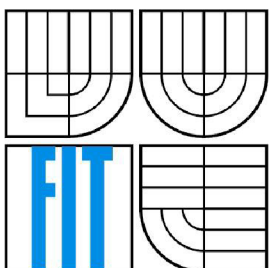


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# VÝVOJ 3D APLIKACÍ V PROSTŘEDÍ BLENDER

BLENDER 3D APPLICATION DEVELOPMENT

## DIPLOMOVÁ PRÁCE

DIPLOMA THESIS

## AUTOR PRÁCE

AUTHOR

Bc. MARTIN OPLETAL

## VEDOUCÍ PRÁCE

SUPERVISOR

prof. Dr. Ing. PAVEL ZEMČÍK

BRNO 2014

## Abstrakt

Práce se zabývá srovnáním a hodnocením dostupných nástrojů pro tvorbu 3D aplikací. Ve zvoleném nástroji (Blender) je vytvořena počítačová hra, která demonstruje jeho možnosti. Postup tvorby je zaznamenán v případové studii pro možné účely vlastní tvorby komplexní hry.

## Abstract

Thesis is focused on evaluation of available tools for 3D application development. In one selected tool (Blender) is created computer game. This game demonstrates Blender's capabilities. The process of creating is recorded in a case study for the possible purposes of own complex games development.

## Klíčová slova

Blender, herní jádro, tvorba her, tvorba grafického prostředí, 3D modelační nástroje.

## Keywords

Blender, game engine, game development, grafic enviroment development, 3D modelling tools.

## Citace

Opletal Martin: Vývoj 3D aplikací v Blenderu, diplomová práce, Brno, FIT VUT v Brně, 2014



## Prohlášení

Prohlašuji, že jsem tuto práci vypracoval samostatně pod vedením prof. Dr. Ing. Pavla Zemčíka. Další informace mi poskytly zdroje uvedené v sekci literatury, zejména zdroje [1] a [2], ze kterých vychází podstatná část práce. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Martin Opletal  
28. května 2014

## Poděkování

Děkuji panu prof. Dr. Ing. Pavlu Zemčíkovi za cenné rady při tvorbě této diplomové práce, profesorskému sboru za znalosti, které mi pomohly zvládnout problematiku práce, a vývojářům nástroje Blender, bez kterých by tato práce nemohla vzniknout.

© Bc. Martin Opletal, 2014

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

1 Úvod.....	1
2 Vývojové nástroje pro 3D aplikace.....	2
2.1 CryENGINE 3.....	2
2.2 Unreal Engine 4.....	3
2.3 Unity 4.....	4
2.4 Blender.....	5
2.5 Další nástroje.....	6
3 Blender a jeho součásti.....	7
3.1 Historie.....	7
3.2 Rozhraní.....	8
3.3 Modelování.....	12
3.4 Modifikátory.....	15
3.5 Materiály.....	15
3.6 Textury.....	16
3.7 Animace.....	18
3.8 Rozšíření.....	20
3.9 Vestavěné enginy.....	21
4 Vyhodnocení nástrojů a volba studie.....	25
4.1 Výběr 3D nástroje.....	25
4.2 Zaměření studie.....	26
5 Případová studie o Blender hře.....	28
5.1 Stažení modelů.....	28
5.2 Tvorba vlastních modelů – ateliér.....	29
5.3 Tvorba vlastních modelů – hlavní postava.....	35
5.4 Materiály a textury.....	38
5.5 Renderování a jeho úprava.....	44
5.6 Animování.....	46
5.7 Herní objekty.....	51
5.8 Blender Game.....	58
5.9 Výdej hry.....	78
6 Závěr.....	81
Reference.....	82
Příloha A.....	84

# 1 Úvod

Dnešní doba umožňuje mít ve většině domácností osobní počítač, notebook či jiná zařízení. Tato zařízení se využívají nejen pro kancelářskou práci nebo přehrávání videí a hudby. Slouží také jako prostředek pro hraní počítačových her, které slouží pro výuku nebo pro zábavu. Hry se dnes skládají převážně z obrazové a zvukové informace ze strany počítače, kterou hráč vnímá pomocí výstupních zařízení: obrazovky, sluchátek, reproduktorů a podle toho náležitě interaguje. Hráč odpověď přenáší do počítače za použití vstupních zařízení, které jsou známy jako klávesnice, myš a další (specializované pro hry). Nové počítačové hry mívají především propracovanou (a stále dále vylepšovanou) obrazovou složku, které se říká grafika, a znamená pro uživatele hlavní zdroj informací. Na vývoj takovýchto složitých her se využívají pokročilé počítačové nástroje, na které se tato práce zaměřuje.

Nástroj pro tvorbu her a jejich běh je označován jako herní engine (v doslovném překladu herní motor). Někdy je používán termín herní jádro, kde právě jádro nebo engine vyjadřuje systém programů umožňující tvorbu a chod videohry. Videohry běží na elektronických přístrojích, jejichž valnou většinu dnes tvoří osobní počítače, herní konzole a aktuálně čím dál častěji se uplatňující mobilní telefony a tablety. Enginy umožňují urychlení vývoje her, jelikož podstatný základ hry je již v enginu vytvořen. Dále umožňují přímý vizuální vjem, jak prostředí hry bude vypadat přímo při tvorbě, čímž značně usnadňují práci. Herní studia vyvíjí mnoho her za relativně krátkou dobu. Zásahu za tuto produktivitu nesou právě technologie herních enginů a jejich zasazení do vývojových nástrojů, které vývojáře distancují od těžkopádného kódu a umožňují tvorbu her v přehledném grafickém rozhraní.

Některé ze známých herních enginů a modelačních nástrojů jsou v práci popsány. Jeden z těchto nástrojů, Blender, byl po zhodnocení výhod zvolen pro případovou studii tvorby 3D aplikace, na kterou se tato práce zaměřuje. Blender je vývojové prostředí určené především pro vytváření 3D modelů a renderování (vykreslování) 3D scény, ale slouží již i pro tvorbu her. Lze využít k vizualizačním projektům, kinematografii, ke tvorbě simulací nebo her. Jedná se o softwarovou sadu nástrojů a systémů, kterými je možné vytvořit trojrozměrnou (dokonce fotorealistickou) virtuální scénu. Blender byl ze všech možných nástrojů určených pro tvorbu 3D aplikací a her k projektu vybrán pro svou výjimečnou pokročilost (vizuální kvalitu výsledného díla a efektivitu tvorby 3D prostředí). Existují, jak se dozvíte, nástroje ještě propracovanější a proto komerčně používané. Blender byl ovšem primárně vybrán na základě toho, že se jedná o open-source nástroj. Je tedy zdarma pro nekomerční, ale dokonce také komerční účely. Pro amatérskou komunitu tvůrců vizualizačních projektů a počítačových her, i pro zkušené profesionály.

Terminologie je někdy v tomto počítačovém oboru složitá a cizí názvy jsou i v české literatuře frekventované. Proto jsou v práci používány i cizojazyčné výrazy, kde jejich význam je předem čtenáři nastíněn. Veškeré obrázky uvedené v práci jsou vlastním přínosem a jsou vytvořeny snímáním obrazovky s rozhraním Blenderu (pouze snímky z filmů a her jsou převzaty z jiných zdrojů).

V následující kapitole jsou uvedeny některé nejdůležitější nástroje určené pro vývoj 3D aplikací (animací, her, simulací) s jejich využitím v praxi, dostupností, efektivitou tvorby a grafickou výkonností. Ve třetí kapitole je důkladněji popsán vybraný nástroj – Blender. Jsou zde rozvedeny také podstatné součásti Blenderu, které jsou využity v případové studii. Ve čtvrté kapitole je provedeno zhodnocení popsaných nástrojů, vysvětlena volba nástroje Blender a zaměření případové studie. V páté kapitole je uvedena samotná případová studie se zvolenou tematikou tvorby počítačové hry.

## 2 Vývojové nástroje pro 3D aplikace

Tato kapitola pojednává o vlastnostech nástrojů, které mohou být využity pro tvorbu 3D aplikace. Na následujících stranách o vývojových nástrojích pro 3D aplikace (primárně hry) je uvedeno, které nástroje jsou aktuálně dostupné na trhu. Jsou upřesněny jejich grafické schopnosti, možnosti využití, k čemu se v praxi již používají a efektivita práce s nimi. Tyto vlastnosti jsou podstatné pro výběr vhodného nástroje k tvorbě 3D aplikace.

### 2.1 CryENGINE 3

CryENGINE 3 je (podle zdroje [13]) vysoce pokročilé řešení vývoje:

- her,
- filmů,
- simulací,
- interaktivních aplikací.

CryENGINE poskytuje mnoho vizuálních efektů a vylepšení, aby vytvářený obraz her byl co nejrealističtější s ohledem na zpracování obrazu člověkem.

- **Adaptace oka a HDR** – využívá se k simulaci reakce člověka na náhlé extrémní změny v osvětlení (přechod postavy z temné místnosti do prosvětleného venkovního prostředí). HDR (High Dynamic Range) zase poskytuje realističtější vykreslení scén s extrémním jasnem a širokou škálou kontrastů.
- **Deferred Lighting** – efektivní vykreslení vysokého počtu světelných zdrojů ve scéně. Využívá per-pixel shading.
- **Screen Space Ambient Occlusion (SSAO)** – poskytuje simulaci, která uchovává rohy a hrany scény tmavé. Výhodou je nízká režie na všech podporovaných platformách s velmi kvalitním výsledkem osvětlení.
- **Tessellation** – CryENGINE podporuje DX11 HW akcelerovanou teselaci objektů, kde novinkou je real-time aktivace teselace. CryEngine podporuje 3 různé typy teselace, které lze prolnout na jediném objektu a vytvořit tak komplexní tvary a siluety. Jedním typem je displacement mapping (komplexní siluety nebylo možné vytvořit pomocí bump mapping). Další dva typy se zaměřují na zaoblení hranatých tvarů objektů.
- **Natural Lighting & Dynamic Soft Shadows** – přirozené osvětlení scény s nízkými nároky na platformy, které vytváří měkké stíny s vysokým rozlišením. Stíny se dynamicky mění v závislosti na pohybu objektů.
- Mnoho dalších efektů, jako real-time Local Reflections (RLR), Parallax Occlusion Mapping (POM), Screen Space Directional Occlusion (SSDO), Color Grading, Particle Shading, Light Propagation Volume, real-time Dynamic Global Illumination, Irradiance Volume, které lze najít na stránkách [13].

Mezi výhody CryENGINE 3 patří následující aspekty.

- CryENGINE 3 je údajně (zdroj [13]) nejrychlejší renderer na světě, který poskytuje téměř fotorealistické prostředí a real-time speciální efekty. Engine zahrnuje implicitní optimální podporu na aktuálně využívané platformě.
- CryENGINE 3 obsahuje také flexibilní systém umělé inteligence, která se skládá z modulárního sensorového systému. Tento systém může pro každou postavu hry implementovat vlastní vidění, slyšení a další vjemy.

- Do enginu byl také zahrnut integrovaný systém pro tvorbu reálně vyhlížejících postav, který zahrnuje systém pro tvorbu jedinečných tváří a animování založené na kostře postav.
- Další výhodou enginu je Live Create. Jedná se o tvorbu práce na jediném editoru ve stylu WYSIWYP (what you see is what you print – co vidíš, to tiskneš) a výsledek lze rovnou vidět a hrát na všech připojených platformách.
- Engine má vestavěn fyzikální systém, který poskytuje realistické reakce objektů na aplikované síly (gravitace, vítr, exploze, kolize s dalšími objekty).

Příklady her vytvořených v CryEngine 3 jsou (zdroj [16]):

- Crysis 3,
- Sniper: Ghost Warrior 2,
- Cabal Online 2,
- Aion,
- MechWarrior Online.

Tento (v řadě třetí) engine, který je vyroben společností Crytek, je využitelný na platformách PC, Xbox 360 a PlayStation 3. Engine lze zakoupit (v době citace z oficiálních stránek Crytek [13]) za 9.90 dolarů/eur na měsíc. Pro nekomerční účely lze celý engine pořídit bezplatně. V současné době se pracuje na následující verzi enginu – CryENGINE 4.

## 2.2 Unreal Engine 4

Unreal Engine 4 (UE4) je dalším vyspělým zástupcem herních enginů. UE4 je (podle zdroje [14]) určen k tvorbě:

- her,
- filmů,
- simulací a vizualizací.

UE4 podporuje pokročilé prvky renderování umožněné v DirectX 11:

- full-scene HDR reflections,
- tisíce dynamických světel na scéně,
- teselaci a displacement,
- fyzikálně založený shading s materiály,
- další.

Unreal Engine 4 obsahuje mnoho pokročilých funkcí a nástrojů, kterými lze efektivně vytvářet herní svět. Některé z nich jsou popsány níže (ze zdroje [14]).

- UE4 zahrnuje kaskádový VFX editor, který poskytuje nástroje k tvorbě detailního ohně, kouře, prachu a jiných jevů. Kaskádová pipeline umožňuje rychlou a HW nenáročnou simulaci částic a poskytuje kolizní systém.
- Dále je enginem poskytován „Blueprint Visual Scripting“, kterým lze vytvářet hry bez použití programovacího jazyka s využitím vytváření uzlů grafu hry. Pro ulehčení ladění hry bylo přidáno „Live Blueprint Debugging“, kterým lze sledovat hodnoty vlastností hry i za jejího běhu. Hru lze kdykoliv pozastavit na základě breakpointů v uzlech grafu.
- UE4 má integrován i „Landscape system“, kterým lze tvořit terén scény. Terén může být tvořen ve větším měřítku než dříve, díky nápomocnému LOD (Level of detail – úroveň

detailu) systému a efektivnímu využití paměti. Také „foliage tool“ pomáhá efektivnějšímu zasazení květin a jiných objektů do terénu.

- Engine obsahuje propracovaný systém umělé inteligence. Postavy lépe reagují na své okolí, lze je navigovat pomocí „navigation mesh“ umístěného ve scéně a případně trasu dynamicky měnit při herních eventualitách.
- V Enginu je velké množství již hotových herních objektů, materiálů a jiných aktiv, které lze procházet v přehledném prohlížeči – „Content Browser“.
- Herní jádro je přístupné a libovolně modifikovatelné pomocí jazyka C++. Lze měnit například fyzikální subsystémy, rendering, audio systém nebo i editor UE4. Je přítomno také „Code View“, kterým lze přímo procházet C++ kód na herních postavách. „Hot Reload“ zase umožňuje měnit kód za běhu hry a přímo tak vidět jeho vliv na hru.

Mezi reprezentativní díla vytvořená v Unreal Engine (převážně UE3, jelikož čtvrtá verze vyšla nedávno) patří (podle zdroje [16]):

- Gears of War,
- Infinity Blade,
- Mass Effect,
- Dishonored,
- Bioshock Infinite a další.

Pomocí UE4 lze tvořit aplikace (převážně hry) pro platformy Windows, Xbox One, OS X, Linux, PS 4, iOS, Android a JS/WebGL. Engine lze pořídit za 19 dolarů na měsíc + 5% z výtěžku (v době citace ze zdroje [14]). Pro nekomerční účely lze získat UDK (Unreal Development Kit) zdarma.

## 2.3 Unity 4

Unity (podle zdroje [15]) slouží pro tvorbu:

- her (2D i 3D),
- interaktivního obsahu.

Unity podporuje DirectX 11 API. Využívá shader model 5.0 a poskytuje mnoho pokročilých grafických efektů.

- Nechybí dnes čím dál častěji používaná teselace pro bohatší detaily modelů.
- Engine poskytuje deferred lighting (zmiňovaný v předchozích enginech), HDR rendering nebo Occlusion Culling, který redukuje počet vykreslovaných objektů.
- Unity engine má zabudován 100 shaderů, od jednoduchých (Diffuse, Glossy a dalších) po složité (Self Illuminated Bumped Specular).
- Unity má integrován „lightmapping tool“, kterým lze „zapékat“ osvětlení do textur pro vyšší výkon hry. Dále lze scénu vylepšit „zapečenou“ Global Illumination, sky lights a emissive materials.
- V Unity 4 se nachází optimalizovaný Motion Blur, Depth Of Field, vinětace, Tonemapping a spousta dalších pokročilých efektů a funkcí.

Unity 4 obsahuje mnoho užitečných nástrojů a možností.

- Do Unity lze importovat modely a animace z téměř jakékoliv 3D aplikace (Maya, 3ds Max, Blender, ...).

- Engine zahrnuje i nástroje na tvorbu terénu a zasazení rostlin a jiných objektů do něj.
- Unity obsahuje fyzikální engine NVIDIA® PhysX® Physics, kterým je možné tvořit rozbitelné objekty, exploze nebo i látky a vlasy/srst vlající ve větru.
- Další užitečnou funkcí je vestavěný pathfinding pro nacházení trasy v terénu s využitím automatického generování navigation mesh.
- Výkon vylepšuje pomocí compute shaderů, kterými lze využít síly GPU jako paralelního CPU.
- Je možné mixovat 3D a 2D hry, jelikož 2D verze jsou tvořeny plochami a k těmto plochám lze jednoduše přidat 3D objekty.
- Hry lze programovat v několika jazycích, jako jsou JavaScript, C# nebo Boo, které se využijí na platformě Open Source .NET, Mono.
- Debugging probíhá v MonoDevelop (pro Windows a Mac). Hru lze pozastavit, nastavit breakpointy a ověřit hodnoty proměnných celé hry.

Mezi díla vytvořená v Unity patří hry:

- Bad Piggies,
- Dead Trigger 2,
- Endless Legend,
- World Zombination,
- další.

Unity Pro lze pořídit za 1 500 dolarů nebo 75 dolarů měsíčně (již se pracuje na Unity 5). Unity Technologies, vývojář Unity, umožňuje i bezplatné využívání volné verze engine, pokud je výdělek z práce na Unity menší jak 100 000 dolarů. Další možností je zkouška trial verze Unity Pro, která je zdarma po dobu jednoho měsíce. Unity umožňuje výdej vytvořených aplikací na mnoho platform, jako jsou Windows Phone, Android, iOS, BlackBerry, Mac, Windows, Linux, webové prohlížeče (Chrome, Firefox, Internet Explorer, Safari – přes Unity Web Player), Sony PS3 a SP4, Xbox One, Wii U.

## 2.4 Blender

Blender je softwarový nástroj pro tvorbu (ze zdroje [2]):

- 3D modelů,
- animací (včetně animovaných filmů),
- počítačových her,
- interaktivních aplikací.

Blender je multiplatformní, takže je možné ho využít na tradičních platformách, jako jsou Windows, Linux, FreeBSD, Mac OS X a další. Spadá pod všeobecnou veřejnou licenci GNU (GNU GPL) a je tedy možné tento nástroj využít zdarma a to i pro komerční účely.

GNU General Public License (GNU GPL) je kvalitně popsána na stránkách GNU [3]. Tato licence umožňující na rozdíl od ostatních omezujících licencí svobodu distribuce, libovolnou úpravu kódů ve všech verzích programů již vzniklých a jejich další distribuci. Tvorbu děl v programech/software pod touto licencí, aniž by na autora nového díla spadala omezující licence ze softwaru, v kterém dílo vytvořil. GNU GPL je tedy velmi populární licence s copyleftovým zaměřením – na odvozená díla spadá opět stejná copyleftová licence. Je to jistě míry opakem copyrightové licence (od toho také vzniklo slovo copyleft, jako hříčka se slovy), která od uživatele práva odebírá. Uživatel softwaru

musí splnit všechny body ustanovené licence, jinak může být trestně stíhán. U copyrightu jde především o zabránění tvorby odvozených programů a jejich další distribuce.

## 2.5 Další nástroje

Výše zmíněné enginy patří mezi nejčastěji se vyskytující středobody pro tvorbu her a vizualizaci diskutované na internetu. Mezi dále využívané nebo v minulosti využitá herní jádra je vhodné zmínit (ze zdroje [16]):

- Frostbite 3 (platformy: PC, Xbox One, Xbox 360, PS 3, PS 4; použito v: Battlefield 3, Battlefield 4 (obr. 1 ze zdroje [17])). Informace o Frostbite 3 jsou uvedeny ze zdroje [17]),
- GameBryo (platformy: Android, iOS, PC, PS3, Wii, Wii U, Xbox 360; použito ve hrách: Warhammer Online: Age of Reckoning, The Elder Scrolls III a IV, Sid Meier's Civilization IV, Fallout 3),
- HeroEngine (platformy: PC; použito v: Star Wars: The Old Republic),
- BigWorld (platformy: prohlížeče, PC; použito v: World of Tanks, Realm of the Titans.),
- mnoho dalších.



Obrázek 1 – Snímek ze hry Battlefield 4

Jedná se ovšem stále o pouhou špičku ledovce, herních enginů je mnohem větší množství a vývojáři mají z čeho vybírat. Ať už jde o grafickou kvalitu, vývojové prostředí nebo cenu těchto nástrojů.



## 3 Blender a jeho součásti

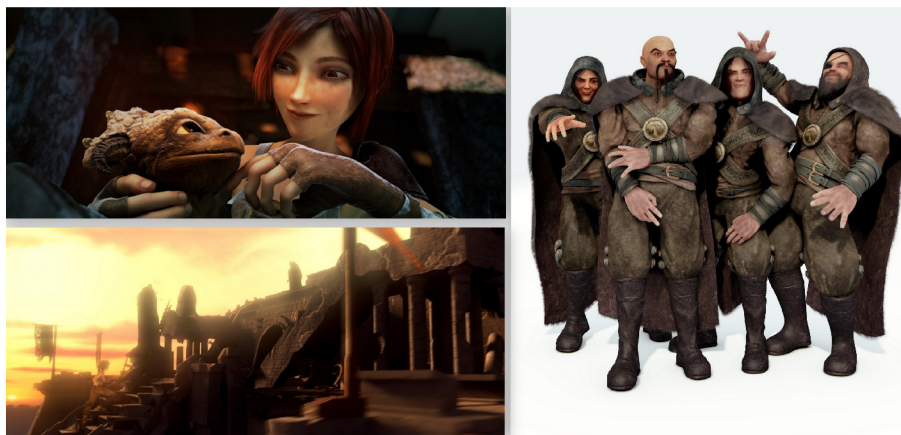
Tato kapitola shrnuje důležité poznatky o Blenderu, nástroji pro tvorbu 3D aplikací. Jsou uvedeny především podstatné součásti Blenderu, které je důležité znát pro implementační fázi práce.

### 3.1 Historie

Hlavní přínos Blenderu se podle [1] a [2] přisuzuje spoluzakladateli holandského animačního studia NeoGeo (1988) – Tonu Roosendaalovi. Roosendaal stál v čele vývoje podnikového softwaru společnosti NeoGeo, který byl od roku 1995 přepisován na softwarový nástroj – prvotního předka Blenderu dnešní podoby. Roosendaal roku 1998 založil společnost Not a Number (NaN), která se dále zaměřovala na Blender. NaN získala investory a při jejich finanční podpoře růst Blenderu pokračoval. Bohužel finanční podpora po pár letech umkla a společnost se roku 2002 dostala do potíží. Vývoj Blenderu byl tímto v NaN pozastaven.

Roosendaal neotálel dlouho a po pár měsících pozastaveného vývoje softwaru našel řešení v neziskové nadační organizaci Blenderu. Jedním z prvních cílů bylo získat 100 000 eur k zaplacení nákladů dosavadních investorů a získání tak jejich souhlasu k volné distribuci Blenderu. Částka se stihla vybrat v rekordních sedmi týdnech, jak tvrdí [2]. Tímto se Blender stal volně dostupným 3D modelačním softwarovým nástrojem s již GNU GPL licencí (13. říjen 2002). Organizace se až dodnes stará s pomocí mnoha dobrovolníků a samotného Tona Roosendaala o vývoj Blenderu v rámci open-source.

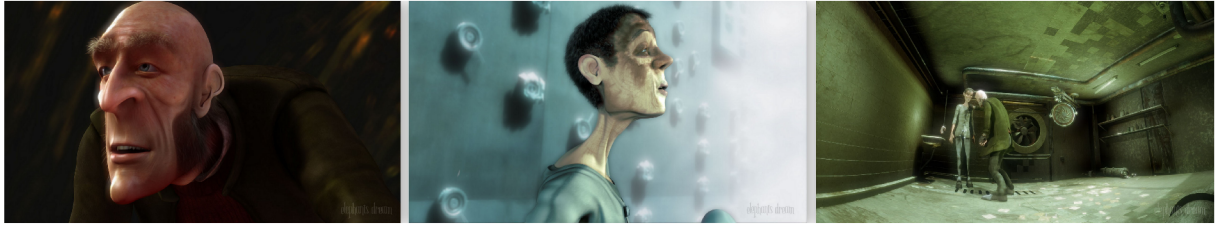
Vzniklo několik krátkých filmů, demonstrujících možnosti i vizuální způsobilost Blenderu. Mezi zdařilé demonstrace patří díla, jako jsou např.: Sintel (2010) – obr. 1 ze zdroje [7], Big Buck Bunny (2008) – obr. Chyba: zdroj odkazu nenalezen ze zdroje [8], Lighthouse (2008) nebo Elephants Dream (2006) – obr. Chyba: zdroj odkazu nenalezen ze zdroje [9]. Též je možné zhlédnout na stránkách Blenderu [2] upoutávky, jako Project London (2009), Kajimba (2009) nebo zajímavé animace: Man in Man (2006) a Esign (2005).



Obrázek 1 – Snímky z filmu Sintel



Obrázek 2 – Snímky z filmu Big Buck Bunny



Obrázek 3 – Snímky z filmu Elephants Dream

## 3.2 Rozhraní

Následující text vytyčí základní poznatky o rozhraní a práci s tímto softwarovým nástrojem. Informace o stažení Blenderu jsou uvedeny v příloze A (Stažení Blenderu – str. ).

### Konvence

V textu jsou využity následující zkratky, které vychází z konvencí literatury anglického jazyka:

- „MW“ – pohyb kolečkem myši („skrolování“),
- „MMB“ – prostřední tlačítko myši/stisknutí kolečka myši,
- „LMB“ a „RMB“ – levé tlačítko myši a pravé tlačítko myši.

### Rozhraní

Vykreslení rozhraní probíhá za využití OpenGL (zdroj [2]). Blender má pracovní plochu prakticky celou nastavitelnou podle požadavků uživatele. Obrazovka je rozčleněna na několik editorů, kde každý z nich může být rozdvojen (tažení pravého horního rohu editoru – obr. 4) a nebo skryt (tažení pravého horního rohu do jiného editoru).

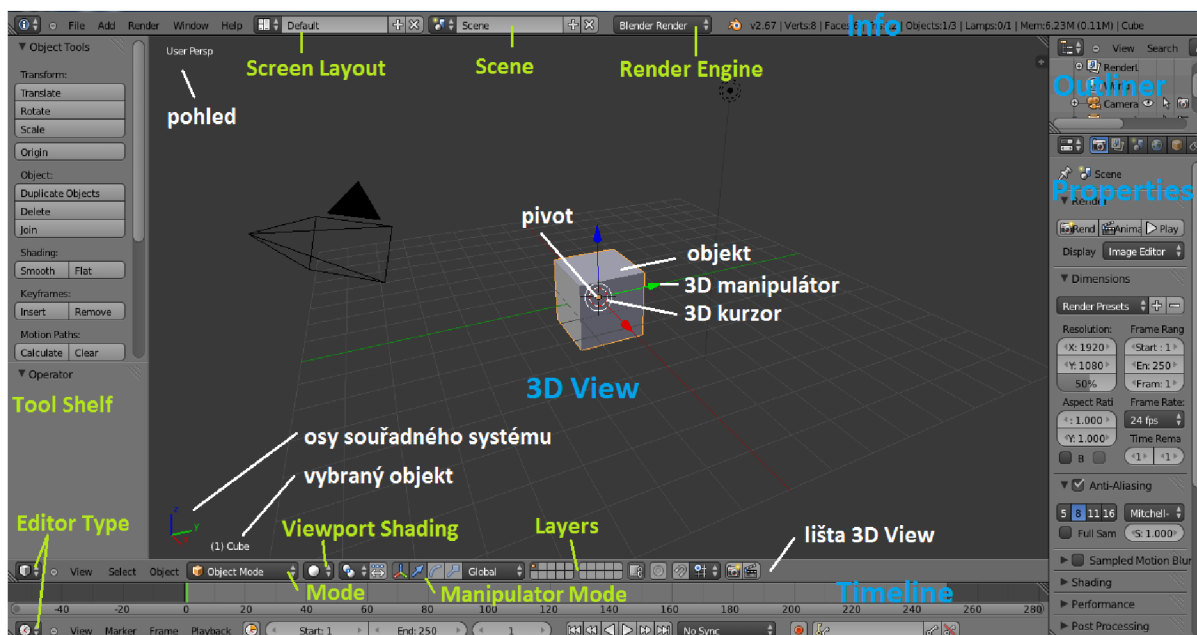


Obrázek 4 – Roh editoru pro rozdvojení

Blender je z velké části orientován na klávesové zkratky, které naleznete v přehledu [5]. Tyto zkratky je důležité znát, jelikož jejich tlačítka hledat v rozhraní není příliš efektivní způsob při častém používání.

Po spuštění Blenderu je v hlavním okně, 3D View Editor, zobrazena úvodní 3D scéna. Scéna je ze začátku uprostřed překryta uvítací obrazovkou, která umožňuje rychlý přístup k dříve vytvořeným souborům Blenderu s příponou „.blend“. Na obr. 5 je úvodní rozvržení rozhraní popsáno (informace vychází ze zdroje [2]). Skládá se z několika editorů, kde editory zahrnují mnoho tlačítek a informací. Popis podstatných částí Blenderu je vyznačen:

- modře – editory,
- zeleně – podstatné tlačítka a lišty editorů,
- bíle – další užitečné informace.



Obrázek 5 – Úvodní scéna

Následující odstavce blíže popisují to nejpodstatnější o editorech (informace vychází z [2]):

- 3D View – náhled scény, editace objektů ve scéně,
- Properties – vlastnosti, úprava vlastností vybraného objektu či scény,
- Info – informace, obsahuje rolovací lišty, typ rozložení obrazovek, zvoleného engine a další,
- Outliner – přehled, seznam organizující datové bloky (objekty, materiály, textury, ...),
- Timeline – časová osa, slouží pro animaci (editor je vysvětlen v sekci o animaci).

### 3D View Editor

3D View obsahuje tato podstatná tlačítka a nastavení.

- Editor Type – tlačítkem lze změnit editor na jiný (obsahuje každý editor).
- View Menu – obsahuje nabídku se základními druhy pohledů ve scéně (pohled zepředu, zleva, zespodu, ...). Změnu pohledu lze provést i pomocí numerické klávesnice (zkratky jsou uvedeny u každého z pohledů a jsou při práci hojně využívány). Hlavní pohledy jsou:
  - zepředu – Numpad 1,
  - zprava – Numpad 3,
  - shora – Numpad 7,
  - přepnutí mezi perspektivním/ortogonálním pohledem – Numpad 5 (perspektivní pohled je pohled, jaký je běžný v reálném světě, tedy vzdálenější objekty se zdají být menší, než bližší objekty, u ortogonálního pohledu tato zákonitost neplatí, paprsky se sbíhají do roviny kamery rovnoběžně, čehož lze efektivně využít při modelování).
- Mode – změna módu 3D View editoru. Nejpoužívanějšími módy jsou:
  - Object Mode – manipulace s objekty ve scéně (výběr, posun, rotace a jiné),

- Edit Mode – pro editování právě vybraného objektu (změna pozic bodů, přidání hrany, ...),
- Pose Mode – úprava pózy armatury (kostry) modelu,
- Weight Paint Mode – změna váhy vertexů (bodů v prostoru) ve skupině vertexů, čímž mění jejich vliv na výsledek (více využití, př.: vliv kostí armatury na vertexy modelu).
- Viewport Shading – změna zobrazení scény ve 3D View. Jednotlivé typy mají různé výpočetní nároky:
  - Textured – zobrazení otexturovaných (obrázek/vzor na povrchu) modelů, nejnáročnější na výkon,
  - Solid – zobrazení modelů v základních barvách (difuzní složka materiálů, bez textury) s vestavěným osvětlením,
  - Wireframe – modely zobrazuje „drátěným modelem“, tedy zobrazuje pouze jejich hrany, které dávají informaci o tvaru modelu,
  - Bounding Box – namísto modelů zobrazuje kvádry dané rozměry modelů.
- Layers (vrstvy) – objekty lze rozčlenit do různých vrstev a zobrazit pouze některé z nich. Zvýší se tak přehlednost scény ve 3D View a nevybrané vrstvy nebudou ani renderovány (vykreslovány).
- 3D kurzor – více využití (především pozice pro vložení nového objektu). „LMB“ lze změnit jeho umístění.
- Pivot – hlavní bod objektu, podle kterého jsou centrovány rotace, změna měřítka a další operace, slouží jako přesná bodová pozice objektu v prostoru.
- 3D manipulátor – slouží pro transformace (posunutí, rotace, změna velikosti) objektů. Transformace v dané ose probíhá tažením manipulátoru myši (vybraným z Manipulator Mode) nebo použitím klávesových zkratk („G“ – Grab/posunutí, „R“ – Rotate/rotace, „S“ – Scale/změna měřítka).
- 3D Tool Shelf – levá lišta, zobrazení/skrytí klávesou „T“ (kurzor myši musí být v daném editoru, což platí pro všechny vstupy, ať už klávesnicí nebo myší). Obsahuje různorodé funkce (transformace, shading, odstranění objektu či nastavení parametrů nově přidaného objektu a další), které jsou často k máni pomocí základních klávesových zkratk.
- Properties Shelf – pravá lišta (není v úvodním rozložení Blenderu na obr. 5), zobrazení/skrytí klávesou „N“. Obsahuje různorodé vlastnosti (přesné hodnoty transformací objektu, nastavení zobrazení scény včetně typu stínování, nastavení obrázku na pozadí scény, ...).

Pohyb ve scéně zobrazené přes 3D View Editor probíhá (podle knihy [1]) následovně (je nutné mít kurzor myši umístěn nad používaným editorem – zde 3D View – jinak může dojít k nechtěnému úkonu v jiném editoru):

- „MW“ – zoom pohledu,
- „Shift + MW“ – vertikální (z aktuální orientace pohledu) posun pohledu,
- „Ctrl + MW“ – horizontální posun pohledu,
- „MMB + pohyb myši“ – změna orientace (rotace) pohledu,
- „Shift + MMB + pohyb myši“ – vertikální i horizontální posun pohledu,
- „Ctrl + MMB + pohyb myši“ – vertikální pohyb myši způsobí plynulý zoom pohledu.

## Properties Editor

Dalším editorem je Properties Editor. Na vrchní liště Properties lze vybrat z několika kontextů (název se zobrazí po najetí myši na ikonu tlačítka), kde základní kontexty jsou následující.

- Render – nastavení pro renderování konečných obrázků (rozlišení snímků, anti-aliasing, „zapékání“ do textury).
- Render Layers – další nastavení pro renderování konečných obrázků (volba vrstev), pro Blender Game nepodstatné.
- Scene – nastavení scény (typu jednotek, barev, audia).
- World – nastavení prostředí scény (barva pozadí, mlha, fyzika).
- Object – nastavení zvoleného objektu (název, transformace, zobrazení).
- Object Constraints – volba omezení pro transformace objektu (sledování cíle a jiné).
- Object Modifiers – úprava objektů pomocí operací (teselace, zrcadlení, decimování). Nemění geometrii objektu, pouze způsob jeho vykreslení.
- Object Data – specifická nastavení pro objekt (barva světla lampy, ohnisková vzdálenost kamery). Ikona se mění podle typu objektu (Text, Lamp, Camera, Mesh).
- Material – nastavení povrchu objektu (barva povrchu, barva odlesku, průhlednost).
- Texture – textura přidává k materiálu další detaily (namapovaný obrázek, procedurální vzor, falešná hloubka povrchu normálovou mapou). Spadá pod zvolený materiál.
- Particles – částicový systém pro tvorbu vlasů na hlavě, květin na terénu a jiných efektů. Je nedostupný v Blender Game.
- Physics – nastavení fyzikálního systému (typ statický/dynamický/rigidní/..., viditelnost, kolize pro Blender Game a látky, kapaliny, kouř a další pro Blender Render a Blender Cycles).

Jednotlivé vlastnosti jsou seskupovány do panelů tlačítek, kterou mohou být rozbaleny/sbaleny (klik myši na ikonu šipky či názvu panelu) nebo posunuty (tažení uchopené ikonky vroubků vpravo od názvu panelu).

## Info Editor

V Info editoru se nachází tyto podstatné aspekty.

- File menu s tlačítky:
  - New – otevřít nový „.blend“ soubor,
  - Save As – uložit soubor,
  - User Preferences... – zobrazí okno s nastavením Blenderu (v sekci Addons lze aktivovat užitečné pluginy).
- Add menu (klávesová zkratka „Shift + A“) – pro přidání objektů do scény.
- Render menu – pro renderování scény. Nejdůležitější je zde tlačítko Render Image („F12“), který vykreslí snímek z pohledu aktivní kamery.
- Screen Layout – výběr z přednastaveného rozvržení obrazovek (Default – po spuštění Blenderu, další důležité jsou Animation a Game Logic pro animování a stavbu herní logiky).

- Scene (obr. 6) – výběr scény ze seznamu vytvořených scén (tlačítko na obr. 6 vlevo), změna názvu (editační pole na obr. 6 uprostřed) a přidání či smazání scény (dvě tlačítka na obr. 6 vpravo). V Blenderu jsou tato tlačítka všude stejná, i jejich význam.



Obrázek 6 – Scene

- Render Engine – výběr z interních renderovacích enginů Blenderu (Blender Render, Blender Game, Blender Cycles). Tyto enginy jsou popsány dále v této kapitole.

## Outliner Editor

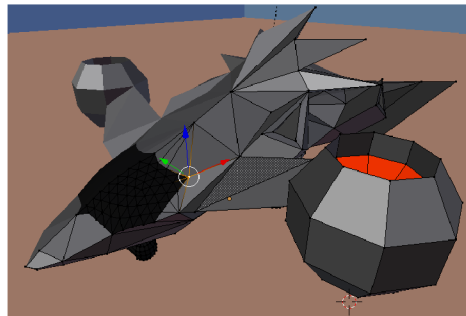
Outliner (přehled) je seznam organizující datové bloky (objekty, materiály, textury, ...) do uspořádané struktury. Z přehledu lze objekty označením vybrat a dále s nimi manipulovat v jiných editorech. Na pravé straně editoru jsou 3 ikonky:

- Viewport Selection – vypnutí/zapnutí možnosti označit objekt ve 3D View Editor,
- Viewport Visibility – vypnutí/zapnutí viditelnosti ve 3D View Editor,
- Rendering – vypnutí/zapnutí viditelnosti v renderovaných snímcích.

## 3.3 Modelování

Po seznámení se s rozhraním Blenderu je možné začít tvorbu vlastního modelu. K modelování lze využít dva hlavní přístupy:

- přímé editování – obr. 7,
- sculpting.



Obrázek 7 – Model vytvořený přímým editováním

### Přímé editování

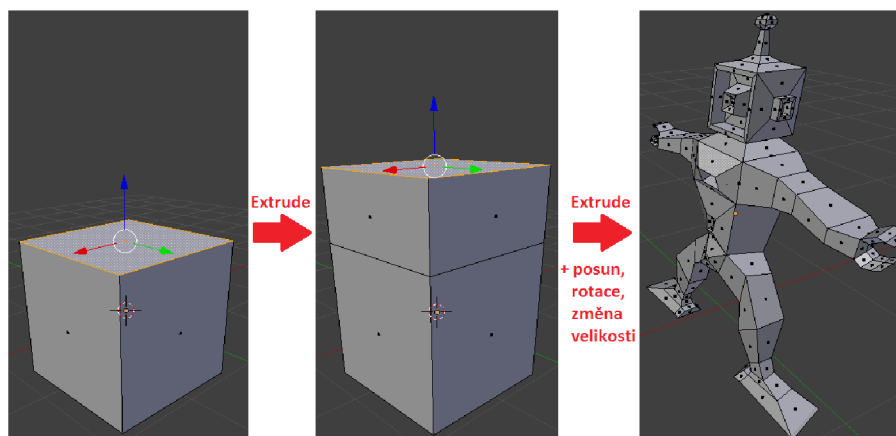
Přímé editování je modelování objektu pomocí práce s vrcholy (obr. 7), hranami a stěnami v editačním módu (do Edit Mode se přepne klávesou „Tab“). Jedná se poměrně o nízkoúrovňové modelování, při kterém je možné využít řady nástrojů k efektivnější práci na modelu. Významná část modelovacích postupů je k dohledání v knize [18]. Mezi nejpodstatnější nástroje, které jsou využity v diplomové práci, patří:

- Extrude,



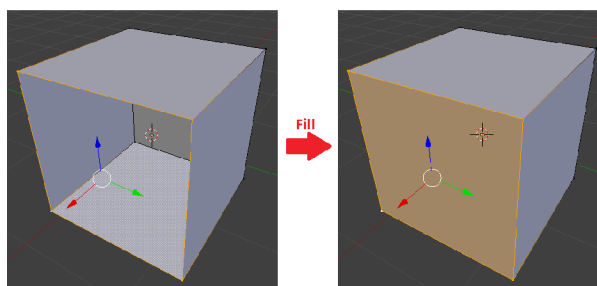
- Fill,
- Add Edgeloop,
- Merging Vetices.

Při použití Extrude se uchopí body, hrany či stěny modelu (záleží v jakém módu je uživatel) a povytáhnou se z povrchu ven nebo dovnitř. (Nástroj se použije po označení daného útvaru klávesou „E“ a tažením myši se povytáhne. Změna se potvrdí „LMB“. Při nepotvrzení RMB zůstane zdvojený útvar na počátečním místě a vytváří nežádoucí duplicity bodů, hran a stěn. Takovýto úkon je nejrozumnější vrátit zpět „CTRL + Z“.) Pomocí používání Extrude s posunem, rotací a změnou měřítka je možné vytvořit jednoduchý model robota – obr. 8.



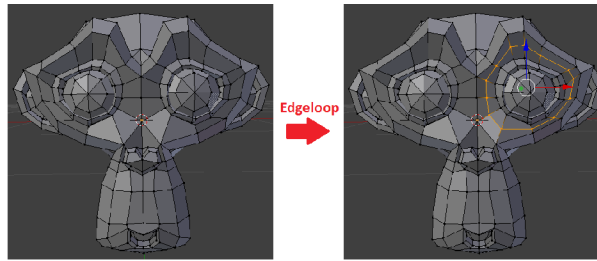
Obrázek 8– Extrude

Dalším praktickým nástrojem je Fill. Jeho pomocí lze vyplnit označené body stěnou – obr. 9 nebo vytvořit mezi dvěma body hranu.



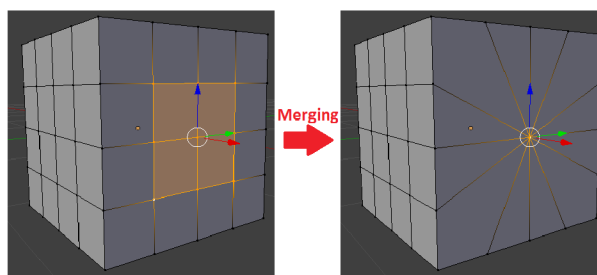
Obrázek 9– Fill

Přes Add Edgeloop lze přidat do modelu další smyčku hran. (Myši se najede mezi dvě již existující smyčky a klávesovou zkratkou „CTRL + R“ se vytvoří v daném místě smyčka nová. Přes „MW“ je možné nastavit celkový počet přidaných smyček hran – volba se potvrdí „LMB“.) Nástrojem se zjemňuje model na požadovaných místech – obr. 10, třeba v místě pro kloubní soustavu, které je podstatně deformováno při animaci.



Obrázek 10– Add Edgeloop

Merging Vertices slouží pro spojení několika bodů do jednoho. (Po označení bodů klávesa „M“. V nabídce se zvolí druh sjednocení, třeba spojení do centra označených bodů.) Nástrojem se převážně zjednodušuje nebo zaceluje model na patřičných místech – obr. 11.

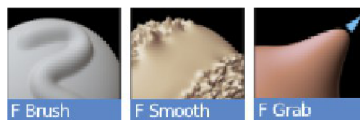


Obrázek 11– Merging Vertices

## Sculpting

Druhým hlavním směrem při modelování je tzv. „Sculpting“ (Sculpt Mode). Je podobný modelování soch z hlíny, přidává se nebo odebírá materiál. Ve skutečnosti zde dochází pouze posunu vrcholů modelu v prostoru, model žádnou tloušťku nemá. Stále se jedná o tu stejnou „tenkou“ plochu. Model se upravuje za použití několika štětců (obr. Chyba: zdroj odkazu nenalezen), jako jsou (zdroj [2]):

- Brush – pro kreslení linek,
- Smooth – vyhlazení povrchu,
- Grab – povytažení oblasti.



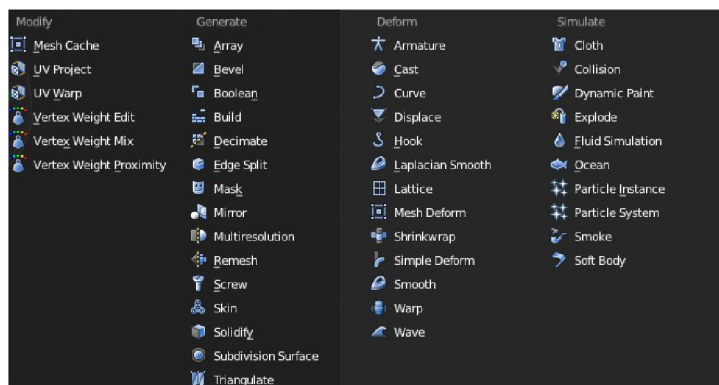
Obrázek 12 – Ikony důležitých štětců

Sculpting se používá na složité modely s hodně vrcholy, které je možné jemným využíváním štětců dotvořit k dokonalosti. Mnohopolygonové modely (modely s hodně stěnami) se ovšem nedají použít pro real-time zobrazování (v reálném čase, prakticky ihned) kvůli jejich výpočetní náročnosti. Lze je využít buď pro offline renderování (jako jsou fotorealistické obrázky), tedy vykreslování, kde na čase výpočtu tolik nezáleží. Další možností je aplikace multiresolution modifikátoru Blenderu na jednodušší model, který je dostatečně jednoduchý pro real-time zobrazení, tím model nabude na vertexech a upraví se pomocí sculpting metody. Složitější povrch se všemi detaily se následně „zapeče“ do textury (uloží se do obrázku) a aplikuje se, třeba ve hře, na jednodušší model.



## 3.4 Modifikátory

Na vytvářený model lze aplikovat Blenderem poskytované modifikátory, jako jsou zmiňovaný multiresolution, subdivision surface (teselace, při které se plošky povrchu, polygony, rozdělí na několik menších a vyhladí tak povrch modelu) nebo Blender poskytuje i různé simulace prostředí, jako jsou simulace kapaliny, oceánu a dalších jevů. Dále modifikátory zahrnují vytvoření zdánlivé tloušťky stěn modelu (duplikací ploch a oddálení od sebe), zrcadlení (efektivní technika pro tvorbu symetrických modelů, kde se modeluje jen jedna strana/půlka modelu a druhá část se zrcadlově zduplikuje) a mnoho dalších, které lze vyčíst z obr. 13. Jejich popis se nachází na stránkách [2].



Obrázek 13 – Modifikátory

## 3.5 Materiály

Materiály slouží v Blenderu pro nastavení vlastností povrchu objektů. Materiály obstarávají (zdroj [2]) barvu, odrazivost, lesk, svítivost, průhlednost. Všechny vlastnosti jsou založeny na fyzikálních principech. Vyrenderovaný obraz je projekcí scény na imaginární podložku, na tzv. desku pohledu, která vnímá simulované světlo. Pro algoritmy výpočtu je výhodnější směr světelného toku otočit a směřovat světlo od čidla (kamery) směrem do scény. Vlastnosti povrchu a úhlu dopadu takového světla na povrch objektů vyjadřují informace o světle, které se dostane do kamery. Na tomto principu funguje renderovací engine. Více informací o osvětlení v Blenderu lze nalézt v knize [19].

Blender zahrnuje 3 základní barevné složky.

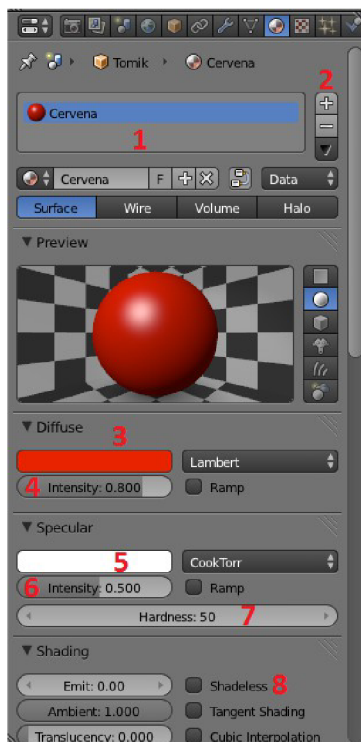
- Diffuse color – základní barva objektu, kterou má při osvětlení bílým světlem (na obr. 14 červená).
- Specular color – barva odlesku z ideálního odrazu světla ze světelného zdroje (na obr. 14 bílý kruh). Odlesk je typický pro lesklé/hladké povrchy.
- Ambient color – ambientní (všudypřítomné) osvětlení určuje ambientní barvu. Simuluje odraz světla od povrchů okolních objektů, které působí jako sekundární zdroje světla. Jedná se o barvu, kterou mají objekty, když nejsou osvětleny. Nastavení této barvy není v záložce Material, ale pod záložkou World. V Blender Game toto nastavení není podporováno.

K nastavení materiálu pro vybraný objekt slouží záložka Material. Při výběru materiálu ze seznamu (obr. 14 – 1), případně vytvoření nového materiálu (2), je umožněno nastavit tyto nejpodstatnější parametry (zdroj [1]), které jsou vyznačeny na obr. 14.

- Diffuse Color (3) – nastavení základní barvy.
- Diffuse Intensity (4) – nastavení intenzity barvy.
- Specular Color (5) – nastavení barvy odlesku.

- Specular Intensity (6) – nastavení intenzity odlesku.
- Hardness (7) – tvrdost materiálu, která určuje velikost odlesku (větší Hardness – menší velikost odlesku).
- Shadeless (8) – při aktivaci Shadeless osvětlení nemá na materiál vliv. Materiál se bude stínovat v jednolitě základní difuzní barvě.

Je možné nastavit také průhlednost materiálu (alfa kanál), jak moc ambientní světlo bude na materiál působit, lze nastavit i osvětlovací model, emitování vlastního světla, materiál potom může dát iluzi, že září (ovšem své okolí ve skutečnosti neosvětluje) a další vlastnosti.



Obrázek 14 – Nastavení materiálu

## 3.6 Textury

Na modely se často aplikují textury, které dokončí vzhled povrchu. V nástroji je umožněno využít mapování obrázkové textury nebo i procedurální texturování, které nevyžaduje žádné externí zdroje (obrázky). Blender podporuje jak barevnou texturu, tak i normal mapping nebo displacement mapping.

Na obr. 16 je lišta s texturami. Na počátku lišty se nachází seznam s vytvořenými texturami, přidání nové textury a další, stejně jak tomu je u materiálů. V seznamu lze navíc texturu zapnout nebo vypnout (obr. 16 – 1). Při výběru textury ze seznamu lze texturu editovat. Nová přednastavená textura má tyto nejpodstatnější parametry, podle kterých se vytváří náhled (3).

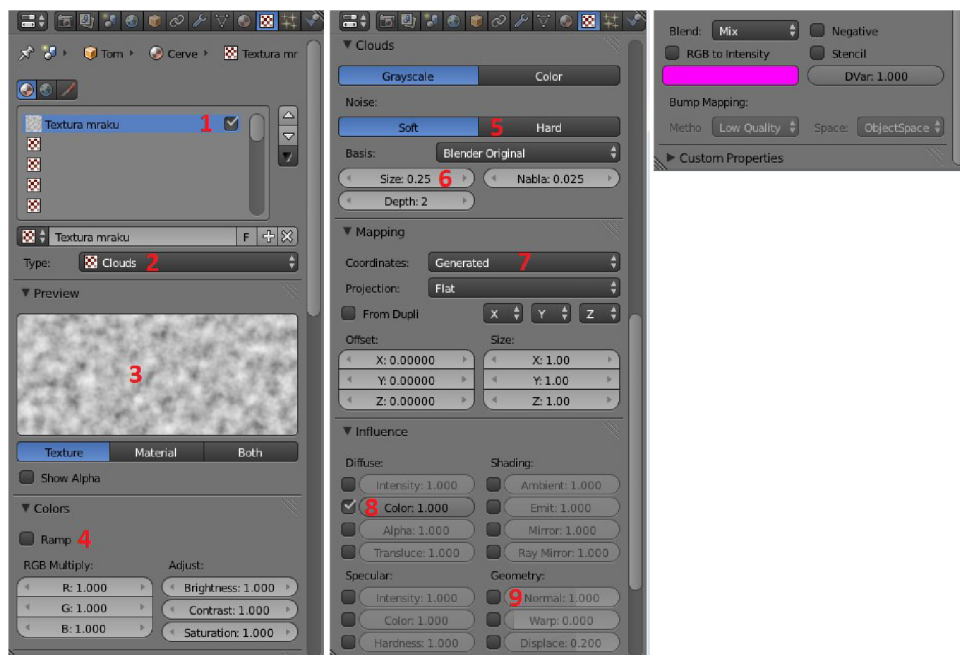
- Type (typ, obr. 16 (2)) – v Blenderu existuje již několik druhů procedurálních a dalších druhů textur. Poměrně používaným typem je procedurální textura Clouds. Pro Blender Game engine je ale použitelná především Image or Movie textura, která pracuje s externími zdroji (obrázky, filmy).

- Preview (náhled, obr. 16 (3)) – poskytuje náhled na aktuální nastavenou texturu.
- Ramp (rampa, obr. 16 (4)) – namísto jedné barvy dané materiálem je možné vytvořit texturu povrch, který obsahuje celou barevnou škálu (obr. Chyba: zdroj odkazu nenalezen) mezi 2 barvami (většinou se mění barva v pravé části rampy a na levé se ponechává barva z materiálu).



Obrázek 15 – Rampa

- Noise (šum, obr. 16 (5)) – určuje jestli přechody procedurálních vzorů v textuře mají být jemné (Soft) nebo ostré (Hard).
- Size (velikost, obr. 16 (6)) – určuje velikost/rozměr vzoru textury.
- Coordinates (koordináty, obr. 16 (7)) – určení způsobu mapování textury na povrch objektů. Při procedurálních vzorech (jako Clouds) lze využít Generated (zachovává stejnou velikost textury, nehledě na velikost modelu, na který se aplikuje). Další možností je Global (texturu roztáhne podle velikosti modelu). U Blender Game, kde se využívá textura z obrázku (Image or Movie), se nastaví UV, čímž Blender bude aplikovat texturu podle vytvořené UV mapy (mapování vertexů 3D modelu na 2D texturu) v UV/Image Editor.
- Color (barva, obr. 16 (8)) – u textur se nastavuje vliv (Influence) textury na různé složky, zde na difuzní barvu.
- Normal (normála, obr. 16 (9)) – nastavená textura se reprezentuje jako normálová mapa (dává zdání pomocí úpravy světla dopadajícího na povrch objektů, že povrch je plastický, má různou hloubku).



Obrázek 16 – Příklad lišty s parametry textury

## 3.7 Animace

Samozřejmě v Blenderu nechybí ani možnost animování, které je podstatné pro hry či filmy. Pro animaci slouží rozvržení obrazovek (Screen Layout) Animation, kde nejdůležitějšími editory jsou Timeline Editor pro základní přehled animace a DopeSheet Editor pro přehledné ukládání a editaci Keyframes zaznamenané akce.

Animace je vytvoření iluze pohybu nebo jiné změny věci (pohyb panáčka po zemi nebo změna barvy světla lampy) pomocí sekvence statických obrázků. Základním kamenem animace v Blenderu jsou tzv. „Keyframes“ (klíčové snímky). Určují hodnoty parametrů objektů v daném čase. Parametry mohou být poloha, rotace, velikost, rychlost, barva, intenzita osvětlení a mnoho dalších. Čas při animaci v Blenderu není určován přímo, ale pomocí Frames (snímků). Snímek je (zdroj [2]) zobrazení jednoho obrazu po určitou dobu projekce. Odsud vyplývá Frame Rate (frekvence snímků), která se uvádí v FPS (Frames Per Second – snímky za sekundu). Pro animaci, kterou člověk vyhodnotí jako plynulou, se považuje minimálně 25 FPS (tedy 40 ms projekce jednoho snímku). Animace je tvořena postupnými změnami parametrů objektů, ovšem měnit zvolené parametry pro animaci snímek po snímku „ručně“ by bylo značně neefektivní. Proto se při animaci využívají Keyframes, kde se pro zvolené snímky „ručně“ zaznačí hodnoty zvolených parametrů a ostatní snímky jsou mezi Keyframes již automaticky interpolovány. Při interpolaci se matematicky nachází přibližná hodnota daného parametru v intervalu ohraničeném klíčovými snímky. Animace se tedy v Blenderu vytváří zaznačením několika kontrolních bodů (Keyframes) a zbytek animace se dotvoří automaticky.

### Timeline Editor

Timeline editor (časová osa, obr. 17) dává přehled o animační části scény (zdroj [2]). Opět jsou uvedeny důležité součásti užitečné znát:

- Time Cursor (1) – časový kurzor, zelená čára v Timeline editoru, slouží k nastavení a zobrazení vybraného snímku (nastavuje se „LMB“ v editoru nebo posunem o 1 snímek klávesami levé a pravé šipky),
- Keyframes (2) – klíčové snímky, jsou zaznačeny v Timeline editoru pro vybraný objekt jako žluté čáry, keyframes zaznamenávají pro model, či jeho části, pozici, rotaci, velikost (případně další atributy: průhlednost, barva, ...) a tyto údaje se zaznačí pro daný snímek (Frame) v Timeline Editoru, mezi jednotlivými klíčovými snímky se při animaci interpoluje a vzniká tak plynulý pohyb (či jiná změna), bez nutnosti zaznamenávat atributy pro každý snímek zvlášť,
- Header Controls – ovládací panel (dolní lišta Timeline editoru), zde se nastavuje první (Start (3)) a poslední (End (4)) snímek animace, aktuální snímek (5) zobrazuje pozici Time Cursor, pro spuštění animace se stiskne tlačítko „přehrát“ (6), které se změní na tlačítko „pauza“ pro pozastavení animace.



Obrázek 17 – Timeline Editor

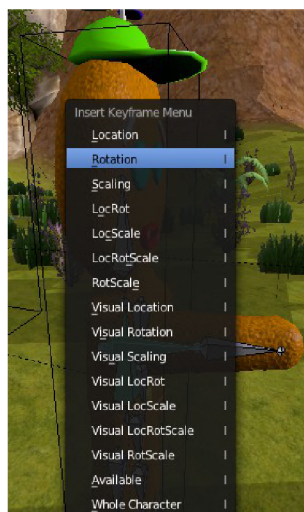
### DopeSheet Editor

Dalším používaným editorem pro animace je DopeSheet (obr. 20). DopeSheet přesně ukazuje, co animátor tvoří (zdroj [2]). Důležité poznatky jsou níže (více informací lze dohledat v knize [12]).

- Main Area (1) – hlavní oblast. Obsahuje keyframes pro všechny kanály akcí. Funkčnost je podobná Timeline Editor, ale kanály pro animaci jsou rozepsány na jednotlivé řádky editoru.
- List-Tree (2) – seznam akčních kanálů. Zde se vyskytují především názvy kanálů pro keyframes vpravo od seznamu.
- Header Bar (3) – dolní lišta editoru. Skládá se z mnoha tlačítek, kde hlavními jsou následující.
  - DopeSheet Mode (4) – editor má několik módů, kde důležité jsou:
    - DopeSheet – pro editaci vícero akcí naráz, přednastavený mód,
    - Action Editor – pro vytvoření a editaci akcí (animací), nejpoužívanější mód (nutné přepnout z DopeSheet).
  - V části (5) se přepíná mezi akcemi, mění název akce, vytváří a ruší akce (tlačítka i jejich funkcionality je obdobná jako v části výběru scény u Info Editor). Navíc se v této části lišty nachází tlačítko „F“, které po jeho zaznačení Blenderu přikáže, aby vybranou akci nemazal, pokud akce není využita žádným objektem (popsané chování platí i pro materiály a další datové bloky, i ty mívají volitelně nastavitelný příznak „F“). Tlačítko nalevo od „F“ svým číslem vyjadřuje, kolikrát daný datový blok (zde akce) je využíván.
  - V části (6) spodní lišty lze uložit označené body Keyframes (označí se v Main Area stejně jako objekty ve 3D View editoru, označené body jsou vykresleny oranžově) a zkopírovat do vybraného snímku (což je místo, kde se nachází zelený posuvník (7)).

V DopeSheet se s body Keyframes manipuluje následovně.

- Nové klíčové snímky se pro vybraný objekt (například kost armatury) vytváří klávesou „I“ s kurzorem myši nad 3D View Editor. Takto lze například zaznamenat transformace (pozici, rotaci, velikost) objektu, což je nejčastější atribut pro animaci. Po stisknutí klávesy „I“ se zobrazí nabídka s atributy k vložení (př. nabídka je na obr. 18).
- Další vlastnosti lze zaznamenat do DopeSheet i v jiných editorech, jako je Properties Editor, s kurzorem myši nad zvoleným nastavením (např. difuzní barva materiálu, po stisknutí „I“ nad atributem se vloží keyframe do DopeSheet a atribut se zvýrazní – obr. 19).
- Zaznačené body z Main Area lze mazat stejně jako objekty ve 3D View (klávesou „X“).

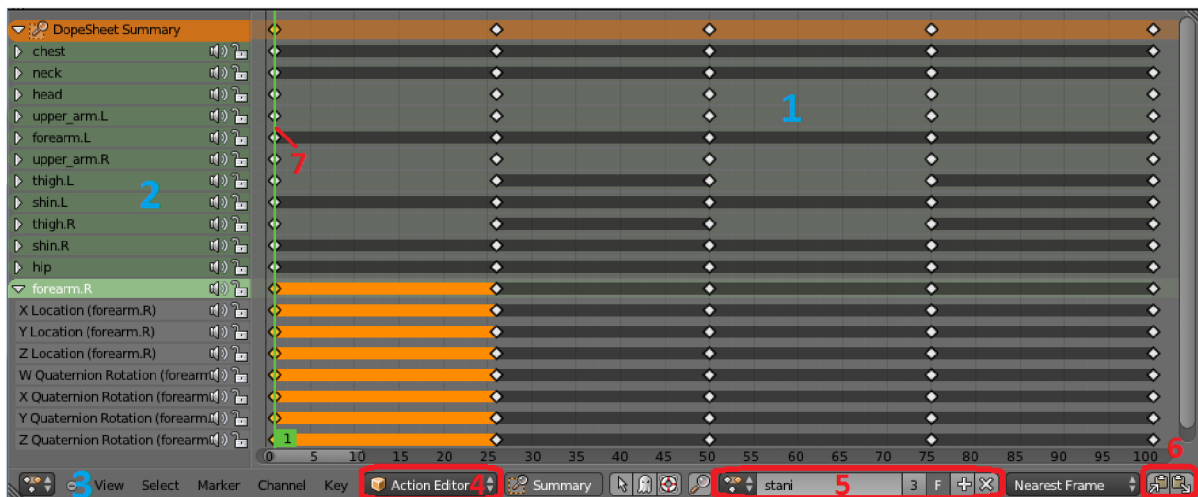


Obrázek 18 – Insert Keyframe Menu



Obrázek 19 – Zaznačený atribut Diffuse Color pro Keyframe





Obrázek 20 – DopeSheet Editor

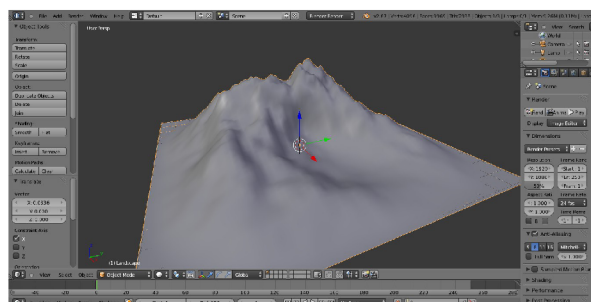
### 3.8 Rozšíření

Jelikož Blender spadá pod open-source, lze číst a zasahovat do jeho kódu. Tímto může uživatel/programátor nástroj rozšířit a novou verzí vydat. Dnes existuje mnoho druhů rozšíření i ve formě pluginů, které je možné přímo v nástroji po jejich aktivaci použít. Mezi zástupce pluginů patří:

- Sapling – slouží pro vytváření stromů podle zadaných parametrů (množství listů, větví, jejich sklon a další vlastnosti), v tomto pluginu byly vytvořeny modely stromů a křoví z obr. 21,
- ANT Landscape – parametrické generování terénu – obr. 22,
- Lightsaber Plugin Effect – přidání světelného meče do obrazu,
- LSD Plugin Effect – snímek bliká mnoha barvami,
- OldMovie Plugin Effect – do obrazu přidá zrnitost a černobílý odstín,
- Save As Game Engine Runtime – export hry do spustitelné verze (s „.exe“ souborem),
- mnoho dalších, jejich přehled je na stránkách Blenderu [2].



Obrázek 21– Sapling plugin



Obrázek 22– ANT Landscape plugin

Blender je velmi všestranný, umožňuje export scény do zařízení s OS Android (přes import existujícího projektu ve vývojovém prostředí Eclipse). Dále je možné vytvořený model, animaci či hru exportovat na web, odkud je spustitelný přímo po zadání webové stránky.

## 3.9 Vestavěné enginey

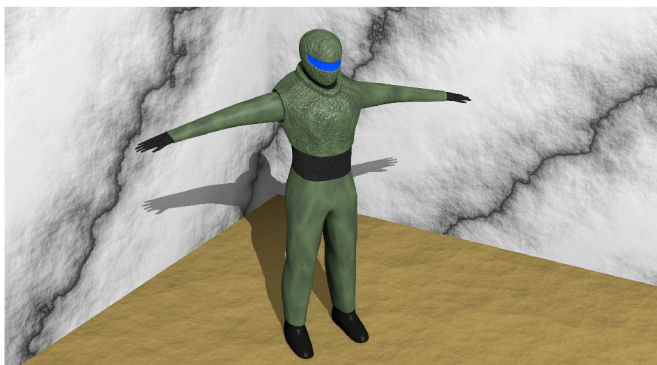
Blender má v sobě obsaženy 3 rozdílné enginey:

- Blender Render,
- Blender Cycles,
- Blender Game.

Každý z engineů je určen pro jinou úroveň realističnosti a jinou rychlost renderování jednotlivých snímků. Blender Game je na rozdíl od ostatních dvou engineů, které jsou určeny výhradně pro vytváření offline snímků scény, zaměřen na tvorbu interaktivních aplikací (kupříkladu her, simulací).

### Blender Render

Blender Render je základním engineem, který byl na počátku nástroje samotného. Je v něm možná tvorba 3D modelů, animací a umožňuje také proces renderování do výsledného snímku. Dává dobré vizuální výsledky – obr. 23 a má obstojně rychlé renderování (v řádu sekund). Zpracování obrazu probíhá offline, bez možnosti změn za běhu vykreslování.



Obrázek 23 – Blender Render

### Blender Cycles

Druhým engineem, který přišel s verzí Blenderu 2.61 [2] je Blender Cycles. Engine je postaven na principu raytracingu (sledování paprsků), přináší nový systém stínování a texturování. Implicitně jsou v něm zasazeny materiály, jako je Glass Material, u kterého je možné nastavit index lomu světla a využít ho pro tvorbu průhledných materiálů (sklo, led, voda a jiné). Blender Cycles přináší i očekávané renderování přes GPU (prozatím staví na podpoře NVidia karet, ale v budoucnu má doplnit i škálu grafických karet mezi ATI a AMD).

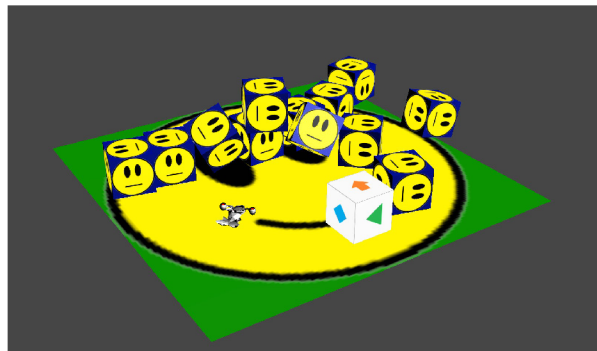


Obrázek 24 – Blender Cycles

Cycles umožňuje kvalitní fotorealistickou vizualizaci, ale na úkor času vykreslení (obrázek skleničky – obr. Chyba: zdroj odkazu nenalezen trval vykreslit kolem hodiny). Jelikož tento engine pracuje v iteracích, kde v každé iteraci zdokonalí výsledný obraz, lze nastavit i menší počet iterací a vykreslení snímku na úkor kvality urychlit.

## Blender Game

Blender Game slouží pro tvorbu her (screenshot z jednoduché hry vytvořené v Blenderu je k vidění na obr. 25), simulací či interaktivních aplikací. Oproti Blender Render engine umožňuje real-time vykreslování a pomocí zabudované herní smyčky lze měnit parametry prostředí za běhu aplikace. K vývoji interaktivních aplikací dopomáhá editor herní logiky, který je zobrazen na obr. 26.

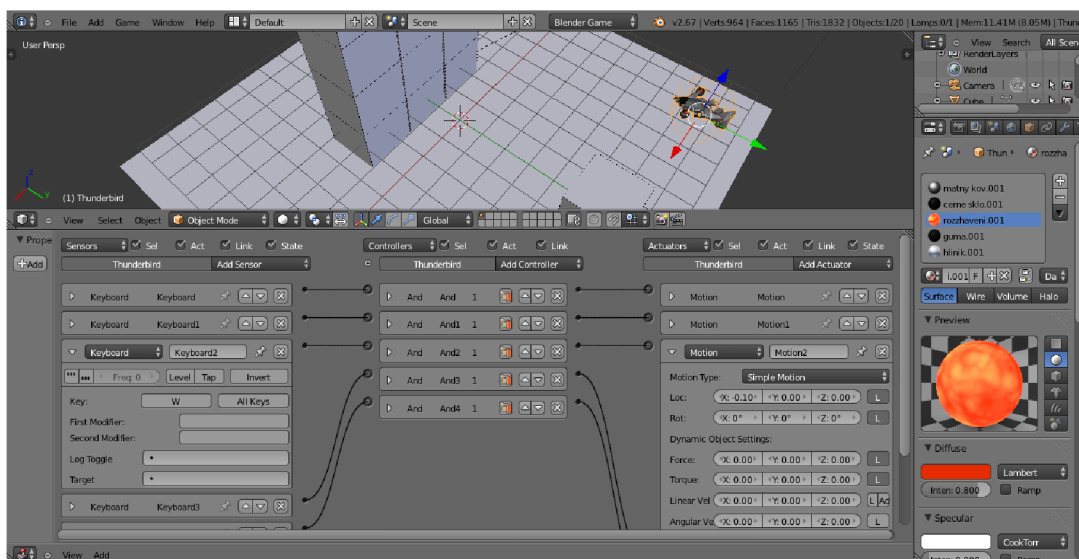


Obrázek 25 – Blender Game

## Editor logiky

Editor se skládá z tzv. „logic bricks“, logických bloků, jejichž nastavením a propojováním je možné herní logiku naprogramovat. V Blenderu jsou tři typy logických bloků, které se spojují do požadovaného efektu (hra s pospojovanými logickými bloky je k vidění na obr. Chyba: zdroj odkazu nenalezen):

- sensors (senzory),
- controllers (kontroléry),
- actuators (aktuátory).

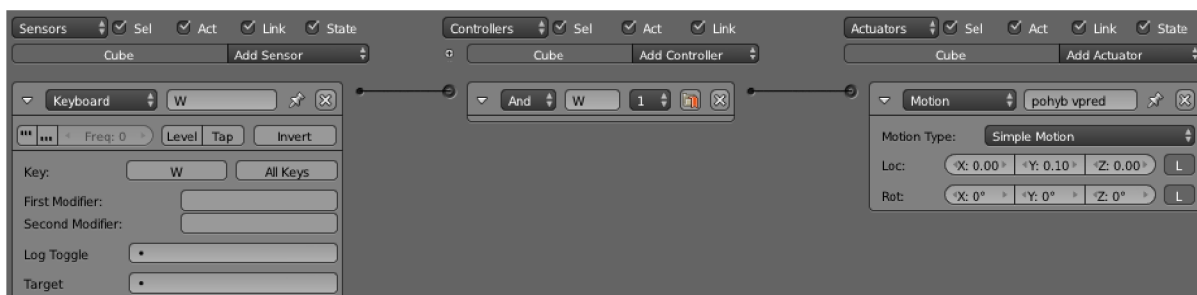


Obrázek 26 – Logic Editor



Senzor registruje změny v aplikaci – uživatelem registrované proměnné (určitá úroveň počtu zdraví hráče, dosaženého skóre), dotyk dvou těles, stisknutí klávesy, tlačítka myši uživatelem atd. Aktuátor má za úkol provést změny v aplikaci na základě vstupů od senzorů (ukončit hru, restartovat scénu, změnit počet životů). Kontrolér je vhodné propojení mezi senzory a aktuátory. Nejčastější spojení (vyjádřeno čarou mezi výstupem a vstupem bloků) je „And“, které vyjadřuje, že všechny impulzy z připojených senzorů musí být aktivní, aby kontrolér „And“ přeposlal impuls do připojených aktuátorů.

Na obrázku 27 je příklad spojení klávesy „W“ (senzor) s pohybem vpřed (aktuátor) přes spojovací „And“ (kontrolér). Toto spojení funguje po celou dobu držení klávesy „W“, tedy v každém logickém tiku hodin (každém vykresleném obrázku hry – při nezatžení HW až 60-krát/s) se objekt, na který jsou logické bloky přichyceny, posune o 0,1 Blender jednotky v ose Y vpřed.



Obrázek 27 – Příklad spoje logických bloků

Blender také podporuje skripty psané v jazyce Python, které lze připojit do kontroléru k aktivačním senzorům. Skript je aktivován při každém pulzu ze senzoru. Je tedy nutné až v Pythonu (místo kontroléru) ošetřit, jaké kombinace aktivací senzorů povedou na spuštění napsaného kódu.

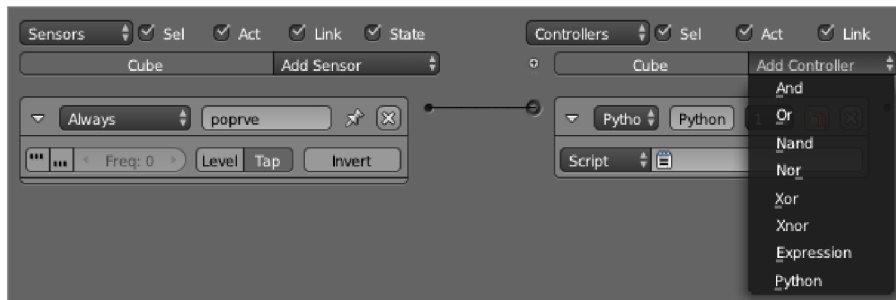
## Python v Blenderu

Předmětem práce není vysvětlení jazyka Python (přehledný český tutoriál k naučení se syntaxe a sémantiky Pythonu je např. na webové stránce [11]). Ovšem pro pochopení pokročilejší logiky hry, která je ve zpracované hře obsažena, je znalost tohoto interpretovaného jazyka žádoucí. Kód je ale komentovaný a pokud čtenář zná jiné objektově orientované jazyky, tak by neměl mít problém s pochopením jeho sémantiky.

Blender obsahuje několik podstatných modulů, které jsou při vývoji v interním herním enginu nezbytné. Dokumentace k implementovaným modulům a API je dostupná na stránkách Blenderu [10]. Nejdůležitějším modulem je „bge.logic“, který obsahuje herní logiku. Přes „bge.logic“ je možné přistupovat k herním objektům, k vytvořeným scénám, lze využít k jejich manipulaci či vytváření. Také lze tímto modulem zjistit, který senzor vytvořený v editoru logiky detekující změny v herní smyčce je právě aktivní a umožnil tak spuštění zbytku skriptu. Přes Python není dostupná většina senzorů (i některé aktuátory), které logické bloky poskytují, proto je nutné je ručně připojit v editoru logiky a v kódu dané senzory (a aktuátory) vyhledat. Existují však výjimky, jako jsou klávesnice a myš. Reakce na vstupní zařízení hráče nepotřebuje přiřazení senzorů na stisk jednotlivých kláves v editoru a je možné dané události detekovat pouhým skriptem (myš je dostupná přes „bge.logic.mouse“ a klávesnice přes „bge.logic.keyboard“).

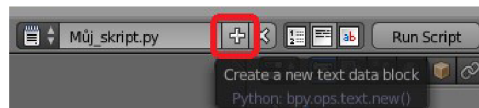
V modulu se vyskytuje globální slovník (dostupný přes „bge.logic.globalDict“, do kterého se ukládají globální proměnné/objekty. Objekty jsou ve slovníku dostupné ve všech napsaných skriptech a mají perzistentní vlastnost. Slovník se zaznamenanými hodnotami je dostupný po celý běh hry. Jeho výhodou navíc je, že pomocí aktuátoru nebo Python kódu (metody: „bge.logic.saveGlobalDict()“ a „bge.logic.loadGlobalDict()“) lze uložit/načíst do/z nového souboru.

Každý skript je spustitelný pouze přes logické bloky nebo z jiného skriptu po importu. V sekci „Sensors“ se zvolí senzory, které budou vysílat impuls po jejich splnění k danému kontroléru v sekci „Controllers“. V tomto případě typu „Python“, kde v nabídce „Add Controller“ (viz obr. 28) se daná položka vybere. Na obrázku 28 je senzor aktivovaný po prvním výskytu objektu ve scéně hry. Vedle tlačítka „Script“ přidaného kontroléru se následně zapíše název skriptu (včetně přípony, pokud má). V Blenderu je umožněno volat i přímo danou funkci ze skriptu. Po kliknutí na tlačítko „Script“ lze přepnout položka na „Module“ a následně do textového pole vpravo zapsat název skriptu a zvolenou funkci (tvar: `nazev_skriptu.nazev_funkce`).



Obrázek 28 – Spuštění skriptu v Logic Block Editor

Skripty se vytváří přidáním nového textového souboru uvnitř Blenderu. Je zvykem jim za jméno přidat příponu „.py“, ať je zřejmé, že se jedná o Python skript a ne o poznámky, které si zde uživatel může poznamenat. Není to však Blenderem vyžadováno. Na obrázku 29 je zvýrazněno tlačítko textového editoru (v editoru Text Editor), kterým lze přidat nový textový soubor. Nalevo od něj je pole pro zadání názvu skriptu, nebo textu a dále od něj je výběr mezi již vytvořenými soubory. Je doporučeno si v případě psaní skriptu zvýraznit řádkování a syntaxi jazyka Python (první a třetí položka napravo od tlačítka pro smazání datového bloku).



Obrázek 29 – Vytvoření datového bloku s textem

## 4 Vyhodnocení nástrojů a volba studie

Již byly shrnuty některé důležité enginy s jejich grafickými možnostmi a pomocnými nástroji, které usnadňují tvorbu různorodých grafických děl. V následujících odstavcích bude uvedeno, který z těchto 3D nástrojů byl vybrán pro tvorbu 3D aplikace a její případové studie. Výběr bude zdůvodněn a následně bude uvedeno, na jakou konkrétní problematiku se práce bude zaměřovat.

### 4.1 Výběr 3D nástroje

V podstatě se všichni vývojáři herních enginů snaží zakomponovat nejmodernější vizuální efekty, nástroje a technologie, což se jim také daří. Po grafické stránce jsou výše zmíněné enginy prakticky rovnocenné. Všechny podporují teselaci, bump mapping, displacement mapping, occlusion culling a mnoho dalších technik. Obsahují i fyzikální jádra, podporu umělé inteligence a skladbu her pomocí uzlů (logických bloků), kterými je možné se vyhnout přímému styku s programovacím jazykem. V tomto ohledu mezi enginy není výrazný technologický rozptyl.

Velká část populace má Blender spojen pouze s modelačním nástrojem, ve kterém lze vytvořit modely pro své aplikace/hry/animace pro následný export do jiné aplikace, do jiného herního enginu. Někteří znalejší vědí, že modely lze v Blenderu vyrenderovat interními enginy (Blender Render a novějším Blender Cycles), případně vytvořit s nimi animaci. Ovšem Blender má již i integrováno poměrně nové, méně známé jádro (Blender Game), které slouží pro vývoj her a jejich následný běh. Prozatím Blender Game nebyl příliš ověřen praxí, jak tomu je u ostatních vyjmenovaných herních enginů. První vytvořené hry (především „Yo Frankie!“) ale dávají najevo, že potenciál pro herní vývoj má.

Blender má následující výhody, které podporují jeho volbu pro diplomovou práci:

- tvorba her, simulací, interaktivních aplikací, fotorealistických obrázků, filmů a jiných animací.
- v samotném nástroji lze modelovat objekty (ostatní enginy touto vlastností nedisponují),
- multiplatformní,
- open-source (zdarma, úprava kódu),
- dokumentace, tutoriály, velká uživatelská komunita,
- vysoká vizuální úroveň,
- mnoho optimalizací určených pro běh hry.

Blender má i své stinné stránky, mezi které patří:

- někdy málo intuitivní ovládání (klávesové zkratky v hojném množství, nováčky se v rozhraní mohou ztratit),
- není stabilní (nutné často ukládat kvůli pádům aplikace),
- podpora pro některé grafické karty se teprve utváří (prozatím může být problém při renderování na AMD/ATI kartách, které vyžadují pro renderování OpenCL – v oficiálních verzích Blenderu není podpora OpenCL, Nvidia karty jsou na tom o poznání lépe se zaměřením na Cuda architekturu, kterou Blender již podporuje).

Blender má oproti ostatním herním jádrům samozřejmě některé nedostatky, které plynou z jeho nekomerčního vývoje. Je ovšem vydáván zdarma v neustále nových verzích a na jejich odstranění se za účasti týmů dobrovolníků neustále pracuje. Práci bych rád také zaměřil open-source směrem. Tedy celý vývoj by měl probíhat v nástrojích zdarma a případně i s modely, na které se nevztahuje žádná

licenční práva. Jelikož Blender je na vysoké úrovni, konkurenceschopný a také je vydáván zdarma, byl zvolen pro tuto práci.

## 4.2 Zaměření studie

Práce se zaměřuje především na demonstraci vývoje 3D aplikací v některém z 3D nástrojů, vytvoření této aplikace a sepsání případové studie její tvorby. Blender, jak bylo zdůvodněno, poslouží jako tento nástroj. Herní jádro Blenderu je poměrně nepříliš probádané a vytvoření hry v něm je nejlepší volbou, jak demonstrovat pokročilou 3D aplikaci. Hra obsahuje herní menu, obchodní nabídky a další obrazovky zaměřené na ovládání myši, tlačítka a vystavěný systém v pozadí aplikace. Navíc samotná hra přímo demonstruje užití herního jádra, modelaci herních objektů a jejich animaci, což je netriviální problematika 3D aplikací. Výběrem počítačové hry bude pokryta velká část spektra tvorby 3D aplikací.

Pro začátek je nutné vědět, jak asi hra bude vypadat. Tato hra bude zpracována ve stylu obrany pevně stanoveného území, také známého pod anglickým názvem „Tower Defence“ (obrana věže, v češtině je používán někdy výraz „věžovky“). Území bude mít venkovní charakter, hra bude zahrnovat obranné jednotky hráče, které brání svou pevnost (chaloupku). Proti hráčově chaloupce budou chodit hordy nepřátel v jednotlivých vlnách a snažit se pevnost zničit. Jako zajímavý přídavek je ve hře obsažena postavička hráče, která bude pod jeho přímou kontrolou. Hráč tak v jednotlivých vlnách nebude jen sledovat, jak jeho věže (obránné jednotky) ničí nepřátele. Sám se aktivně zapojí do obrany chaloupky. Výsledkem hry bude dosažené skóre při prohře (chaloupka zničena) nebo úplné vítězství (poraženy všechny vlny nepřátel). Když je znám scénář, může být hra také vhodně pojmenována. Ponese název „Perníček TD“.

Pro hru bude nutné vymodelovat herní účinkující (hráče, nepřátele, věže, zbraně, munice, chaloupku) i okolní prostředí (terén, oblohu, dekorace, rostliny, stromy). Modelace bude provedena primárně pro účely hry, tedy s nižšími výpočetními nároky (modely z méně polygonů). Sekundárně budou modely, včetně celé scény, renderovány do kvalitnějších prezentačních obrázků. Jejich modelace bude od počátku zaměřena i na eventuální teselaci, kde její úroveň bude možné přepínat. Teselaci bude také možné poskytnout kvalitnější (méně hranaté) modely hráčům s výkonnějšími platformami. K herním objektům se musí vytvořit materiály. K demonstraci této tvorby se započne procedurálními texturami, které si uživatel může přímo v Blenderu vytvořit sám. Naváže se na ukládání těchto textur do obrázků, které (jakožto textury) jsou podporovány v herním jádře Blenderu (oproti procedurálním texturám). Pro modely bude nutné vytvořit animace pohybu, útoku atd. Krom animací je vhodné ke všem herním úkonům přiřadit i zvukovou složku. Pro některé méně podstatné modely (dekorace) nebo zvuky a textury se využije případné stažení z internetových stránek, které poskytují tato aktiva zdarma. Tím se čtenáři studie demonstruje i možnost tvorby her bez nutnosti vytváření vlastních herních aktiv.

Jednotlivé věže se dají zakoupit v obchodním terminálu. V terminálu jsou poskytovány pod dalšími nabídkami i zbraně, munice a různá vylepšení. Mezi těmito typy zboží se musí přepínat, což je učiněno pomocí tlačítek. Při detekci kurzoru myši nad položkou zboží se zobrazí ke zboží detailní popis, který si vybere data z uložených atributů herního systému. Také tlačítka pro koupi daných produktů nesmí chybět. Tato část hry představuje grafické rozhraní a systém jako pro běžnou interaktivní aplikaci, což je pro čtenáře případové studie značným přínosem ve znalostech tvorby a pochopení univerzality Blenderu. Na podobném „klikacím“ principu bude vytvořeno i herní menu, nabídka s nastavením hry (především rozlišení výstupního obrazu hry a možnost vypnout rozsáhlou dekoraci náročnou na HW) a prezentace s ovládáním hry (klávesové zkratky a namapované tlačítka myši).

Celý systém hry bude naprogramován v jazyce Python, který Blender podporuje. Jelikož herní senzory a aktuátory nejsou možné vytvořit ve skriptech, bude herní logika vytvářena také v logických blocích. Aby uživatel pochopil princip logických bloků, bude v nich vytvořen pohyb hlavní postavy hráče, který by měl být ideální pro úvodní demonstraci. Skripty budou rozděleny do přehledných

modulů, ve kterých bude kód patřičně komentován pro čtenáře, který bude chtít pochopit herní logiku hlouběji a implementovat si vlastní složitější hru. Pro účely efektivnějšího ladění hratelnosti hry bude vhodné vytvořit skript zaujímající pouze přednastavitelné herní hodnoty (útok věží, rychlost pohybu nepřátel, ...). Tím bude hra snadno modifikovatelná.

Hra by měla primárně fungovat na platformě Microsoft Windows, jejíž export na spustitelnou verzi na tomto OS Blender přímo podporuje. Pro účel distribuce hry i studie budou vytvořeny webové stránky, ze kterých si uvedené dílo lidé mohou stáhnout zdarma.

# 5 Případová studie o Blender hře

Kapitola obsahuje případovou studii (jakožto součást vlastního přínosu diplomové práce) na téma tvorby 3D aplikace ve zvoleném vývojovém nástroji – Blenderu. Proč je zvolen zrovna tento nástroj bylo vysvětleno v předchozí kapitole. Studie je zaměřena na vývoj 3D počítačové hry, která nejlépe vystihuje ve vší komplexnosti tvorbu 3D aplikace.

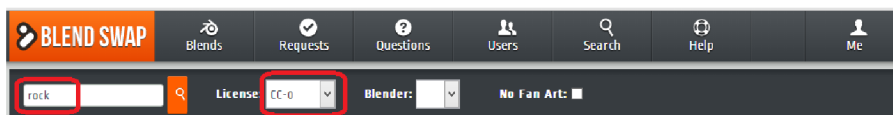
## 5.1 Stažení modelů

Hra se musí skládat z určitých objektů, které budou zaplňovat herní plochu – dekorace, podstatné objekty pro hru (hráč, nepřítel, použitelné předměty). Tyto objekty budou buď vytvořeny v Blenderu nebo staženy z volně dostupných zdrojů. Jelikož je tato práce myšlena jako příspěvek open-source komunitě, bude podstatná část objektů vytvořena vlastnoručně v Blenderu a následně nahrána na distribuční server. Všechny objekty ovšem nemusí herní vývojáři dělat sami a mohou je z takovýchto serverů i stáhnout. V následujícím textu bude vysvětleno, jakými způsoby lze nahrát již vytvořené objekty do vývojářova Blenderu se hrou.

### Blender modely

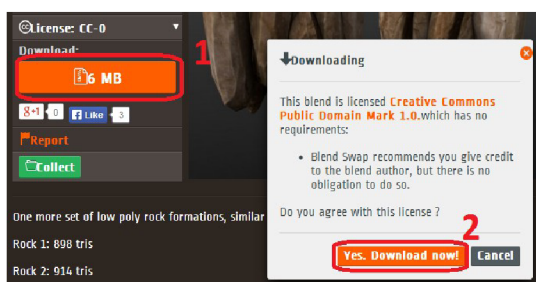
V systému Blend Swap, který umožňuje stažení 3D modelů přímo v Blender formátu (\*.blend), lze stáhnout mnoho povedených modelů. Lze je třdit i podle vztažené licence, což může usnadnit výběr modelu pro vlastní potřebu. Jelikož se jedná o přímo Blenderem podporovaný formát, celý postup bude jednoduchý. Bez nutnosti konvertování formátů a importu. Nevýhodou je nutná registrace s omezeným stahovacím limitem (aktuálně 200 MB/měsíc a při nahrání na jejich server alespoň jednoho modelu 300 MB/měsíc + 10 MB za každý model až do 400 MB/měsíc).

1. Přejde se na stránku: <http://www.blendswap.com/category/staff-picks>.
2. Zadá se do vyhledávače hledaný objekt (pro hru se využije skála, tedy anglicky „rock“ – obr. 1) a nejlépe se využije hledání objektů pod licencí CC-0 – obr. 1, na které se nevztahují žádná licenční práva.



Obrázek 1 – Vyhledávání modelů na stránkách Blend Swap

3. Vybere se ze zobrazené nabídky vhodný model pro hru (povedený je „Low Poly Rock Formation #2“ od uživatele OliverMH).
4. Stáhne se na disk počítače (nejprve se klikne na tlačítko pod Download – obr. 2 (1) a následně na potvrzení stažení, u kterého se stručně vysvětlují požadavky licence – obr. 2 (2)).



Obrázek 2 – Model ke stažení z Blend Swap

Je třeba brát ohled na to, aby model nebyl příliš složitý. Především by neměl obsahovat příliš mnoho polygonů (stěn), jinak by hra nadměrně utrpěla na rychlosti vykreslování a začala by být nehratelná na méně výkonných herních platformách.

## Nepodporované formáty s modely

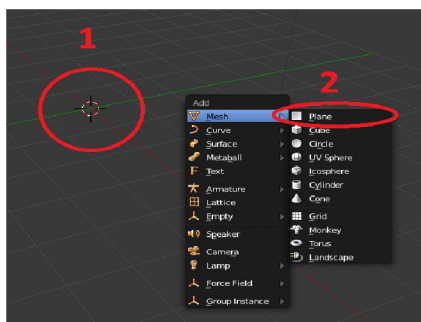
Další možností je stáhnout modely ze SketchUp, který již obsahuje velké množství 3D modelů. Většina z uložených souborů je ve formátu, který Blender přímo nepodporuje. Je tu ovšem možnost přeformátování vytvořených modelů. Celý postup importování nekompatibilních modelů do Blenderu je uveden v příloze A (Import nepodporovaných formátů s modely – str. Chyba: zdroj odkazu nenalezen).

## 5.2 Tvorba vlastních modelů – ateliér

První modely jsou hotovy díky podpoře dobrovolníků, kteří ukládají zdarma své modely na internet. Další model se vytvoří vlastními silami. Začne se s tvorbou jednoduchého prostředí, ve kterém budou vytvářené modely vystupovat. Takovýto „ateliér“ zajistí jednodušší orientaci v prostoru a lepší představu, jak model bude vypadat zasazený do scény. Také je podstatné scénu patřičně osvětlit, jinak by model při vyrenderování nebylo vidět. Přece jenom modely nejsou dělány pouze pro tuto hru, ale také pro open-source komunitu, která si je může stáhnout. Tito lidé budou chtít použít modely pro různorodé účely a je vhodné model připravit i pro renderování mimo real-time, kde se předpokládá vyšší kvalita. Takto vytvořený „screenshot“ modelu zasazený do ateliérového prostředí bude použit jako ilustrační obrázek.

### Vytváření objektů

1. Otevře se nový soubor Blenderu a smaže se přednastavená krychle (kamera i světlo budou zapotřebí).
2. Do středu scény se přidá plocha („SHIFT + A“ zobrazí Add nabídku pro přidání objektu na pozici 3D kurzoru – obr. 3 (1), dále se zvolí Add→Mesh→Plane – viz. obr. 3 (2)), která bude představovat podlahu.

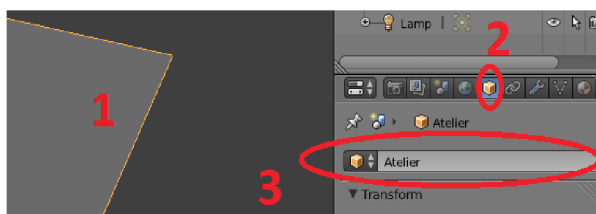


Obrázek 3 – 3D kurzor (1) a přidání plochy (2)

3. Plocha se zvětší na dostatečnou velikost. Dvacet jednotek na šířku bude bohatě stačit („S“, „10“ – číselně na numerické klávesnici, „ENTER“).
4. Následně se vytvoří dvě stěny ateliéru. Podlaha se zduplikuje („SHIFT + D“).
5. Duplikát plochy se otočí k podlaze do pravého úhlu, aby představoval jednu ze stěn („R“, „Y“, „90“, „ENTER“).
6. A posune se o 10 jednotek dozadu, aby přiléhá k podlaze („G“, „X“, „-10“, „ENTER“).



7. Podobně se vytvoří druhá stěna:
  - duplikace podlahy („SHIFT + D“),
  - rotace („R“, „X“, „90“, „ENTER“),
  - posun („G“, „Y“, „10“, „ENTER“).
8. Obě stěny se spojí v jeden objekt. Označí se obě stěny („SHIFT + RMB“ na obě stěny) a spojí se („CTRL + J“).
9. Ještě se tato rohová clona posune nahoru, ať přiléhá i v z-souřadnici prostoru k podlaze („G“, „Z“, „10“, „ENTER“).
10. Celek lze spojit do 1 objektu (výběr stěnového objektu + podlahy a „CTRL + J“) – obr. 4 (1).
11. Pro jednodušší orientaci se všechny zanesené objekty pojmenovávají. Vzniklý ateliér se tedy pojmenuje jako „Atelier“ (název objektu – obr. 4 (3) je dostupný z Object properties – obr. 4 (2)).



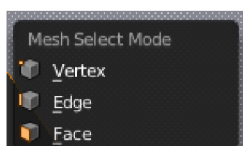
Obrázek 4 – Vytvořený ateliér (1) a v Object Properties (2) změněn název objektu (3)

V předešlém procesu výroby ateliéru byla procvičena tvorba nových objektů (vytvoření, spojení a přejmenování) a jejich manipulace (posun, rotace, změna velikosti). Alternativní postup je uveden v příloze A (Alternativní tvorba ateliéru – str. Chyba: zdroj odkazu nenalezen).

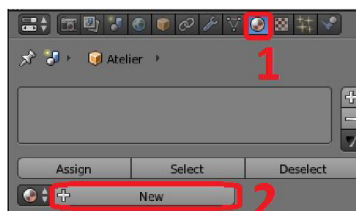
## Materiály

Následuje změna materiálu a textury vzniklých ploch. Ateliér se přetvoří na pouštní terén a stěny na skálu.

1. Přepne se do Edit Mode (v Object Mode musí být označen ateliér, než se přejde k editaci).
2. Vybere se podlaha (přepne se výběrový mód na výběr stěn „CTRL + TAB“ → Mesh Select Mode → Face – obr. 6 a stěna se označí „RMB“).
3. Přejde se do Material Properties – obr. 5 (1) a vytvoří se nový materiál pro podlahu (tlačítko „New“ – 5 (2)). Zbarví se celý objekt do odstínu šedé, ne pouze stěna. To je v pořádku, objekt musí zaujmout nějaký materiál, tedy si sám zvolil první a zatím jediný materiál.



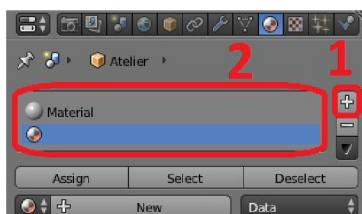
Obrázek 6 – Výběr stran objektu



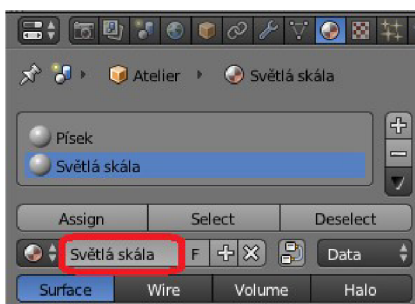
Obrázek 5 – Material Tab (1) a tlačítko New (2)



4. Přejde se z označení podlahy na označení stěn a přidá se další materiál (tlačítko „+“ – obr. 7 (1)). Vytvoří se nový druh materiálu (opět tlačítko „New“, jak tomu bylo u obr. 6 (3)).
5. Nyní se materiál přiřadí k již vybraným stěnám (Material→Assign). Aktuálně jsou objektu Ateliér přiřazeny 2 různé materiály ze seznamu materiálů – obr. 7 (2).
6. Materiál „Material“ se přejmenuje na „Písek“ a materiál „Material.001“ na „Světlá skála“ (textové pole s názvem materiálu je zobrazeno na obr. 8). Názvy jsou opravdu důležité pro orientaci, především pak v rozsáhlejších projektech.



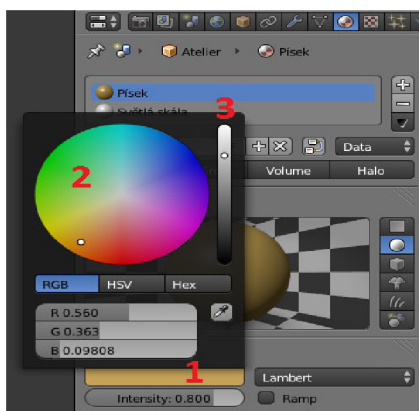
Obrázek 7 – Tlačítko pro další materiál (1) se seznamem materiálů (2)



Obrázek 8 – Změna názvu materiálu

Přichází část transformace materiálů (materiály jsou prozatím ve výchozí šedé barvě). Bude potřeba v Blederu zařídit, aby materiály vypadaly co nejvěrohodněji, aby se blížily skutečnému světu.

1. Přepne se zpět do Object Mode. Jelikož materiály již byli přiřazeny jednotlivým plochám ateliéru, není třeba pokračovat v editaci.
2. Vybere se materiál „Písek“ a změni se jeho Diffuse složka na hnědou barvu („LMB“ na okénko ilustrující barvu difuzní složky – obr. 9 (1), dále jen Material→Diffuse→Colour, otevře se editace barvy, kde se nastaví patřičná hnědá v barevné paletě – obr. 9 (2)). Podlaha ateliéru se automaticky zbarví do hnědé barvy, jelikož má přednastavený materiál „Písek“, který se nyní změnil z šedé na hnědou.
3. Požadovaný efekt bude ovšem o něco tmavší hnědá (posuvníkem na obr. 9 (3) lze plynule nastavit intenzita barvy, při nižší hodnotě se materiálová barva ztmaví). Hodnoty barvy RGB modelu (i dalších, které Blender podporuje, jako je HSV) není třeba odhadovat z barevné palety a intenzity. Barvu je možné zadat přímo číselnými hodnotami, které se pro jednotlivé barevné modely nacházejí pod barevnou paletou.

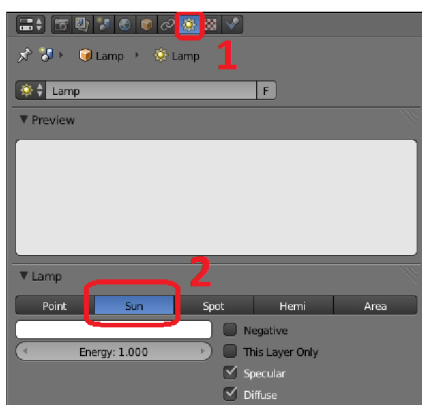


Obrázek 9 – Diffuse Color materiálu (1), barevná paleta (2), posuvník intenzity barvy (3)

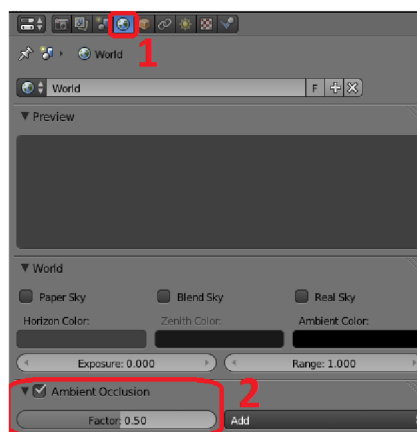
## Osvětlení

Při vyrenderování obrázku („F12“) se zjistí, jak materiál vypadá ve výsledném obrázku. Scéna je ovšem zatím špatně osvětlena, protože se využívá přednastavené bodové světlo s nízkou úrovní energie. Pro scénu se využije sluneční světlo, které dostatečně kvalitně osvětlí scénu a nasimuluje venkovní prostředí.

1. Označí se přednastavené světlo („RMB“) a přenastaví se jeho typ z Point na Sun (Object Data→Lamp – obr. 10 (1) se přepne na Sun – obr. 10 (2)).
2. Světlo se posune (pomocí šipek) mimo prostory ateliéru, ať neruší při pozdějším vývoji modelu.
3. Dále se nasměruje úhel slunečních paprsků (čárkovaná čára u světla reprezentuje směr záření, ten bude upraven pomocí rotace světla), ze kterého světlo bude dopadat na scénu. Předejde se nechtěnému clonění světla stěnami.
4. Stíny, které model bude vrhat, budou v aktuálním stavu osvětlení velmi tmavé. Je nutné je zesvětlit a uměle se tak přiblížit odrazu světla od okolí i do míst, kam slunce přímo nesvítí. K tomuto přiblížení se využije ambientní osvětlení (v záložce World – obr. 11 (1) se označí Ambient Occlusion a změní jeho faktor na polovinu – obr. 11 (2), jinak by byly stíny příliš světlé).



Obrázek 10 – Object Data (1), Sun (2)



Obrázek 11 – Nastavení Ambient Occlusion

5. Je nutno podotknout, že ambientní světlo je všudypřítomné a nepůsobí pouze na stíny, ale na celou scénu. Pokud by se scéna jevila kvůli ambientnímu světlu přesevětlena a zároveň je nutné uchovat stíny světlejší, lze snížit energii světla ze slunečního osvětlení.

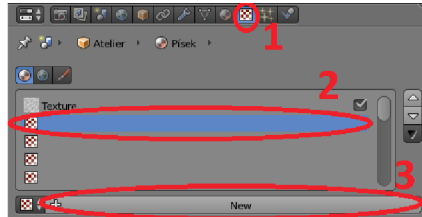
Názorný přehled jednotlivých změn osvětlení je obrázkově demonstrován v příloze A (Přehled změn osvětlení – str. Chyba: zdroj odkazu nenalezen).

## Textura podlahy

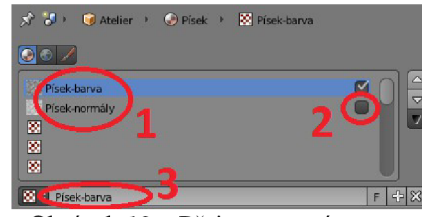
Podlaha barvou může připomínat písek, ale jen vzdáleně. Plocha je bez jakýchkoliv vzorů, jednoduše a působí uměle. K účelu zdokonalení vzhledu materiálu lze využít procedurální textury (bez nutnosti aplikace externí, pevně dané textury/obrázku). Jak barevné textury, tak i normálové.

1. Přejde se do Textures Properties – obr. 12 (1) a vytvoří se nová textura (nová textura se přidává obdobně jako materiál, tlačítkem „New“). Přidá se ještě druhá textura (která se vybere z nabídky textur – obr. 12 (2) a opět „New“ – obr. 12 (3)).

2. Textury se pojmenují jako „Písek–barva“ a „Písek–normály“ – obr. 13 (1) (umístění názvu u textur je obdobné jako u materiálů – obr. 13 (3)).
3. Prozatím se odznačí textura „Písek–normály“ – obr. 13 (2), ať nekoliduje při vývoji textury s barvou „Písek–barva“, kterou se začne.

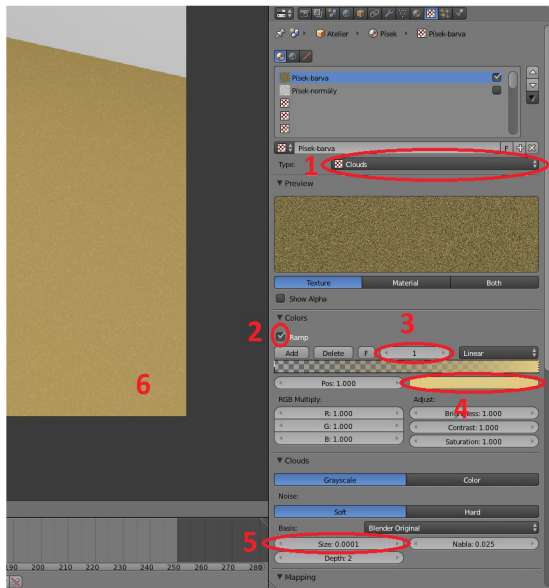


Obrázek 12 – Textures (1), další textura v nabídce (2), vytvoření nové textury (3)

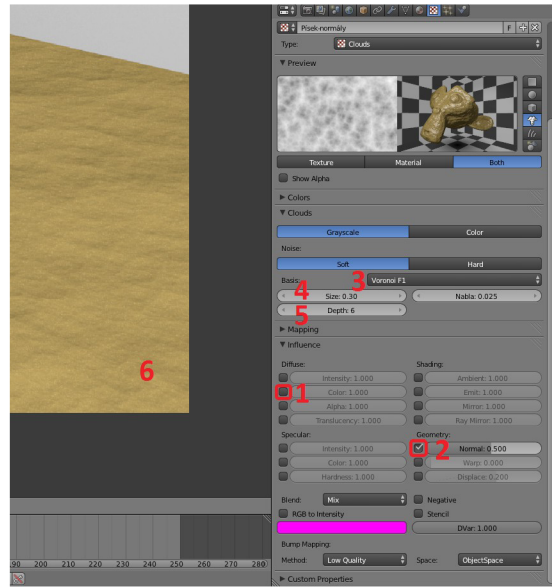


Obrázek 13 – Přejmenované textury (1), odznačená textura (2), pole pro změnu názvu textury (3)

4. Vybere se textura „Písek–barva“ a upraví se tak, aby písek obsahoval jistou zrnitost v podobě různobarevných zrníček. Bude využita procedurální textura typu „Clouds“ – obr. 15 (1) s barevným přechodem. Provede se následující postup:
  - označí se barevný přechod Textures→Color→Ramp – obr. 15 (2),
  - barevná zarážka „0“ se ponechá nezměněna (využívá alfa kanál, barva se využije z již vytvořeného materiálu),
  - vybere se barevná zarážka „1“ – obr. 15 (3),
  - upraví se barva na světle hnědou/žlutou – obr. 15 (4),
  - zmenší se velikost procedurálního vzoru na minimum – obr. 15 (5),
  - po vyrenderování scény („F12“) bude písek vypadat již zrnitě – obr. 15 (6),
  - zefektivnit práci lze postupem z přílohy A (Náhled na texturu – str. Chyba: zdroj odkazu nenalezen).
5. Označí se textura „Písek–normály“ a přejde se na její tvorbu.
  - Označí se možnost výběru barvy (Textures→Influence→Diffuse→Color) – obr. 14 (1).
  - Označí se normály (Textures→Influence→Geometry→Normal) – obr. 14 (2) a nastaví se na hodnotu 0,5. Touto hodnotou se nasimuluje méně drsné zvlnění terénu.
  - Změní se základ Clouds textury (Textures→Clouds→Basis) na Voronoi F1 – obr. 14 (3).
  - Upraví se velikost vzoru textury (Textures→Clouds→Basis→Size) na 0,3 – obr. 14 (4).
  - Hloubka (Textures→Clouds→Basis→Depth) se přenastaví na hodnotu 6 – obr. 14 (5). Se základní hloubkou 2 by byla textura příliš ostře a uměle tvarovaná.
6. Výsledný písečný povrch („F12“) již obsahuje různobarevná zrníčka i zvlnění písku – obr. 14 (6).



Obrázek 15 – Clouds (1), Ramp (2), Color Stop (3), Color (4), Size (5), zrnitý písek (6)

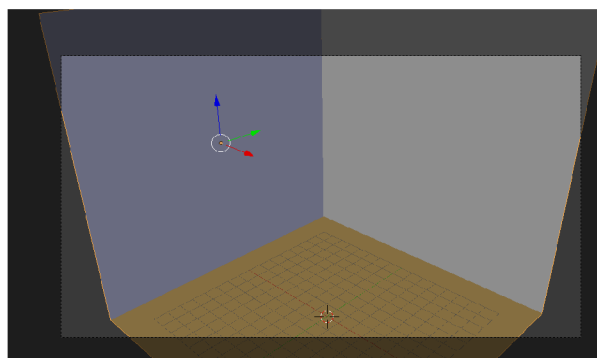


Obrázek 14 – Barvy (1), Normály (2), Voronoi F1 (3), Size (4), Depth (5), výsledný písek (6)

## Posunutí kamery

Přejde se k výrobě textury pro stěny, které budou zhotoveny ze skalního materiálu. Ještě než se začne, posune se kamera výše, aby byla zeď na vyrenderovaném obrázku vidět.

1. Posune se pohled ve 3D View na místo, které má být vyobrazeno kamerou/při vyrenderování (k tomu zle využít:
  - „MW“ pro přiblížení/oddálení,
  - „MMB + pohyb“ pro volnou rotaci kolem nastaveného středu,
  - „SHIFT“ + MMB + pohyb“ pro volný pohyb nahoru/dolů i doprava/doleva,
  - „SHIFT + MW“ pro pohyb nahoru/dolů,
  - „CTRL + MW“ pro pohyb doprava/doleva).
2. Zafixuje se kamera na střed obrazovky – obr. 16 („CTRL + ALT + 0“). Zde záleží na vzdálenosti od středu (čím dále, tím kamera bude dále a pokryje větší oblast).
3. Z pohledu kamery se odejde („0“).



Obrázek 16 – Scéna v pohledu kamery

## Dokončení ateliéru

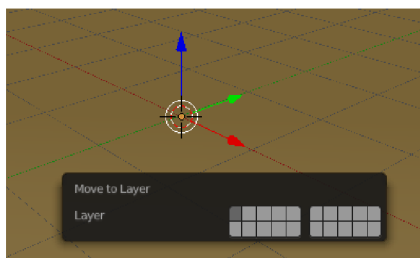
Kamera je přenastavena a na vyrenderovaném obrázku již bude vidět i velká část stěny. Pustí se do její tvorby.

1. Tvorba využívá opět rozhraní textur, jen s jiným nastavením (mramor). Celý postup je zachycen v příloze A (Textura stěn – str. Chyba: zdroj odkazu nenalezen).
2. Nyní je ateliér plně připraven pro vytváření modelů, které budou do něj usazeny na konečný renderovací proces. Ateliér se zvláště uloží jako soubor „ATELIER.blend“ (File→Save As...) a bude následně představovat novou scénu, do které se budou vytvářené modely aplikovat.

## 5.3 Tvorba vlastních modelů – hlavní postava

Začne se s tvorbou hlavní postavy, kterou bude ovládat hráč. Bude nutné vytvořit model postavy a vytvořit na jeho plochy vhodné materiály s texturami. Bude následovat postup vytvoření kostry pro animaci postavy a vytvoření několika animací nutných ke hře (běh postavičky apod.). Hlavní postava bude panáček z perníku a bude se jmenovat „Tomik“.

1. Otevře se dříve uložená scéna „ATELIER.blend“ (File→Open...) a rovnou se uloží jako nový model (File→Save As...) „Pernicek TD.blend“. Včasným uložením scény se předejde nechtěnému přepsání ateliéru, který je základní formou i pro další modely. Další informace o nechtěných chybách při modelaci a jejich opravě jsou v příloze A (Oprava chyb – str. Chyba: zdroj odkazu nenalezen).
2. Přidá se do scény plocha („SHIFT + A“→Add→Mesh→Plain), přejde se do Edit Mode („TAB“) a odstraní se veškeré vertexy z plochy („A“ k označení/neoznačení všech bodů a „X“→Delete→Vertices k odstranění). Zbude jediný bod, který představuje počátek objektu (Pivot).
3. Objekt se přejmenuje na „Tomik“ (změna názvu v Object TAB).
4. Teď se rozčlení vrstvy pro přehlednost modelování. Model se přesune do 6. vrstvy, kde nebudou při modelaci překážet ostatní objekty scény (v Object Mode: „M“→výběr vrstvy, jak jde vidět na obrázku 17). První vrstva bude ponechána celá pro osvětlení a kameru. Ateliér bude přesunut do vrstvy pod první (11. vrstva – zde se později budou odkládat neviditelné objekty). Přejde se do vrstvy s modelem (výběr dané vrstvy na spodní liště 3D View), ve kterém se bude pokračovat.



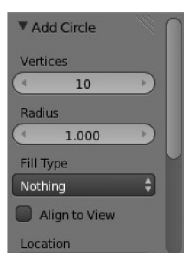
Obrázek 17 – Přesun objektu do jiné vrstvy

5. Přejde se do pohledu z vrchu („CTRL + C“ vycentruje 3D kurzor do středu prostoru a zaměří pohled na něj a „7“ se převede pohled na shora). To umožní efektivní modelování v souřadnicích, na které je člověk zvyklý z grafů. Perníček bude rozložen na ploše os x/y a v ose z bude jeho hloubka. Tímto se modelář vyhne problémům s nevhodnou orientací modelu (po dokončení modelu je třeba model zpět rotací převrátit do roviny s ateliérem).

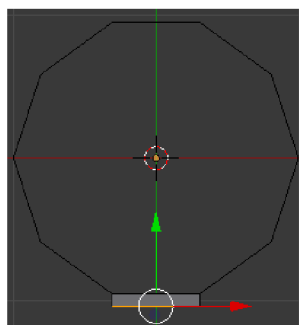
## Hlava

Začne se s modelováním hlavy. Postava bude permíček, takže se vymodeluje jeho čelní strana a ke konci se přidá hloubka modelu.

1. Přidá se k postavě kruh („SHIFT + A“→Mesh→Circle). Změní se počet bodů v kruhu na 10 (v Object Tools se zobrazí Add Circle nabídka – viz. obr. 19), což bude v herním engine vhodnější pro nižší výpočetní náročnost. Právě ta musí být při tvorbě hry neustále na paměti.
2. Hlava se natočí tak, aby spodní hrana hlavy byla rovnoběžná s osou x (je nutné být v módu výběru bodů). Dostane se tak prostoru pro „extrude“ krku. Přejde se do výběrového módu hran a spodní hrana hlavy se pomocí „extrude“ („E“) vysune dolů za vzniku krku – obr. 18.



Obrázek 19 –  
Add Circle



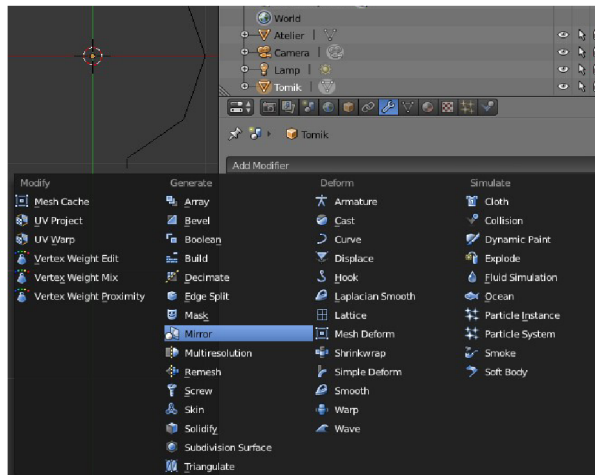
Obrázek 18 – Hlava s  
krkem

## Tělo

Tělo, které bude navazovat na krk, má již dvě symetrické strany (levá, pravá), které by se zvlášť modelovaly s udržením symetričnosti obtížně. Proto se využije modifikátorů, které Blender poskytuje.

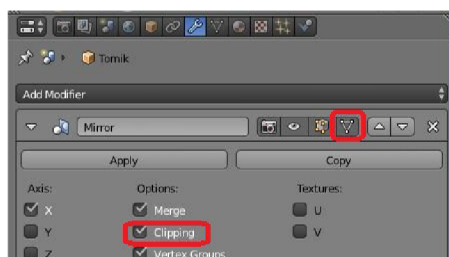
1. Odstraní se levá polovina hlavy. Ve vertex módu („CTRL + SHIFT + 1“) se vyberou body z levé poloviny dosavadního modelu (pomocí box-select tool „B“) a odstraní se.
2. Smazané body levé strany (i následující, co budou vytvořeny pro tělo) budou vytvářeny zrcadlově podle pravé strany objektu. Toho se dosáhne využitím zrcadlového objektového modifikátoru – Mirror (Object Modifiers→Add Modifier→Mirror – obr. 20).



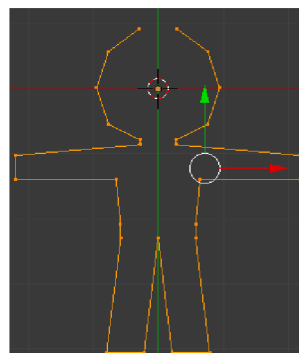


Obrázek 20 – Přidání modifikátoru Mirror

3. V nastavení zrcadlení se přidá funkce zachytávání bodů v rovině zrcadla k sobě, jinak by při posunutí těchto bodů vznikaly mezery, zrcadlené strany by nepřiléhaly k sobě (Object Modifiers→Mirror→Clipping). Zároveň přidáme zobrazování zrcadlení i v editačním módu, ať je celý model patrný už při modelování. Obě nastavení jsou zobrazena níže na obrázku 22.
4. Označí se pravý bod (pracuje se pouze s pravou polovinou modelu) krku a přes „extrude“ se vytváří zbytek těla. Není již nutné využívat žádné elementární tvary k symetričnosti, jak tomu bylo u hlavy. Model se zakončí v rozkroku, kde se na základě zapnutého připínání „Clipping“ bod nerozlučně spojí v jeden s levou polovinou těla. Prozatím stačí načrtnout základní tvar – obr. 21. Vrch hlavy se spojí až po aplikování zrcadlení.



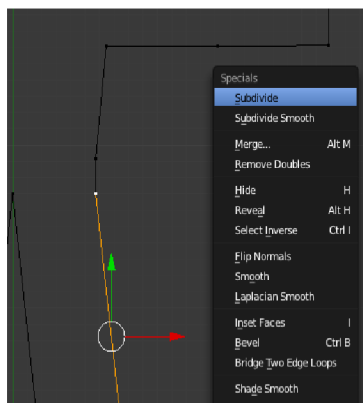
Obrázek 22 – Úprava modifikátoru Mirror



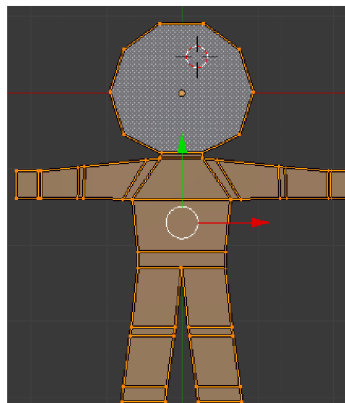
Obrázek 21 – Základní tvar modelu

5. Už v této fázi je třeba myslet na pozdější animaci postavy ve hře. Model se bude pohybovat a kroutit v kloubních oblastech těla. Při animování pohybu by vznikaly nevzhledné lomy stěn, kterým je třeba se vyhnout. Dokud je aplikováno zrcadlení, využije se k vytváření nových hran v místech kloubů. Když je do existující hrany třeba přidat bod (zde do oblasti kloubů kolen a loktů, využije se rozdělení (označí se hrana, „W“→Specials→Subdivide – obr. 24). Nový bod přibude uprostřed hrany, na jinou pozici lze přemístit transformacemi („G“).
6. Model by měl mít v kloubních pozicích alespoň 2 hrany pro animaci (nevzhledné minimum je 1). Nově vzniklé body v kloubech je třeba spojit (označí se protilehlé body a aplikuje se „fill“ – „F“). Protilehlé body v pasu a body vrchu hlavy se spojí až po aplikaci modifikátoru

zrcadla (tlačítko „Apply“ u modifikátoru „Mirror“). Výsledná přední strana postavy bude vypadat tak, jak je zobrazeno na následujícím obrázku 23.

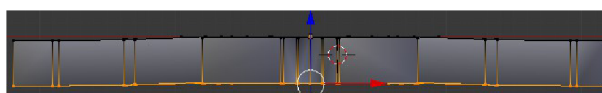


Obrázek 24 – Subdivide hrany modelu



Obrázek 23 – Výsledná přední strana modelu

7. Zbývá modelu přidat hloubku. Přejde se do čelního pohledu („Num 1“), označí se všechny body a model se „extruduje“ ve směru osy z – obr. 25. Některé operace vytvářejí nechtěné vertexy – informace o nich a případné odstranění je uvedeno v příloze A (Duplicitní vrcholy – str. Chyba: zdroj odkazu nenalezen). Tímto je základ modelu celého těla hotov.



Obrázek 25 – Extrude přední strany modelu do hloubky

## Obličej a doplňky

K tělu je ještě nutné přidat obličej (oči, ústa, ...) a doplňky (knoflíky, čepice a další). Detailnější postup je již v příloze A (Obličej a doplňky – str. Chyba: zdroj odkazu nenalezen).

## 5.4 Materiály a textury

V následující kapitole je ukázán způsob vytváření materiálů a textur pro model. Textury jsou nejdříve vytvořeny procedurálním postupem (tedy pomocí nastavených parametrů se matematicky vytvoří vzor textury) a poté „zapečeny“ (uloženy do externího obrázku). Textury z obrázku jsou již využitelné i pro herní jádro Blender Game.

Vytvoření základních materiálů pro model i vytvoření složitějších materiálů s procedurálními texturami (perník) není o moc odlišnější oproti tvorbě materiálů pro ateliér. Tyto postupy jsou uvedeny v příloze A (Materiály a textury modelu – str. Chyba: zdroj odkazu nenalezen).

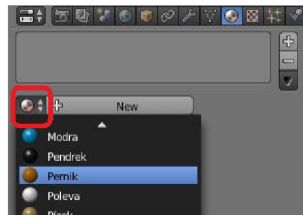
## Textura z externího obrázku

Prozatím se textury zobrazují pouze po vyrenderování klávesou „F12“. Ve skutečné hře se ale renderovat tímto způsobem nebude. Promítání textur na model probíhá real-time a proto z dobrého důvodu (výkon) není ani podporováno v herním engine zpracování procedurálních textur. V Blender Game Engine je nutné mít na modelu namapovány textury z již vytvořených obrázků. Procedurální textura perníku vytvořená dříve ale nepříjde vniveč. Blender umožňuje techniku „zapékání textur“,

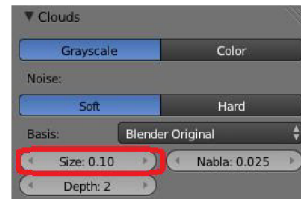


při které je možné procedurální texturu uložit do nového rastrového obrázku. Obrázek je poté využit k texturování založeném na UV mapování. Postup vytvoření takovéto textury je následující.

1. Vytvoří se 2 nové plochy nad sebou (Add→Mesh→Plane). Na tuto činnost se využije volné vrstvy scény (layer 2).
2. Ke spodní se zvolí materiál perníku (obr. 26). Protože se bude „zapékat“ jen přibližný detail textury, aby se snížila paměťová náročnost obrázku (a je to znát i na výkonu stroje při hře), zvýší se velikost vzoru obou textur (Clouds→Size: 0,1 – obr. 27). Jelikož textura bude na modelu poměrně malá a má schopnost se bez viditelných znaků opakovat (tzv. „seamless texture“), bude působit detailně. Malá kapacita potřebné paměti s udržením detailnosti bude velkou výhodou ve hře.

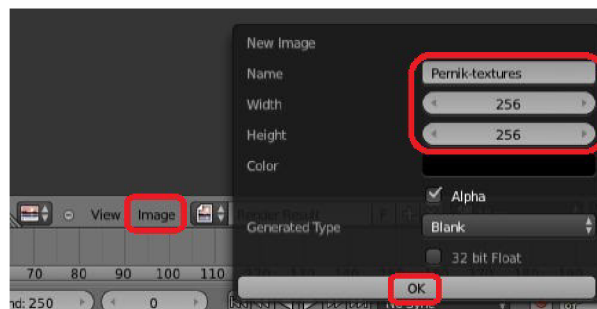


Obrázek 26 – Výběr materiálu perníku



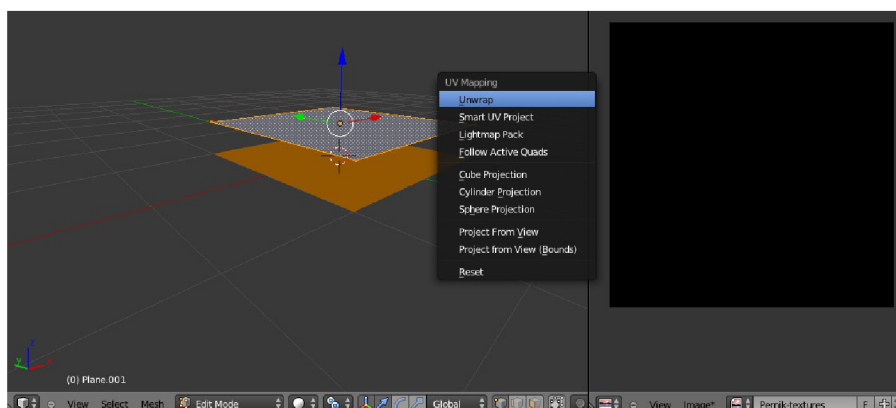
Obrázek 27 – Změna velikosti textury

3. Pro vytváření textury se rozdvojí obrazovka, kde druhá část se přepne na UV/Image Editor.
4. V UV/Image Editor se vytvoří nový obrázek (Image→New Image – obr. 28). Upraví se název obrázku, rozlišení na 256x256 (pro opakující se texturu bude stačit) a potvrdí se „OK“.



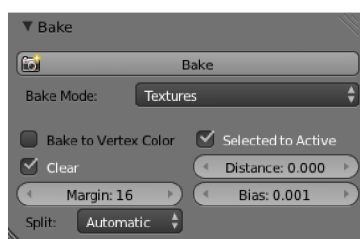
Obrázek 28 – Vytvoření nového obrázku

5. Ve 3D view se označí plocha bez materiálu. Přejde se do Edit Mode. Označí se všechny body plochy a provede se rozbalení – „Unwrap“ („U“ – obr. 29).



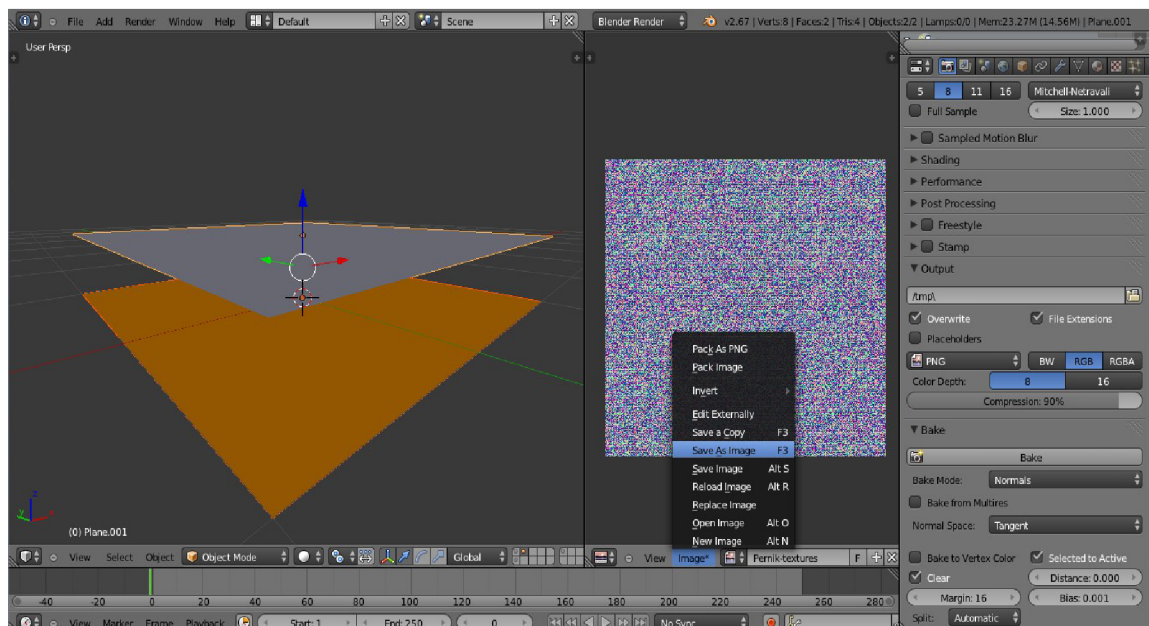
Obrázek 29 – Rozbalení plochy

6. Přejde se zpět do Object Mode, ve kterém se označí nejdříve plocha s materiálem „Pernik“ a k ní plocha bez materiálu (obě musí být označeny). Do této plochy (vytvořeného obrázku plochy) se bude ukládat textura.
7. V Properties Editor se vybere tabulka Render. Na jejím konci je položka Bake, která se po rozvinutí nastaví tak, jak je tomu na obrázku 30 (označí se Selected to Active a Bake Mode se přepne na Textures). Přes Bake Mode se dává Blenderu najevo, jaký typ textury se má „zapéct“. V tomto případě to je barevná složka.



Obrázek 30 – Nastavení Bake

8. Po kliku na tlačítko „Bake“ se textura uloží do obrázku.
9. Následně je možné v UV/Image Editor hotový obrázek uložit (Image→Save As Image, opět obrázek pojmenovat, vybrat vhodnou lokaci pro obrázek v souborovém systému a stisknout Save As Image).
10. Pro zapečení normál se v Render→Bake změní Bake Mode na Normals. Opět je nutné vytvořit nový obrázek v Blenderu (tentokrát název: „Pernik-normals“, rozlišení: 256x256) a stisknout tlačítko Bake.
11. Po zapečení se obrázek uloží pod stejným názvem („Pernik-normals“). Celek je možné vidět na obrázku 31.
12. Po procesu se plochy smažou („X“) a vrstva se opět uvolní.

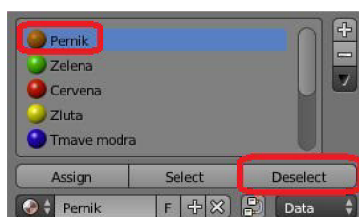


Obrázek 31 – Uložení obrázku s vytvořenou texturou

## Mapování textury

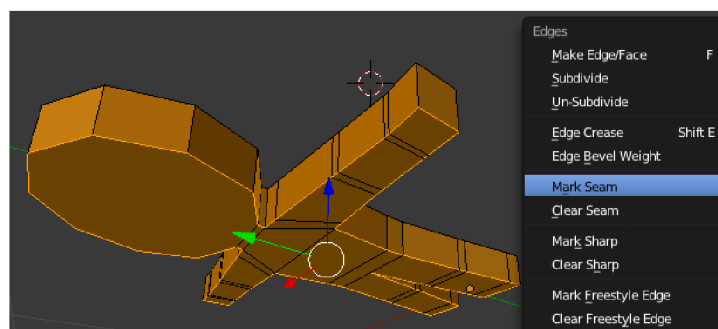
S vytvořenými obrázky textur je možné textury aplikovat na model. Bude zapotřebí model perníčku rozložit pomocí švů („seams“) na jednotlivé ostrůvky stran, které následně bude možné namapovat k textuře.

1. Ve vrstvě s perníčkem se vybere jeho model a přejde se do Edit Mode.
2. Ponechá se viditelné pouze tělo modelu z perníku. Označí se všechny vrcholy modelu. V tabulce s materiály se vybere materiál „Pernik“ a klikne se na „Deselect“ – obr. 32.



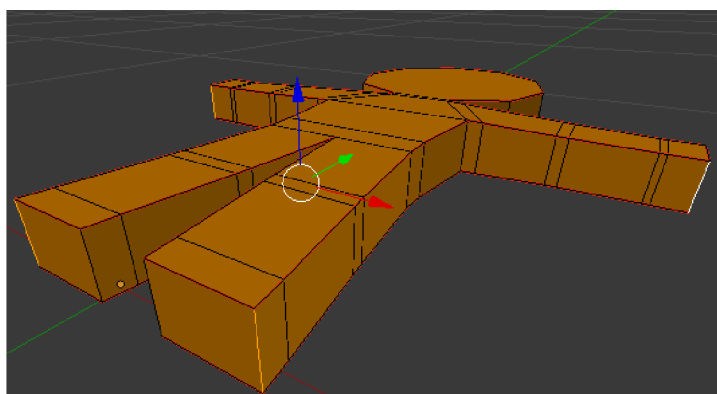
Obrázek 32 – Odznačení vertexů spojených s materiálem perníku

3. Aktuálně jsou vybrány všechny vrcholy krom perníku a je možné je schovat (s myší ve 3D View se stiskne klávesa „H“). Zůstane v modelu viditelné pouze perníkové tělo a zbytek modelu nebude překážet při zaznamenávání švu.
4. Označí se hrany spodní poloviny těla a zaznamenají se jako šev („Ctrl + E“ → Edges → Mark Seam) – obr. 33.



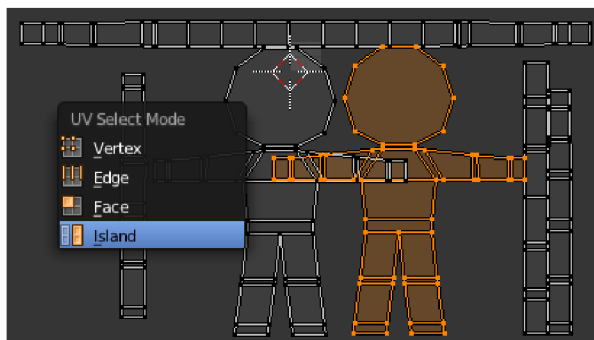
Obrázek 33 – Zaznačení švů jedné strany modelu

5. To samé se provede s vrchní stranou modelu. Tímto jsou označeny dva ostrovy stěn a třetí se nachází mezi nimi. Tvoří ovšem smyčku podél celého těla, což je nežádoucí.
6. Smyčka bočních stěn modelu se rozkouskuje na několik menších. Lze vybrat hrany, jak je tomu na obrázku 34 a opět je zaznamenat jako šev.



Obrázek 34 – Zaznačení švů boků modelu

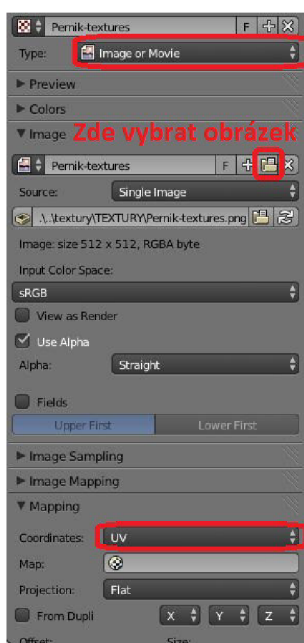
7. Označí se všechny vrcholy modelu a model se rozbálí („U“→UV Mapping→Unwrap).
8. V UV/Image Editor se zobrazí vertexy rozbaleného modelu. V něm lze vybrat všechny vrcholy modelu a přes „Scale“ mapování roztáhnou, případně jinými transformacemi upravit.
9. Jelikož v tomto případě se jedná o jednolitou, opakující se texturu, tak na pozici a rotaci mapovaných vrcholů nezáleží. Mohou se dokonce mapování přes sebe překrývat nebo zasahovat mimo obrázek (obrázek se automaticky opakuje). V módu výběrů nad UV/Image Editor přibyl typ Island – obr. 35, kterým lze celý ostrov vrcholů označit a transformovat dle potřeby.



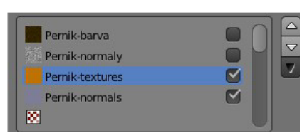
Obrázek 35 – Výběr ostrovů vertexů UV mapy

Vrcholy jsou namapovány na obrázek s texturou, ale při vyrenderování scény se zobrazí pouze model s původní procedurální texturou. Je třeba tyto textury vypnout a vytvořit nové s UV mapováním koordinátů vertexů do obrázku.

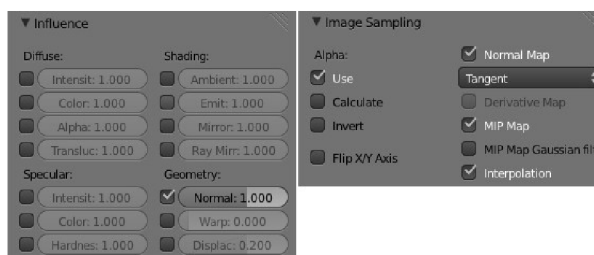
1. Mezi texturami materiálu „Pernik“ se vytvoří 2 nové. Označí se a původní textury se vypnou – obr. 36.
2. „Pernik–textures“ se nastaví podle obr. 37. Ikonou se složkou se vybere požadovaná textura s barvou – „Pernik–textures“.
3. Pro „Pernik–normals“ se provede stejné nastavení, jen se změní výběr obrázku ze souborového systému na „Pernik–normals“ a ve slotu Influence se přepne na Normal (obr. 38 vlevo). Hodnotou Normal se ovlivňuje hloubka plasticity oproti originálnímu obrázku. V Image Sampling je potřeba zapnout Normal Map a nastavit na Tangent (na obr. 38 vpravo).



Obrázek 37 – Nastavení textury „Pernik–textures“



Obrázek 36 – Přenastavení textur materiálu na nové



Obrázek 38 – Speciální přenastavení pro normálovou texturu

4. Po vyrenderování scény se dostane ke zhlédnutí model s novými texturami z obrázků. Pokud je textura příliš malá nebo velká, jde ji upravit v UV/Image Editor pomocí „Scale“ rozbalených ostrovů modelu.

## 5.5 Renderování a jeho úprava

V následujících odstavcích je představeno přepnutí „shadingu“ (způsobu stínování a vykreslování) modelu i celé scény. Získá se tak vizuálně přitažlivější model a také real-time náhled na scénu s modely v podobě, jak bude vypadat při spuštění hry. Po provedených změnách je ukázán i způsob vytvoření reprezentativního obrázku se scénou.

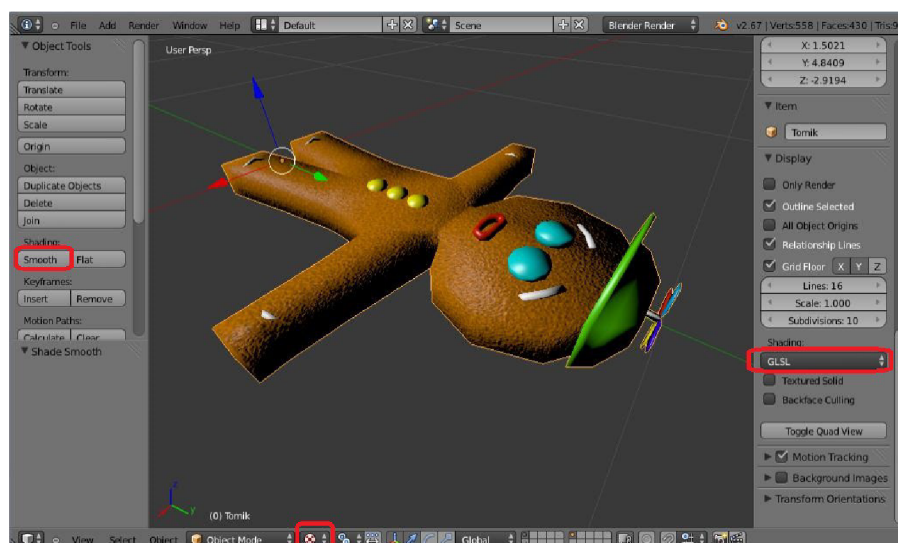
### Shading

Neustálé renderování „F12“ je k zjišťování vzhledu textur na modelu nepraktické. Blender umožňuje GLSL shading, při kterém se scéna ve 3D View zobrazuje zhruba jako při samotném vyrenderování. Navíc se tento mód bude aplikovat pro chod hry.

Dalším vylepšením bude zapnutí Smooth shading místo aktuálního Flat shading. Model se bude stínovat vyhlazeně. Ovšem ne na všechny modely se hodí. Ostrost hran modelu za Smooth shading mizí (stěny, které mají být ostře řezané, jsou stínovány jako zaoblené – nevhodné pro krabice apod.).

1. Zapne se GLSL shading namísto Multiresolution, které doposud bylo ve scéně aktivní („N“→Display→Shading→GLSL) – obr. 39.
2. Přepne se Viewport Shading ze Solid na Texture (na spodní liště 3D View) – obr. 39.
3. Aktivuje se Smooth Shading („T“→Object Tools→Shading→Smooth) – obr. 39.

Práce hned pouhými pár změnami vypadá o poznání lépe. Ovšem GLSL funguje jenom s texturami z obrázků. Procedurální textury nebudou v tomto módu vyobrazeny.



Obrázek 39 – Přepnutí zobrazení modelu i scény

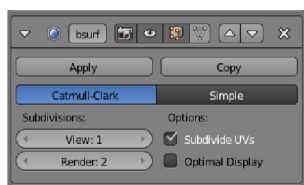
### Teselace

Ze základu byl model vytvářen jako „low-poly“, tedy z málo polygonů, málo vrcholů. Účelem tohoto provedení je udržet co nejvyšší FPS při hře. Pro plynulou hru by hra měla fungovat s alespoň 25 FPS. Na silnějších strojích se hladiny FPS mohou pohybovat zbytečně vysoko a hráč by si za vizuálně stejné plynulosti hry mohl užít kvalitnější grafiky, více detailů modelu bez kostřbatých hran. K tomuto účelu je možné v Blenderu nastavit teselaci modelu s různými úrovněmi.

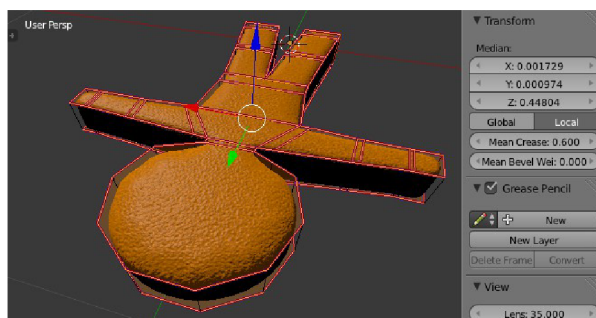
1. Vybere se model perníkového panáčka.



2. V Object Modifiers se přidá modifikátor Subdivision Surface (Object Modifiers→Add Modifier→Subdivision Surface). Modifikátor je přednastaven na Subdivisions: 1 ve 3D View editoru a na Subdivisions: 2 při renderování – obr. 40. Také má užitečnou vlastnost úpravy UV map při teselaci, tedy i mapování textury s perníkem a jiných, která je opět přednastavena v základu.
3. Přenastaví se View Subdivisions na 2. Hlava se vyhladí z kostrbatého osmistěnného válce do kulatějšího tvaru.
4. Ne vždy je objekt teselován podle představ modeláře. Je zde ale možnost úpravy nastavení, podle kterého se má teselovat. V Edit Mode se vyberou přední a zadní stěny modelu.
5. Zvýší se Mean Crease v liště napravo („N“→Transform→Mean Crease: 0,6) – obr. 41, čímž se teselované stěny přitáhnou k původní stěně.



Obrázek 40 – Subdivision Surface



Obrázek 41 – Úprava Mean Crease

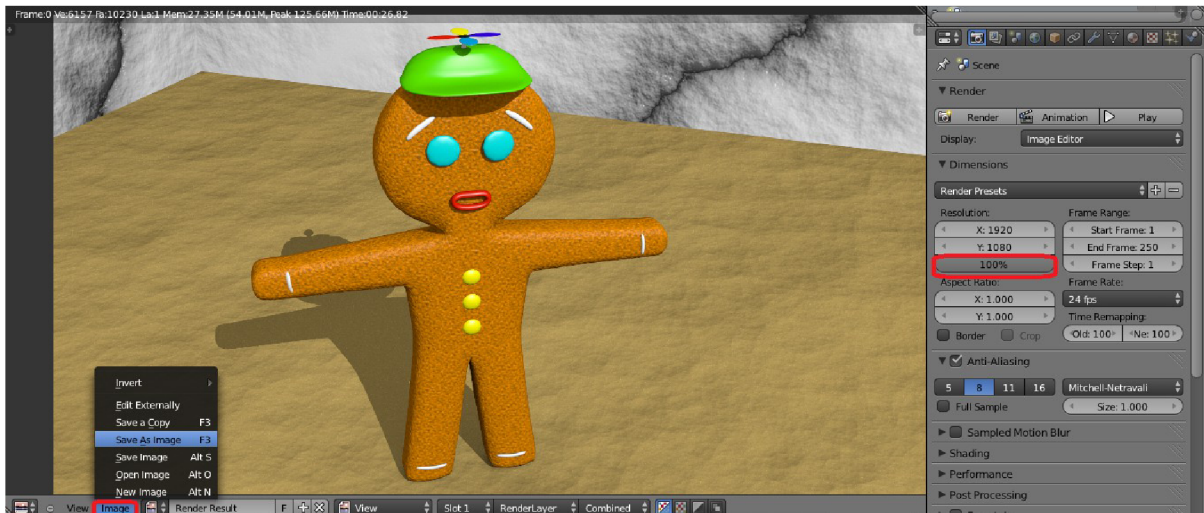
6. S teselací ovšem přichází další problémy. Jednotlivé části modelu nemusí vždy přiléhat k sobě. Někdy je nutné po zapnutí teselace i model upravit, pokud nestačí nastavení Mean Crease. Případně je možné udělat kompromisy mezi různými úrovněmi teselace s umístěním částí modelu. Další možností je vytvořit několik rozdílných modelů (jeden pro hru, druhý pro vykreslení obrázku). Blender umožňuje změnu Mean Crease nejen pro stěny modelu, ale také pro hrany. V tomto duchu se upraví celý model (v Edit Mode je stále skryt zbytek modelu, jen tělo je zobrazeno, skrytí lze zvrátit přes „Alt + H“), pokud různé úrovně teselace mají být ve hře podporovány. Pro vyrenderování kvalitního obrázku budou vysoké úrovně teselace k užítku.

## Vytvoření obrázku

Model je prakticky hotov. Zbývá nastavení scény, jeho vyrenderování a uložení obrázku. Obrázek může sloužit pro reprezentaci modelu na serverech poskytujících 3D modely nebo také jako pozadí pro menu hry.

1. Umístění do ateliéru (výběr všech vrstev scény) se zapnutou teselací pro render (Object Modifiers→Subsurf→Subdivisions→Render: 2).
2. Upravení scale, translace, rotace modelu, aby jeho předeek směřoval z ateliéru.
3. Úprava směru osvětlení (u slunce stačí rotací).
4. Nastavení kamery („Ctrl + Alt + Num 0“ z pozice ve 3D View, která je ideální pro snímek).
5. Vykreslení reprezentačního obrázku (v Render→Dimensions→Resolution přenastavit z 50%, polovičního rozlišení, na 100% – obr. 42 a „F12“).
6. Vykreslený obrázek uložit do počítače (zobrazený UV/Image editor s vykresleným obrázkem, Image→Save As Image – obr. 42).





Obrázek 42 – Úprava rozlišení a uložení obrázku

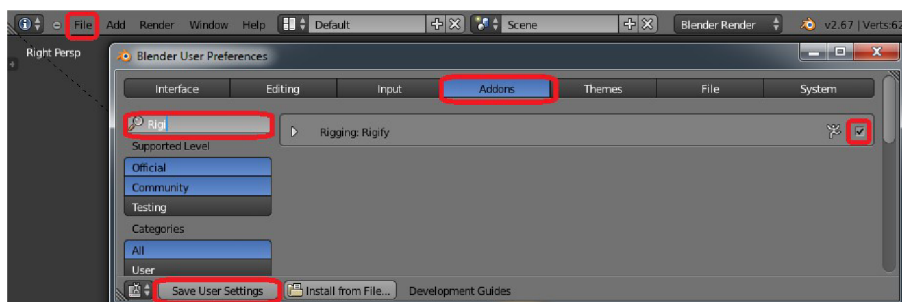
## 5.6 Animování

Nyní se přejde ke kroku, který bude sloužit jednak pro jednodušší úpravu pózy pro vytváření obrázků, ale především pro vytváření pohybů modelu (běh, skok) pro hru. Vytvoří se kostra pro model (armatura), určí se váhy všech částí modelu pro jednotlivé kosti armatury a nakonec se vytvoří pohyb pomocí snímkování pozic a rotací kostí.

### Tvorba armatury

Započne se vytvořením armatury, kterou je možné získat pomocí pluginu. Jedná se o kostru pro člověka, bude tedy nutné provést úpravy, aby odpovídala kostře pro perníkového panáčka.

1. Aktivuje se plugin s názvem „Rigging: Rigify“ (File→User Preferences..., pod Addons se do vyhledávacího pole napíše podřetězec názvu pluginu a plugin se aktivuje – vše je vyznačeno v obr. 43). Pro uchování aktivity pluginu i při novém spuštění Blenderu se zadá Save User Settings – obr. 43.



Obrázek 43 – Aktivace pluginu pro lidskou armaturu

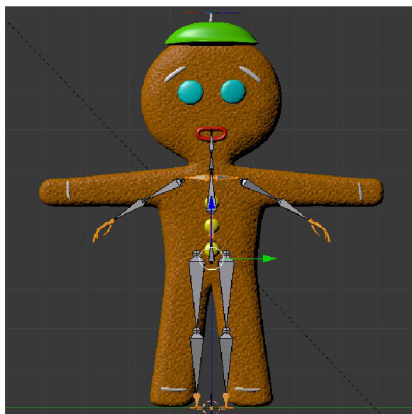
2. K modelu perníkového panáčka se přidá nově aktivovaná položka „Human (Meta-rig)“ (Add→Armature→Human (Meta-rig)).
3. Změní se název objektu na „Tomik–kostra“.

4. Kostra bude zasazená do těla modelu, protože je ale nutná editace kostry, která by jinak v těle nebyla vidět, je užitečné zapnout X-Ray mód pro tento objekt – obr. 44. Kostra bude prosvítat přes ostatní objekty.

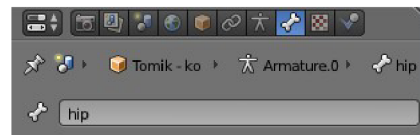


Obrázek 44 – Aktivace X-Ray

5. Oba objekty se natočí do stejné roviny (perníček lze rotovat do původního stavu zkratkou „Alt + R“ – pohled zprava). Armatura se posune na pozici panáčka (měla by být přesně vycentrovaná) a zvětší na jeho velikost.
6. Z armatury se smažou přebytečné kosti (tyto kosti jsou označeny na obr. 45), které panáček oproti člověku nemá. Půjde o dlaně a chodidla (včetně prstů). Ani klíční kosti nebudou potřebné a ani páteř (panáček vystačí s hrudní kostí protaženou k pánvi („hip“, názvy kostí lze nalézt v Bone tabulce – obr. 46). Editace kostí je dostupná opět v Edit Mode. Pose Mode se bude prozatím vyhýbat, dokud kostra není hotova.

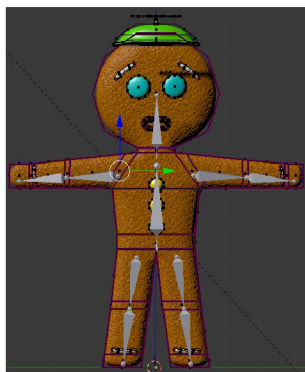


Obrázek 45 – Označené přebytečné kosti

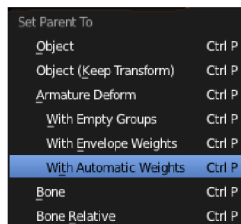


Obrázek 46 – Bone Tab

7. Dále je nutné upravit pozice kostí. Klouby by měly sedět u hran modelu určených pro klouby – obr. 49.
8. Model se přiřadí ke kostře, jakožto k rodiči (vybere se v Object Mode model, pak k němu kostra a „Ctrl + P“) s možností automatického vážení modelu (Set Parent To→Automatic Weight – obr. 47).
9. Pro barvení vah bude kostra v Pose Mode a s modelem se přejde do Weight Paint Mode – obr. 48. Tím, že je armatura v Pose Mode, bude možné přímo z modelu označovat jednotlivé kosti. Viewport Shading se přepne na Solid – obr. 48, aby bylo vidět obarvení modelu pro označenou kost.



Obrázek 49 – Úprava kostí

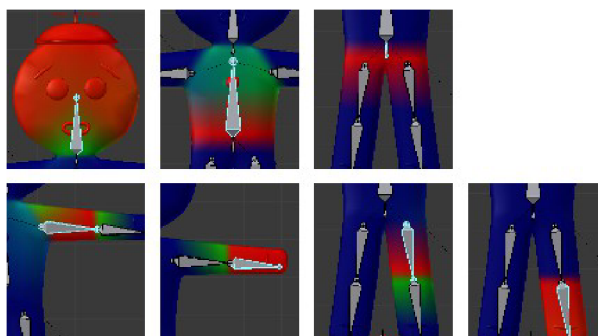


Obrázek 47 –  
Nabídka Set Parent  
To



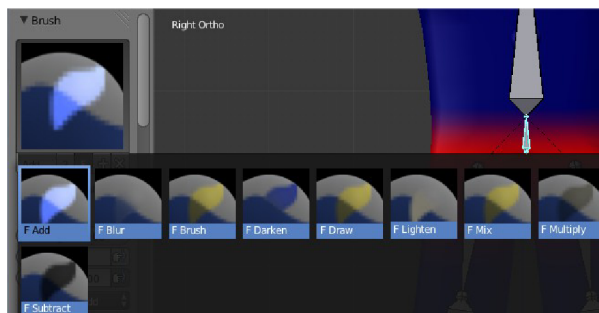
Obrázek 48 – Weight  
Paint a Solid módy

10. Při dříve zvolením Set Parent To With Automatic Weights je model částečně rozumně obarven podle okolních kostí. Při pohybu vybrané kosti (rotaci), ovšem ne všechny oblasti se chovají ve všech úhlech podle představ modeláře a je potřeba je doladit. Zpravidla platí, že část modelu kolem kosti má největší váhu a části modelu za klouby menší, ale přece jen nějakou, aby nedocházelo k ostrým lomům při pohybu kostí. U hlavy, trupu, ... musí být plně obarveny (červeně) všechny přilehlé doplňky, aby se pohybovaly s kostí v celku a nedeformovaly se podle okolí (př.: protahující se knoflík trupu, který nemá být pružný). Obě strany modelu (levá i pravá) je vhodné obarvit stejně. Nesmí se zapomenout na obarvení i zadní strany modelu, které bude stejné jako z přední strany. Na následující koláži obrázků 50 jsou obarvení všech kostí panáčka.

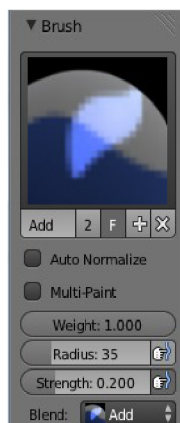


Obrázek 50 – Obarvení všech kostí panáčka

11. Barví se štětci Add (přidání váhy – do červena) a Subtract (odebrání váhy – do modra), které jsou k nalezení na Tool Shelf („T“). Přepíná se mezi nimi kliknutím na ikonu štětce (obrázek 51 ukazuje výběr štětce). Intenzitu přidávané/odebírané váhy lze upravit posuvníkem Weight, poloměr štětce posuvníkem Radius (obrázek 52).



Obrázek 51 – Výběr ze sady štětců

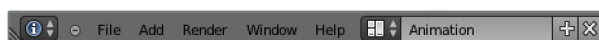


Obrázek 52 –  
Nastavení štětce

## Animace pohybu

S vytvořenou armaturou a obarveným modelem pro jednotlivé kosti armatury je možné začít vytvářet animace pohybů. Využijte se snímkování polohy a především rotace kostí v závislosti na čase (přesněji snímku). Vytvoří se jako první animace chůze vpřed. Další animace (chůze zpět, úkrok, běh vpřed, běh vzad, skok, střelba z několika typů zbraní, ...) se vytvoří podobným způsobem.

1. Model musí být v Pose Mode.
2. Přepne se rozložení obrazovek do animačního rozložení (z Default do Animation – obr. 53).



Obrázek 53 – Animation Layout

3. Zobrazí se 4 podstatné editory: DopeSheet, Graph, Timeline a 3D View. V DopeSheet se přepne z módu DopeSheet na mód Action Editor a tlačítkem New (obr. 54) se vytvoří nová akce.



Obrázek 54 – Nastavení DopeSheet

4. Nová akce se přejmenuje z přednastaveného názvu „Action“ na „chuze vpred“ (obr. 55) a označí se příznak „F“ – obr. 55. V příloze A je tento příznak vysvětlen (Nepoužívané databloky – str. Chyba: zdroj odkazu nenalezen)

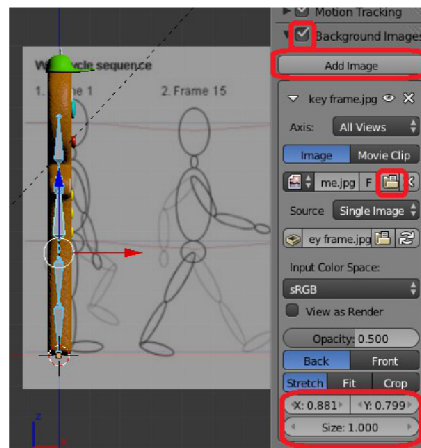


Obrázek 55 – Přejmenování  
akce a označení „F“

5. Ve 3D View lze na pozadí vykreslit pomocný obrázek s průběhem animace. Pokud není požadovaný obrázek nalezen (nebo se jedná o unikátní pohyb), musí být animace provedena bez šablony, což je poměrně náročný proces. V pravé boční liště („N“) se nachází kolonka

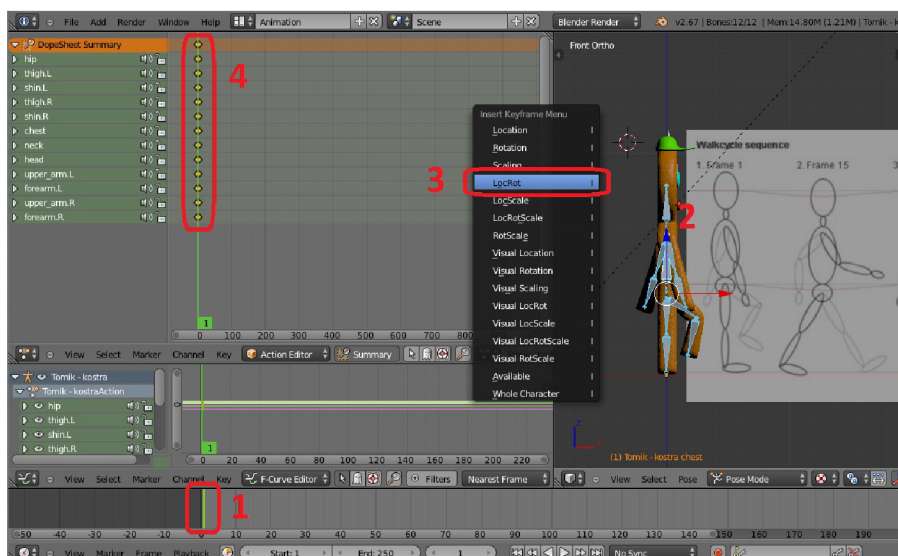
Background Images, kterou je nutné aktivovat, kliknout na Add Image a vybrat obrázek ze souborového systému. Obrázek se šablonou chůze panáčka byl nalezen na webové stránce: [http://www.erain.com/blog/archives/2004/08/introductory\\_wa\\_1.html](http://www.erain.com/blog/archives/2004/08/introductory_wa_1.html).

- Poté je potřeba nastavit velikost obrázku (Size), aby odpovídal model ze šablony vlastnímu modelu a pozici obrázku (X a Y) – obr. 56. Obrázek v pozadí se bude zobrazovat při základních pohledech dostupných ve View na spodní liště (Right, Top, Bottom ...) při ortogonálním nastavení.



Obrázek 56 – Obrázek na pozadí scény

- V Timeline Editor se označí požadovaný snímek, pro který bude vytvářena póza modelu (první snímek bude Frame 1 – obr. 57 (1)).
- Transformacemi (zde rotací) se v Pose Mode armatura modelu nastaví do první polohy (2), označí se změněné kosti (zde všechny) a vytvoří se snímek (nad 3D View klávesa „I“→Insert Keyframe Menu→LocRot (3)). Přidaná data se zobrazí v ostatních editorech (4). Pokud je nutné některé z uložených dat smazat, je možné je vybrat v DopeSheet Editor a odstranit jako běžné objekty v Blenderu.



Obrázek 57 – Zaznačení Keyframe



- Vytvoří se pózy pro všechny klíčové snímky (zde je další snímek na pozici Frame 15, 30, 45 a 60). Obrázek se šablonou se posouvá přes položku uvedenou dříve – X. Při opakovaném pohybu, jako je chůze, je nutné, aby první snímek a poslední byly stejné, jinak nebude animace plynulá. K tomuto účelu je možné označit všechny body prvního snímku v DopeSheet (Box Select „B“), „Ctrl + C“, najetí v časové ose na poslední pozici (Frame 60) a „Ctrl + V“.
- Animace je hotova. Její vizualizaci lze přehrát a ověřit v Timeline editoru. Nastaví se první a poslední snímek (Start: 1 a End: 59 – poslední snímek je stejný jako první, při End: 60 by se totožný snímek 1 a 60 přehrál dvakrát, což je nežádoucí – obr. 58 (1)) a animace se spustí (2). Zastavuje se tlačítkem pauzy, které po spuštění animace nahradí tlačítko pro přehrání. Pokud některý pohyb neodpovídá představě uživatele, lze se vrátit na daný snímek v DopeSheet a data vymazat, případně s označeným snímkem přepsat novým „Keyframe“ ve 3D View.



Obrázek 58 – Lišta Timeline Editor

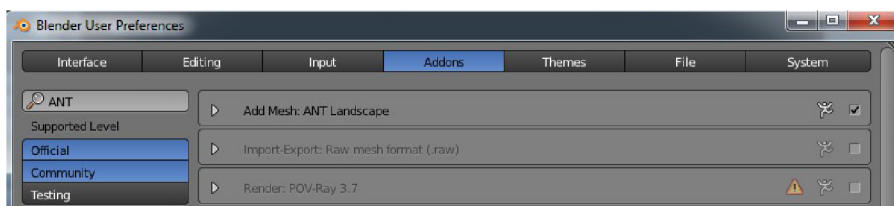
## 5.7 Herní objekty

Ještě před programováním hry a jejím skládáním z logických bloků bude nutné vytvořit další herní objekty. Je nutné vymodelovat nepřátelské jednotky a jelikož se jedná o „Tower Defence“ hru, vytvoří se i vlastní perníková chaloupka, kterou bude hráč bránit. Tvorbou těchto herních aktiv je umístěna v příloze A (Další herní objekty – str. Chyba: zdroj odkazu nenalezen).

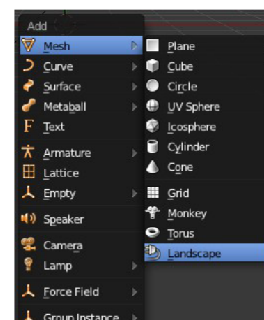
### Terén

Poměrně unikátním modelem hry (co se týče tvorby) je terén, u kterého se využije pluginu pro tvorbu terénu, modelování pomocí sculpting štětců a mapování textury pomocí stencil mapy.

- Aktivuje se plugin ANT Landscape (File→User Preferences...) – obr. 59.
- Přidá se model terénu, který bude sloužit jako základ pro následnou detailnější modelaci („Shift + A“→Add→Mesh→Landscape) – obr. 60. Umístí se do 2. vrstvy („M“).

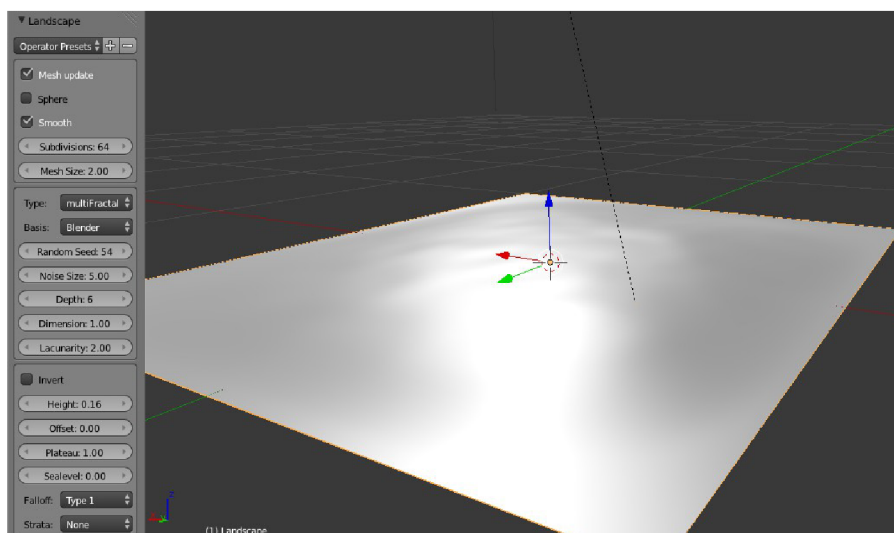


Obrázek 59 – Aktivace ANT Landscape



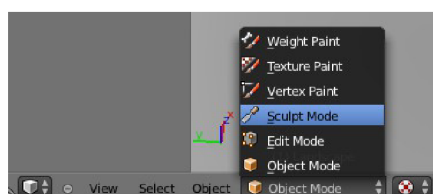
Obrázek 60 – Přidání terénu

- Nastavení pro přidání Landscape mesh jsou na levé liště (Tool Bar) v sekci Landscape – obr. 61. Plugin se využije pro zvrásnění terénu podle využitých nastavení na obrázku 61. Položkou Random Seed lze změnit základ terénu a vybírat si mezi vícero vygenerovanými terény. Zvýšením hodnoty položky Noise Size se roztáhne šum (vyvýšení) terénu a bude tak hladší. Položkou Height se mění výška terénu (pro tento případ se využije nízká hodnota, okolní skály se vytvoří později štětcem).



Obrázek 61 – Nastavení Landscape objektu

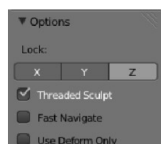
4. Terén má 2 krát 2 Blender jednotky. Pro herní mapu se zvětší 30 krát („S“, „30“).
5. Poté se označený terén přepne do Sculpt Mode – obr. 65.
6. S vybraným Brush štětcem (obr. 64) se terén zvýší v oblastech hrany terénu. Vytvoří se tak skaliska, aby se hráči znemožnilo projít na konec mapy a přepadnout dolů. Pro vysoký terén je vhodné zvýšit hodnotu Strength (kliknutím na hodnotu Strength lze přepsat hodnota nad 1, posuvníkem pouze mezi nulou a jedničkou), což je síla štětce a zvýšit posuvníkem Radius tak, aby štětec pokrýval šířku zamýšleného skaliska. V sekci Options pro Brush je možné zamknout pohyb v x a y ose (viz. obr. 63), aby docházelo při aplikaci štětce pouze ke změnám v ose z (při pohledu shora na Landscape při ortogonálním zobrazení terén zůstává ve tvaru čtvercové matice – viz. Obr. 62).



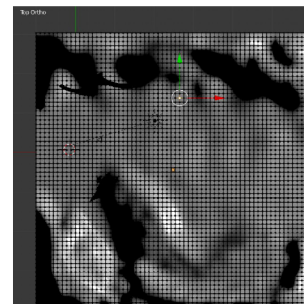
Obrázek 65 – Přepnutí do Sculpt Mode



Obrázek 64 – Štětec Brush



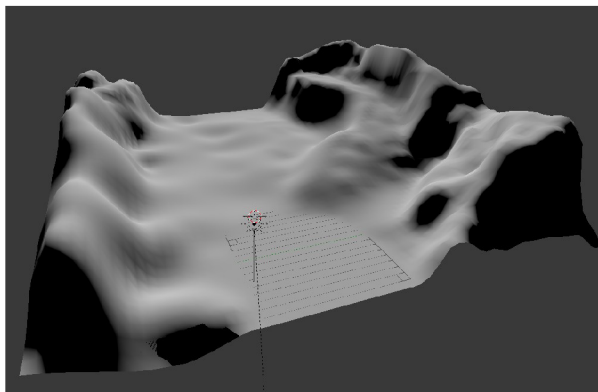
Obrázek 63 – Uzamknutí posunu bodů v X a Y ose



Obrázek 62 – Čtvercová matice

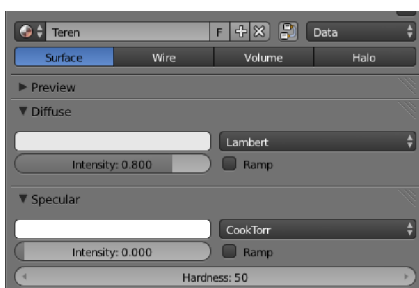
7. Dvě protilehlé strany se nechají nevyvýšené. Zakryjí se později dalšími objekty, ať není vyvýšený terén jedinou překážkou. U jedné ze stěn je možné udělat vyvýšenou plošinu s rampou, terén bude tak zajímavější. Terén se dotvaruje štětcem Smooth (pro vyhlazení) a Brush s nastavením Subtract (pro snižování terénu na rozdíl od přidávání na výšce). Hotový půdorys je na obrázku 66.





Obrázek 66 – Hotový půdorys terénu

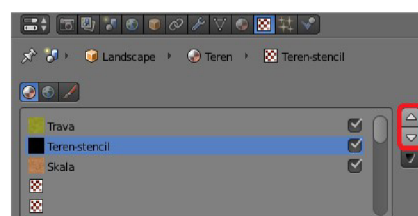
8. Přidá se terénu materiál „Teren“ a nastaví se intenzita Specular na 0 (obr. 68), ať se neleskne ideálně odraženým světlem.
9. V texturách se přidají k materiálu „Teren“ 2 nové textury. Textura trávy a textura skály. Půjde o běžné obrázkové textury (type: Image or Movie, v sekci Image se vybere ze souborového systému obrázků, Mapping→Coordinates: UV). Oba obrázky pro textury musí být „seamless“ – při poskládání vedle sebe nesmí být viditelné spoje. Takovéto obrázky lze nalézt na stránkách <http://cgtextures.com/> po vyhledání slov „grass“ a „rock“ (obr. 67). U textury musí být příznak „Tiled“. Pro hru stačí stáhnout texturu s velikostí „Small“. Některé textury jsou možné k vidění ve 3D náhledu s poskládáním několika dlaždic vedle sebe (3D Tiling Preview).
10. Mezi tyto dvě textury se přidá ještě jedna se jménem „Teren–stencil“. Pozice textur se upravuje šipkami vedle výběru textur (obr. 69).



Obrázek 68 – Nastavení Specular Intensity na 0

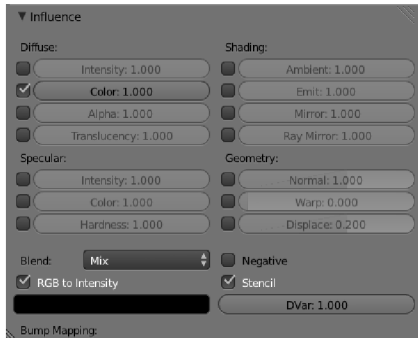


Obrázek 67 – Textura skály

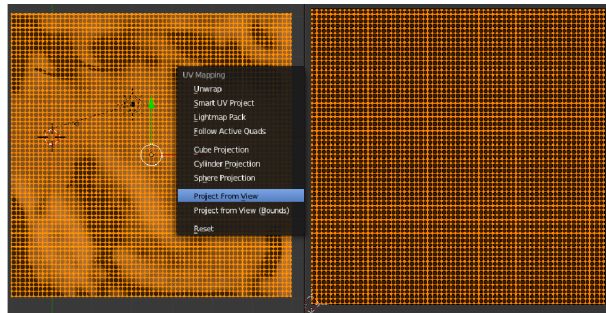


Obrázek 69 – Úprava pozice textury

11. Pro texturu „Teren–stencil“ se vytvoří nový obrázek v UV/Image Editor s názvem „Teren“, který bude mít přednastavené rozlišení 1k na 1k pixelů (půjde o černobílý obrázek, který i při této velikosti nebude zabírat moc paměti). Obrázek se uloží do počítače (v UV/Image Editor: Image→Save As Image, opět název „Teren“).
12. Textura „Teren–stencil“ bude mít nastavení, jako běžná obrázková textura, krom nastavení v sekci Influence, kde se přidají vlastnosti Stencil. Tato nová nastavení jsou vidět na obr. 70.
13. K textuře „Teren–stencil“ je potřeba UV mapa objektu Landscape, která se vytvoří ortogonálním pohledem shora na Landscape (ve 3D View: „U“→UV Mapping→Project From View). Vzniklé mapování se v UV/Image Editor upraví, aby sedělo na celý obrázek (Scale, Grab) – obr. 71.

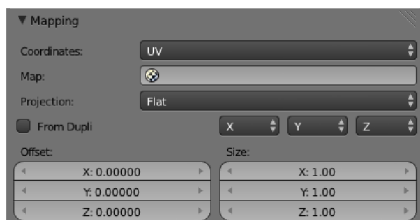


Obrázek 70 – Speciální nastavení pro Stencil

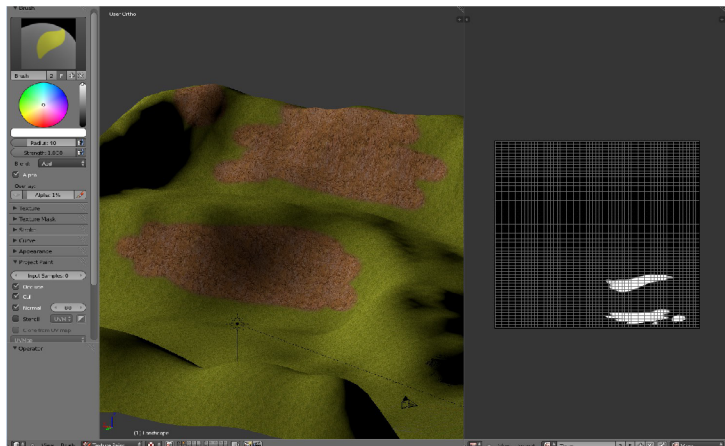


Obrázek 71 – UV mapa

14. V nastavení textury se upraví opakování textur trávy a skály (Mapping→Size→X: 50, Mapping→Size→Y: 50 pro trávu a X: 20, Y: 20 pro skálu). Lze takto nastavit frekvenci opakování textury na objektu pro každou texturu zvlášť (na rozdíl od Scale UV mapy v UV/Image Editor). Posuvníky pro nastavení Size jsou vidět na obr. 72.
15. Terén je prozatím otexturován pouze texturou trávy, která se nachází nad texturou se Stencil nastavením (černý obrázek pro Stencil znamená, že se má na celou plochu terénu aplikovat vrchní textura). Přejde se k aplikování textury, co se nachází pod „Texture–stencil“ (tedy skály). Ve Vertex Paint Mode se vybere štětec Brush. Nastaví se jeho vlastnost Blend na Add a barva štětce na bílou (obrázek 73 vlevo nahoře). Při kreslení na terén ve 3D View se Strength: 1 dojde k obarvování černé Stencil mapy v UV/Image Editor na bílo – obr. 73 vpravo. Bílá v tomto případě představuje aplikování textury pod „Terén–stencil“ texturou (skály). Pro plynulé přechody mezi trávou a skálou se Strength štětce sníží. Dojde tak ke kreslení odstínu šedi do Stencil mapy, který představuje aplikování obou textur (trávy, skály) v daném poměru. Při změně Blend: Subtract dojde k opačným účinkům kreslení. Mapa se začne obarvovat do černa (textura trávy).

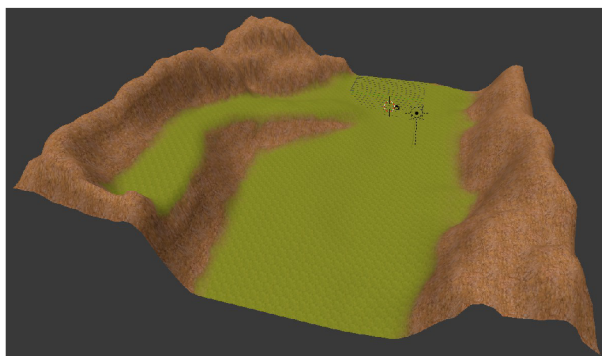


Obrázek 72 – Posuvníky pro Size



Obrázek 73 – Kreslení do Stencil mapy

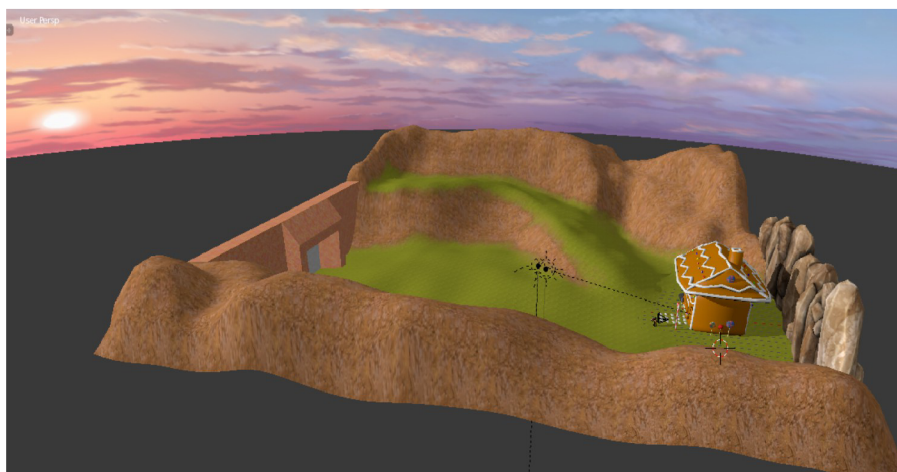
15. Po dokončení kreslení ve Vertex Paint Mode, je nutné nově vytvořenou mapu znovu uložit (UV/Image Editor: Image→Save Image). Dokončený terén bude vypadat následovně (obr. 74). Blender poskytuje touto technikou nanést pouze dvě textury – více informací o alternativách pro více textur je v příloze A (Alternativy texturování terénu – str. Chyba: zdroj odkazu nenalezen).



Obrázek 74 – Dokončený terén

Herní mapa je téměř hotova, ale chybí k ní ještě pár úprav. Ohraničení ve zbývajících částech terénu a obloha.

1. K terénu se přidá pro ohraničení herní mapy skála ze strany chaloupky. Skála byla na začátku případové studie stažena z Blend Swap.
2. Z druhé strany (odkud budou chodit nepřátelé) se přidá brána. Využijí se běžné modelovací techniky uvedené dříve.
3. Nad terén se přidá Skydome (kopule s texturou nebe). Lze nalézt např. na Blend Swap stránkách nebo vytvořit vlastní. Konec kopule nesmí být vidět z jakékoliv hráčovy pozice. Přidané objekty jsou na obr. 75.

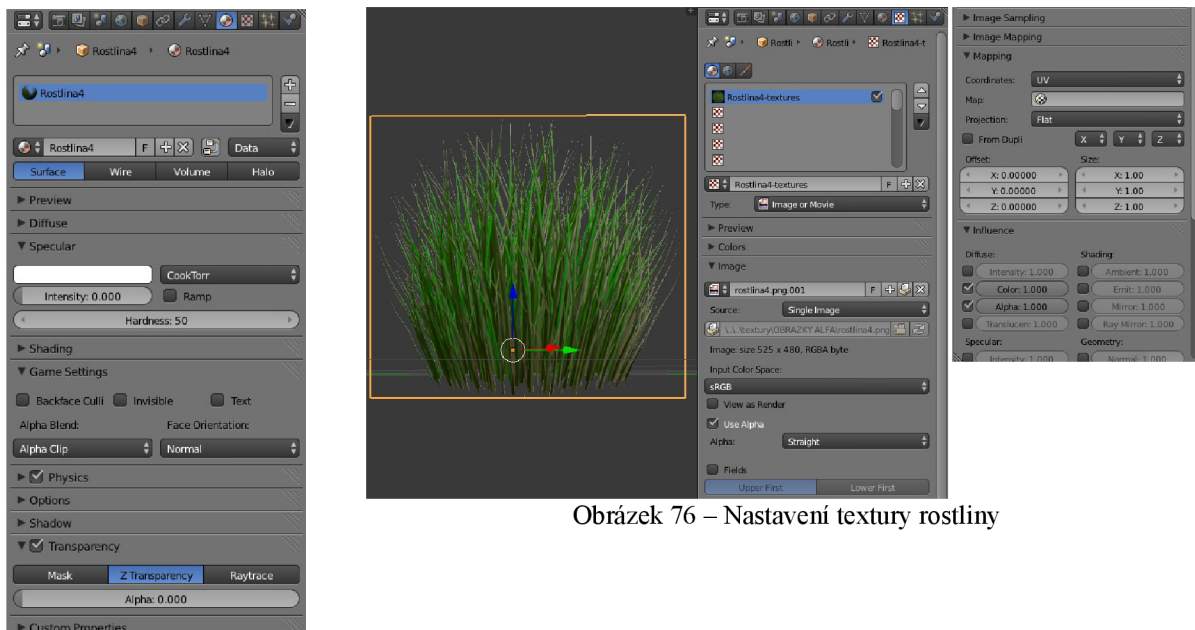


Obrázek 75 – Přidané zátarasy a Skydome

## Rostliny

Jako dekorace se pro terén využijí rostliny a stromy. Započne se s rostlinami.

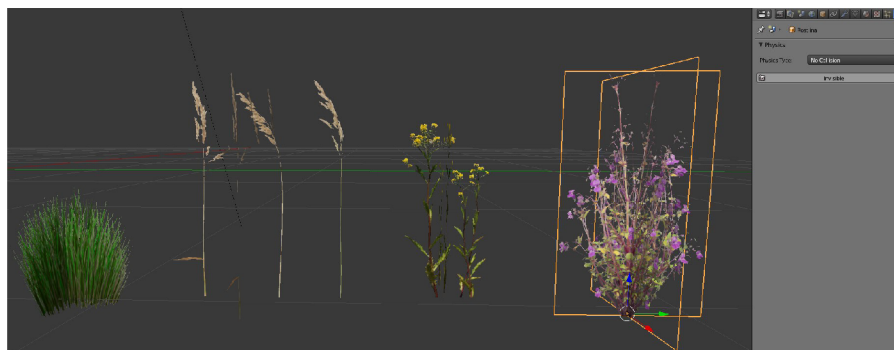
1. Vytvoří se nová plocha (Add→Mesh→Plane).
2. Přidá se k ní nový materiál podle nastavení na obr. 77.
3. K materiálu se přidá textura obrázku rostliny s alfa kanálem. Nastavení je na obrázku 76 (běžná obrázková textura roztažená UV mapou roztaženou po celé ploše + přidáný parametr Influence→Alpha: 1). Výsledná plocha s obrázkem rostliny je na obrázku 76 vlevo.



Obrázek 76 – Nastavení textury rostliny

Obrázek 77 – Nastavení materiálu rostliny

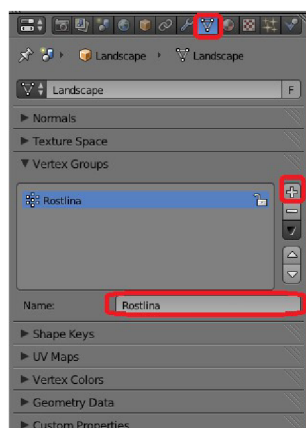
4. Objekt plochy se v Edit Mode zdublikuje, aplikuje se rotace podle osy z o 90 stupňů a obě plochy se spojí do jediného objektu. Vznikne tak nízko-polygonální a přitom detailní rostlina ze 2 billboardů – obr. 78.
5. Physics Type musí být No Collision, aby bylo možné přes rostliny procházet (vpravo na obr. 78).
6. Pro rozmanitost flóry se vytvoří několik druhů rostlin (výsledek viz. obr. 78).



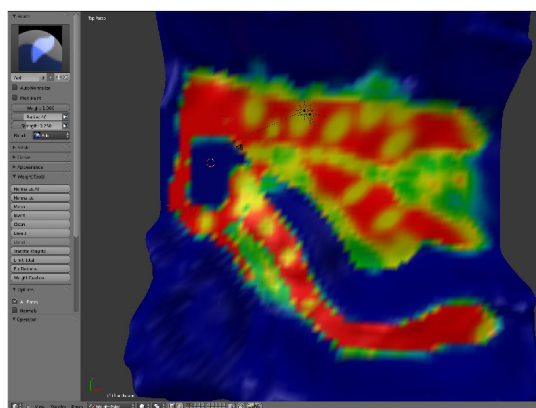
Obrázek 78 – Vytvořené Rostliny s No Collision fyzikou

7. Přidání rostlin do terénu se provede pomocí Weight Paint (pro určení pozic rostlin) a částicového systému. Pro objekt Landscape se vytvoří nová skupina vrcholů (Objekt Data→Vertex Groups→tlačítko „+“) s názvem „Rostlina“ – obr. 79.
8. Přejde se do Weight Paint a Solid Mode.
9. Ve 3D View se do terénu vyznačí oblasti, kde se květiny budou vyskytovat. Barva více dočervena bude znamenat větší hustotu výskytu rostlin. Vyznačené oblasti v terénu jsou viditelné na obr. 80.





Obrázek 79 – Vertex Group pro rostliny

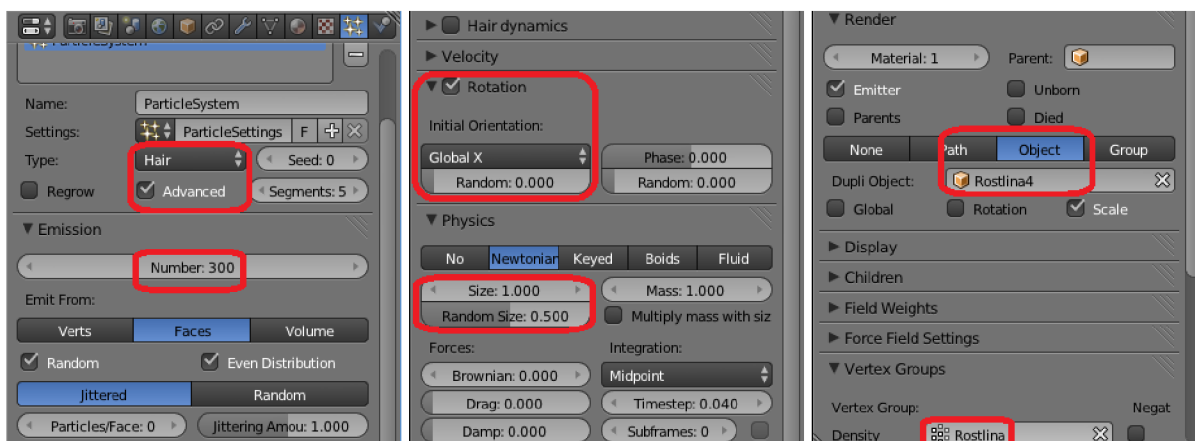


Obrázek 80 – Označený terén pro rostliny

10. Nastavený renderovací engine musí být stále Blender Render (herní engine neumožňuje práci s částicovým systémem). V Properties→Particle System se přidá nový systém.

11. Nový Particle System se nastaví podle obr. 81.

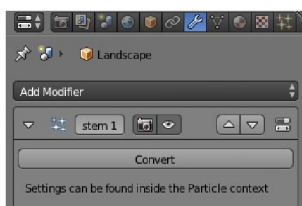
- Typ bude Hair a zaznačí se Advanced možnost, aby se odkryla položka pro rotaci částic.
- V Emission se určí počet částic (zde rostlin) vložených na terén.
- V zaznačené položce Rotation se přenastaví počáteční orientace na „Global X“ (implicitně je rotace rostlin vzhledem k terénu špatná).
- V Physics se upraví velikost částic (Size) a také se vytvoří náhodnost ve velikost rostlin, ať nejsou všechny stejně velké (Random Size).
- V Render se přepne na Object a určí se, který objekt se má použít jako částice.
- Ve Vertex Groups se určí mapa hustoty vrcholů (položka Density).



Obrázek 81 – Nastavení Particle System pro rostliny

12. V Properties→Modifiers se provede konvertování částic na Mesh (tlačítko Convert – obr. 82). Poté lze s rostlinami manipulovat jako s běžnými objekty.

13. Rostliny se přesunou do vlastní vrstvy – obr. 83. Při přílišném zatížení hry lze vrstvu s dekorací (rostlinami) jednoduše vypnout. Na hru pak nemá vliv.
14. Proces se provede pro všechny vkládané rostliny.



Obrázek 82 – Tlačítko Convert u modifikátoru Particle System



Obrázek 83 – Přenesení rostlin do jiné vrstvy

15. K rostlinám se přidají stromy. Buďto se stáhnou z úložiště modelů nebo se vymodelují vlastní pomocí pluginu Sapling. Je podstatné, aby model byl dostatečně jednoduchý pro hru (listy pomocí velkých billboardů, dřevnatá část by neměla být příliš složitá v počtu polygonů).

## 5.8 Blender Game

Hlavní herní objekty jsou vytvořeny a je možné přejít k programování herní logiky. Podkapitola se již nebude odehrávat pod enginem Blender Render, ale pod enginem Blender Game. Pro tuto podkapitolu se předpokládá spolupráce s příloženým souborem „Pernicek TD.blend“, především kvůli Python skriptům hry (jejich umístění je popsáno v závěru kapitoly 3.9). Většina práce bude probíhat pod rozložením obrazovek Game Logic, ve kterém je možné upravovat scénu (3D View Editor), skládat herní logiku z bloků (Logic Editor) i psát herní skripty v Pythonu (Text Editor).

1. Engine Blender Render se přepne do herního enginu Blender Game – obr. 84.
2. Rozložení obrazovek se přepne do Game Logic – obr. 84.

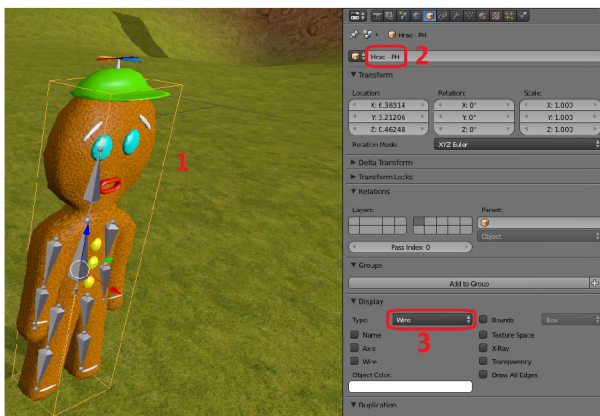


Obrázek 84 – Game Logic a Blender Game

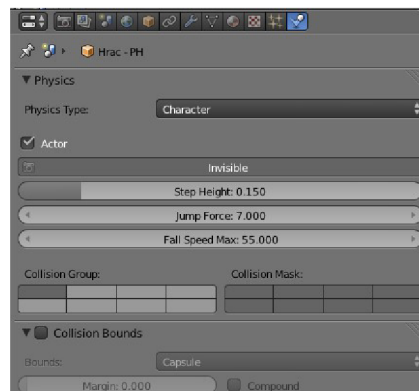
## Fyzika

Hráčem nebude přímo objekt s postavičkou, ale využije se tzv. „placeholder“. Placeholder je objekt, který zastupuje nějaký jiný objekt, který zpravidla bývá strukturálně složitější (jako objekt perníkového panáčka).

1. Okolo hráče se vytvoří kvádr, který objekt bude těsně obklopot (obr. 85 (1)).
2. Název objektu se změní na „Hrac – PH“ (2).
3. Upraví se zobrazení kvádru ve 3D View na drátěný model, bude tak vidět skrz kvádr (Properties→Object→Display→Type: Wire (3)).
4. Armatura postavičky se přiřadí k „Hrac – PH“ jako k rodiči. Při pohybu hráče (což je „Hrac - PH“) se bude stejně tak pohybovat i armatura a k armatuře připojený model postavičky.
5. Přejde se v Properties Editor na Physics a upraví se nastavení fyziky (obr. 86) na hráče (Physics Type: Character, aktivuje se Actor a vypne se vykreslování kvádru ve hře – Invisible).



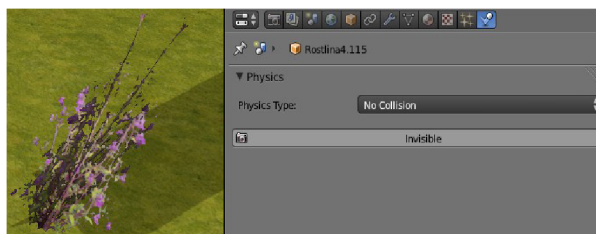
Obrázek 85 – Nastavení objektu hráče



Obrázek 86 – Nastavení fyziky hráče

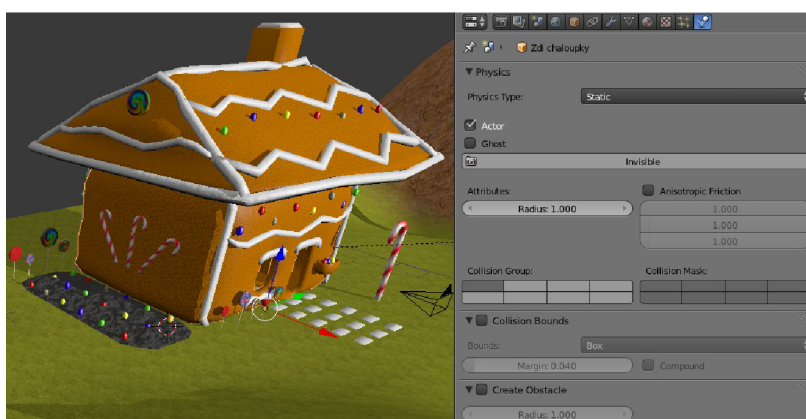
Tímto je nastavena fyzika pro hráče. Ovšem každý objekt ve hře musí mít nějaký typ fyziky.

1. Malé rostliny, listí stromů a podobné objekty, kterými hráč a další objekty (nepřátelé) mohou procházet, se nastaví ve Physics na Physics Type: No Collision – obr. 87.



Obrázek 87 – No Collision

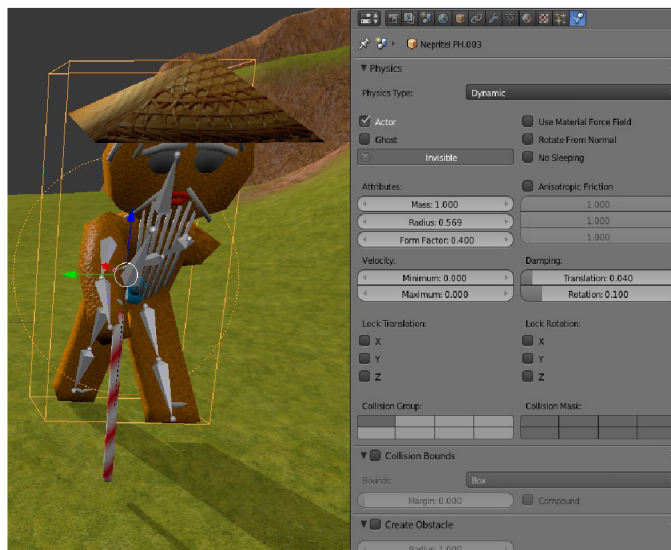
2. Objekty kterými se nesmí projít a tvoří pevné, nepohyblivé překážky (terén, chaloupka, skála, kmen stromu) – Physics Type: Static (obr. 88). U Chaloupky a dalších objektů, které budou hrát roli při detekci senzory (nepřítel musí zjistit, jestli je v blízkosti chaloupky, aby mohl zaútočit), se nastaví příznak Actor.



Obrázek 88 – Static

3. Pro pohyblivé předměty (nepřítel, krabice zasažitelná střelou, ...) – Physic Type: Dynamic (obr. 89).





Obrázek 89 – Dynamic

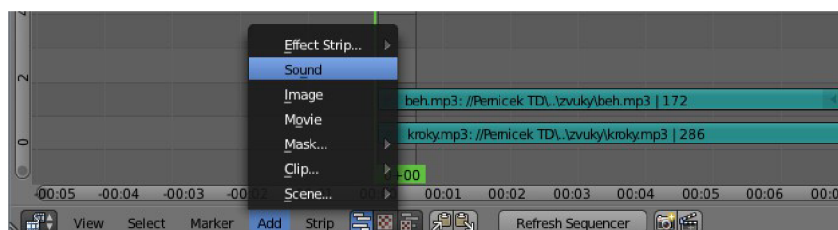
## Zvuk

Hra je doplněna samozřejmě i zvukovou složkou. Její tvorba je na rozdíl od tvorby animace podstatně jednodušší, když není nutné ji tvořit osobně. Zvuky, stejně jako textury, jsou poskytovány zdarma ke stažení na internetu. Na stránkách Freesfx je možné stáhnout audio stopy z poměrně rozsáhlé databáze. Je nutná registrace a následně uvedení této stránky, že z ní bylo čerpáno.

1. Přejde se v internetovém prohlížeči na stránku: <http://www.freesfx.co.uk/sfx/>.
2. Do vyhledávače se zadá hledaný zvuk. Pro hru bude nutné vyhledat zvuk chůze („walk“), běhu, střelby a další zvuky.
3. Vybere se vhodný zvuk z nabídnutého seznamu. U chůze je vhodný delší zvuk, ať se neopakuje příliš často ve smyčce, do které bude vložen. U střelby je naopak vhodná krátká zvuková stopa, jelikož se bude přehrávat celá při každém výstřelu. Také se tím bude šetřit paměť.
4. Zvuk se uloží do složky, kde je umístěn vytvářený soubor se hrou „Pernicek TD“. Po konečném exportování hry do spustitelné verze (postup uveden na konci studie) se tato složka zkopíruje do složky s exportovanou hrou.

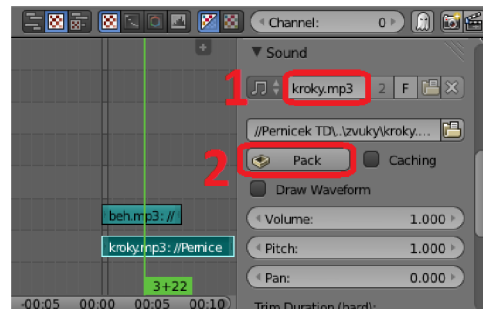
Zvuky již jsou k dispozici pro hru. Teď je nutné je vložit do hry. Pro spuštění ze skriptu se lze odkázat na pozici zvukového souboru přímo. Jelikož ale bude část zvuku vložena pomocí logických bloků, musí se zvuk v Blenderu přilinkovat.

1. Přejde se do rozložení obrazovek Video Editing. Scéna se nastaví „Scene“.
2. Ve Video Sequence Editor se přidají zvukové stopy chůze a běhu (Add→Sound) – obr. 90.



Obrázek 90 - Přidání zvuků do Blenderu

- Zvuky je přejmenují. Na pravé liště Video Sequence Editor je pro označenou zvukovou stopu nabídka vlastností. V sekci Sound se nachází její název – obr. 91 (1). Podle těchto názvů se později zvuky naleznou v Logic Editor.
- Zvuky vkládané tímto způsobem (nepřidávané skriptem) je praktičtější importovat do „\*.blend“ souboru přímo, než je přidávat do vedlejší složky zvlášť. Zaznačí se tlačítko „Pack“ – obr. 91 (2), kterým se zvuk vloží do souboru Blenderu.

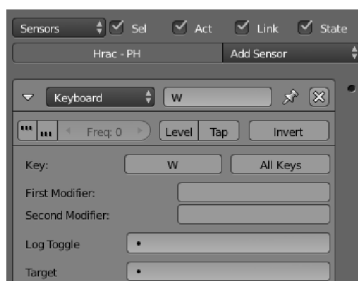


Obrázek 91 - Úprava zvuku

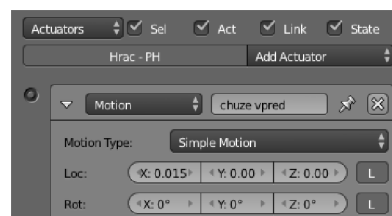
## Pohyb

Započne se s uvedením hráče do pohybu. Tento proces obsahuje spojení kláves na klávesnici s posunem hráče po herní ploše a aktivací patřičných animací pohybu. Také zvuk je zakomponován. Jedná se o základní skladbu herní logiky, kterou je možné uskutečnit i bez znalosti jazyka Python (tedy skládáním logických bloků v editoru logiky). Začne se s chůzí vpřed a postupně se přidají další pohyby, které jsou hráči s postavíčkou umožněny.

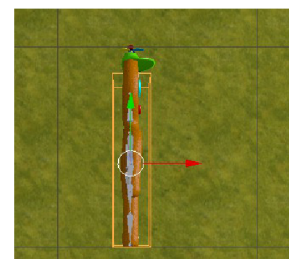
- V Logic Editor se pod sekci Sensor přidá senzor Keyboard (Add Sensor→Keyboard) a nastaví se na kontrolu klávesy „W“ (Key: W). Logické bloky je také užitečné pojmenovávat – obr. 92. Stejným způsobem se přidá i senzor pro kontrolu klávesy „SHIFT“ (Key: SHIFT).
- V sekci Actuator se přidá ke hráči aktualizace pozice (Add Actuator→Motion). Nastaví se posunutí hráče po lokální (zapnuto tlačítko s příznakem „L“) ose x o hodnotu 0.02 (obr. 93). Hodnota záleží na velikosti hráče. Zde hráč měří 1 Blender jednotku – viz. obr. 94. K aktualizaci bude docházet při každém snímku hry (až 60-krát za vteřinu), kdy hráč bude držet klávesu „W“. Proto stačí takto malá hodnota.
- Ve 3D View se vybere k objektu „Hrac – PH“ navíc armatura hráče, čímž se ve stejném logickém editoru zpřístupní oba objekty. Jejich aktuátory potom bude možné spojit ke společnému senzoru.



Obrázek 92 – Logický blok se senzorem klávesy

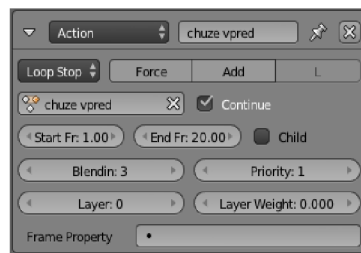


Obrázek 93 – Aktuátor pohybu



Obrázek 94 – Hráč měří 1 jednotku

- V sekci Actuator se přidá pod objekt „Tomik – kostra“ aktuátor Action. Jeho nastavení jsou viditelná na obrázku 95 (animace pohybu byla pro hru modifikována – vytvořena vlastní, pokud se postupovalo přesně podle obrázku se šablonou animace pohybu, bude End Frame: 59). Každá animace by měla mít jedinečnou prioritu, aby animace nekolidovaly (př.: aktivace chůze vlevo a zároveň vpřed). Blender při stejné prioritě (Priority) i vrstvě (Layer) několika animací aktivních naráz reaguje chybně. Animace ve vyšších vrstvách překrývají animace ve vrstvách nižších.
- Ke kostře (nebo i jinému objektu) se připojí zvuková složka (Add Actuator→Sound) – obr. 96. Konkrétní zvuk se vybere ze seznamu zvuků (tlačítko s obrázkem noty). Důležité je nastavit přehrávání zvuku ve smyčce a pouze po dobu držení klávesy (Play Mode: Loop Stop).

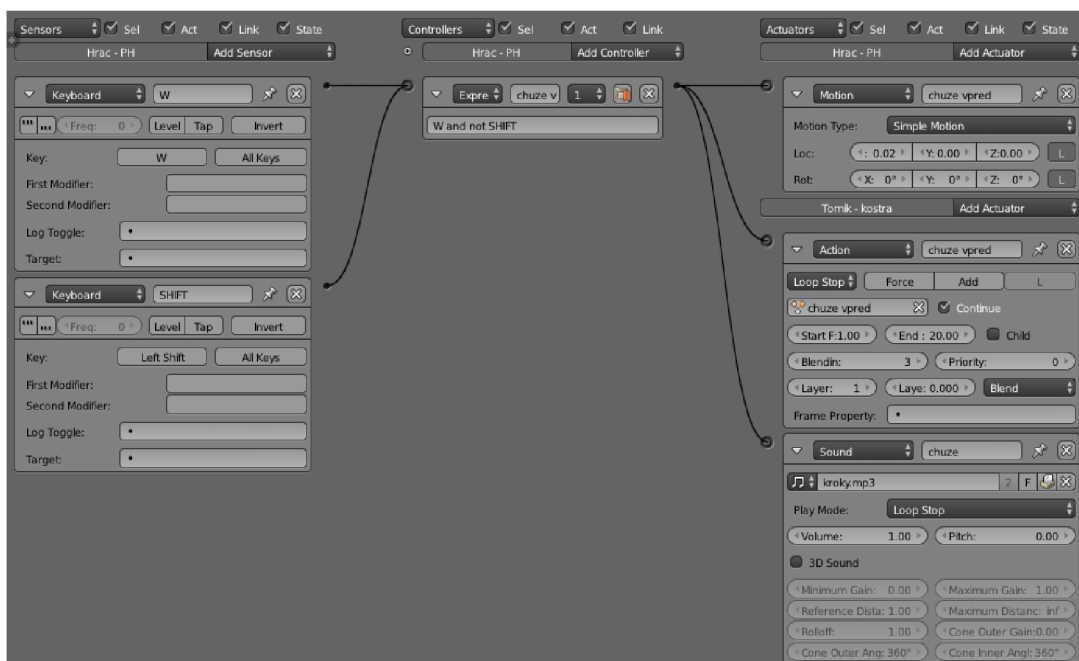


Obrázek 95 – Aktuátor Action pro chůzi vpřed



Obrázek 96 - Zvuk chůze

- V sekci Controller se přidá kontrolér „Expression“, který bude spojovat klávesu „W“ s aktualizací pozice hráče i přehráním animace a zvuku. Do Expression lze vkládat výrazy, kterými lze určit vlastní logické kombinace. Výrazem „W and not SHIFT“, kde „W“ a „SHIFT“ jsou názvy připojených senzorů, se kontrolér nastaví tak, že propustí impuls do připojených aktuátorů jen při aktivním držení klávesy „W“ a deaktivované klávese „SHIFT“. Tento krok je učiněn kvůli skutečnosti, že běh panáčka (implementován později) je odlišen oproti chůzi držením klávesy „SHIFT“. Celek tedy bude mít tvar jako na obr. 97.



Obrázek 97 – Logické bloky pouze pro chůzi vpřed

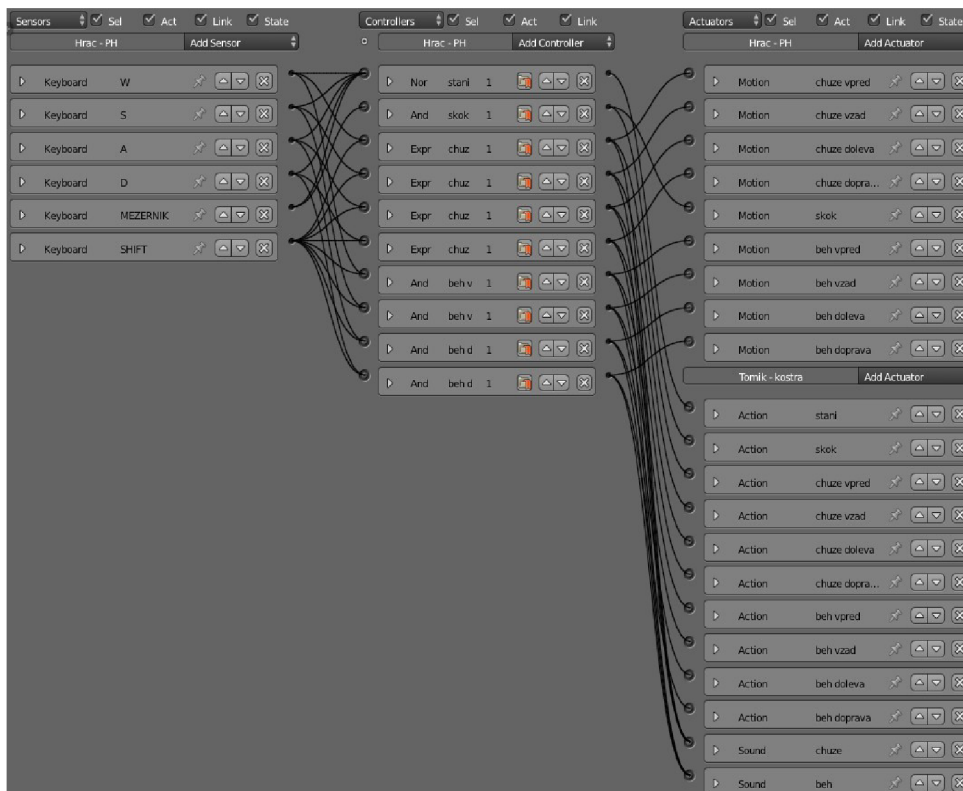
- Přidají se logické bloky (LB) pro ostatní reakce na klávesy (dozadu – klávesa „S“, vlevo „A“, vpravo „D“) ve stejném stylu jako klávesa „W“. I Expression kontroléry budou obdobné, jen s jinou klávesou (namísto „W“). Pro zvuk se využije stejný logický blok z obr. 96.
- Pro běh se klávesy „W“, „S“, „A“, „D“ spojí ve svých vlastních kontrolérech, tentokrát „And“ (všechny senzory musí být aktivní), s klávesou „SHIFT“. Ke kontrolérům se přiřadí nové aktuátory Motion s vyššími hodnotami posunu hráče ve stejných osách, jako tomu je při chůzi. Pro běh se využije nová zvuková stopa s vhodnou nahrávkou běhu.
- Vytvoření skoku je trochu jiné. Ke klávese „SPACEBAR“ se připojí přes „And“ aktuátory podle obr. 98. Nastavení Play u Action znamená, že se animace má přehrát vždy celá.



Obrázek 98 - Aktuátory (Motion a Action) pro skok

- Při nečinnosti hráče se aktivuje také animace, aby hráč nestál strnule. Jeho priorita i vrstva ovšem je minimální (ostatní akce budou tuto animaci překreslovat). Spoje jsou viditelné na obr. 99. Využívají tentokrát aktuátor „Nor“ (všechny senzory musí být deaktivovány), aby se animace spustila.

Všechny pospojované logické bloky (pouze pro pohyb hráče) vypadají následovně (obr. 99).

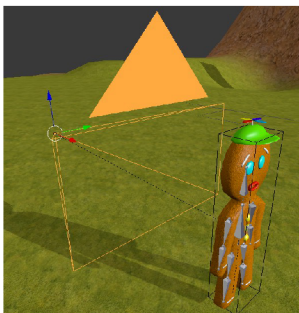


Obrázek 99 – Všechny spojené LB pro pohyb

## Kamera

Při pohybu hráče se s ním musí pohybovat i kamera. V této hře bude kamera za zády hráče (tzv. „Third-person shooter“ – střílečka z pohledu třetí osoby).

1. Kamera se umístí za hráče tak, aby byl vidět hráč i prostředí před ním – obr. 100. Kamera se přejmenuje na „Kamera“.
2. Kamera se připojí k hráči, což je objekt „Hrac – PH“ (Parent To).
3. Volitelně je možné do scény přidat další kamery („Shift + A“ Add→Camera) a připojit k hráči. Jejich aktivita se bude přepínat numerickými klávesami (vytvořený herní skript „Kamera.py“ aktivovaný z „Hrac.py“).



Obrázek 100 – Kamera

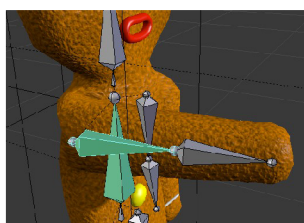
## Pohyb myši

Při tomto stylu hry, bude nutné zprovoznit i ovládání/míření myši. Přímá podpora v Blenderu pro tento účel není. Proto vznikla funkce „rotace\_hrace“ ve skriptu „Mys.py“, která je volána při pohybu myši ze skriptu „Hrac.py“. Obstarává rotaci hráče i k němu připojené kamery „Kamera“ podle směru pohybu myši.

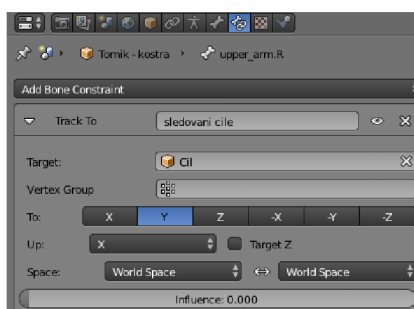
## Míření

Míření zbraní směrem na cíl se zajistí omezením („Constraint“) kosti armatury. Při pravém tlačítku myši hráč zamíří, jinak zbraň sklopí. Při následujícím procesu hrají podstatnou roli ostatní animace, které přepíší napřimenou ruku k cíli při míření do uvolněné pozice, když hráč přestane mířit. Žádná z animací při míření nesmí ovlivňovat předloktí pravé ruky (kost „forearm.R“), předloktí musí při míření zůstat napřimené spolu s paží.

1. V Pose Mode se ke kosti „upper\_arm.R“ (pravá paže, obr. 101) přidá v Properties→Bone Constraints omezení Track To (Add Bone Constraint→Tracking→Track To). Nastavený Constraint je na obr. 102.



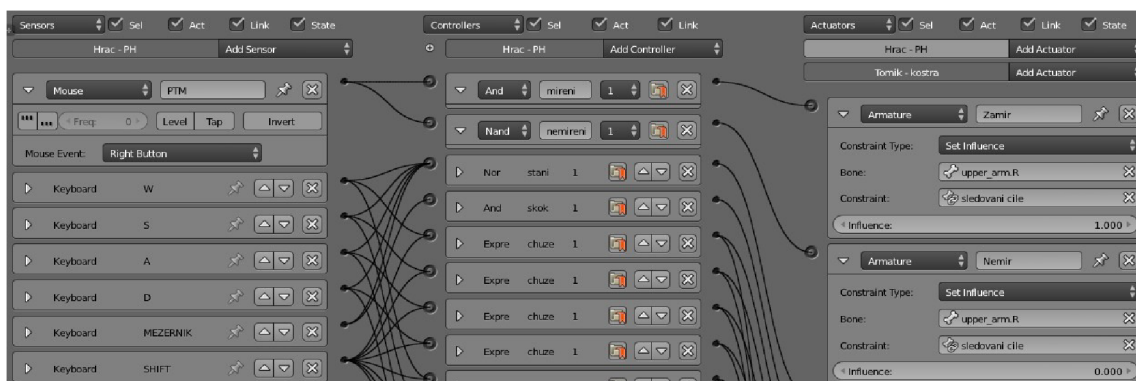
Obrázek 101 – Track To pro ruku i torzo



Obrázek 102 – Track To pro ruku



- Pro kost torza „chest“ (obr. 101) se přidá stejné omezení s nastavením podle obr. 104.
- V základním stavu (při nemíření) bude vliv omezení paže vypnut (Influence: 0). Při míření bude omezení zapnuto (Influence: 1). Toho lze dosáhnout v Logic Editor (s označenou kostí) podle obr. 103. Sledování kosti „chest“ zůstane neustále aktivní, žádné úpravy tedy nejsou nutné. Hráč se tak neustále naklání k cíli.



Obrázek 103 – Nastavení vlivu míření na kost

- Při míření je možné animací pozice kamery kameru posunout před zbraň hráče. Při nemíření se kamera vrátí do původní pozice za hráčem. Toho se docílí přiřazením aktuátoru Action s animací posunu kamery (obr. 105) ke kontroléru s názvem „míření“ (na obr. 103). Typ akce Flipper zajistí návrat kamery za hráče (reverzi animace) při puštění pravého tlačítka myši.



Obrázek 104 – Track To kosti „chest“



Obrázek 105 – Animace kamery při míření

## Zbraně

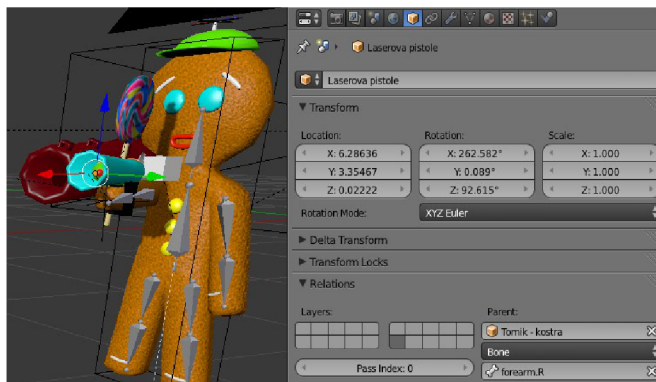
Následující text bude pojednávat o přidání zbraní do hry. Již je možné natáčet do stran kameru i hráče. Zbývá přidat do scény cíl a zbraň, kterými hráč na cíl bude mířit. Také při zamíření se kamera posune před zbraň a při výběru zbraně se zbraň musí přepnout. Musí existovat proměnné pro munici, časové prodlevy při nabíjení zbraně a střelbě, zvukové efekty a další. Vytvoření zbraňového systému se může jevit jako nikterak složité, ale jde o rozsáhlou část vývoje hry. Mnoho těchto procesů je vytvořeno skripty „Munice.py“ a „Zbrane.py“ ve výsledné hře.

- Vymodelují se nové objekty se zbraněmi. Pivot objektu bude umístěn před hlavní zbraň (viz. obr. 106).
- Zbraně se přiřadí k hráčově ruce (Properties→Object→Relations→Parent – nastavení viz. obr. 107 vpravo) a poté přesunou tak, aby je hráč držel v ruce (viz. obr. 107 vlevo).





Obrázek 106 – Modely zbraní



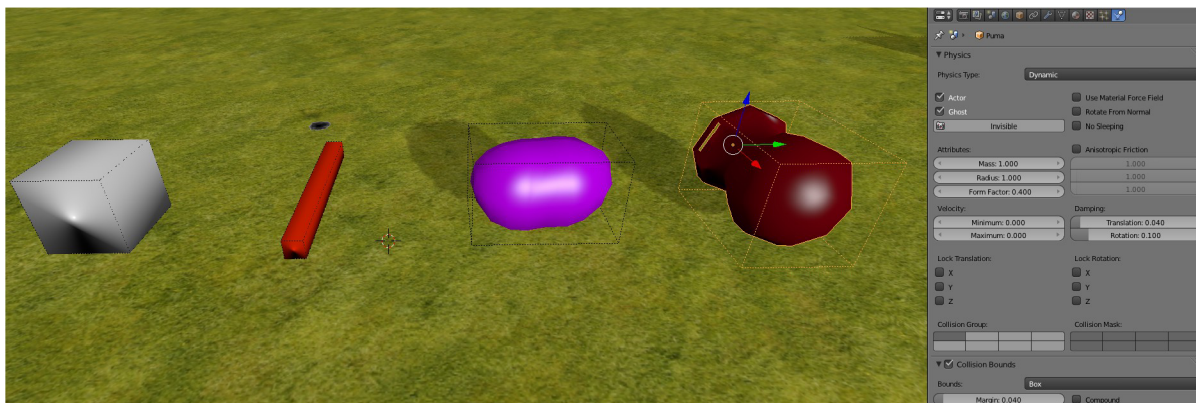
Obrázek 107 – Držení zbraně (vlevo) a nastavení Parent

3. Zbraně se nataví v Properties→Physics jako No Collision.
4. Reakce na přepínání zbraní (číslice, kolečko myši) jsou kontrolovány ve skriptu „Hrac.py“. Změna zbraní, stejně tak zbraňový systém, je vytvořen skriptem „Zbrane.py“.

## Munice

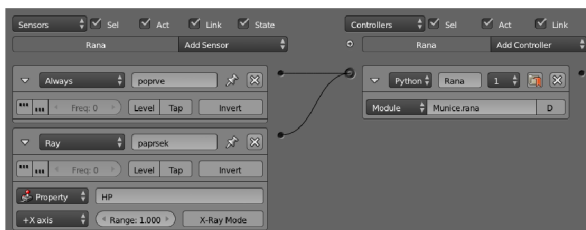
Munice bude mít odlišný systém. Náboje se generují ze zbraňového systému („Zbrane.py“) při výstřelu (detekce levého tlačítka myši v „Hrac.py“). Při každém výstřelu se vytvoří nový objekt/projektíl, který musí být umístěn ve skryté vrstvě scény (neoznačená vrstva). Chování pro každý druh projektilu je popsáno ve skriptu „Munice.py“.

1. Vytvoří se projektily pro každou zbraň (rána lízátkem, laser, fazole, puma – na obrázku 108 odleva doprava). Projektily rána lízátkem a laser nejsou nezbytně nutné (fungují na principu okamžité akce – vyslání pomocného paprsku před objekt, zda netrefí zničitelný objekt, zrušení objektu – vše v jediném snímku hry). Aby ovšem veškerý útok fungoval na stejném principu a byl jednotně popsán v jediném skriptu „Munice.py“, budou i pro tyto útoky vytvořeny projektily. Při rozsáhlejšímu projektu je přehledná struktura programu velmi cenná.

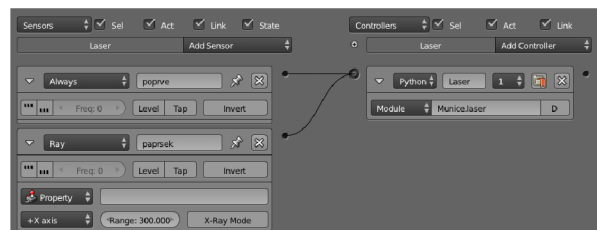


Obrázek 108 – Vytvořené projektily s nastavenou fyzikou

2. Fyzika (Properties→Physics) projektilů bude mít nastavení podle obrázku 108 (vpravo). Podstatné je Physics Type: Dynamic, aby projektil mohl být uveden do pohybu. Laser i rána lízátkem budou mít ve Physics nastavenou vlastnost Invisible, aby ve hře nebyly vykreslovány.
3. Zbraň v podobě lízátka slouží jako zbraň zblízka. Pro ni (a případně další zbraně zblízka) je vytvořen před hráčem objekt (krychle) s Physics: No Collision a Invisible. Na pozici tohoto objektu budou přidávány rány (objekty) lízátkem. Vyhne se takto problémům s pozicí vytvoření rány, která by na takto pohyblivé zbrani dělala problémy.
4. Sensory detekující vzdálenost, kolize a další nelze vytvořit v Blenderu pouhým Python skriptem. Musí být vytvořeny v logických blocích a ke skriptu navázány. Ve skriptu lze k navázaným sensorům přistoupit pro zjištění výsledků. Rána lízátkem má následující nastavení v Logic Editor (obr. 109).
5. Laser má sestavení v Logic Editor podle obr. 110.

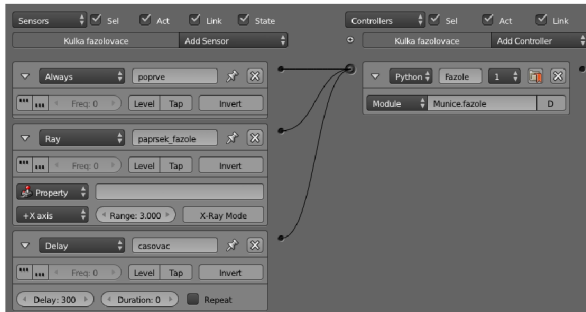


Obrázek 109 – LB rány lízátkem

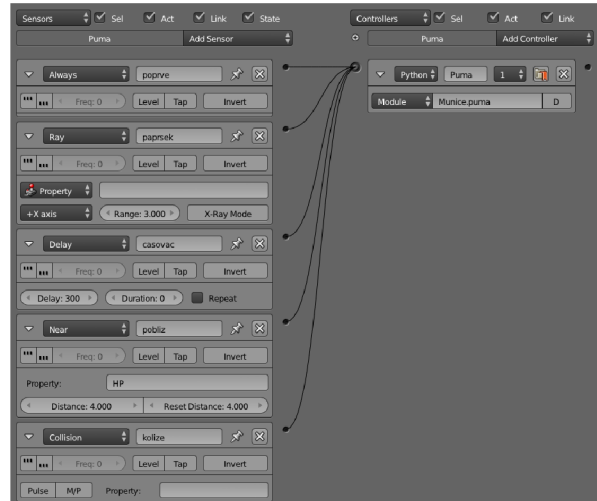


Obrázek 110 – LB laseru

6. Fazole má sestavení v Logic Editor podle obr. 112.
7. Puma má sestavení v Logic Editor podle obr. 111.



Obrázek 112 – LB projektilu fazole



Obrázek 111 – LB pumy

## Head Up Display

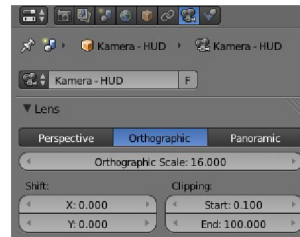
Ve skriptech je sice definován počet celkové munice a munice v zásobníku, ale hráč o tomto množství zatím nemá přehled. Neví ani kolik mu zbývá ještě životů, jaké má skóre apod. Aby hráč měl přehled o dění hry, vytvoří se HUD (head-up display), který bude tvořen překrývající scénou. Tato scéna bude poskytovat hráči veškeré potřebné informace o hře. Výhodou vytvoření nové scény,

kteřá překrývá herní scénu („Scene“), je vlastnost textu (i text je objekt scény) a dalším objektů v ní, že budou vždy v popředí. Při vytvoření objektů v „Scene“, by mohly být schovány za okolní objekty).

1. Vytvoří se nová scéna (v Info Editor: Scene→tlačítko „+“→New) s názvem „HUD“ (obr. 114) a přepne se do Blender Game. Stínování se přepne do GLSL z přednastaveného Multitextured.
2. Do scény se přidá kamera a nastaví se do Orthographic módu (viz. obrázek 113). Orthographic Scale: 16 zabere celou pomocnou mřížku ve scéně (16 Blender jednotek velká).



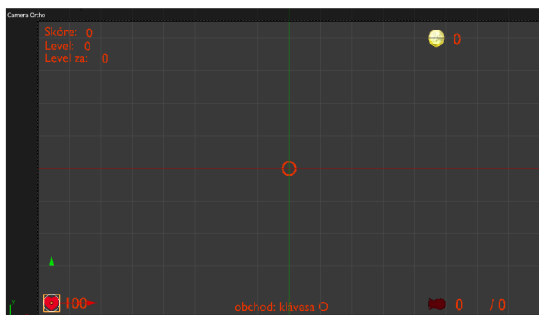
Obrázek 114 – Vytvoření scény HUD



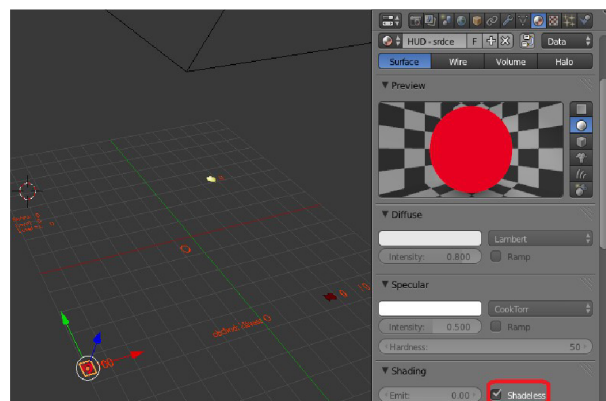
Obrázek 113 – Nastavení kamery HUD

3. Přidají se textové objekty („Shift + A“→Add→Text), které se umístí před kameru s patřičnou velikostí (Scale) na vhodných pozicích – obr. 115. Munice do dolního pravého rohu, životy do dolního levého rohu, skóre (a další podobné údaje) do horního levého rohu, peníze do horního pravého rohu. Je nutné udělat místo pro přibývající cifry.
4. Pro rozdělení počtu projektilů v zásobníku a celkové munice se využije nový textový objekt nesoucí text „/“ – obr. 115. Text se zapisuje do textového objektu v Edit Mode.
5. Před texty se umístí obrázky vyjadřující význam daného textu (průhledné plochy otexturované obrázkem s alfa kanálem) – obr. 115. Samotný obrázek lze v Blenderu vytvořit jako 3D model, který se „zapeče“ do textury. Postup je prakticky stejný jako v podkapitole o „zapékání“ procedurální textury, jen namísto plochy s materiálem se bude pracovat s modelem.
6. Do středu scény, přesně do středu záběru kamery, se umístí zaměřovač – obr. 115.
7. Všechny objekty ve scéně budou mít nastaven v materiálu příznak Shadeless (viz. obr. 116). Dává se tak Blenderu najevo, aby daný materiál nestínoval. Jelikož ve scéně HUD nebude osvětlení, materiály by byly vykresleny černě.

Výsledné rozvržení objektů scény z pohledu kamery je viditelné na obr. 115 i 116.



Obrázek 115 – Rozmístění objektů ve scéně „HUD“

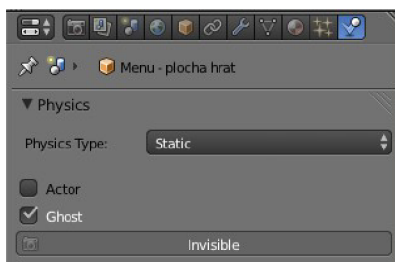


Obrázek 116 – Nastavení materiálu pro ikonky

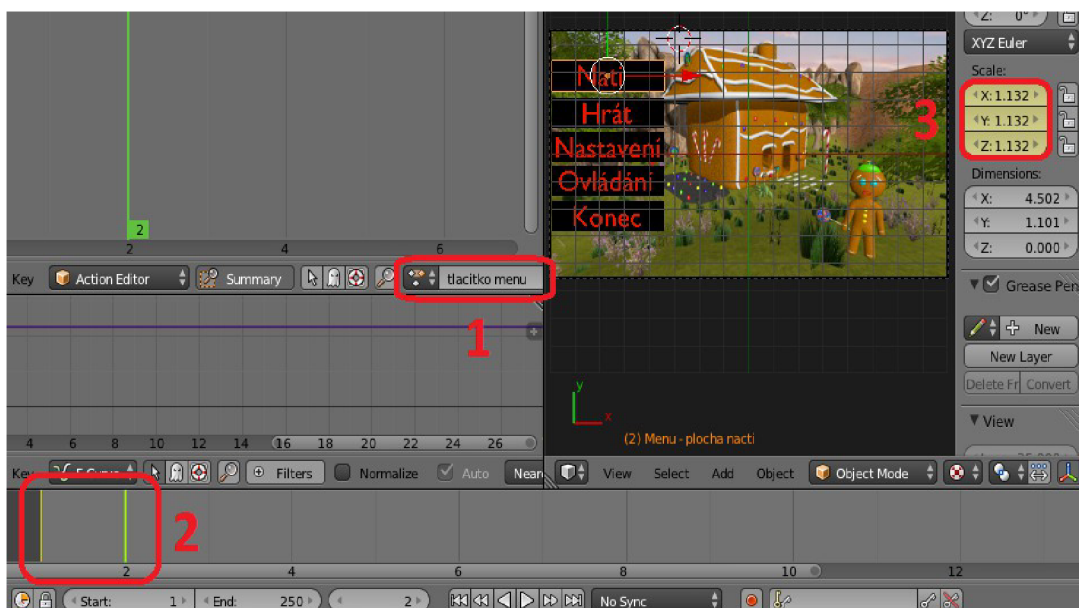
## Menu

Při spuštění hry se zobrazí herní menu, kde si hráč může vybrat, jestli chce začít novou hru, načíst dříve uloženou hru, změnit nastavení hry, podívat se na ovládání a nebo hru ukončit. Menu využívá tlačítka s předdefinovanou akcí a myš s viditelným kurzorem. Herní menu je prakticky interaktivní aplikace, které je možné v Blenderu také (krom her) tvořit.

1. Vytvoří se nová scéna „Menu“. Nastavení scény a vytvoření kamery je stejné jako v „HUD“ scéně.
2. Vytvoří se nová plocha, která se roztáhne tak, aby pokrývala celé zorné pole kamery. Otexturuje se obrázkem, který se pořídí z naaranžované scény hry. Vznikne tak pozadí pro scénu „Menu“ (obr. 118).
3. Nad plochu s obrázkem se přidá textový objekt a v Edit Mode se přepíše na text tlačítka.
4. Mezi text a pozadí se přidá obdélníková plocha (tlačítko), která bude podkladem pro text tlačítka. Fyzika tlačítka se nastaví tak, aby byla statická (takto Blender pozná kliknutí myši), neviditelná a neinteragovala s okolím (obr. 117).
5. Vytvoří se pro tlačítko animace, která zvětší tlačítko, když nad ním bude kurzor myši. Postup je stejný, jako u animací pohybu postavíčky. Vytvoří se nová animace (obr. 118 (1)), kde ve Frame: 1 (obr. 118 (2)) bude Scale aktuální velikosti. Velikost se zaznačí do snímku (ve 3D View: „I“→Insert Keyframe Menu→Scaling). Ve Frame: 2 se lehce zvětší velikost tlačítka (obr. 118 (3)) a opět se aplikuje Scale do snímku.



Obrázek 117 – Nastavení fyziky tlačítka



Obrázek 118 – Animace tlačítka při najetí myši nad něj



- Po Parent To textu k tlačítku se bude transformovat i text.
- Aby animace zvětšení tlačítka fungovala při kurzoru myši nad tlačítkem, v Logic Editor se vytvoří spoje podle obrázku 119. V pospojovaných logických blocích se vyskytuje také akce tlačítka při kliknutí levého tlačítka myši (senzor „LMB“) nad tlačítkem („MYS NAD“). Akce jsou vytvořeny již skriptem „Menu.py“. Do LB se vkládají jako moduly, ve kterých se přistupuje na funkci danou názvem tlačítka („nacti“, „hrat“, „nastaveni“, „ovladani“) – po vzoru z obr. 119.



Obrázek 119 – LB tlačítka „Načti“

- Pro ukončení hry se logické bloky vytvoří velmi jednoduše podle obrázku 120. Místo skriptu s několika potřebnými příkazy se využije aktuátor.



Obrázek 120 - LB tlačítka „Konec“

- Tímto je scéna herního menu hotová. Při spuštění hry vypadá následovně (obr. 121).

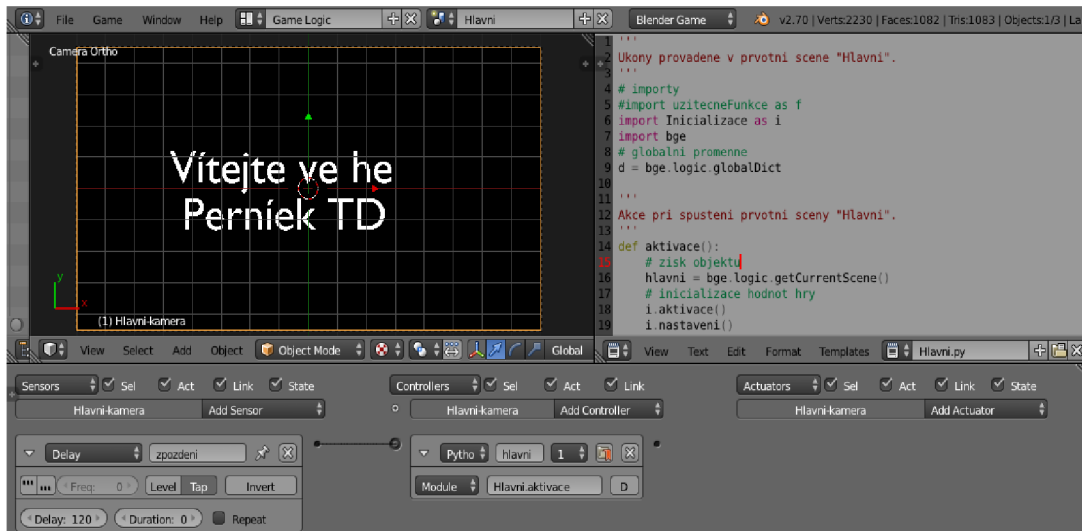


Obrázek 121 – Hotové menu při spuštění hry

## Úvodní scéna

Jako úvodní scéna hry se využije nová scéna „Hlavní“. Postup vytvoření scény je obdobný tvorbě „HUD“ (Blender Game, GLSL, pojmenování scény, ortogonální kamera). „Hlavní“ obsahuje černou

plochu na pozadí (Material Tab – Diffuse: černá, nastaven příznak Shadeless) a text „Vítejte ve hře Perníček TD“. Při spuštění hry bude tato scéna aktivní jako první a po určité době, zde 120 logických tiků (dvě sekundy), se přejde na scénu „Menu“. Logické bloky se pospojují podle obr. 122. Využívají funkci ze skriptu „Hlavni.py“. Tyto bloky lze přiřadit libovolnému trvalému objektu scény „Hlavni“ (např. kameře).

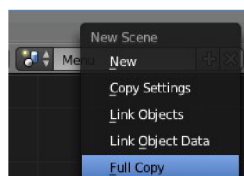


Obrázek 122 – Nastavení scény „Hlavni“

## Druhé herní menu

Vytvoří se ještě jedno herní menu, které se zobrazí při stisknutí klávesy „Esc“ při spuštění hry (aktivní scéna „Scene“). Pokud by se přešlo do základního menu (výše vytvořenému), celý postup hry by zanikl spolu se scénou hry „Scene“, kterou by nahradila scéna „Menu“. Proto se vytvoří druhé menu, které se bude aktivovat ve hře na způsob Overlay Scene. Hra („Scene“) zůstane aktivní (pouze se pozastaví) a překreslí se novým menu, ze kterého lze odejít zpět do spuštěné hry (hra se odpauzuje). Druhé menu má také jinou stavbu tlačítek.

1. Vytvoří se nová scéna pomocí kompletní kopie prvotního menu (u nabídky scén tlačítko „+“ → New Scene → Full Copy) – obr. 124.
2. Scéna se přejmenuje na „Menu\_hry“.
3. Zmenší se plocha pozadí – obr. 123. Při aktivaci této scény zůstane na pozadí vidět pozastavená hra.



Obrázek 124 – Menu New Scene



Obrázek 123 – Vytvořená scéna s druhým menu hry



- Upraví se text tlačítek. Blender ve 3D View sice nezobrazuje diakritiku korektně (chybí „č“ ve slově „Zpět“ a další), ale při spuštěné hře se vykreslí správně. Také se přidají co scény tlačítka nová. Tlačítko „Zpět“ pro návrat z menu do hry a „Ulož“ pro uložení aktuálního dění ve hře. Scéna po tomto kroku vypadá dle obr. 123.
- Logické bloky se vytvoří jako při prvním menu, ale tentokrát i tlačítko „Konec“ má vlastní funkci ve skriptu (obr. 125). Pro druhé menu se využívá nový skript – „Menu\_hry.py“. Volané funkce mají stejné názvy, jako jsou názvy tlačítek.

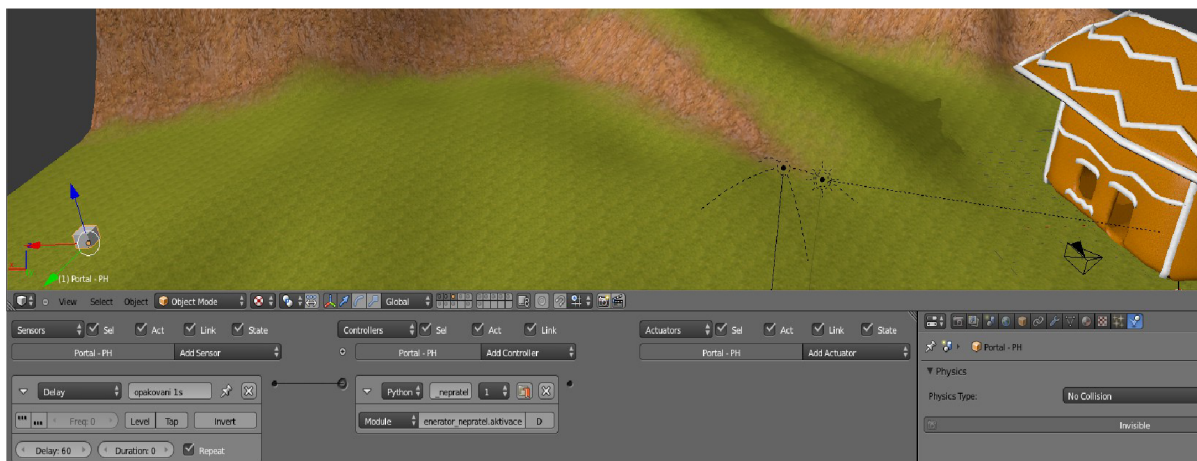


Obrázek 125 – LB tlačítka „Konec“

## Generování nepřátel

Hra je již spustitelná přes herní menu, hráč se může pohybovat ve hře a střílet. Zatím ovšem nemá žádné cíle. Přidají se do hry tedy nepřátelé. Ti budou chodit k chaloupce, kterou po jejím dosažení budou poškozovat až do úplného zničení. Pak hráč prohrál.

- Vytvoří se objekt (krychle) do scény hry, který bude představovat počátek pro generování nepřátel (obr. 126). Krychle se ve Physics nastaví jako No Collision a Invisible (jedná se pouze o pomocný objekt) – obr. 126 vpravo.



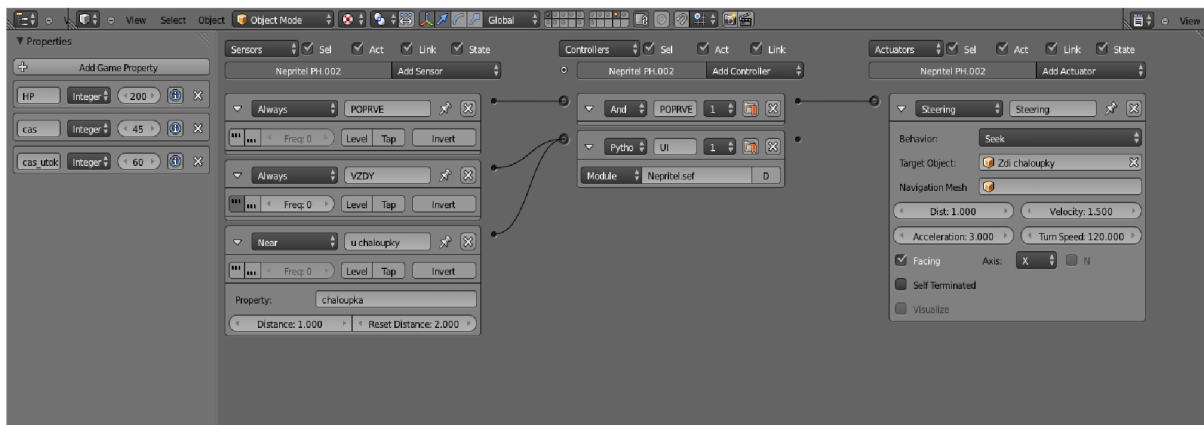
Obrázek 126 – Počátek pro generování nepřátel

- V Logic Editor se vytvoří senzor Delay, který každých (příznak Repeat) 60 logických tiků (při nezpomalené hře 60 tiků odpovídá 1 sekundě) pošle impuls do připojeného kontroléru. Kontrolér má přiřazenu funkci skriptu „Generator\_nepreatel.py“ (obr. 126), která obstarává generování nepřátel v určitých časech od začátku hry.
- Nepřátelské postavičky musí být ve skryté vrstvě scény, jelikož budou skriptem přidávány do hry jako nové objekty a metoda „addObject“ vyžaduje tyto objekty ve skryté vrstvě.

4. Ke každé nepřátelské jednotce se přidají vlastnosti:

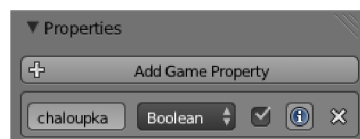
- „HP“ (životy postavičky),
- „cas“ (odpočet do odstranění postavičky při dosažení HP 0 a méně),
- „cas\_utok“ (odpočet do dalšího útoku na chaloupku).

Každá přidaná postava do hry má tyto vlastnosti své vlastní a lze udržovat ve hře hodnoty těchto vlastností pro každou postavu zvlášť. Jejich přidání je viditelné na obr. 127 vlevo. Nastavení hodnoty pro vlastnost „HP“ není důležité. Je nastavována pro postavy při jejich přidání do hry ve skriptu „Nepritel.py“. Hodnoty životů si skript bere ze skriptu „Inicializace.py“, kde jsou všechny nastavitelné hodnoty hry.



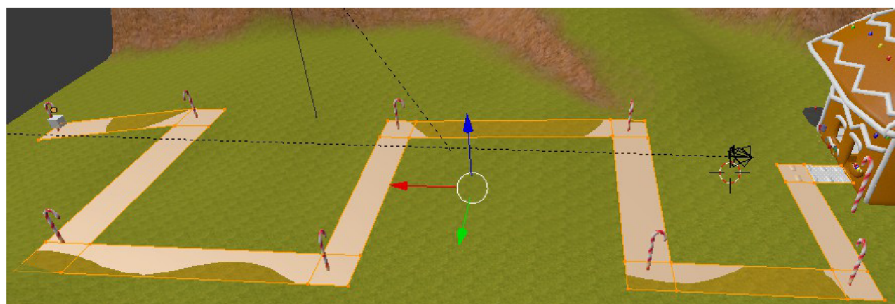
Obrázek 127 – LB pro nepřátelskou postavu „Sef“

5. V Logic Editor se upraví bloky (zde pro postavu „Sef“) podle obrázku 127. Near senzor sepne při dosažení objektu s vlastností „chaloupka“ na vzdálenost danou vlastností senzoru Distance (zde 1 Blender jednotka). V aktuátoru Steering, který se po přidání postavy do hry poprvé aktivuje, se upravuje rychlost pohybu (Velocity) opět dle hodnot ve skriptu „Inicializace.py“. Steering znamená pronásledování objektu daného v Target Objekt – zde chaloupky.
6. K chaloupce je nutné vlastnost „chaloupka“ přidat, aby nepřítel rozpoznal svůj cíl. Postačí typ Boolean – obr. 128.



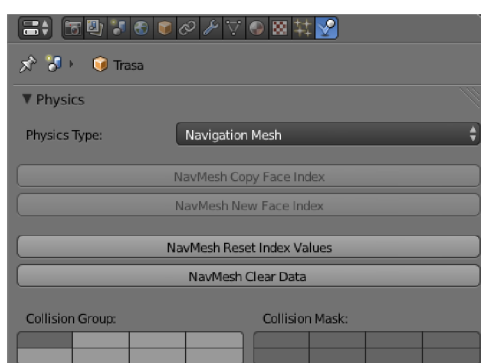
Obrázek 128 – Nastavená vlastnost „chaloupka“

7. Při tomto nastavení Steering aktuátoru nepřítel běží od svého vložení do hry přímo k chaloupce. U Tower Defence her je ovšem spíše zvykem, aby nepřítelé procházeli po určité klikaté trase. Věže pak mají více času k jejich zničení a záleží i na pozici umístění věže. Do hry je možné přidat Navigation Mesh – objekt, podle kterého se budou další objekty navigovat. Navigation Mesh lze upravit přímou trajektorií běhu nepřátel k chaloupce na složitější cestu dle představ návrháře. Do scény se vloží plocha, která se v Edit Mode vytvaruje do tvaru cesty, po které mají nepřítelé běžet. Umístí se těsně k terénu – obr. 129.

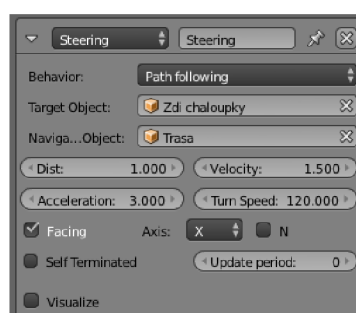


Obrázek 129 – Plocha upravená do Navigation Mesh

8. Fyzika plochy se přenastaví na Navigation Mesh – obr. 130.
9. Steering se upraví na Path following podle obr. 131.

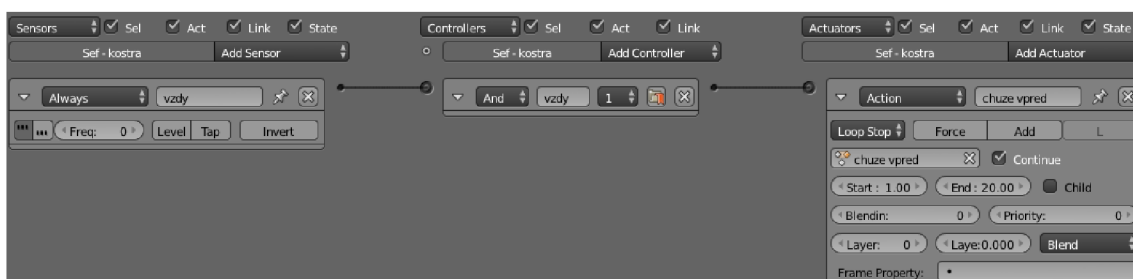


Obrázek 130 – Physics plochy pro Navigation Mesh



Obrázek 131 – Upravený Steering

10. Všechny postavy mají svou vlastní UI (umělou inteligenci). Ta je určena funkcí skriptu „Nepritel.py“, která nese jejich jméno. Skriptem se obstarává kontrola, zda je postava poražena (animace padnutí a ukončení objektu/postavy), útok na chaloupku (další animace a ubírání životů chaloupky), prvotní nastavení hodnot (životy, rychlost pohybu) atd.
11. Ke každé z nepřátelských postav se přidá animace pohybu (LB na obr. 132), která se bude ve smyčce přehrávat od jejich počátečního vložení do hry. Hodnota Layer je 0. Animace bude překreslována jinými animacemi ze skriptu „Nepritel.py“ (padnutí, útok), které mají hodnoty Layer vyšší.

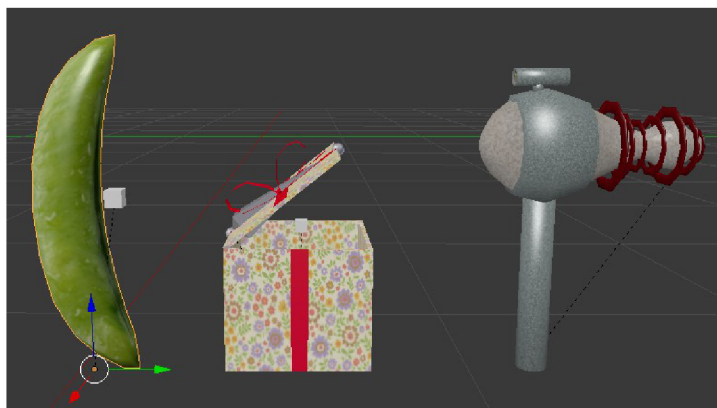


Obrázek 132 – Animace pohybu nepřítele „Sef“

## Věže

Jelikož hra je zaměřena ve stylu Tower Defence, nesmí chybět ani obranné věže, které se postarají o likvidaci nepřátel bez přímé účasti hráče. Hráč na zem pokládá věže koupené přes obchodní terminál, jinak se může připojit do obrany se svou koupenou zbraní jako podpora.

1. Vymodelované věže (fazolová, laserová, pumová), které se nacházejí v neoznačené vrstvě scény, se natočí v Edit Mode tak, aby směřovaly čelní stranou ve směru osy y (dále připojený aktuátor „Track to“ bere čelo objektu ve směru osy y). Fyzika věže se nastaví jako No Collision. Pivoť věži musí být na pozici země, jelikož objekty jsou přidávány k nohám hráče. Vše je vyznačeno na obr. 133.



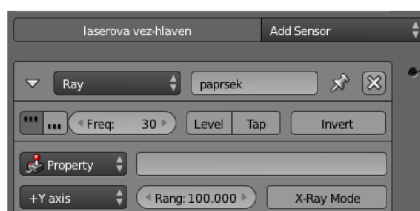
Obrázek 133 – Věže s natočenou čelní stranou i hlavními

2. Ke každé z věží se připojí (Parent To) vlastní objekt (krychle – obr. 133), který bude představovat počáteční pozici výstřelu (hlaveň). Objekt musí mít nastavenou fyziku: Invisible, No Collision.
3. Fazolová a pumová věž se doplní LB podle obr. 134. Využívaným skriptem pro UI věží bude „Vez.py“. V tomto skriptu se po přidání věže do hry upravují i hodnoty Distance, Reset Distance (na jakou vzdálenost věž útočí) a frekvence útoků Freq u Near senzoru. Object u „Track to“ aktuátoru se upravuje dynamicky skriptem (čelně sleduje nepřítele poblíž). Fazolová věž oproti obrázku 134 využije funkci „Vez.fazolovka“.



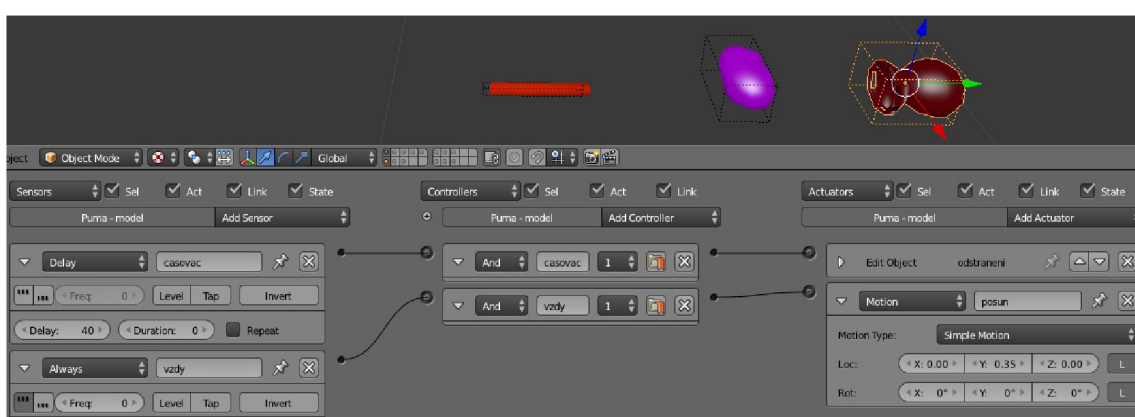
Obrázek 134 – LB pumové věže

4. Laserová věž má navíc na své hlavní připojen senzor paprsku pro vykreslování laserové čáry ve skriptu „Vez.py“ – obr. 135. Senzor je umístěn ke kontroléru s funkcí „Vez.laserovka“ (stejně jako Near a Always senzory).



Obrázek 135 – Ray senzor pro hlaveň laserové věže

- Munice se pro věže vytvoří podobná munici pro zbraně hráče. Musí být ale bez pomocné logiky pro zásah cíle (zásah proběhne po výstřelu vždy, o ten se stará skript „Vez.py“). Navíc (kvůli aktuátoru věží „Track to“) je i munice nastavena čelní stranou ve směru osy y – obr. 136.



Obrázek 136 – Munice věží s LB a nastavení její orientace

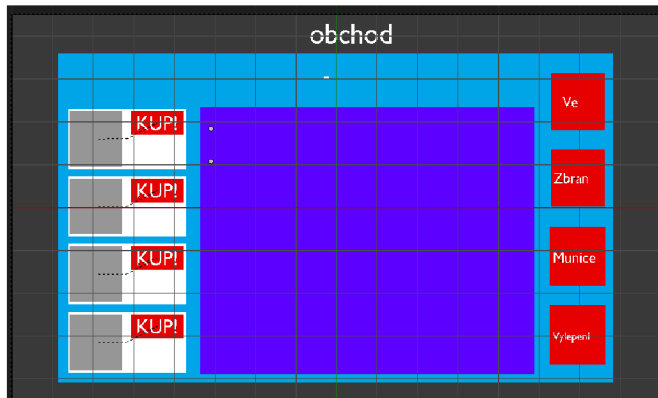
## Obchod

Hráč po zničení nepřítele získá nejen skóre, ale i peníze. Peníze využije k nakoupení věci z obchodu, který se aktivuje po stisknutí klávesy „O“. Zda-li klávesu hráč stiskl, se detekuje ve skriptu „Hrac.py“. V tomto skriptu se kontrolují veškeré úkony hráče (krom pohybu, který je postaven z LB). Tvorba obchodu není o moc jiná než tvorba menu ve hře. Opět půjde o interaktivní aplikaci s myší na Overlay Scene.

- Vytvoří se další scéna „Obchod“. Nastavení je opět stejné jako u „HUD“ scény.
- Do scény se přidá plocha pro pozadí obchodu.
- Přidají se nová tlačítka (viz. obr. 137) na pravé straně obchodu, která budou přepínat mezi typem zboží, co se má zobrazit v nabídce (věže, zbraně, munice, vylepšení). Jejich funkcionalita je popsána skriptem „Obchod.py“.
- Vytvoří se plochy pro zboží. Nad každou plochou budou plocha, kam se umístí model zboží (př.: rotující zbraň) a tlačítko „KUP!“ pro zakoupení položky (na obr. 137 vlevo).
- Popis zboží bude umístěn uprostřed scény (obr. 137 uprostřed) na nové ploše, kam se bude vypisovat text s detailem zboží. Texty uvedené na obr. 137 jako „-“ nebo jen modrou tečkou se budou skriptem ve hře dynamicky měnit podle volby hráče (typ obchodu, popisek zboží).
- Modely se vytvoří pro obchod zvlášť. Již existující modely lze zkopírovat z jiné scény pomocí „Ctrl + C“ a vložit do scény obchodu přes „Ctrl + V“ Nesmí se poté zapomenout



upravit – změnit velikost, pivot umístit do středu objektu (pro rotaci uvedenou v postupu níže), zrušit LB k nim připojené). Modely pro obchod jsou zobrazeny na obr. 138.



Obrázek 137 – Rozložení scény „Obchod“



Obrázek 138 – Modely obchodu

7. Modely objektů se budou skriptem dynamicky přenastavovat (podobně jako text) podle volby kategorie obchodu. Po přidání modelů nad šedou plochu se pomocí LB, které jsou připojené na modely zboží, budou uvádět do rotace (pro efekt). Modely musí být v neoznačené vrstvě (pro metodu addObject ve skriptu „Obchod.py“), jinak nebudou skriptem přidány do scény.
8. V LB nad všemi tlačítky jsou umístěny senzory pro detekci myši nad tlačítkem a senzory pro detekci stisku „LMB“ (tedy obdobně jako u scény „Menu“). Senzory vedou do kontroléru s funkcí „Obchod.aktivace“. Tato funkce obstarává veškerou kontrolu nad úkony hráče v obchodě. Navíc se zde vyskytuje rotace nad vkládanými modely. Logická struktura scény „Obchod“ kvůli množství objektů je poměrně rozsáhlá a je proto k vidění jen v souboru „Pernicek TD.blend“.
9. Ve hře bude obchod vypadat následovně (obr. 139).



Obrázek 139 – Obchod ve hře

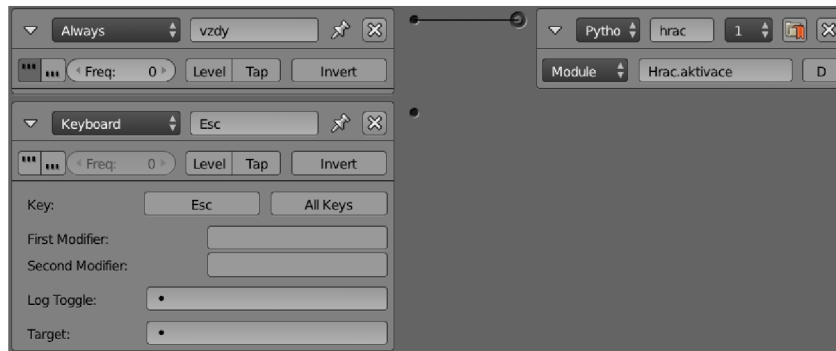
## Pomocná logika hry

Zbývá ještě pár detailů pro běh hry. Je třeba kontrolovat vstupy hráče (krom pohybu, který je již vytvořen pomocí LB). Kontrola stisku „Esc“ pro herní menu, kontrola stisku „O“ pro obchod, kontrola výstřelu ze zbraně „LMB“, atd.

1. Pro tento účel se aktivuje skript „Hrac.py“ – obr. 140.



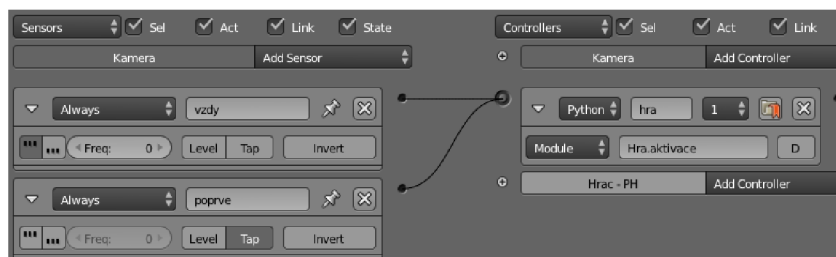
2. Aby se zamezilo vypnutí hry při stisku klávesy „Esc“, vytvoří se volný senzor registrující klávesu „Esc“ – obr. 140.



Obrázek 140 – Registrování dalších úkonů hráče

Poslední logikou je kontrola hry. V ní se zjišťuje prohra hráče, výhra hráče, aktualizují se časovače a nastavuje hra při jejím spuštění.

1. Ke kameře (nebo jinému stálému objektu ve hře) se připojí LB podle obr. 141.



Obrázek 141 – Hlavní logika hry

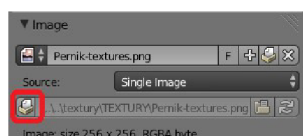
## 5.9 Výdej hry

Následující podkapitola uvede postup exportování hry na platformu OS Windows. Také po úspěšném exportu hra musí být ověřena (validována a verifikována), než může být vydána pro širší veřejnost.

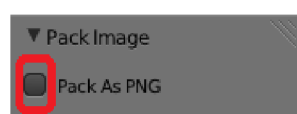
### Export hry

Aby hra byla spustitelná nejen v Blenderu (klávesou „P“), ale i mimo něj, je nutné provést její export a učinit některá nastavení.

1. Uloží se externí textury přímo do „\*.blend“ souboru. V každé obrázkové textuře (s barvami i normálami) se použije tlačítko pro zabalení textury k souboru (Textures→Image – obr. 142).
2. V Tool Shelf→Pack Image se po zabalení označí „Pack As PNG“ – obr. 143.

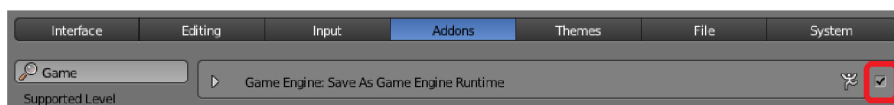


Obrázek 142 – Tlačítko zabalení textury



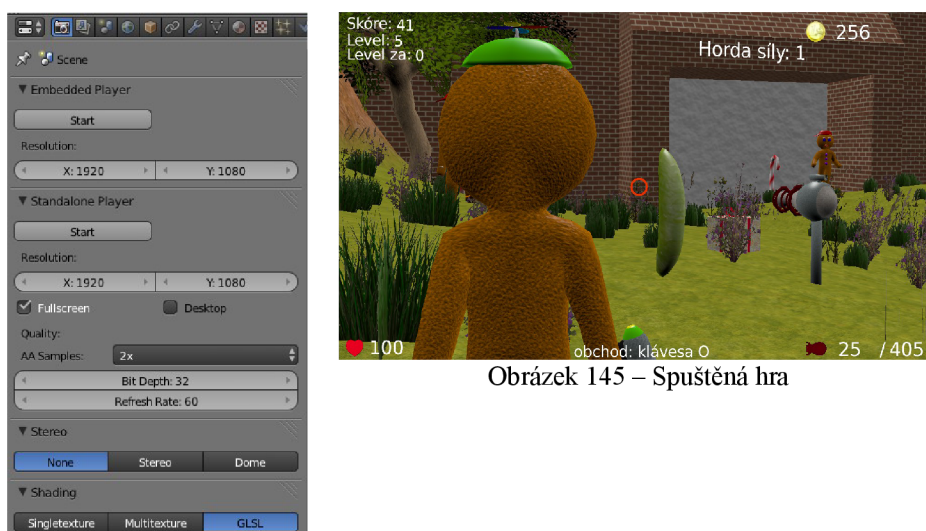
Obrázek 143 – Označení Pack As PNG

- Aktivuje se plugin „Save As Game Engine Runtime“ (ve File→User Preferences...) – obr. 144. Po aktivaci se stiskne „Save User Settings“ pro trvalé použití pluginu v Blenderu.



Obrázek 144 – Aktivace Runtime pluginu pro hru

- Ve všech scénách se aktivují vrstvy, co mají být vidět ve hře. Skryté vrstvy (s přidávanými objekty – projektily, nepřátelé a další) se deaktivují.
- Aktivuje se scéna, která bude ve spuštěné hře jako první, tedy scéna „Hlavní“.
- Všechna nastavení pro exportovanou hru jsou taková, jaká jsou aktuálně v Blenderu (GLSL, rozlišení obrazu, Fullscreen, antialiasing, ...). Podstatné je nastavit parametry v Render Tab, které jsou na obr. 146.



Obrázek 145 – Spuštěná hra

Obrázek 146 – Nastavení Render pro hru

- Exportuje se hra do spustitelného souboru pro Windows platformy (File→Export→Save As Game Engine Runtime). Do vybrané složky souborového systému se automaticky umístí všechny soubory pro spuštění hry.
- Do složky se hrou se umístí složka s externími zvuky, které jsou spuštěny skripty.
- Souborem „Pernicek TD.exe“ se hra spustí. Obr. 145 demonstruje, jak spuštěná hra vypadá. Hru je možné také distribuovat na další platformy MS Windows pouhým zkopírováním složky.

## Ověření funkčnosti hry

Po exportu hry stále není jisté, jestli hra funguje správně. Pro tento účel je možné hru hrát od začátku sám. Zkoušet všechny možnosti, které hra poskytuje a zjišťovat tak její trhliny. Při jejich nalezení je v „\*.blend“ souboru opravit. Dalším stupněm je poskytnout hru svým přátelům nebo rodině (jakožto drobný vývojář), než se poskytne širší veřejnosti.

Tato hra byla průběžně testována několika lidmi, kteří během jejího vývoje dávali zpětnou odezvu. Podle připomínek testerů je dobré hru upravit, aby se jim líbila a vyhovovala jim hratelnost. Vývojář není ten, kdo primárně má hru hrát, ale jsou to ostatní lidé, kteří si ji chtějí zahrát.

Po vydání hry (například na svých webových stránkách a odkazy jinde na tyto stránky) se již hra poskytne pro veřejnost. Pokud je u hry uveden odkaz na vývojáře, lze hru podle připomínek dále upravovat. Zdokonalovat ji pro co nejlepší herní zážitek, který lidé chtějí.

## 6 Závěr

Cílem diplomové práce bylo vybrat vhodný 3D nástroj pro tvorbu 3D aplikací a v tomto nástroji 3D aplikaci vytvořit a sepsat k ní případovou studii. Tento cíl byl splněn.

Pro diplomovou práci byla nastudována literatura na téma „Blender“ a vývoj aplikací v tomto nástroji. Jako primární zdroj informací sloužila dokumentace/manuál Blenderu. Některé knihy sloužily jako vzor pro výklad případové studie. Tímto byl splněn první bod zadání.

Byla navržena metodika tvorby aplikací a postup pro tvorbu případové studie ve čtvrté kapitole. Tímto byl splněn druhý bod zadání.

Byla vypracována případová studie pro tvorbu aplikací v nástroji Blender v páté kapitole. Za studii byla zvolena 3D počítačová hra, která v největší možné míře pokrývá komplexnost tvorby 3D aplikací. Studie zároveň vysvětluje tvorbu složitější hry v Blenderu, která je v literaturách a návodech na webu v tomto měřítku nedohledatelná. Tímto byl splněn také třetí bod zadání.

V průběhu práce byla vytvořena jednoduchá hra pro otestování způsobilosti Blenderu pro následující komplexní hru pro diplomovou práci. Tato složitější hra byla také vytvořena (a tvorba popsána ve výše zmíněné případové studii). Její hratelná verze, obrázky ze hry a celý okomentovaný kód s logikou hry a vytvořenými 3D modely jsou zdarma poskytnuty na webových stránkách pro veřejnost, kterou tato tematika zajímá. Hra byla průběžně testována a upravována podle odezvy testerů. Aktuálně (pro diplomovou práci) se nachází ve funkčním stavu, ale bude i nadále průběžně zdokonalována podle odezvy hráčů.

Dále byl vytvořen snímek z Blender Cycles (sklenička s měsícem) a vzniklo mnoho modelů v Blender Render (model vojáka, stromy a křoviny za použití pluginu Sapling a jiné), kde jejich část je do hry a diplomové práce (jako ilustrační obrázky) zahrnuta.

Jako vylepšení a pokračování práce je možné vytvořit další herní mapy a vymodelovat více nepřátelských jednotek a věží. Také vyvážení hry (získané mince za zničení nepřítele a množství mincí pro nákup zdrojů ve hře) by chtělo úpravy, pro což jsou skripty hry již nachystány. Hra může být ještě více optimalizována a snížit tak nároky na zdroje HW. Je možné vytvořit vícero nastavení pro vykreslování scény hry ke zvýšení „frame rate“ na méně výkonných strojích. Praktické by bylo hru exportovat i na další platformy (krom aktuálního OS Windows) – Android, iOS, Mac, Linux nebo přímé spuštění hry v internetovém prohlížeči.

# Reference

- [1] FLAVELL, Lance. *Beginning Blender: open source 3D modeling, animation, and game design*. New Edition. New York: Distributed to the book trade worldwide by Springer Science Business Media, c2010, xviii, 420 s. ISBN 978-143-0231-271.
- [2] Blender Foundation: *Blender* [online]. [cit. 2013-08-01].  
Dostupné z: <http://www.blender.org>
- [3] Free Software Foundation: *GNU General Public License* [online]. 2013 [cit. 2013-08-01].  
Dostupné z: <http://www.gnu.org/licenses/gpl.html>
- [4] BACONE, Victor Kuller. *Blender Game Engine beginner's guide: the non programmer's guide to creating 3D video games*. New Edition. Birmingham. ISBN 18-495-1702-9.
- [5] *Blendertips* [online]. [cit. 2014-1-8].  
Dostupné z: <http://blendertips.com>
- [6] *Blender3D* [online]. 2013 [cit. 2013-10-22].  
Dostupné z: <http://www.blender3d.cz>
- [7] *Sintel* [online]. 2013 [cit. 2013-10-26].  
Dostupné z: <http://www.sintel.org>
- [8] *Big Buck Bunny* [online]. 2013 [cit. 2013-10-26].  
Dostupné z: <http://www.bigbuckbunny.org>
- [9] *Elephants Dream* [online]. 2013 [cit. 2013-10-26].  
Dostupné z: <http://orange.blender.org>
- [10] Blender Foundation: *Blender Python API* [online]. [cit. 2014-04-02].  
Dostupné z: [http://www.blender.org/documentation/blender\\_python\\_api\\_2\\_67\\_release](http://www.blender.org/documentation/blender_python_api_2_67_release)
- [11] *Učíme se programovat v jazyce Python* [online]. [cit. 2014-04-02].  
Dostupné z: <http://howto.py.cz/cap01.htm>
- [12] VASCONCELOS, Virgilio. *Blender 2.5 character animation cookbook: 50 great recipes for giving soul to your characters by building high-quality rigs and understanding the principles of movement*. Birmingham, U.K.: Packt Open Source, 2011, iii, 289 s. ISBN 978-1-849513-20-3.
- [13] Crytek: *CryENGINE* [online]. 2014 [cit. 2014-05-05].  
Dostupné z: <http://www.crytek.com/cryengine>
- [14] Epic Games: *Unreal Engine* [online]. 2014 [cit. 2014-05-06].  
Dostupné z: <https://www.unrealengine.com>
- [15] Unity Technologies: *Unity 3D* [online]. 2014 [cit. 2014-05-07].  
Dostupné z: <https://unity3d.com>
- [16] Aaron Lee: *The top 14 game engines: The list in full* [online]. 2013-03-22 [cit. 2014-05-07].  
Dostupné z: <http://develop-online.net/news/the-top-14-game-engines-the-list-in-full/0114330>
- [17] Electronic Arts Inc.: *Battlefield* [online]. 2014 [cit. 2014-05-07].  
Dostupné z: <http://www.battlefield.com>

- [18] HERREÑO, John E. *Blender 2.5 hotshot: challenging and fun projects that will push your Blender skills to the limit*. Birmingham, U.K.: Packt Pub., 2011, iii, 314 s. ISBN 978-1-849513-10-4.
- [19] VAN GUMSTER, Jason. *Blender for dummies: challenging and fun projects that will push your Blender skills to the limit*. 2nd ed. Hoboken, N.J.: Wiley, c2011, xv, 430 s. ISBN 04-705-8446-7.



# **Příloha A**

V příloze A se vyskytují některé dodatečné informace pro případovou studii. Jejich znalost je užitečná pro tvorbu počítačové hry, ale ne nezbytně nutná.

Příloha se nevešla do 12MB potřebných pro odevzdání do IS WIS. Celá práce je umístěna v příloženém DVD k diplomové práci.