Terrace“. Implementujeme tedy do našeho šumového základu terasovitost, která je v přírodě poměrně častá a dodá do výsledku realismus. Do tohoto zařízení však ještě vstupuje zařízení „Gradient“, které nám pravidelnost terasovitosti „rozbíjí“ a přiblíží více tendenci přírody, „působit nahodile“. Třetím krokem je připojení a nastavení zařízení „Erosion“. Přestože má také nastavení tohoto zařízení mnoho parametrů, které lze nastavit a kombinovat, zařízení nabízí možnost několika použitelných předvoleb, které pro počátky tvorby postačují. Prozatím je to k programu WorldMachine vše. Významnou zmínku si však ještě zaslouží v kapitole určené tvorbě textur, respektive masek. Právě tyto, včetně výsledného objektu terénu, budeme exportovat z programu k navazující tvůrčí práci. Přestože je evidentní, že autoři programu vyvinuli maximální úsilí, aby vytvořili „generátor“ terénu uživatelsky přívětivým, bude začínajícímu uživateli trvat mnoho hodin, než se naučí s programem dobře pracovat. Mnohdy nezbude než parametrická nastavení zkusit osvědčenou metodou „pokus-omyl“. Ani tato metoda však není bezchybnou. Opětovné generování terénu po změně jednotlivých parametrů, může trvat i mnoho minut, a tak nezbyvá než se obrnit značnou dávkou trpělivosti a vyzbrojit

výkonným výpočetním systémem. Obrázek č. 22 obsahuje finální objekt terénu, vykreslen přímo vykreslovacím systémem WorldMachine. Hora na levé části obrázku je charakteristická jemně stupňovitou topologií, díky použitému zařízení „Terrace“ a jeho nastavení. Nežádoucí pravidelnost teras, jsme „rozbili“ použitým zařízením „Gradient“. Z hlediska použitelnosti terénu do naší a vůbec do drtivé většiny her je stav údolí. Ten je díky zařízení „Erosion“ poměrně hladký, pouze s několika málo kameny nebo vrásněním. To nám umožňuje, aby se po takovém terénu pohybovala herní postava nebo v našem případě kutálel golfový míček. Žák by měl chápat dvojici podmínek správné tvorby. Měl by být schopen realizovat esteticky svůj vlastní tvůrčí záměr, například z jednoduchého náčrtku a současně by měl být terén ve hře použitelný – a to s minimem úprav v aplikaci 3ds Max.



Obrázek 22: *Výsledný objekt terénu*

Dalších několik málo objektů (strom, brána), které jsou ve hře zobrazovány, modelovat na tomto místě nebudeme. Postupy, které je zapotřebí použít, byly již zmíněny v teorii modelování nebo v typických ukázkách jednotlivých modelovacích modifikátorů 3ds Max. Čtenář si tak vymodeluje vlastní verze objektu strom a brána nebo použije modely, které jsou součástí, přílohou této metodické příručky. Studium těchto přiložených modelů může přímo posloužit jako edukační materiál pro naše žáky. Postup tvorby základních objektů nezbytně nutných pro realizaci hry, tj míček a terén, jsme však objasnili čtenáři detailně.

3.3 Materiálové inženýrství



Kdyby bylo našim cílem naučit žáky modelovat základní objekty, například pro tisk modelů na 3D tiskárně, měli bychom v tomto okamžiku již splněno.

K tvorbě multimediálního projektu 3D hry však budeme potřebovat, osvojit si další znalosti, a sice základy tvorby materiálů, které objektům následně přiřadíme. Bez této kapitoly by byly naše modely spojitě šedivé nebo měly jinou výchozí, spojitou barvu. To by však bylo na multimediální zážitek docela málo, a proto pojďme s chutí na další náročnou ale zajímavou kapitolu.

Dříve než se pustíme do tvorby tohoto materiálů, objasníme tři základní termíny, které budou dále v textu často zmiňovány. Konkrétní tvorbu materiálu si dále v textu ukážeme na nejsložitějším objektu ve hře, tedy terénu.

Pod termínem **textura**, rozumíme běžně rastrový zdroj – obrázek (textura kůry stromu, ovoce..), který je pouze jednou součástí všech činitelů, kteří se na výsledném vzhledu objektu podílejí.

Termín **mapa** si můžeme představit opět jako „texturu“, tedy rastrový obrázek nebo procedurálně generovaný obraz, který je ale umístěn v konkrétním kanálu, často jednom z mnoha, která spoluutváří výsledný vzhled. Tento zmíněný kanál může například zajišťovat nerovnost povrchu objektu, potom hovoříme o bump mapě a podobně. Termín mapa v sobě tedy zahrnuje logicky uspořádanou texturu, jako prostý rastrový obraz, který však můžeme dále modifikovat, například opakovat obraz v ose x a podobně. Účelem mapy je ovlivnit konkrétní charakteristiku povrchového vzhledu objektu (hrbolatost, barva, odlesky a podobně).

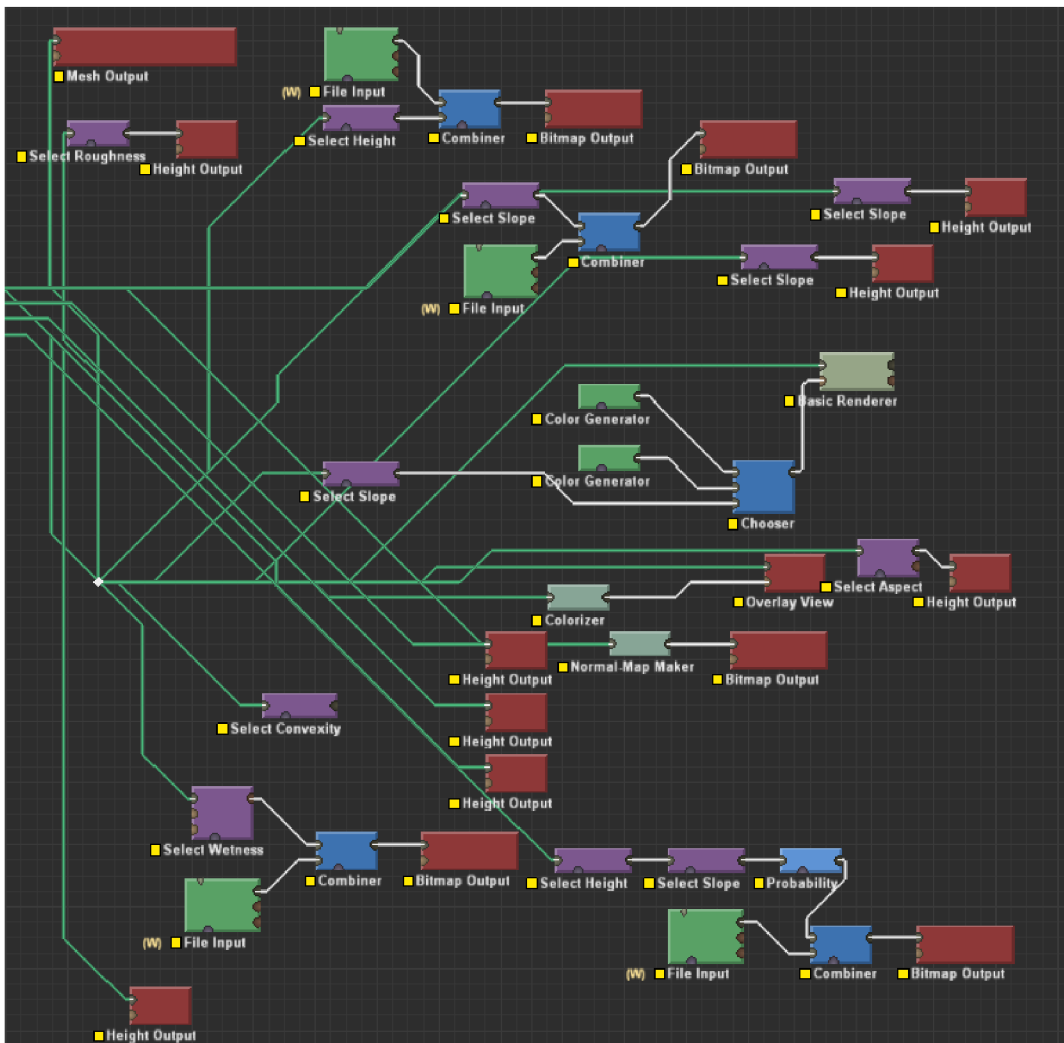
Termín obsahově nadřazený textuře ale také mapě je **materiál**. Materiál se skládá obvykle z několika map a tvoří tak komplexní způsob, jakým bude výsledný materiál vykreslován systémem na 3d objektu. V našem případě vykreslovacím systémem programu 3ds Max nebo později Babylon.js - WebGL. Vytváříme-li tedy například materiál plastu, je našim cílem vytvořit takovou kompozici map, která při určitém nastavení světla, stínů a dalších vnějších vlivů ve scéně, povede k realistickému vykreslení tvořeného materiálu plastu.

3.3.1 Export textur z WorldMachine

V našem úsilí, kterak vytvořit dobrý materiál pro náš herní terén, započneme v programu WorldMachine. Zde totiž můžeme získat cenné textury nebo spíše masky, z nichž

textury vytvoříme a ty potom použijeme do map výsledného materiálu, komponovaného v aplikaci 3ds max. Pojdme ale vše prozkoumat pěkně popořadě.

Na obrázku č. 23 vidíme zařízení, díky kterým zmíněné masky získáme. Bude však lépe, když si také žáci otevrou zdrojový soubor: **teren.tmd** a budou jednotlivá nastavení procházet s učitelem. Všem těmto zařízením, předchází jedno zařízení „Erosion“ (hnědé barvy). Toto zařízení poskytuje tedy svoji čtveřici výstupních portů typu „Heightfield“.



Obrázek 23: Soustava zařízení pro tvorbu textury terénu

Odtud získáme rastrové masky v odstínech šedi, které selektují některou z charakteristik našeho pohoří. Může se jednat místa nížin, štítů pohoří nebo oblasti, kde docházelo v průběhu času k erozím atp. Jsou to tedy místa, která je potřeba oddělit, neboť v přírodě jsou rovněž odděleny, například jinou barvou. Volitelně můžeme na výstup zařízení „Erosion“ napojit vstupní port fialových zařízení. Ta nám umožní ještě větší kontrolu nad selektováním zvláštností našeho terénu. Jestliže jsme použili výstupní port zařízení „Erosion“ pro selekci erozních míst, potom díky fialovému zařízení, můžeme tuto selekci ještě více konkretizovat například nadmořskou

výškou a vybrat tak erozivní plochy například pouze v údolí nebo ta erozivní místa, která svírají se sousedícími polygony specifikovaný úhel. WorldMachine splní téměř každou naši představu. Když jsme s maskou spokojeni, napojíme výstupní port zařízení „Erosion“ nebo výstupní port některého z fialové kategorie zařízení se vstupním portem červené barvy, například „Height Output“. Rozkliknutím zmíněného červeného zařízení už jenom nastavíme umístění pracovní složky našeho projektu a masku vygenerujeme tlačítkem „Write Output to Disk“ k navazujícímu pracovnímu procesu. Konečně nastal také čas, abychom exportovali terén, jako síť polygonů. Provedeme to snadno. První výstupní port zařízení „Erosion“, napojíme na vstupní port červeného zařízení „Mesh Output“. V jeho nastavení je důležitý zejména parametr, určující počet polygonů našeho terénu. Vyšší hodnota je předpokladem většího detailu, realismu objektu. Hodnota musí reflektovat schopnosti hardware, pro které je hra určena (PC, mobilní zařízení a podobně). V našem případě jsme nastavili hodnotu 0.5 milionu polygonů.



Jestliže chceme, aby žák vytvářel skutečně dobře vyhlížející masky, které vytvoří později realistický materiál, doporučujeme inspirovat se referencemi nejrůznějších terénů a pozorovat jejich konkrétní přírodní charakteristiky.

Zda tvoří pohoří terasy, jakou míru eroze pozorujeme, míra vrásnění a podobně. Vzhledem k tomu, že kvalitní tvorba 3D obsahu je mnohahodinovým maratónem před obrazovkou, doporučujeme, udělat si také s žáky čas a vyrazit přímo do přírody a nejlépe s fotoaparátem. Prozkoumat inspirativní zákoutí naší vlasti a poté nabytou inspiraci proměnit v úžasnou, novou „realitu“, vytvořenou na počítači.

3.3.2 Tvorba textury terénu v Adobe Substance Painter



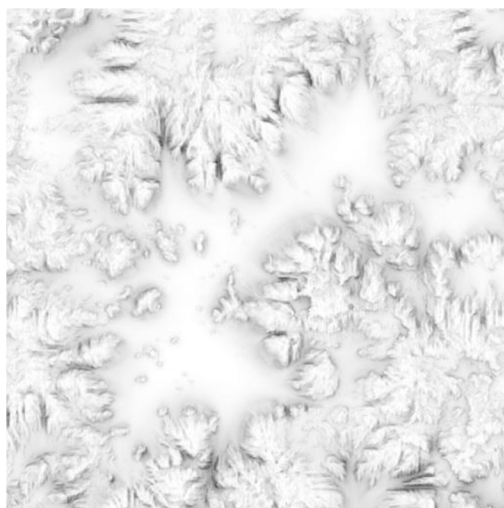
V Programu Substance Painter, vytvoříme nyní finální vzhled textur, který následně v programu 3ds max dopracujeme v konečný materiál terénu. Velká výhoda tohoto produktu, používaného profesionálními tvůrci pro malování textur, je jeho interaktivita. Každá akce, kterou provedeme na objektu, který do programu importujeme, například tah štětcem nebo aplikace naší masky z WorldMachine, provádíme názorně přímo na objekt a výsledek vidíme v reálném čase. Kreslení textur můžeme sice provádět a běžně se tak děje v aplikacích pro tvorbu rastrového obrazu, příkladem Adobe Photoshop nebo Gimp. Chybí zde ovšem zmíněná interaktivita. Nekreslíme názorně přímo na objekt a výsledky naší tvůrčí kompozice musíme nejdříve uložit v programu, následně načíst v 3D programu a v případě nutných úprav tak stále dokola. Efektivita takové práce je samozřejmě rozporuplná, přestože i mezi profesionálními tvůrci poměrně častá. Následujícím

detailním postupem, představíme nejenom postup tvorby textur pro náš objekt terénu, ale cíleně volíme takový postup, který je pro žáka, coby začátečníka, dobře srozumitelný a použitelný okamžitě a obecně na kterýkoli jiný objekt.



1. Zvolíme menu File > New Project. V zobrazeném okně, klikneme na tlačítko “Select”. V tomtéž okně klikneme na tlačítko “Add”, kterým do nového projektu Substance Painter, přidáme obrázky masek vytvořených dříve v programu World Machine.

2. V pravém postranním panelu vybereme záložku “TEXTURE SET SETTINGS”, ponecháme výchozí nastavení a klikneme na tlačítko “Bake Mesh Map”. Program na základě 3D dat z importovaného objektu terénu, vytvoří několik map, které můžeme při tvorbě výsledných textur využít. Pro naše účely zmíníme nejpraktičtější z nich - “ambient occlusion” mapu. Tato mapa je vhodná pro emulaci vzhledu skutečného globálního osvětlení, které vypočítává rozsah,



Obrázek 24: *Ambient Occlusion*

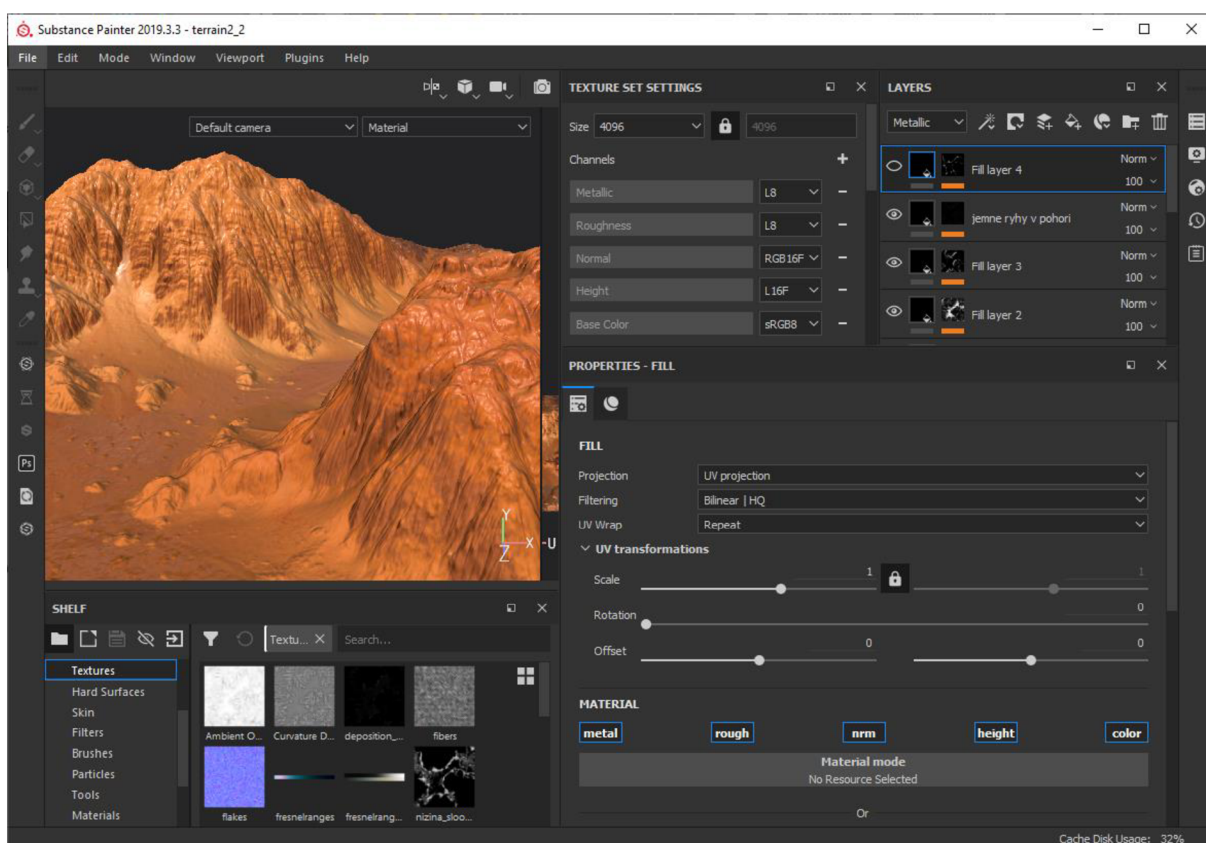
ve kterém je oblast uzavřena nebo brání přijímání přicházejícího světla. Místa objektu reálně zastíněná, mají barvu v odstínech šedi nejtmaší. Výsledek ilustruje obrázek č. 24. Tímto způsobem lze simulovat také míru opotřebení objektů.

3. Layers > Add Fill Layer > Base Color, změníme na zamýšlenou převládající barvu terénu. Stejným způsobem přidáme tolik vrstev, kolik jsme vygenerovali masek z programu WorldMachine.

4. Pravým tlačítkem myši klikneme na vrstvu v záložce layers a z nabízených možností vybereme „Add Black Mask“ Klikneme pravým tlačítkem na čtvereček symbolizující právě vytvořenou masku a zvolíme možnost „Add Fill“. Do právě vytvořené prázdné masky, můžeme

přiřadit masku z WorldMachine. To uděláme tak, že v levém postranním panelu vybereme složku projektu „Textures“ a požadovanou texturu přetáhneme do pravého panelu, konkrétně do tlačítka s popiskem „grayscale“. Tentýž postup zopakujeme pro všechny masky, které plánujeme aplikovat.

5. Výsledky své práce, vytvořené textury jednoduše exportujeme do rastrových obrazů, příkazem v menu File > Export textures. Uživatelské rozhraní a zmíněné příkazy ilustruje obrázek č. 25.



Obrázek 25: *Tvorba textur terénu*

3.3.3 Materiálové sloty a dokončení v 3ds Max



Tam, kde naše výprava za poznáním modelovacích technik započala – tedy v programu 3ds max, tam také končí. Tentokrát se však zaměříme na to, jak vytvořené textury, vytvořené v aplikaci Substance Painter, spojit v jeden výsledný materiál, který na objekt terénu aplikujeme. Stejným způsobem může čtenář postupovat při tvorbě dalších materiálů k herním objektům.

V prostředí 3ds max můžeme zvolit z několika materiálů, přičemž každý z nich má své specifické využití. Disponuje vlastní sadou mapových slotů, do kterých můžeme vkládat připravené textury a také další sadou parametrů, kterými je možno výsledný vzhled materiálu

opatřit. Vzhledem k tomu, že použití našich modelů s přiřazeným materiálem směřujeme do prostředí Babylon.js, tedy rámce, který bude s našimi modely pracovat, vykreslovat je na obrazovky uživatelů, využijeme vývojáři nejvíce doporučený materiál. To s ohledem na parametrické možnosti nastavení a realistiku výsledku. Tímto materiálem je Arnold Standard Surface.

Nejprve tento materiál popíšeme v obecné rovině. V jeho charakteristických rysech, stejně tak jako mapové sloty, které budeme v naší hře v rámci materiálu používat. V posledním kroku budeme detailně explarovat, kterak to v prostředí 3ds max provedeme a grafiku hry tak můžeme dokončit.

Arnold Standard Surface, je fyzicky založený materiál, schopný produkovat realisticky mnoho typů materiálů z našeho světa, ale také materiály zcela fantaskní. Obsahuje sloty pro základní difúzní mapu (vrstvu), metalickou mapu s komplexním Fresnelovým efektem pro simulaci kovových povrchů, skla a další. Naši hry se týká níže popsaná čtveřice map.



Base color, neboli difúzní charakteristika materiálu, představuje základ většiny materiálů. Můžeme si ji představit jako základní barevný popis objektu. Obrázek č. 26 ilustruje tuto naši mapu vytvořenou v programu Substance

Painter.

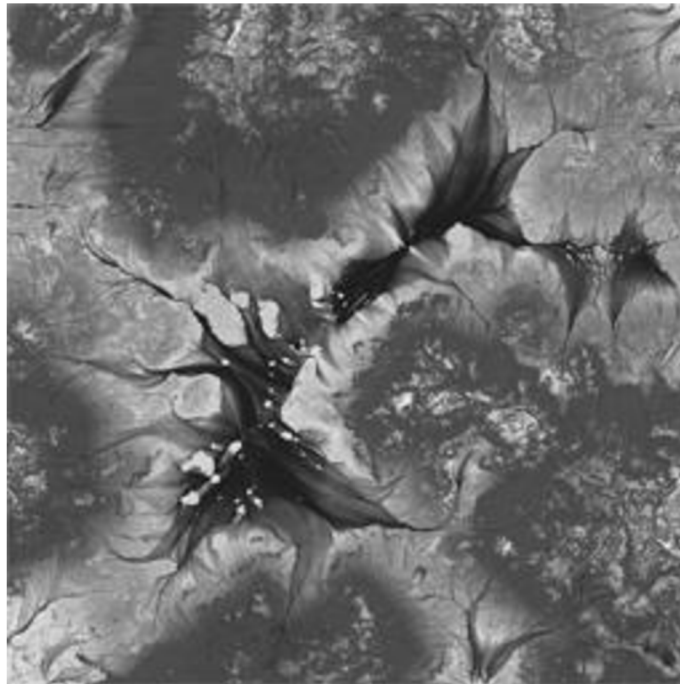


Obrázek 26: *Výsledná textura terénu*

Specular Roughness, neboli spekulární drsnost, řídí míru lesklosti povrchu objektu až do míry zrcadlových odrazů. Čím je hodnota nižší, tím je odraz ostřejší. V intervalu nastavitelných hodnot $\langle 0, 1 \rangle$, nám hodnota 0 poskytne dokonale ostrý zrcadlový odraz (odraz okolních objektů ve scéně). Zatímco hodnota blízká jedné, vytvoří odrazy, které jsou blízké difúznímu základu. V naší hře přiřazujeme nikoliv jednu hodnotu ze zmíněného intervalu ale opět rastrový obraz,

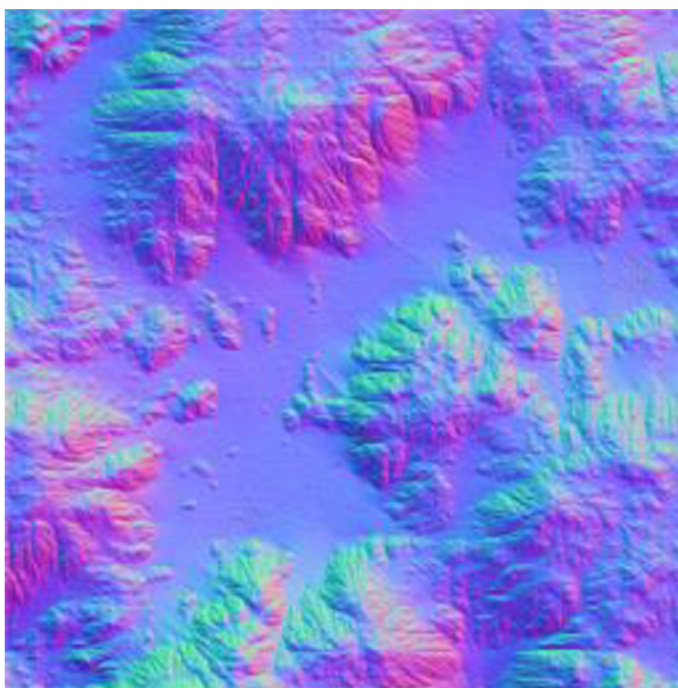
vytvořený z předešlých kapitol. Každý obrazový bod z odstínů šedi tohoto obrázku tak představuje jinou hodnotu - míru odlesku. Použití této mapy nám ve hře umožní simulovat deštěm namoklý, lesknoucí se povrch pohoří. Obrázek č. 27 ilustruje texturu, umístěnou v tomto mapovém slotu.

Posledním využitým mapovým slotem, je mapa Normal. Podle Contora (2018) se jedná o velmi výkonnou a populární mapu, techniku mezi aplikacemi vykreslující 3D grafiku



Obrázek 27: *Spekulární textura terénu*

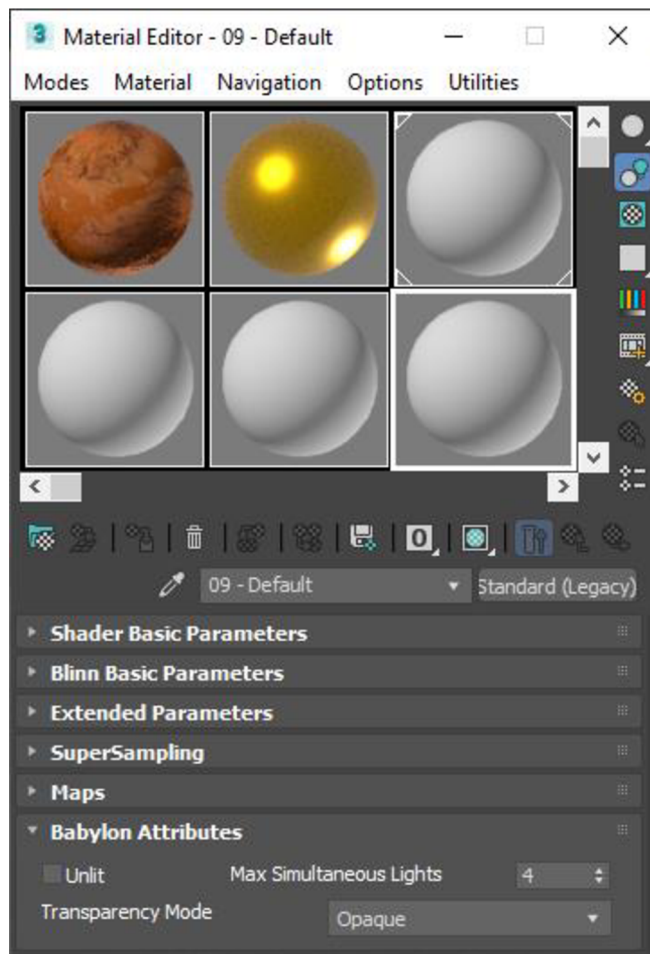
v reálném čase. Normálové mapování vytváří iluzi vysoce detailní geometrie na low-poly modelu tím, že ukládá normály povrchu do textury, kterou lze poté použít k výpočtu polygonálně detailnějšího objektu. Tato metoda je obzvláště populární v moderních hrách, protože vývojářům umožňuje dosáhnout rovnováhy mezi vysokým výkonem a zobrazenými detaily objektů ve scéně. Při mapování pomocí normálové mapy, jsou normály povrchu reálných polygonů objektu „low-poly“, nahrazeny sub-normálami, které jsou zakódovány do textury tohoto typu. Výsledkem je iluze geometrie zachycující větší detail. Obrázek č. 28 ilustruje normálovou mapu. Tu jsme získali na základě „high-poly“ modelu z aplikace World Machine.



Obrázek 28: *Normálová mapa*

Nyní uvedeme detailní postup, kterak umístit jednu z textur do mapového slotu v aplikaci 3ds max. Další textury umístíme stejným postupem, pouze do mapového slotu jiného příslušného názvu.

1. Otevřeme aplikaci 3ds max. V hlavním menu File > Import > Import, vybereme soubor obsahující model terénu, vytvořený v aplikaci World Machine.
2. Stiskem klávesy „M“, otevřeme okno Material Editoru, alchymistické dílny, kde všechny úžasné materiály vznikají.
3. Klikneme na tlačítko Standard (Legacy). V nově otevřeném okně se proklikáme stromovou strukturou až k možnosti “Standard Surface”, cestou: Arnold > Surface > Standard Surface. Standard Surface je tedy označení pro materiál (shader), který v sobě nese příslušné mapové sloty, do nichž budeme vkládat připravené textury z předešlých kapitol. Okno materiálového editoru ilustruje obrázek č. 29. Součástí obrázku je také šestice koulí v horní části obrázku. Každá z nich představuje náhled materiálu, jak tedy náš materiál bude přibližně vypadat. Jak budete přidávat postupně nové textury, bude se postupně měnit i tento náhled.



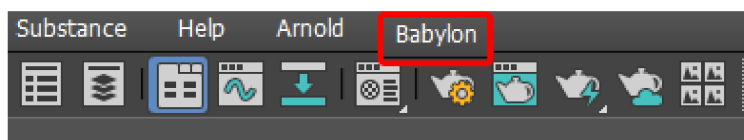
Obrázek 29: *Material Editor 3ds Max*

4. Rozklikneme v Material Editoru paletku Maps, následně rozklikneme tlačítko „Base color“. V záložce „General“ vybereme možnost „Bitmap“. V nově zobrazeném okně, vybereme příslušný rastrový soubor.

3.4 Export objektů a materiálu do engine Babylon.js

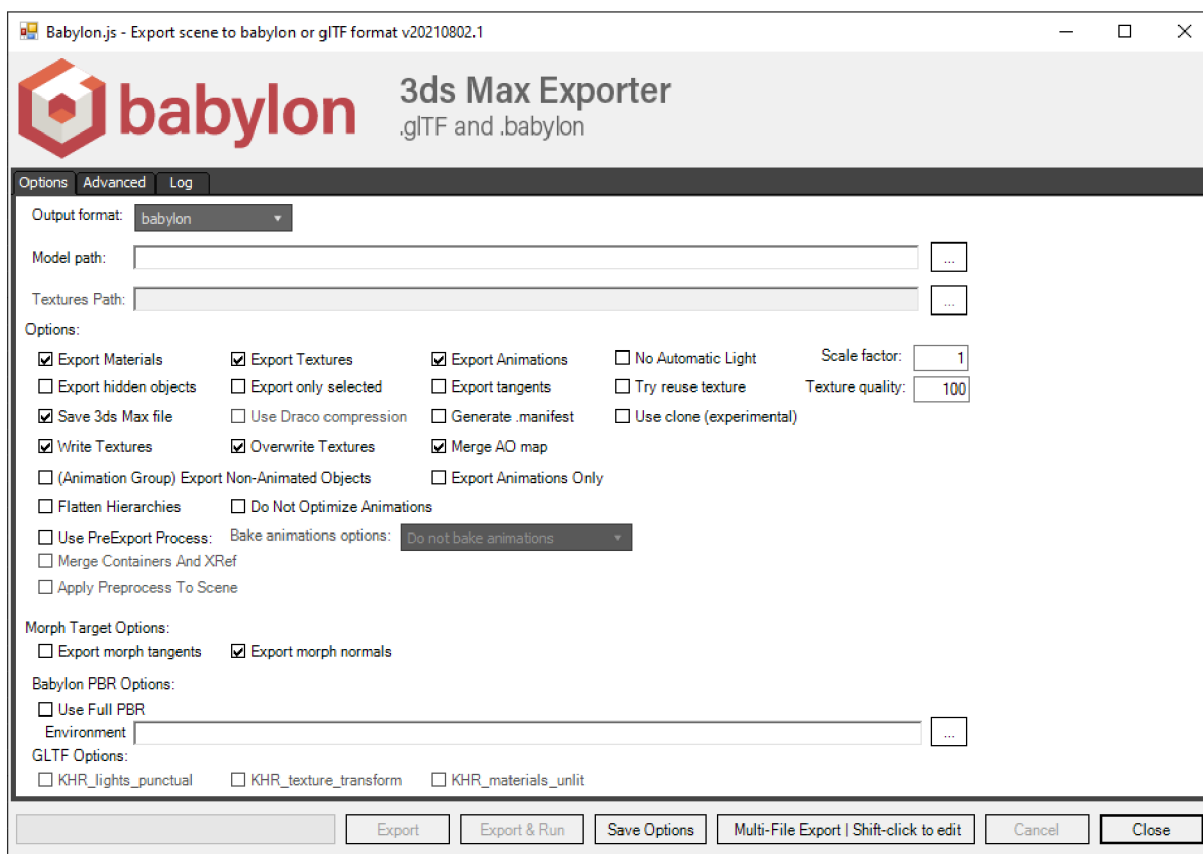
Finální export objektů, provedeme opět v aplikaci 3ds max. Na rozdíl od náročných témat tvorby objektů a malování textur, je samotný export objektů již snadnou úlohou, kterou objasníme stručně ve dvou krocích.

Plugin pro slíbený export zdrojů, je možno snadno vyhledat a stáhnout na stránkách github.com pod heslem Babylon.js. Exportér je dostupný pro verze 3ds Max 2015 -2022. Po instalaci se v aplikaci 3ds Max vytvoří nová položka v horním menu, kterou ilustruje obrázek č. 31.



Obrázek 31: Správně nainstalovaný plugin Babylon.js

V pracovním výřezu 3ds Max, vybereme objekt, který plánujeme exportovat. Z nového menu „Babylon“, vybereme položku „Babylon File Exporter“. Zobrazené okno s nastavením ilustruje obr. č. 30.



Obrázek 30: Exportér Babylon.js v 3ds Max

V okně exportéru změním pouze několik málo nastavení. Samozřejmě zvolíme cestu a název exportovaného souboru „Model path“. Změním ale také výstupní soubor z výchozího nastavení „babylon“ na formát „glb“. Zatrhneme ještě volbu „Exported only selected“ – a nastavení potvrdíme tlačítkem „Export“. V dolní části okna se zobrazí protokol o prováděném exportu. Jestliže export proběhne bez potíží, jsme připraveni využít exportované soubory v rámci Babylon.js. O tom bude ovšem až další kapitola, kapitola čtvrtá.

3.5 Učební úkoly a shrnutí třetí kapitoly



Následující baterie učebních úkolů, je pro žáka koncipována s rostoucí obtížností, a to od fáze porozumění učivu až po tvůrčí kompetenci. Jejich zvládnutí je předpokladem k vlastní, samostatné tvůrčí cestě žáka, v aplikacích určených pro tvorbu prostorové grafiky. Úkoly jsou koncipovány do značné míry jako úkoly divergentní. Žák může správně odpovědět vícero způsoby. S žáky je potom přirozeně možno diskutovat výběr nejlepší odpovědi z odpovědí akceptovatelných. Úkoly jsou ale také koncipovány jako problémové. Žák tedy s největší pravděpodobností nezná odpověď na zadaný úkol z hlavy a musí systematicky odpověď hledat v dalších zdrojích, například v doporučené nápovědě k aplikaci 3ds Max.

1. Vyjmenujte obvyklý postup tvorby polygonálního modelu a pokuste se vlastními slovy vyjádřit, co je obsahem těchto kroků?
2. Vysvětlete význam jednotlivých parametrů všech modifikátorů uvedených na obrázku č. 7.
3. Objektu krychle, přiřazujte metodou pokus-omyl další modifikátory ze seznamu modifikátorů. Jestliže je modifikátor určen k deformaci objektů, vysvětlete jeho význam i jeho parametrů.
4. Navrhněte seznam všech věcí, které by bylo možno modelovat pomocí modifikátoru Lathe?
5. Navrhněte seznam všech věcí, které by bylo možno modelovat pomocí modifikátoru Loft?
6. Aplikujte postup z řešených úkolů k modifikátoru Lathe nebo Loft a pokuste se některý z navrhovaných věcí realizovat.
7. Na dvojici objektů krychle a válec, vyzkoušejte všechny parametry modelovacího nástroje ProBoolean.
8. Analyzujte možnosti modelovacích nástrojů Lathe, Loft a ProBoolean a rozeberte, která by s kombinací těchto modifikátorů, bylo možno realizovat další složitější objekty.
9. Uveďte klady a zápory modelovacích nástrojů z úkolu č. 8, ale také modifikátoru Edit Poly a argumentujte možnosti jejich kombinovaného použití v rámci práce na jednom objektu.
10. S dopomocí modelovacích nástrojů Lathe, Loft, ProBoolean a Edit Poly, navrhněte a vytvořte libovolný model, přírodního nebo technického charakteru, který bude využívat a spojoval v jednom objektu všech výhod uvedených modelovacích nástrojů.
11. Otevřete projektový soubor **baby_bottle.max**. V pracovním výřezu uvidíme scénu s dětskou lahví, a právě s touto scénou budeme nyní pracovat. V tomto úkolu, budete objevovat zákonitosti správného nastavení a) trojice použitých materiálů, b) prostředí

vizualizace v aplikaci 3ds Max. Výslednou vizualizaci po vykreslení scény, vidíte na obrázku č. 32.

- a) Otevřete okno Materiálového Editoru (pomocí klávesy M). Metodou pokus-omyl, nastavujte různé číselné hodnoty u jednotlivých parametrů materiálů a sledujte účinky změn parametrů. Četnost pokusů, redukuje intenzivní práci se skvěle napsanou nápovědou k programu, která význam parametrů objasní (otevřete klávesou F1). Zhodnoťte a shrňte praktický význam jednotlivých parametrů materiálů, pro vlastní budoucí tvorbu.
- b) Otevřete okno Environment and Effects, pomocí horního menu Rendering > Environment, jehož účelem je, vytvořit vhodné prostředí obklopující vizualizovaný objekt. Reflexní materiály naší scény tedy reflektují zde nastavené okolní prostředí. Postupujte stejně jako v případě zkoumání parametrů trojice materiálů, následně zhodnoťte a shrňte význam jednotlivých parametrů, pro vlastní budoucí tvorbu.



Obrázek 32: *Vizualizace dětské láhve*

Σ

V průběhu třetí kapitoly žáci postupně porozuměli trojici kroků, vedoucích nakonec k vytvoření 3D modelu polygonální metodou modelování, která využívá možnosti manipulace se sub-objekty modelu. Dokázali také použít sadu základních modelovacích nástrojů – Lathe, Loft, Displace, ProBoolean, Displace. Dostatečným procvičováním modelovacích nástrojů dokázali žáci analyzovat a syntetizovat možnosti a limity jednotlivých nástrojů a tím volit jejich efektivní kombinace k tvorbě

náročnějších modelů. Navázali ale také na předchozí znalosti z rastrových editorů, typu Adobe Photoshop, především tedy na práci s vrstvami, maskou. K tvorbě úžasných materiálů aplikovaných následně na 3D model herního terénu, využili k tomu optimální aplikaci Adobe Substance Painter.

4 Tvorba hry v Babylon.js

Cíle kapitoly

Žák bude:

- schopen navrhnout a naprogramovat funkční projekt v Babylon.js.
- aplikovat poznatky z výuky nebo nápovědy k Babylon.js. Načte a zobrazí základní 3D primitiva v Babylon.js.
- analyzovat požadavky na osvětlení scény a nastavení kamery a zhodnotí situační požadavky na osvětlení scény a nastavení kamery.
- schopen shrnout technické a estetické požadavky kladené na zamýšlený multimediální program v ose – implementace Babylon.js - osvětlení – kamera – fyzika hry a kolizní systém – logika hry.
- schopen iniciovat tvorbu originálního multimediálního projektu.

Motivace: úvodem lze motivovat nabytím výsledných kompetencí v ose: 3D grafika – programování – multimediální projekt. Hraní her je nepochybně celosvětovým fenoménem napříč věkovými kategoriemi nebo pohlavím. Atraktivita možnosti vytvořit si jednoduchou vlastní hru je nasnadě. Průběžná motivace by měla být generována z dílčích úspěchů žáků. Ze schopnosti použít další a další vizuální, herní možnosti Babylon.js. Další využití nabývaných dovedností ve školní praxi (školní web, prostory školy v 3D..) nebo soutěžích. Soutěže jsou výrazným motivátorem, především nadanějších žáků. Nutno však respektovat obecné principy motivovanosti žáka k práci, viz kapitola věnovaná motivaci. Nutno je také respektovat obor SŠ a identifikovat vlastní preference, potřeby, zájmy žáků a tomuto zjištění plán motivace organizovat.

Metody: problémové vyučování viz blíže v kapitole 2.4, dále brainstorming, výklad, názorná ukázka, prezentace, procvičování pod dohledem učitele, skupinová práce. Metodickou příručku lze ale také využít jako inspiraci k projektovému vyučování.

Metodický postup: čtvrtá kapitola metodické příručky, obsahuje doporučený způsob explanování nejfrekventovanějších, základních témat spojených s tvorbou počítačových her a jiných multimediálních projektů v prostředí Babylon.js. Jednotlivé kapitoly jsou řazeny postupně tak, jak by měly být žákům explanovány. Učivo je mimořádně náročné na představivost, logické a inovativní myšlení. Příložený, kompletní zdrojový kód hry s bohatým komentářem by měl být žákům k dispozici, jako didaktická pomůcka. Porozumění učivu bude pro žáky mimořádně obtížné, jestliže nebude správně realizována kapitola věnovaná tvorbě grafického obsahu, tedy kapitola třetí. Tato kapitola žákům neobjasňuje

pouze tvorbu 3D modelů nebo materiálů. Vytváří také nezbytnou terminologickou základnu z oblasti počítačové grafiky, která umožní žákům porozumění významům a vztahům jednotlivých tříd, metod a jejich parametrům v Babylon.js. Žákům je potřeba nabídnout dostatek příkladů a prostoru k procvičování ve škole a nejlépe také doma.

Program realizace výuky

Kapitola 4.1 | Učitel názorně explanuje základní pojmy a jejich vztahy, související s platformou Babylon.js (význam termínů: JavaScript (už znají) – webová aplikace (už znají) WebGL – Visual Studio). Uplatní projektor. *[vysvětlování, názorná ukázka, rozhovor]*. Učitel promítne programový kód základního projektu Babylon.js (Programový kód č. 1). Žáci kód reprodukují v prostředí Visual Studio 2017. Učitel závěrem explanuje nezbytné a obvyklé části hry, multimediální aplikace (tyto části jsou také vyjmenovány v diskutovaném zdrojovém kódu) *[názorná ukázka, procvičování pod dohledem učitele, vysvětlování, rozhovor]*.

Kapitola 4.2.1 – 4.2.2 | Učitel explanuje možné zdroje objektů v projektu Babylon.js (vytvořené programovým kódem v Babylon.js nebo export z modelovacího 3D programu) *[vysvětlování, názorná ukázka]*.

Žák si následně zobrazí oficiální stránky Babylon.js, sekci „Playground“ (aktuálně dostupné na elektronické adrese: playground.babylonjs.com). Nyní může pracovat s „live editorem“ rámce Babylon.js. Zobrazí tvůrci Babylon.js připravenou, editovatelnou programovou ukázkou věnovanou postupně: a) vytvoření objektu programovým kódem, b) importu objektů. Žák si zde procvičí vysvětlené učivo změnou parametrů volaných metod. Učitel postupně žákům zadá také taková problémová zadání, která žák realizuje pouze v součinnosti s oficiální nápovědou Babylon.js *[názorná ukázka, procvičování pod dohledem učitele, řešení problémů, výuka podporovaná počítačem]*.

Kapitola 4.3 | Učitel vysvětlí základní teoretické koncepce osvětlení ve světě CGI a uvádí je do vztahu k možnostem, které poskytuje rámec Babylon.js *[vysvětlování, názorná ukázka, rozhovor]*. Žák si následně zobrazí oficiální stránky Babylon.js, sekci „Playground“ (aktuálně na elektronické adrese: playground.babylonjs.com). Nyní může pracovat s „live editorem“ rámce Babylon.js. Zobrazí tvůrci Babylon.js připravenou, editovatelnou programovou ukázkou věnovanou postupně všem podporovaným typům osvětlení a základním parametrům. Žáci si procvičí vysvětlené učivo změnou parametrů volaných metod. Učitel postupně žákům zadá také taková problémová zadání, která žák realizuje pouze

v součinnosti s oficiální nápovědou Babylon.js [*názorná ukázka, procvičování pod dohledem učitele, řešení problémů, výuka podporovaná počítačem*].

Kapitola 4.4 | Učitel explanuje základní teoretické koncepce použití kamery ve světě CGI a uvádí je do vztahu k možnostem, typům kamer, které poskytuje rámec Babylon.js [*vysvětlování, názorná ukázka, rozhovor*]. Žák si následně zobrazí oficiální stránky Babylon.js, sekci „Playground“ (aktuálně na elektronické adrese: playground.babylonjs.com). Nyní může pracovat s „live editorem“ rámce Babylon.js. Zobrazí tvůrci Babylon.js připravenou, editovatelnou programovou ukázkou věnovanou postupně všem podporovaným typům kamer. Procvičí si vysvětlené učivo změnou parametrů volaných metod. Učitel postupně žákům zadá také takové problémové zadání, které žák realizuje pouze v součinnosti s oficiální nápovědou Babylon.js [*názorná ukázka, procvičování pod dohledem učitele, řešení problémů, výuka podporovaná počítačem*].

Kapitola 4.5 | Učitel postupně vysvětlí praktické využití: a) implementace uživatelského vstupu (klávesnice, myš, gamepad) a b) možnosti kolizního systému a fyziky v rámci Babylon.js [*vysvětlování, názorná ukázka, rozhovor*]. Žák si následně zobrazí oficiální stránky Babylon.js, sekci „Playground“ (aktuálně na elektronické adrese: playground.babylonjs.com). Nyní může pracovat s „live editorem“ rámce Babylon.js. Zobrazí tvůrci Babylon.js připravenou, editovatelnou programovou ukázkou věnovanou postupně možnostem uživatelského vstupu z klávesnice a myši. Později otevře také ukázkou věnovanou koliznímu systému a fyzice. Procvičí si vysvětlené učivo změnou parametrů volaných metod. Učitel postupně žákům zadá také takové problémové zadání, které žák realizuje pouze v součinnosti s oficiální nápovědou Babylon.js [*názorná ukázka, procvičování pod dohledem učitele, řešení problémů, výuka podporovaná počítačem*].

Závěrečné zadání učebního úkolu pro kapitoly č. 4 | Analyzujte předložený zdrojový kód počítačové hry Fantasy golf. Implementujte v něm objekty vytvořené v kapitoly č. 3. Opravte chyby ve zdrojovém kódu, navrhněte možnosti jeho vylepšení, vylepšení hry samotné a realizujte je. Presentujte výsledek své práce před třídou a vlastní postup práce i výsledek obhajte. Na úkolu se podílejí stejné skupiny žáků, známé ze závěrečného úkolu třetí kapitoly. Učitel individuálně zváží a) rozsah chyb v poskytnutém zdrojovém kódu a b) četnost, přesnost zanechaných poznámek zdrojového kódu hry. [*skupinová výuka, řešení problémů, kritické myšlení, výuka podporovaná počítačem*]. Žák smí a aktivně využívá všech relevantních informačních kanálů informační společnosti.

Časová dotace: na každou z kapitol připadá jedna vyučovací jednotka. Procvičování předmětných dovedností a kompetencí však může představovat násobky naznačené časové náročnosti.

Vstupní požadavky: žák dokáže aplikovat základní principy programování v rozsahu práce s proměnnými, psaní metod a funkcí s porozuměním oboru jejich platnosti. Vhodně aplikuje rozhodovací, iterační příkazy, spravuje chyby a výjimky. Volání Babylon.js API, vyžaduje také porozumění termínům třída, objekt – instance třídy, metoda a její parametry. Procedurální programování hry, které bylo použito, představuje pro žáky určité zjednodušení. Hru však lze poměrně snadno transformovat do podoby objektově strukturovaného zdrojového kódu. Takto „dvojstupňově“, postupuje například Žára (2015).

Didaktické prostředky ICT: projektor, PC s OS Windows 10 a internetovým připojením pro každého žáka. Minimální konfigurace PC stanice, determinovaná HW požadavky na akcelerovanou 3D grafiku, grafická karta tedy alespoň NVIDIA Geforce 640 a novější. Procesor Intel Core i5, operační paměť 8 GB DDR4.



V čem budeme programovat naši hru Fantasy golf a co pro nás znamená Babylon.js? Než odpovíme, zmíníme nejprve termín WebGL. Jedná se o 2D a 3D grafickou knihovnu, určenou pro webové prohlížeče. Knihovna WebGL tedy vývojářům na celém světě umožňuje zdarma použití API funkcí, které je možno volat z programovacího jazyku JavaScript. Naši žáci, jako dychtiví vývojáři, mohou vykreslovat interaktivní 2D a 3D grafiku v jakémkoli z nejrozšířenějších webových prohlížečů, napříč platformami. To znamená, že WebGL, je nativně integrován do prohlížečů, bez potřeby jakýchkoli externích doplňků. Už toto nám dává naději v dlouhý život diskutované technologie (porovnáme-li například s Adobe Flash). S WebGL můžeme tedy využít GPU grafické karty, na klientských počítačích. Hlavní nevýhodou tohoto API však je jeho složitost použití pro vývojáře.

Programování s WebGL přímo z Javascriptu, je tedy složitým procesem, který však lze významně usnadnit použitím některého z takzvaných „frameworků“, což chápeme jako jakési „rámce“, postavené nad WebGL. Frameworky nám tak mohou například zprostředkovat volání několika metod původní knihovny WebGL a tyto logicky kumulovat do jediné metody. Takto dojde pochopitelně k zjednodušení a použití bude pro vývojáře mnohem více „přátelské“.

Dateme k této práci, existuje hned několik takových rámců. Na tomto místě se zaměříme na jeden z nich, jak tušíme - na Babylon.js. Babylon.js můžeme chápat, jako skutečně účinný

nástroj k tvorbě graficky náročných, plně interaktivních 2D nebo 3D aplikací, vykreslovaných před zraky uživatele v reálném čase. Představovaný framework tak můžeme použít pro tvorbu náročných multimediálních projektů, včetně výukových programů, her a některé možná překvapí, že také k tvorbě aplikací virtuální reality. Výraznou výhodou popsané technologie je také fakt, že je ve výchozím nastavení prohlížečů obvykle povolena, a to i přes teoretické možnosti zneužití této technologie. Chceme-li vykreslit jednoduchý objekt koule nebo roviny? Stačí zavolat metodu `CreateBox (...)` nebo `CreatePlane (...)` a to je báječné, obzvláště když na vývoj aplikace nemáme celé týmy vývojářů. I přes nesporné zjednodušení práce, které nám framework `Babylon.js` umožní, je potřeba si uvědomit, že volání trojice metod místo deseti, od tvůrců aplikací stále jisté porozumění termínům z ranku počítačové grafiky vyžaduje. Neznáme-li například termíny `polygon` nebo `bodové světlo`, bude tak jako tak obtížné metody úspěšně a smysluplně volat.



Výše uvedeným se dostáváme k připomenutí, že činnost tvorby hry ve školním prostředí, je minimálně intradisciplinárním oříškem, propojující více témat téhož oboru (například multimédia a programování) nebo mnohem spíše interdisciplinární výzvou techniky ale také matematiky nebo fyziky, protože tvorba logika hry si to vyžádá v menší nebo větší míře vlastně vždy.

4.1 Základní projekt `Babylon.js` v HTML a JavaScriptu



Nejprve otevřeme zvolený textový editor/IDE a vytvoříme nový soubor/projekt. Zkopírujeme do něj níže uvedený programový kód č 1. V našem případě používáme vývojové prostředí Microsoft Visual Studio 2017 Community. Zdrojový kód uložíme do nového souboru s názvem `index.html`. Výsledek našeho úvodního snažení si můžeme prohlédnout uložením výstupního souboru na úložiště webového serveru, případně simulací serveru lokálně.

HTML část ukázky, bude pro naše žáky skutečně jednoduchá. Na začátku můžeme vidět načtení externí knihovny `Babylon.js` z oficiálního serveru CDN a také načtení knihovny pro herní fyziku - `Ammo.js`. Upozorníme, že u složitějších projektů, budeme muset načítat hned několik dalších knihoven. Jejich názvy nám však ochotně prozradí, poměrně dobře zpracovaná on-line nápověda k `Babylon.js`. Malá část CSS, slouží k úpravě výchozího nastavení okrajů stránky HTML. Nejdůležitějším počinem, je zřejmě použití prvku „`canvas`“. Tento prvek, budeme v naší hře používat k 2D vykreslení všech 3D objektů, přesněji té části scény, aktuálně zabírané aktivní kamerou. Dílčí výsledky našeho programování vždy uvidíme přímo

na zobrazené webové stránce, uvnitř webového prohlížeče. Neopomenutelným pomocníkem našim žákům jistě bude, v prohlížečích integrovaná ladící konzole. Po stlačení klávesy F12, se ve většině nejrozšířenějších prohlížečů otevře slíbená nabídka pro vývojáře. Takto zkontrolujeme, zda nám nějaká chyba v našem programovém kódu, nebrání ve sledování kýžených výsledků naší tvrdé práce.

Už ale žádné další zbytečné napětí, popíšeme tady žákům, krok za krokem tu zajímavější, JavaScriptovou část zdrojového kódu. Jak jsme již řekli dříve, prvek „plátna“, nebo-li canvas - je povinným prvkem pro vykreslování 3D scény a jakožto vývojáři, musíme zajistit, aby byl načten prohlížečem. Prvek canvas, jehož zvolené ID je „FantasyGolfGameCanvas“, je použit k vytvoření **BABYLON.Engine**. Tato třída Engine (motor), respektive tento vytvořený objekt, je zodpovědný za plně funkční propojení s nízkoúrovňovým rozhraním API WebGL a Audio. Je to právě tento diskutovaný objekt, který bude vykreslovat (renderovat) scénu na našich stránkách. Prvním parametrem jeho konstruktoru je vytvořený canvas - plátno, tedy objekt rozhraní DOM¹⁰. Druhým parametrem zapínáme nebo vypínáme vyhlazování kreslené scény a můžeme tak ušetřit něco výpočetního času, na úkor vyhlazení hran vykreslovaných objektů. Babylonská scéna vzniká předáním motoru scéně.

Termín scéna můžeme žákům přiblížit jako kontejner, který ukládá všechny objekty, které potřebuje naše hra. Tento soupis je uveden v poznámkách viz programový kód č. 1. Poznámky s hvězdičkou znamenají, že tento objekt je reálně potřeba definovat vždy. Metoda `runRenderLoop()` zajišťuje, aby byla scéna zobrazována opakovaně, v reálném čase a obvykle rychlosti 60 fps, tedy šedesátkrát za sekundu, jestliže to ovšem výpočetní výkon použitého počítače umožní. Popsané se stane díky registraci a spuštěním vykreslovací smyčky.

¹⁰ DOM, objektový model HTML dokumentu, API umožňující přístup a modifikaci k HTML elementům stránky.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>FantasyGolfGameCanvas Babylon.js</title>
  <script src="http://cdn.babylonjs.com/2-5/babylon.js"></script>
  <script src="https://preview.babylonjs.com/ammo.js"></script>
  <style>
    html, body {
      overflow: hidden;
      width: 100%;
      height: 100%;
      margin: 0;
      padding: 0;
    }
    #FantasyGolfGameCanvas {
      width: 100%;
      height: 100%;
      touch-action: none;
    }
  </style>
</head>
<body>
  <canvas id="FantasyGolfGameCanvas"></canvas>
  <script>
    var canvas = document.getElementById("FantasyGolfGameCanvas");

    var createScene = function () {
      scene = new BABYLON.Scene(engine);
      // *tvorba a/nebo import objektů hry,
      // *nastavení kamery,
      // *nastavení světla,
      // nastavení materiálů,
      // *nastavení klávesnice a/nebo gamepadu,
      // nastavení fyziky a kolizí,
      // nastavení animace pomocí "bones",
      // tvorba audia,
      // post-process, částicové systémy a alší efekty,
      // *logika hry.
      return scene;
    }

    var engine = new BABYLON.Engine(canvas, true, { preserveDrawingBuffer: true,
stencil: true });
    var scene = createScene();

    engine.runRenderLoop(function () {
      if (scene) {
        scene.render();
      }
    });
  </script>
</body>
</html>

```

Programový kód 1: *Základní projekt Babylon.js*

4.2 Implementace herních objektů



Termíny 3D objekty, 3D modely nebo také „mesh – 3D síť“, chápeme obvykle jako synonyma. Ve skutečnosti jsou 3D objekty nebo modely tím, čemu říkáme (polygonální) 3D síť. Jejich struktura ukládaná do paměti počítače, je poměrně jednoduchá. Děje se tak pomocí tzv. vertex bufferu objektu. Jedná se o pole 3D bodů – vrcholů, které objekt v prostoru definují. Tato data, která se používají k popisu geometrie, jsou využita pro jeho následné vykreslení. Vrcholy představují body v prostoru a jsou tedy potřebné k vytváření geometrických tvarů. Jedna strana krychle, bude obvykle vykreslena pomocí dvojice trojúhelníků. Grafické karty jsou totiž optimalizovány pro akceleroovaný výpočet a vykreslování trojúhelníků na obrazovce počítače. Popsané použije Babylon.js, respektive WebGL k vykreslení našich úžasných objektů. Síť můžeme rozlišovat dvojího typu. Prvním typem jsou jednoduchá, základní 3D primitiva, příkladem kostka nebo koule. Tyto mohou být vytvořeny přímo v Babylon.js - programovým kódem, přesněji voláním jediné metody. Druhým typem je import obvykle složitějších 3D sítí, přímo z některého z modelovacích programů, jakými jsou například 3ds Max, Maya, Blender a mnoho dalších.

4.2.1 Implementace základních 3D primitiv v Babylon.js



Jednoduché 3D objekty, můžeme tedy vytvořit přímo v Babylon.js. K vytváření sítí nám tento framework poskytuje statické metody ve třídě BABYLON.MeshBuilder, které nám umožní vytvářet základní síť, jako jsou krychle, koule nebo rovina. Celkem je nyní možno vytvořit 18 základních objektů, které jsou vývojáři k dispozici. Snadnost vytvoření krychle ilustruje programový kód č. 2. Podobně vytvoříme další objekty. První parametr metody CreateBox(...), definuje název sítě pro pozdější identifikaci ve zdrojovém kódu. V druhém a dalších parametrech, můžeme definovat řadu volitelných parametrů, podrobněji vždy viz on-line nápověda Babylon.js. K takové praxi, tedy vyhledávání v nápovědě, případně z jiných zdrojů, je potřeba vest také naše žáky. Předkládání výhradně hotových řešení není vhodné. Učitel není encyklopedií a komplexní nápovědou mimořádně rozsáhlého frameworku Babylon.js už vůbec ne. Posledním parametrem předáváme odkaz na scénu (kam přidat síť). Scén může být totiž vytvořeno podle potřeby hned několik a je možné mezi nimi „přepínat“.

```
var box = BABYLON.MeshBuilder.CreateBox("box", options, scene);
```

Programový kód 2: *Vytvoření objektu krychle*

Jakmile vytvoříme síť, můžeme přistupovat k jejím vlastnostem. K těm nejzákladnějším a nejpoužívanějším patří transformace v prostoru. Zdrojový kód č.3, ilustruje příklad nastavení pozice krychle v naší scéně.

```
box.position = new BABYLON.Vector3(0, 2, 5, 0);
```

Programový kód 3: *Situování objektu v herním světě*

Ty bystřejší z našich žáků možná napadne, položit si otázku, proč vytvářet tyto jednoduché objekty, když můžeme vše importovat přímo z modelovacích programů. Důvodů je hned několik. Ve fázi osnování idejí, základních principů hry, můžeme vytvořit přímo v Babylon.js tyto jednoduché objekty, jako zástupné objekty a operativně u nich měnit počty polygonů, podle předpokládané a často se také měnící náročnosti finálních objektů. Tento způsob práce je velmi flexibilní a efektivní. Nahrazení zástupných objektů finálními, je úkonem nenáročným a stejně postupuje například badatel Chenard (2017). V naší hře využíváme základní 3D primitiva jako zástupné objekty ale také jako pomocné objekty. Původně byl použit pomocný objekt kvádru, z něhož byly emitovány animované částicové systémy¹¹, pro tvorbu mlhoviny v oblasti mimo herní terén. Objekt golfového míčku o vysokém počtu polygonů, pozicí i rotací, následuje použitý pomocný objekt koule o nízkém počtu polygonů, který je účasten kolizí s terénem. Důvod je jasný, úspora výpočetního výkonu, při zachování realistiky simulace. To vše bez povšimnutí hráčem. A konečně jednoduchý objekt roviny, který detekuje okamžik kolize s míčkem v herní jamce. Žák by měl pochopit, že se účastní vývoje aplikace, typického problémového vyučování a ani dobrá znalost vyhledávání na internetu nemusí přinést odpovědi na všechny jeho otázky. Některé otázky musí tvůrčím způsobem vyřešit žák sám, a ne vždy to bude „na počkání“. Bez tréninku trpělivosti se žák neobejde.

4.2.2 Import objektů do Babylon.js



Způsobů, kterak importovat síť do Babylon.js je hned několik. Pro začínající žáky - vývojáře je na první pohled nejvíce atraktivním způsobem načtení všech prvků scény z aplikace typu 3ds Max, použitím metody `BABYLON.SceneLoader.Load(...)`. Úspěšné volání této metody nám totiž vrátí rovnou objekt scény. Druhým způsobem je použití metody `BABYLON.AssetsManager(...)`. Volání

¹¹ Částicový systém je specifickou modelovací a animační technikou objektů, obtížně reprezentovatelných jedním povrchem 3D sítě. Často se jedná o logicky uspořádané, pohybující se shluky miniaturních objektů (částic), opatřených speciálně navrženými materiály, které zpodobňují děje typu mlhovina, exploze nebo tok tekutin.

této metody bude automaticky doprovázeno výchozím preloaderem¹², informující hráče o stavu načítání dat. Posledním způsobem, který jsme použili v naší hře, je volání metody `BABYLON.SceneLoader.ImportMesh(...)`. Tato metoda importuje všechny sítě, materiály, animované objekty, kosti¹³ a částicové systémy nalezené v právě importovaném souboru. Tato operace trvá úměrně složitosti scény, použitým výpočetním systémem nebo rychlosti použitého internetového připojení. Úspěšným voláním vývojář získá kontrolu nad načtenými sítěmi a dalšími prvky scény. Zásadními parametry metody jsou cesta k souboru, odkaz na scénu a název funkce, která bude provedena po dokončení diskutované metody. Její volání je asynchronním procesem¹⁴ (prostřednictvím požadavku XHR¹⁵). To znamená, že pokud s načítaným obsahem chceme za voláním metody `ImportMesh(...)` pracovat, potom musíme ověřit, že tyto zdroje již byly plně načteny. V opačném případě dojde k vyvolání chybového hlášení.

4.3 Osvětlení



Cílem osvětlení v počítačových hrách, není pochopitelně vytvořit jakékoliv světlo. Implementace světla do velkolepých herních světů, má svoje technické ale také umělecké determinanty. Zapotřebí je zvolit typ světla, nejvhodnější umístění, případně zvolit barvu světelné emise, intenzitu, stín a případně další parametry. Před vytvořením a nastavením prvního světla, ve světě Babylon.js, nevykreslíme o mnoho více než canvas s jednou spojitou barvou. To je jistě málo. Jako CGI¹⁶ tvůrci, potřebujeme více. Světlo by nám mělo sloužit k tvorbě důmyslné iluze skutečnosti nebo našich představ a záměrů. Uměle použité světlo by mělo v hrách podporovat a vyvolávat emoční reakce s očekávaným atmosferickým prožitkem. Vývojář bohužel nemůže s úspěchem zvolat legendární „budiž světlo“, jako se stalo v knize Genesis, věnované stvoření reálného světa. Podíváme se proto raději blíže na zoubek otázce, která s osvětlením scény začít.

Loose (2000) uvádí k našemu tématu něco užitečné úvodní teorie. Světelnou informaci naše oko zpracovává pomocí speciálních buněk, tyčinek a čípků, rozmístěných na sítnici oka. Okem vnímáme pouze omezené světelné spektrum v rozsahu barev: červená, oranžová, zelená,

¹² Preloader poskytuje graficky zpracovanou obrazovku, někdy také animovaně zpracovanou, informující o průběhu načítání zdrojů programu. Předchází spuštění programu.

¹³ Kostmi (bones), rozumíme systém kloubovitě fungujících pomocných objektů, tvořících hierarchická propojení, která lze použít k animaci objektů, nejtýpější herních postav.

¹⁴ Asynchronní proces je událostí prováděnou nezávisle na hlavním toku programu.

¹⁵ XHR je rozhraní, které umožňuje webovým aplikacím transfer dat typu klient-server.

¹⁶ CGI (computer-generated imagery), jsou s podporou počítače, počítačových aplikací, generované grafické výstupy.

modrá, tmavě modrá a fialová. Mezi světlo, které již nevidíme, patří rentgenové záření, ultrafialové nebo infračervené světlo.

V praxi vyzařovaná barva značně ovlivňuje naladění, tedy atmosferické podání celé scény. Odstíny modré působí chladněji a s takovým světlem se setkáme často například na toaletě nebo v kuchyni. Naopak v obývacím pokoji používáme odstíny nebo různé teploty žlutých odstínů. Jako tvůrci vytváříme určitou vizuální náladu. Dodáme, že se v naší populaci můžeme běžně setkat s poruchami barvocitu a tato porucha může být rozpoznána také u našich žáků.

Výhodou osvětlování scény ve světě CGI, jsou jednoznačně nekonečné možnosti, kterak scénu nasvítit. Před volbou parametrického nastavení světel, bychom se měli zamyslet nad základní charakteristikou světla. Stanovit si cíl, který postupně precizujeme. Mezi tyto základní otázky podle Bella (2000) patří následující. Je scéna osvětlena přirozeně, tj. sluncem, měsíčním svitem a podobně? Jestliže ano, potom o jakou část dne se jedná? Je jasno nebo zataženo? Zatímco přirozené osvětlení má poměrně jasné realistní charakteristiky, které se snažíme napodobit, potom umělé osvětlení, může být téměř jakékoliv. Může se tedy jednat o osvětlení elektrickým zdrojem světla nebo ohněm, svíčkou nebo osvětlení ve fotografickém studiu.

Kategorizace osvětlení na přirozené a umělé, může být do určité míry dobrým vodítkem našemu výsledku. Představa autora o atmosféře nebo naladění scény je však podle našeho názoru nadřazená. Mezi jiné typy světla ve světě CGI patří nejrůznější efekty typu záře - „glow“ nebo světlo vyzařující z objemu objektu, případně jejich materiálu. Tohoto efektu bylo v naší hře využito například při ztvárnění znaku Univerzity Palackého nebo golfového míčku, který se ocitne v cílové jamce nebo mimo hranice herního světa.

4.3.1 Metody nasvícení scény



Existuje nepřehledné množství způsobů nasvícení scény, především scény ateliérové. V následujícím textu vybíráme nasvícení podle Bella (2000).

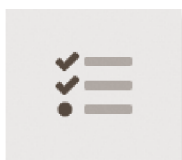
Do scény situujeme přednostně tzv. hlavní světlo – key light. Toto světlo je ve scéně často pouze jedno, bývá zdrojem základního osvětlení o největší intenzitě. Světlo se umísťuje obvykle v úhlu 45 stupňů a tytéž stupně svírá s čelním pohledem na scénu. Druhým světlem je světlo výplňové – „fill light“. Tato světla vyjasňují zastíněná místa scény, nedostatky světla hlavního. Zatímco světlo hlavní má obvykle největší intenzitu, výplňová světla působí lokálněji, přidávají však do scéně hloubku a tím zdání reality. Posledním typem světla tohoto

konceptu je vlasovka – „kicker“. Nejedná se o kadeřnickou terminologii, jak by se mohlo zdát, ale o světlo poskytující zadní osvětlení scény, vzhledem ku kameře. Světlu se také někdy říká „obrysově“. Pomáhá oddělit cílové objekty od pozadí, pomocí ozáření jejich obrysů. Typickým příkladem využití tohoto světla je východ nebo západ slunce.

V naší hře využíváme nepravidelnou matici bodových světel, přičemž jedno z nich je světlem hlavním o nejvyšší intenzitě, které simuluje sluneční záření. Výhodou takového uspořádání je kvalitní osvětlení, napodobující radiozitu¹⁷. Nevýhodou je všesměrovost bodových světel, a tedy náročnost na výpočetní systém. Loose (2000) takový typ osvětlení označuje termínem „rozptýlené světlo“.

Podle našeho názoru by měl každý tvůrce předvést vlastní svébytný pohled na osvětlování scény a vytvořit tak atmosféru scény, kterou chce zprostředkovat svým uživatelům.

4.3.2 Nasvícení scény, hry v Babylon.js



Nastal čas, objasnit žákům, jaká světla mohou vytvářet pomocí Babylon.js a realizovat tak osvětlení způsobem naznačeným v předcházející podkapitole.

Nyní se podíváme na podporované typy světel a pokusíme se zvážit jejich výhody a nevýhody pro vývojáře.

Bodové světlo je světlo definované jedinečným bodem v naší scéně. Z tohoto bodu je světlo vyzařováno do všech směrů. Dobrým příkladem bodového světla je stolní lampa, respektive klasická žárovka v ní. Výhody i nevýhody tohoto typu světla byly již zmíněny dříve.

Směrové světlo je určeno konkrétním směrem vyzařování. Světlo je vyzařováno z jakéhokoli místa ve scéně, pouze však v určitém směru. Světlo vychází z pozice umístění světla a pokračuje ve stanoveném směru scénou, teoreticky až do nekonečna nebo do okamžiku interakce s objektem ve scéně.

Další typ světla se podobá bodovému reflektoru. Je definováno polohou, směrem, úhlem a exponentem. Tyto hodnoty definují světelný kužel, vycházející z jeho polohy ve scéně a pokračující zvoleným směrem. Hemisférické světlo je posledním typem osvětlení v Babylon.js. Jeho použitím snadno simulujeme jednoduché osvětlení okolního prostředí. To v praxi znamená, že scéna bude osvětlena stejnou intenzitou v rámci celé scény. Hemisférické světlo je definováno směrem, obvykle z oblohy. Často až nastavením barevných vlastností,

¹⁷ Radiozitu chápeme jako osvětlovací techniku nepřímého osvětlování scény. Do výpočtu výsledného osvětlení, jsou například zahrnuty několikanásobné odrazy světelného paprsku mezi povrchy (objekty) ve scéně.

docílíme akceptovatelných výsledků. Nejedná se však rozhodně o světlo, které by se realistikou osvětlení podobalo radiozitě (globální iluminaci) známé z 3ds Max (vykreslovací systémy Arnold nebo V-Ray a jiné). Jemného, rozptýleného světla se tedy nedočkáme.



Jestliže je ve světě počítačových her potřeba vykreslit v reálném čase takovou kvalitu osvětlení objektů, jenž se podobá diskutované realistice světla reálného světa, použijí vývojáři obvykle metodu zvanou „light map“. Účinky realistického světla se „zapečou“ do rastrového obrazu, který se stává součástí materiálu. Objekt s takovým materiálem může být v Babylon.js osvětlen například jednoduchým světlem hemisferickým. Atmosferický výsledek však bývá mnohem působivější. Jedno upozornění závěrem. Maximální počet současně použitých světel ve scéně, je nastaven na hodnotu čtyři. Toto nastavení je našťastí pouze výchozí a lze jej změnit. Změnu ilustrujeme programovým kódem č. 4.

```
var material = new BABYLON.StandardMaterial("mat", scene);
material.maxSimultaneousLights = 6;
```

Programový kód 4: *Nastavení maximálního počtu světel ve scéně*

Správné pochopení, použití diskutovaných světel, jejím parametrům, doporučujeme pro naše žáky realizovat v aplikacích typu 3ds Max. Světla zde totiž snadno vytvoříme. Stejně tak snadno měníme jejich parametry, které zde nesou podobná nebo stejná pojmenování jako v Babylon.js. Výsledky nastavení potom vidíme ihned v pracovním výřezu. Ve zdrojovém kódu naší hry Fantasy golf, mohou žáci objevit celkem devět bodových, tedy všesměrových světel s rozdílnou intenzitou, případně barvou světla a pochopitelně jejich strategicky nastavenou polohou v prostoru. Jedno ze světel disponuje výrazně větší světelnou intenzitou a sehrává zde roli osvětlení sluncem. Programový kód č. 5, ilustruje nastavení hlavního světla. Význam parametrů je zřejmý.

```
//Klíčové světlo, simulující sluneční svit
var light2 = new BABYLON.PointLight("Omni1", new BABYLON.Vector3(0, 5, 0), scene);
light2.intensity = 100000;
light2.position = new BABYLON.Vector3(-53.76, 101.06, 12.74);
light2.diffuse = new BABYLON.Color3(249, 215, 28);
```

Programový kód 5: *Nastavení vlastností hlavního světla - "slunce"*

4.4 Kamera



Zásady používání kamery, byly formulovány již dávno. U počátečních verzí tohoto zařízení se jednalo spíše o celé temné místnosti s malými otvory. Obraz přenášený pomocí světla se zvětšoval na protější zeď.

Kamerami dneška, reprezentujeme pohled na scénu z určitého úhlu, pohledu. Tento konkrétní, zvolený pohled kamery je strůjcem vyprávění vizuálního příběhu, ať už se jedná o vizualizaci objektu, animovaný film nebo v našem případě počítačovou hru.

Poté, co vytvoříme kameru, můžeme nastavit výřezy pro zobrazení úhlu pohledu kamery. Pomocí výřezu kamery, můžeme kameru nastavit tak, jako kdybychom se dívali jejím objektivem. V Babylon.js můžeme vytvořit více kamer. Získáme tak více klíčových pohledů na scénu. Můžeme tak kamery přepínat a získat různé záběry na tutéž scénu. Pokud chceme animovat pohyb pohledem kamery, můžeme vytvořit kameru a animovat její polohu. Můžeme například chtít přeletět krajinu nebo projít budovou. Můžeme animovat i další parametry kamery. Můžeme například animovat zorné pole objektivu, abychom zvýšili efekt přiblížení scény.

Kamera, podobně jako světla, musí být ve světě Babylon.js vytvořena a řádně nastavena. Bez vytvoření kamery nebude naše hra ve Visual Studiu vůbec přeložena a objeví se chybové hlášení. Klíčové objekty scény, situujeme obvykle ve středové kompozici nebo zlatém řezu. V konečném výsledku je nutno řídit se vkusem. Podle Looseho (2000) – bez ohledu na působivost nebo bouřlivost scény by měla kamera pozorovat akci a vyprávět tak nerušeně příběh. Neměla by se otáčet kolem svých os, neboť si pozorovatel připadá jako na rotující ploše.

Z naší zkušenosti je nejlépe, když si divák kamery a její bouřlivosti ani nevšimne. Kamera by neměla být v hlavní roli, ale vždy hlavní postava – míček. Ve středové kompozici nebo zlatém řezu. Babylon.js používá kamerové pohledy také jako nositele zvláštních efektů. Jedná se například o efekt rozmazání pohybem (motion blur). Míra rozmazání objektu je potom přímo závislá na rychlosti jeho pohybu. S kamerou můžeme korigovat hloubku perspektivy (depth of field.). Volíme míru rozostření objektů v pozadí, někdy také v popředí nebo kombinaci obojího.

4.5 Závěrečné poznámky

Doposud jsme zmínili implementaci minimálního kódu Babylon.js, tvorbu světla a kamery. Všechny zmíněné součásti tvorby hry mají společné, že bez nich není možné výsledek našeho snažení spustit. Zmíníme nyní několik dalších oblastí tvorby, bez kterých je sice možné náš výtvar kompilovat, ovšem s omezenou nebo žádnou možností interakce ve smyslu uživatel – počítač.

4.5.1 Možnosti uživatelského vstupu



Žák by měl porozumět možnostem ovládání našeho herního projektu. Uživatelský vstup je samozřejmě akceptován z klávesnice a myši. Babylon.js však nabízí víc. Implementoval také Gamepad API, které spolupracuje s každým detekovaným moderním gamepad současnosti. Díky tomuto API můžeme snadno mapovat tlačítka nebo nastavit osy kříže gamepadu. Podporována je možnost také gamepadu pro herní konzoli XBOX. Uživatelský vstup je téměř nezbytný v projektech virtuální reality. V tomto případě máme na mysli technologie WebVR a WebXR. Hlavními rozdíly mezi WebVR a WebXR, je podpora různých typů uživatelských ovladačů. Zatímco WebVR podporuje rozšířené gamepad API a několik vybraných ovladačů, WebXR již podporuje mnoho různých typů uživatelských vstupů, včetně dotykových obrazovek a také ovladačů pro zachycení pohybu rukou. V projektu Fantasy golf je řešeno ovládání pomocí klávesnice. Ovládáme směrový kříž, indikující směr odpalu golfového míčku. Stlačením jednoho z tlačítek na klávesnici také zaznamenáváme dobu, po kterou bylo stlačeno tlačítko. Takto získáme míru odpalu míčku. Další uživatelský vstup nebyl programován, přestože by uživatel hry mohl takového dojmu nabýt. S implementovanou kamerou typu „ArcRotate“, můžeme totiž pomocí myši automaticky rotovat kamerou kolem zvoleného objektu (míčku). Tutéž funkcionalitu získáme pomocí tlačítek, šipek na klávesnici. Programový kód č. 6 ilustruje implementaci události „OnKeyDownTrigger“.

```
scene.actionManager = new BABYLON.ActionManager(scene);
scene.actionManager.registerAction(new
BABYLON.ExecuteCodeAction(BABYLON.ActionManager.OnKeyDownTrigger, (evt) => {

if (evt.sourceEvent.key == "p") {
// po stisku tlačítka "p", dojde k nastavení barvy světla "Light2"
Light2.diffuse = new BABYLON.Color3(249, 215, 28);
}

}

}
```

Programový kód 6: *Implementace stisku tlačítka klávesnice*

4.5.2 Fyzika a kolize hry



Babylon.js poskytuje vestavěný kolizní systém a několik variant fyzikálních motorů. Žák by měl chápat jejich základní odlišnost a správné použití. Kolizní systém lze použít ku zjištění, zda se dva nebo více objektů střetávají nebo již dokonce protínají navzájem. Detekci kolize lze také nastavit mezi objektem a stanoveným bodem v prostoru. Tam kde je tato druhá varianta možná, uspoříme něco výpočetního času. Dalším krokem je obvykle vytvoření obsluhy události, detekující kolizi. Příkladem může být

požadavek zčernání obrazovky, jestliže míček opustí oblast vymezenou pro hru. V naší hře kolizní systém aplikujeme například na aktivní kameru. Objekt kamery obklopíme neviditelným pomocným objektem koule zvoleného poloměru. Přestože je koule neviditelná pro hráče, můžeme na ni aplikovat kolizní systém. Výsledkem bude, že kamera nebude procházet skrze hlavní objekt hry – terén, nebo jakékoliv další zvolené objekty. Programový kód č. 7 ilustruje postupně vše od vytvoření kamery, zapnutí detekce kolize až po přiřazení cíle kamery. Tímto cílem je proměnná pozici míčku.

```
// Vytvoření kamery a nastavení základních parametrů: pozice, rychlosti a pozičních
limitů kamery v ose X a Y
var camera = new BABYLON.ArcRotateCamera("ArcRotateCamera", 0, 0, 6, new
BABYLON.Vector3(-17.06, 3.70, 18.45), scene);

camera.speed = 8;
camera.lowerRadiusLimit = 6;
camera.upperRadiusLimit = 12;

// Kamera přiřazena k plátnu hry
camera.attachControl(canvas, true);

// Zapnutí možnosti detekovat kolize ve scéně
scene.collisionsEnabled = true;

// Zapnutí detekcí kolize kamery
camera.checkCollisions = true;

// Rádus kolidující koule
camera.collisionRadius = new BABYLON.Vector3(1, 1, 1);

// Nastavení objektu míčku jako cíle kamery
camera.setTarget(scene.getMeshByName("golf_ball").position);
```

Programový kód 7: *Nastavení kamery*

Fyzikální motor Babylon.js, který realizuje fyzikální simulace hry se od diskutovaného systému kolizí značně liší. Fyziku nelze aplikovat na kamerách, avšak na modelech samotné scény. Jinými slovy, pokud je fyzika povolena (a nastavena) - můžeme například nechat kouli padat z výšky, působením gravitace, dokud se nesrazí s jinou sítí (například terénem) a podle předem daných parametrů se bude ve scéně koule odrážet a kutálet. Bude tedy interagovat s herním prostředím. Z řečeného by mělo být pro žáka evidentní, že fyzikální motor nenabízí pouze prostou detekci kolize objektů ve scéně. Nabízí také fyzikální simulaci uvnitř herního prostředí, kterou hráč vnímá jako samozřejmou animaci, interakci objektů.

Babylon.js zdařile integroval tyto fyzikální motory do podoby pluginů. Tyto umožňují vývojářům přidat fyzikální interakce mezi objekty scény. Na rozdíl od systému interních kolizí fyzikálních systémů, zde dochází k výpočtu dynamiky objektů a emulaci interakce mezi nimi „v reálném životě“. Pokud se tedy dva objekty srazí, navzájem se také „odrazí“. Dojde k tomu

přesně tak, jak bychom očekávali od skutečného objektu. Systém pluginů v Babylon.js nám umožnil používat osvědčené fyzikální motory a integrovat je do vykreslovací smyčky Babylon.js. Kromě velmi pokročilého použití není třeba komunikovat přímo s fyzikálním motorem. Babylon.js dělá práci za nás. Úpravy jsou však možné. V našem případě bylo například potřeba omezit pohyb míčku, při určité nízké rychlosti jeho pohybu. Hra by byla jinak poměrně nehratelná a hráč by musel čekat neúnosně dlouhý čas na reálné zastavení míčku interagujícího s terénem. Babylon.js aktuálně integruje čtveřici motorů. Mají řadu odlišností, jako rychlost zpracování simulace, maximální počet polygonů interagujících objektů, způsoby odrazivosti nebo v realistice. Jejich podrobný popis najde žák i učitel v nápovědě k Babylon.js. Zmíníme jejich názvy, podle nichž je možno podrobnosti dohledat. Jedná se o Cannon.js - nádherný fyzikální motor napsaný zcela v JavaScriptu. Dále Oimo.js nebo Ammo.js. Motor Energy.js není aktuálně veřejně dostupný. Jedná se o port fyzikálního motoru z C++. Každý z motorů má své vlastní vlastnosti a svůj vlastní způsob výpočtu dynamiky těles. V naší hře byl použit motor Ammo.js pro svoji schopnost pracovat s 3D sítěmi vysoké hustoty a nabízet tak přesné – realistické simulace. I přes to, byl objekt míčku a terénu duplikován a zneviditelněn. Sítě duplikátů byla zjednodušena. Interpolací byl tedy redukován počet polygonů (modifikátor Optimize) těchto objektů a tyto duplikáty byly přiřazeny do výsledné simulace. Posledním krokem potřebného triku bylo, naprogramování následování pozic a rotací duplikovaného objektu originálem. Programový kód č. 8 ilustruje implementaci fyziky naší hry.

Implementované motory fyziky jsou pro hru Fantasy golf mimořádně důležité, neboť nám vytváří podstatnou část logiky naší hry. I přesto bylo zapotřebí, naprogramovat logiku herní šipky ovladatelné pomocí klávesnice tak, aby hráč mohl měnit směr šipky v prostoru a stanovit tak směr, kudy golfový míček nakonec poletí. Intenzita odpalu míčku je řešena pomocí jediného tlačítka klávesnice. Delší podržení stlačeného tlačítka, předurčuje silnější odpal míčku.

```

// Asynchronní volání
Ammo().then(
  (onResolved) => {
    // Volba motoru
    physEngine = new BABYLON.AmmoJSPlugin(false);

    // Zapnutí možností fyziky ve scéně
    scene.enablePhysics(new BABYLON.Vector3(0, -9.8, 0), physEngine);

    // Přiřazení objektů do fyzikální simulace
    terenPhysics.physicsImpostor = new BABYLON.PhysicsImpostor(terenPhysics,
    BABYLON.PhysicsImpostor.MeshImpostor, { mass: 0, friction: 10 });
    gate.physicsImpostor = new BABYLON.PhysicsImpostor(gate,
    BABYLON.PhysicsImpostor.MeshImpostor, { mass: 0, friction: 10 });
    sphereMetal.physicsImpostor = new BABYLON.PhysicsImpostor(sphereMetal,
    BABYLON.PhysicsImpostor.SphereImpostor, { mass: 1, friction: 10 });

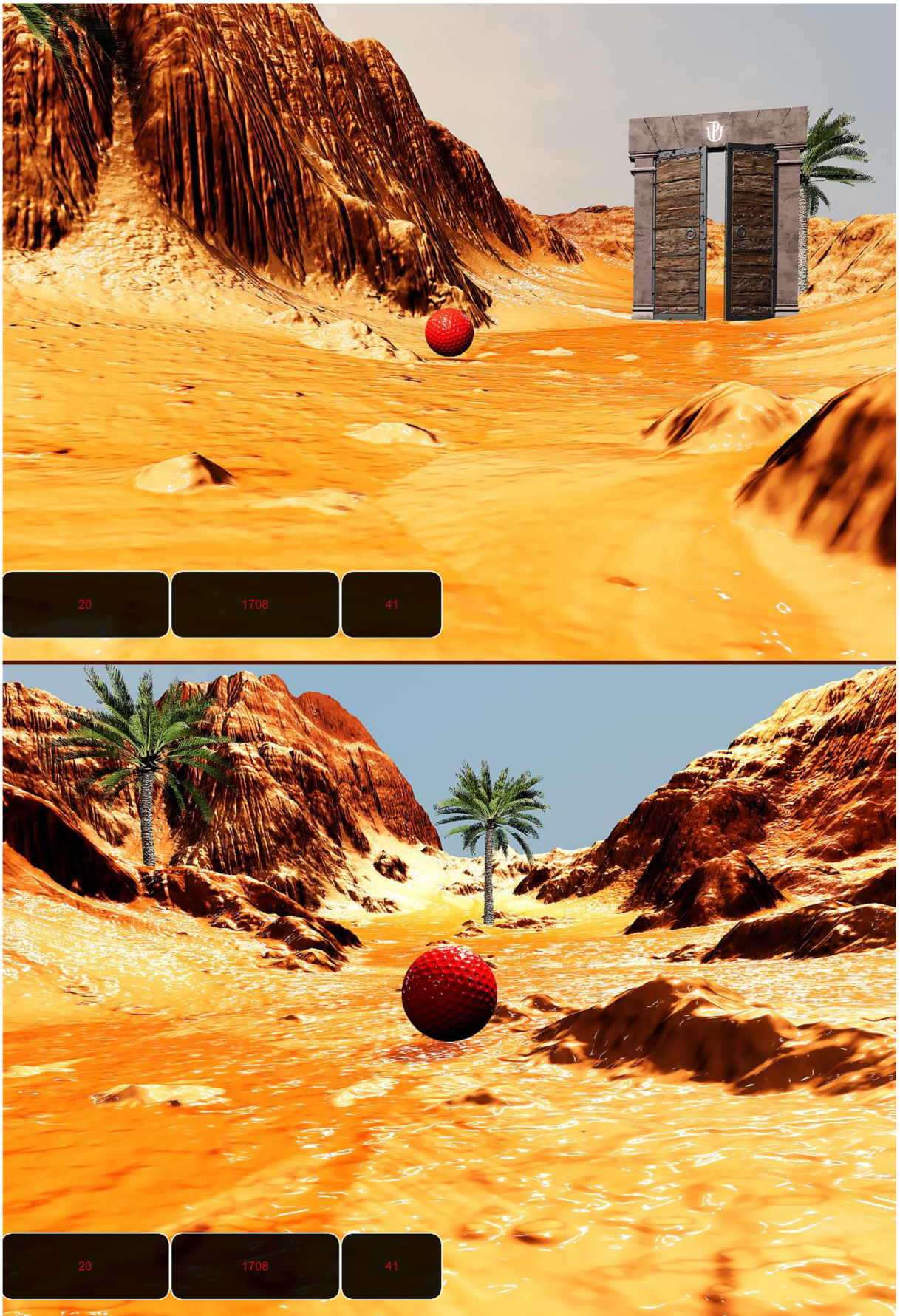
    // Nastavení rychlosti simulace
    physEngine.setTimeStep(1 / 60);

  },
  (onRejected) => {
    // Zdrojový kód pro případ selhání volání
  }
)

```

Programový kód 8: *Implementace fyziky hry*

Zdárné naprogramování herní šipky, bude od žáků předpokládat nejenom obecné základy programování, ale také matematické znalosti z oblasti analytické geometrie a vektorové algebry. Koncový bod šipky (a nakonec směrový vektor odpalu), vypočítáváme jako průnik roviny a koule. Pochopení této logiky nemusí být pro všechny žáky hned samozřejmé, a proto je vhodné, podrobně je seznámit s příloženým zdrojovým kódem a zejména jeho podrobným komentářem. Stejně tak si učitel ověří, že v matematice byly již tyto kapitoly probrány a učivo bylo osvojeno. Konečný výsledek vývoje hry Fantasy golf, ilustrujeme na tomto místě obrázkem č. 33.



Obrázek 33: Dvojice obrázků ze hry Fantasy golf

Závěr

Hlavním cílem diplomové práce bylo, vytvořit návrh metodické příručky, věnované tvorbě nenásilné 3D počítačové hry s pracovním názvem - Fantasy golf. Použité postupy vývoje hry měly být co nejvíce blízké postupům profesionálů. Metodická příručka je obsahem třetí a čtvrté kapitoly. Konstatujeme tedy, že cíl diplomové práce byl splněn. Naším tvůrčím předpokladem také bylo, vytvořit účinnou a inspirativní metodickou příručku, která intradisciplinárně nebo spíše interdisciplinárně zpracovává tvorbu počítačové hry. Metodická příručka byla plně a úspěšně použita pro řízení vlastní výuky. Můžeme proto konstatovat, že byl tento předpoklad naplněn.

Teoretická kapitola první a druhá se věnuje zodpovězení výzkumné otázky, která je možno kvalitní metodickou příručku na pozadí relevantní pedagogické teorie a naší pedagogické zkušenosti vypracovat. Jako základní teoretické východisko nám posloužil výzkum badatelky Jany Staré (2021), zejména výzkumná část zaměřená na požadavky pedagogů na metodické příručky. Podobně významným přínosem byly texty badatele Slavomíra Rysa (79), konkrétně jeho pojednání o třetím typu přípravy učitele na vyučování. Naše předložená metodická příručka tedy není kopií žádné existující příručky a její podoba byla dána výhradně vlastním výzkumem. Z naší rešerše k tématu metodických příruček vyplynulo, že doposud neexistuje dostatek autoritativních příkladů metodik tvorby metodické příručky. Proto jsme se vydali cestou vlastní.

Kapitola třetí a čtvrtá je obsahem samotnou metodickou příručku. Tyto kapitoly jsou rozděleny na teoretickou část, která odpovídá na výzkumnou otázku, jak vytvářet interaktivní 3D obsah s žáky a co nejlépe profesionálnímu pracovnímu postupu. Praktickým výstupem těchto kapitol je spustitelná počítačová hra Fantasy golf. Dostupná literatura, která tvorbu her zpracovává, nabízí téměř vždy pouze velmi základní pohled na problematiku a výstupní projekt je pro mnoho zájemců o danou problematiku graficky příliš jednoduchý, nezajímavý už od pohledu na výsledek snažení. Klasickou metodickou příručku pro školy k dané problematice je prakticky nemožné nalézt. O to více nás těší, že se nám zamýšlenou výzvu podařilo realizovat.

Tvorba této metodické příručky byla mimořádně časově náročná a nebýt skutečnosti, že bylo toto téma garantem studia předběžně odsouhlaseno už v červnu 2020 a ve stejném čase byly zahájeny první práce na ní, bylo by pro nás pravděpodobně nemožné, práci včas dokončit.

Využití této metodické příručky je nasnadě. Vzhledem k prvnímu úspěšnému nasazení ve vlastní praxi, budeme příručku i nadále využívat. Stejně tak mohou text využít kolegové, vyučující technické předměty na střední škole. Byli bychom ale stejně tak rádi, pokud by tato diplomová práce, ať už její teoretická nebo praktická část, posloužila jako inspirace dalším diplomantům.

Na tuto diplomovou práci by bylo samozřejmě možno dále navázat. Efektivita výuka s použitím této metodické příručky by mohla být například zhodnocena evaluačním výzkumem. Šetřeními bychom tak například odpověděli na hypotézu, zda žáci dosahovali lepších studijních výsledků, nežli při výhradním používání učebnice. Mohli bychom také zkoumat, zda intradisciplinární, tematické spojení počítačového programování a počítačové multimediální tvorby, povede k hlubšímu osvojení učiva našimi žáky v čase - v komparaci s klasickým separovaným přístupem. Zjištění z takového výstupu mohou posloužit k hodnotné aktualizaci naší metodické příručky a dalším konsekvencím.

Užitá literatura

AMES, Carole. *Classrooms: Goals, Structure, and Student Motivation*. APA Central Feature, s. 261-269.

BELL, Jon A. *3D Studio MAX verze 3: Efekty a design*. Praha: SoftPress, 2000. ISBN 80-902824-2-3.

BELL, Jon A. *3ds max 6: tipy a triky*. Praha: SoftPress, c2004. ISBN 80-86497-70-4.

BERTRAND, Yves. *Soudobé teorie vzdělávání*. Praha: Portál, 1998. Studium (Portál). ISBN 80-7178-216-5.

BIRN, Jeremy. *Digital Lighting & Rendering*. 3. San Francisco: New Riders Pub, 2013, 453 s. ISBN 978-0321928986.

BLAHUTKOVÁ, Marie, Jiřina KLENKOVÁ a Dana ZICHOVÁ. *Psychomotorické hry pro děti s poruchami pozornosti a pro hyperaktivní děti*. Brno: Masarykova univerzita v Brně, 2005. ISBN 80-210-3627-3.

BOARDMAN, Ted. *3DS MAX 5: podrobná příručka*. Brno: Computer Press, 2004. ISBN 80-7226-798-1.

BROOKER, Darren. *Essential CG Lighting Techniques with 3ds Max 3*. Londýn: Routledge, 2012, 393 s. ISBN 1138400815.

BRUNER, J. S. *A Study of Thinking*. Praha: Science Editions, 1967. ASIN B007KL3O1E.

CANGELOSI, James S. *Strategie řízení třídy: jak získat a udržet spolupráci žáků při výuce*. Vyd. 4. Přeložil Milan KOLDINSKÝ. Praha: Portál, 2006. Pedagogická praxe (Portál). ISBN 80-7367-118-2.

CANTOR, Diego. *Real-Time 3D Graphics with WebGL 2*. Birmingham: Packt Publishing, 2018, 896 s. ISBN 978-1-78862-969-0.

CARBONARO, Greg. *3D Studio MAX: kouzla s efekty*. Brno: UNIS Publishing, 1999. ISBN 80-86097-34-X.

ČAPEK, Robert. *Moderní didaktika: lexikon výukových a hodnoticích metod*. Praha: Grada, 2015. Pedagogika (Grada). ISBN 978-80-247-3450-7.

ČAPEK, Robert. *Líný učitel: jak učit dobře a efektivně*. Praha: Raabe, [2017]. Dobrá škola. ISBN 978-80-7496-344-5.

ČÁP, Jan. *Psychologie mnohostranného vývoje člověka: vysokoškolská příručka pro studenty filozofických a pedagogických fakult studijních oborů 77-01-8 psychologie, 75-04-8 pedagogika a psychologie a 76-12-8 učitelství všeobecně vzdělávacích předmětů aprobace psychologie*. Praha: Státní pedagogické nakladatelství, 1990. Učebnice pro vysoké školy (Státní pedagogické nakladatelství).

ČÁP, Jan a Jiří MAREŠ. *Psychologie pro učitele*. Vyd. 2. Praha: Portál, 2007. ISBN 978-80-7367-273-7.

DEWEY, John. *Demokracie a výchova*. Přeložil Josef HRŮŠA. V Praze: Jan Laichter, 1932. Laichterova filosofická knihovna.

DORAN, George. *There is a S.M.A.R.T way to write managements goals and objectives*. Management Review, s. 35-36. ISSN 1468-2370.

DVOŘÁKOVÁ, Markéta. *Projektové vyučování v české škole: vývoj, inspirace, současné problémy*. Praha: Karolinum, 2009. ISBN 978-80-246-1620-9.

FENSTERMACHER, Gary D. a Jonas F. SOLTIS. *Vyučovací styly učitelů*. Praha: Portál, 2008. ISBN 978-80-7367-471-7.

FISHER, Robert. *Učíme děti myslet a učit se: praktický průvodce strategiemi vyučování*. Praha: Portál, 1997. Pedagogická praxe. ISBN 80-7178-120-7.

FOTR, Jiří, Jan KŘENEK a Martin VLACH. *Tipy, efekty a kouzla v Adobe Photoshopu*. Praha: Computer Press, c2001. DTP & grafika. ISBN 80-7226-438-9.

FRASER, Bruce, Fred BUNTING a Chris MURPHY. *Správa barev: průvodce profesionála v grafice a pre-pressu*. Brno: Computer Press, 2003. DTP & grafika. ISBN 80-7226-943-7.

GLASSER, William. *Choice Theory: A New Psychology of Personal Freedom*. Londýn: HarperCollins e-books, 2010. ASIN B003YCOPAM.

HENDL, Jan. *Kvalitativní výzkum: základní teorie, metody a aplikace*. Čtvrté, přepracované a rozšířené vydání. Praha: Portál, 2016. ISBN 978-80-262-0982-9.

CHENARD, Julian. *Learning Babylon.js*. Liverpool: Independently published, 2017, 213 s. ISBN 978-1520730448.

CHRÁSKA, Miroslav. *Metody pedagogického výzkumu: základy kvantitativního výzkumu*. 2., aktualizované vydání. Praha: Grada, 2016. Pedagogika (Grada). ISBN 978-80-247-5326-3.

KALHOUS, Zdeněk a Otto OBST. *Školní didaktika*. Vyd. 2. Praha: Portál, 2009. ISBN 978-80-7367-571-4.

KAYSER, Wendy. *Four levels od training evaluation*. Alexandria: ATD Press, 2016, 290 s. ISBN 978-1-60728-008-8.

KENDALL, John. *The New Taxonomy Educational Objectives*. London: The University of Corwin Press, 2007. ISBN 978-1-4129-3628-6.

KENNEDY, Sanford. *3ds max 6: animace a vizuální efekty*. Brno: Computer Press, 2004. DTP & grafika. ISBN 80-251-0328-5.

KLIČKOVÁ, Marie. *Problémové vyučování ve školní praxi*. Praha: Státní pedagogické nakladatelství, 1989. Knihovnička učitele.

KOHN, Alfie. *Punished by Rewards*. London: Mariner Books, 2018. ASIN 0618001816.

KŘÍŽ, Jan. *3ds max 6: praktické postupy*. Brno: Computer Press, 2004. ISBN 80-251-0329-3.

LAMMERS, Jim a Lee GOODING. *Maya 4: kompletní průvodce*. Praha: SoftPress, c2002. ISBN 80-86497-30-5.

LEE, Kim. *Mistrovství v 3ds max 4*. Praha: SoftPress, 2002. ISBN 80-86497-19-4.

LOOSE, Duan. *3D Studio Max 3.0 Workshop*. Chicago: Hayden Books, 2000, 600 s. ISBN 978-0789723437.

LOOSE, Duane. *3DS MAX 4*. Praha: Computer Press, 2001. DTP & grafika. ISBN 80-7226-587-3.

LUNA, Frank a Andreas SCHWILL. *Introduction to 3D Game Programming with DirectX 11*. 2012. Herndon: Mercury Learning & Information, 2011, 600 s. ISBN 978-1936420223.

MACCONVILLE, Ruth. *Building Happiness and Motivation in Adolescents: A Positive Psychology Curriculum for Well-Being*. Londýn: Jessica Kingsley Publishers, 2012. ISBN 978-1849052610.

MAGER, Robert. *Preparing Instructional Objectives*. California: Fearon Publisher, 1962. ASIN B000NXREO4.

MAŇÁK, Josef a Vlastimil ŠVEC. *Výukové metody*. Brno: Paido, 2003. ISBN 80-7315-039-5.

MAREŠ, Jiří. *Styly učení žáků a studentů*. Praha: Portál, 1998. Studium (Portál). ISBN 80-7178-246-7.

MARZANO, Robert. *Designing & Teaching Learning Goals & Objectives*. Bloomington: Marzano Research Laboratory, 2009, 896 s. ISBN 978-0-9822592-0-7.

MILLER, Tom. *Programujeme 3D hry v jazyce C#: [tři kompletní hry od úplných základů]*. Brno: Computer Press, 2006. ISBN 80-251-1126-1.

MOREAU, Julien. *Babylon.js Essentials*. Birmingham: Packt Publishing Ltd, 2016, 293 s. ISBN 978-1-78588-479-5.

MURDOCK, Kelly. *3ds Max 9 Bible*. 1. Chichester: John Wiley, 2007, 1296 s. ISBN 13: 978-0470100899.

NAKONEČNÝ, Milan. *Sociální psychologie*. Praha: Stanislav Juhaňák - Triton, 2020. ISBN 978-80-7553-842-0.

OAKES, Jeanne a Linda DARLING. *Preparing Teachers for Deeper Learning*. 1. Cambridge: Harvard Education Press, 2019, 416 s. ISBN 978-1682532928.

OKOŇ, Wincenty. *K základům problémového učení*. Praha: Státní pedagogické nakladatelství, 1966. Pedagogická teorie a praxe.

PASCH, Marvin. *Od vzdělávacího programu k vyučovací hodině: jak pracovat s kurikulem*. Praha: Portál, 1998. ISBN 80-7178-127-4.

PEHLIVANIAN, Ara a Don NGUYEN. *JavaScript okamžitě*. 2. vydání. Přeložil Ondřej BAŠE. Brno: Computer Press, 2021. ISBN 978-80-251-5025-2.

PETTY, Geoffrey. *Moderní vyučování*. 6., rozš. a přeprac. vyd. Přeložil Jiří FOLTÝN. Praha: Portál, 2013. ISBN 978-80-262-0367-4.

PRŮCHA, Jan. *Moderní pedagogika*. Šesté, aktualizované a doplněné vydání. Praha: Portál, 2017. ISBN 978-80-262-1228-7.

PRŮCHA, Jan, Jiří MAREŠ a Eliška WALTEROVÁ. *Pedagogický slovník*. 4. aktualiz. vyd. Praha: Portál, 2003. ISBN 80-7178-772-8.

REISER, Brian. *A Learning Goals Driven Design Model for Developing Science Curriculum*. Michigan: University of Michigan, 2007, 34 s.

RYS, Slavomír. *Příprava učitele na vyučování*. Praha: Státní pedagogické nakladatelství, 1979. Pedagogická teorie a praxe.

ŘEHÁČEK, David. *3D Studio MAX 2: uživatelská příručka*. Brno: Computer Press, 1998. DTP & grafika. ISBN 80-7226-104-5.

SHELDRIK, Wayne. *Motivating students*. Ottawa: ICT Books, 2017, 77 s. ASIN B00D60U8D0.

SHREINER, Dave. *OpenGL: průvodce programátora*. Brno: Computer Press, 2006. DTP & grafika. ISBN 80-251-1275-6.

SCHUBERT, Sigrid a Andreas SCHWILL. *Didaktik der Informatik*. 2011. Heidelberg: Spektrum Akademischer Verlag, 2011, 416 s. ISBN 978-3-8274-2652-9.

SKALKOVÁ, Jarmila. *Obecná didaktika: vyučovací proces, učivo a jeho výběr, metody, organizační formy vyučování*. Praha: Grada, 2007. Pedagogika (Grada). ISBN 978-80-247-1821-7.

STARÁ, Jana. *Práce učitelů s učebnicemi*. Praha: Univerzita Karlova, Pedagogická fakulta, 2021. ISBN 978-80-7603-081-7.

STEED, Paul. *Animace postav: 3ds max a character studio*. Brno: Computer Press, 2004. ISBN 80-251-0030-8.

STEJSKALOVÁ, G. *Tvorba učebnice IKT pro ZŠ Křižná*. Zlín. Diplomová práce. Univerzita Tomáše Bati, 2013. Vedoucí práce doc. RNDr. Ladislav Dušek, Ph.D.

ŠUMAVSKÁ, G. *Jak a čím motivovat žáky ke studiu a vést je k odpovědnosti*. Kurikulum S [online], 2010. Dostupné na: <http://www.nuov.cz/kurikulum/jak-a-cim-motivovat-zaky-ke-studiu-a-vest-je-k-odpovednosti>.

TYLER, Ralph. *Basic Principles of Curriculum and Instruction*. Chicago: The University of Chicago Press, 1949. ISBN 0-226-82031-9.

UHER, Jan. *Základy americké výchovy*. Praha: Čin, 1930. Knihovna české mysli.

VÁVRA, Jaroslav. *Didaktika geografie 1: od vzdělávacího programu k vyučovací hodině v zeměpisu na ZŠ, na příkladu tématu Světový oceán*. Liberec: Technická univerzita v Liberci, 2006. ISBN 80-7372-083-3.

WALNUM, Clayton. *Programujeme grafiku v Microsoft Direct3D*. Brno: Computer Press, 2004. ISBN 80-251-0136-3.

WAYNE, Freeze. *Windows Game Programming with Visual Basic and DirectX*. Calgary: Que, 2001, 381 s. ISBN 978-0789725929.

ŽÁRA, Jiří. *Moderní počítačová grafika*. 2., přeprac. a rozš. vyd. Brno: Computer Press, 2004. ISBN 80-251-0454-0.

ŽÁRA, Ondřej. *JavaScript: programátorské techniky a webové technologie*. Brno: Computer Press, 2015. ISBN 978-80-251-4573-9.

WAYNE, Freeze. *Windows Game Programming with Visual Basic and DirectX*. Calgary: Que, 2001, 381 s. ISBN 978-0789725929.

Seznam obrázků

Obrázek 1: <i>Didaktické texty (Převzato z: Stejskalová 2009)</i>	8
Obrázek 2: <i>Preferované vlastnosti metodických příruček (Převzato z: Stará 2021)</i>	10
Obrázek 3: <i>Command panel</i>	40
Obrázek 4: <i>2D a 3D primitiva 3ds Max</i>	41
Obrázek 5: <i>Základní modifikátory 3ds Max a jejich účinky</i>	42
Obrázek 6: <i>Příklad užití modifikátoru Lathe</i>	43
Obrázek 7: <i>Pracovní výřez s aplikovaným modifikátorem Loft</i>	45
Obrázek 8: <i>Vizualizace šroubováku</i>	46
Obrázek 9: <i>Příklad použití ProBoolean</i>	48
Obrázek 10: <i>Příklad použití modifikátoru Displace</i>	50
Obrázek 11: <i>Pohled na základní funkce modifikátoru Edit Poly</i>	51
Obrázek 12: <i>Sub-objekty modelu</i>	52
Obrázek 13: <i>Aplikovaný MeshSmooth</i>	52
Obrázek 14: <i>Základní funkce modifikátoru Edit Poly</i>	53
Obrázek 15: <i>Nastavení funkce Chamfer</i>	55
Obrázek 16: <i>Redukce velikosti polygonu</i>	56
Obrázek 17: <i>Na míček aplikovaný MeshSmooth, vpravo vizualizován</i>	57
Obrázek 18: <i>GUI programu WorldMachine</i>	59
Obrázek 19: <i>Příklad zařízení</i>	60
Obrázek 20: <i>Zařízení generující herní terén</i>	62
Obrázek 21: <i>Perlinův šum, kolorovaný</i>	63
Obrázek 22: <i>Výsledný objekt terénu</i>	64
Obrázek 23: <i>Soustava zařízení pro tvorbu textury terénu</i>	66
Obrázek 24: <i>Ambient Occlusion</i>	68
Obrázek 25: <i>Tvorba textur terénu</i>	69
Obrázek 26: <i>Výsledná textura terénu</i>	70
Obrázek 27: <i>Spekulární textura terénu</i>	71
Obrázek 28: <i>Normálová mapa</i>	72
Obrázek 29: <i>Material Editor 3ds Max</i>	73
Obrázek 30: <i>Exportér Babylon.js v 3ds Max</i>	74
Obrázek 31: <i>Správně nainstalovaný plugin Babylon.js</i>	74
Obrázek 32: <i>Vizualizace dětské láhve</i>	76
Obrázek 33: <i>Dvojice obrázků ze hry Fantasy golf</i>	96

Seznam programových kódů

- Programový kód 1: *Základní projekt Babylon.js*..... **Chyba! Záložka není definována.**
- Programový kód 2: *Vytvoření objektu krychle* **Chyba! Záložka není definována.**
- Programový kód 3: *Situování objektu v herním světě* **Chyba! Záložka není definována.**
- Programový kód 4: *Nastavení maximálního počtu světel ve scéně* ... **Chyba! Záložka není definována.**
- Programový kód 5: *Nastavení vlastností hlavního světla - "slunce"*. **Chyba! Záložka není definována.**
- Programový kód 6: *Implementace stisku tlačítka klávesnice* **Chyba! Záložka není definována.**
- Programový kód 7: *Nastavení kamery* **Chyba! Záložka není definována.**
- Programový kód 8: *Implementace fyziky hry* **Chyba! Záložka není definována.**

ANOTACE

Jméno a příjmení:	Ing. Jan Mikulčík
Katedra:	Katedra technické a informační výchovy
Vedoucí práce:	Mgr. Jan Kubrický, Ph.D.
Rok obhajoby:	2022

Název práce:	Návrh didaktického textu pro učitele SŠ k problematice tvorby 3D počítačových her
Název v angličtině:	Design of a didactic text for high school teachers on the issue of creating 3D computer games
Anotace práce:	<p>Diplomová práce se zabývá oblastí tvorby počítačových her ve výchovně vzdělávací realitě středních škol.</p> <p>Teoretická kapitola první a druhá se věnuje zodpovězení výzkumné otázky, která je možno kvalitní metodickou příručku k tématu tvorby počítačových her vypracovat. Děje se tak na základě relevantní pedagogické teorie.</p> <p>Kapitola třetí a čtvrtá je obsahem samotnou metodickou příručku. Tyto kapitoly jsou rozděleny na teoretickou část, která odpovídá na výzkumnou otázku, jak vytvářet interaktivní 3D obsah s žáky a co nejbližší profesionálnímu pracovnímu postupu. Praktickým výstupem těchto kapitol je spustitelná počítačová hra Fantasy golf</p>
Klíčová slova:	Tvorba metodické příručky, videohry, tvorba počítačových her, modelování 3D objektů,

	počítačová animace, tvorba textur, programování počítačových her, Babylon.js, WebGL.
Anotace v angličtině:	<p>The diploma thesis deals with the area of creating computer games in the educational reality of secondary schools.</p> <p>The theoretical chapters of the first and second are devoted to answering the research question of how a quality methodological manual on the topic of creating computer games can be developed. This is done on the basis of relevant pedagogical theory.</p> <p>The third and fourth chapters are the content of the methodological manual itself. These chapters are divided into a theoretical part, which answers the research question of how to create interactive 3D content with students and as close as possible to a professional workflow. The practical output of these chapters is the executable computer game Fantasy golf.</p>
Klíčová slova v angličtině:	Creation of methodical manual, video games, creation of computer games, modeling of 3D objects, computer animation, creation of textures, programming of computer games, Babylon.js, WebGL.
Přílohy vázané k práci:	DVD obsahující programový kód hry a pracovní soubory z 3ds Max
Rozsah práce:	107 stran, DVD příloha
Jazyk práce:	Český jazyk