VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
FACULTY OF MECHANICAL ENGINEERING

ÚSTAV AUTOMATIZACE A INFORMATIKY
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

SIMULACE SESTAVOVÁNÍ KOMPONENT VE VIRTUÁLNÍ REALITĚ

SIMULATION OF COMPONENT ASSEMBLY IN VIRTUAL REALITY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. Karel Škubal
AUTHOR

VEDOUCÍ PRÁCE — Ing. Tomáš Hůlka
SUPERVISOR

BRNO 2023

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ

Zadání diplomové práce

Ústav:	Ústav automatizace a informatiky
Student:	**Bc. Karel Škubal**
Studijní program:	Aplikovaná informatika a řízení
Studijní obor:	bez specializace
Vedoucí práce:	**Ing. Tomáš Hůlka**
Akademický rok:	2022/23

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Simulace sestavování komponent ve virtuální realitě

Stručná charakteristika problematiky úkolu:

Úkolem studenta bude nejprve stručné stručná rešerše možností sestavování modelů ve VR a interakcí jednotlivých součástí. Ve vhodně zvoleném prostředí pak student vytvoří VR model, který bude doplněn o aktivní a interaktivní prvky a otestuje ho na reálném VR headsetu.

Cíle diplomové práce:

Stručná rešerše problematiky.
Návrh konceptu VR modelu sestavovací stanice (např. pro PC komponenty či mikroelektroniku).
Praktická realizace navrženého VR modelu v rámci simulačního prostředí (např. Unity).
Otestování funkčnosti na reálném VR zařízení.

Seznam doporučené literatury:

WANG, S., MAO, Z., ZENG, C., GONG, H., LI, S. and CHEN, B.: A new method of virtual reality based on Unity3D, 2010 18th International Conference on Geoinformatics, Beijing, 2010, pp. 1-5.

TRENHOLME, D., SMITH, S.P.: Computer game engines for developing first-person virtual environments, Virtual Reality (2008) 12: 181.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2022/23

V Brně, dne

L. S.

doc. Ing. Radomil Matoušek, Ph.D. ředitel ústavu	doc. Ing. Jiří Hlinka, Ph.D. děkan fakulty

ABSTRAKT

Virtuální realita je vytvořena oklamáním mozku pomocí zobrazení jedné scenérie na dvourozměrném obraze, přičemž pro každé oko je generován obraz z mírně posunuté perspektivy. Tímto je vytvořena iluze trojrozměrného prostoru, se kterým je za pomocí vhodného software možné i simulovat interakci s virtuálními objekty. Mezi možnou interakci objektů patří jejich sestavování, od obyčejného skládání krychlí na sebe, po různé propojování věcí určováním podmínek. V teoretické části této diplomové práce jsou vysvětleny a popsány možnosti práce s objekty ve virtuální realitě, včetně příkladů možných využití, a poté jsou porovnány některé softwarové nástroje pro tvorbu virtuálního prostředí. V praktické je více přiblížen proces vytváření modelů a podmínek pro jejich sestavování, včetně konkrétního postupu použitého pro vytvoření simulace montáže tří testovacích počítačových sestav s různou kombinací zvolených komponent s danými omezeními.

ABSTRACT

Virtual reality is created by tricking the brain by displaying a single scene on a two-dimensional plane, generating an image from a slightly shifted perspective for each eye. This creates the illusion of a three-dimensional space, with which, by the help of suitable software, it is also possible to simulate interactions with virtual objects. Possible interactions between objects include their assembly, from simple stacking of cubes on top of each other, to various linking of objects by determining link conditions. In the theoretical part of this thesis, some possibilities of working with objects in virtual reality are explained, including examples of possible uses. After that there is a comparison of some software tools for creating a virtual environment. In the practical part, the process of creating models and the conditions for their assembly is more detailed, including the specific procedure used to create a simulation of the assembly of three test computer builds with different combinations of selected components with designated restrictions.

KLÍČOVÁ SLOVA

Virtuální realita, simulace, sestavování komponent, virtuální prostředí, 3D modelování, CAD modelování, Unity

KEYWORDS

Virtual reality, simulation, component assembly, virtual environment, 3D modelling, CAD modelling, Unity

ÚSTAV AUTOMATIZACE A INFORMATIKY

BIBLIOGRAFICKÁ CITACE

ŠKUBAL, Karel. *Simulace sestavování komponent ve virtuální realitě.* Brno, 2023. Dostupné také z: https://www.vut.cz/studenti/zav-prace/detail/145858. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky. Vedoucí práce Tomáš Hůlka.

PODĚKOVÁNÍ

Chtěl bych poděkovat vedoucímu práce Ing. Tomášovi Hůlkovi za podporu při tvorbě této diplomové práce. Dále bych chtěl také poděkovat mým nejbližším za jejich podporu během psaní práce.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že, že tato práce je mým původním dílem, vypracoval jsem ji samostatně pod vedením vedoucího práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury.

Jako autor uvedené práce dále prohlašuji, že v souvislosti s vytvořením této práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následku porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestně právních důsledků.

V Brně dne 20. 5. 2023 ……

Karel Škubal

OBSAH

1 ÚVOD

Virtuální realita je v základu pokus o oklamání mozku tak, aby jím byla neexistující věc interpretována jako reálná. Počátky této snahy lze vidět již v 19. století, kdy pomocí zaplnění zorného pole pozorovatele rozsáhlým obrazem bylo dosaženo imerzivního vjemu dané scenérie. Později se začalo používat k navození iluze prostoru posunutí perspektiv jedné scény pro každé oko, což zapříčinilo prohloubení vjemu této scény do třetího rozměru.

V moderní době tento efekt nemusí být používán pouze pro zobrazování, ale zároveň pro interakci s takto vytvořeným prostorem. Tento koncept virtuální reality se v posledních desetiletích velice rychle rozšířil z čistě industriálního využití i k běžným uživatelům. Tím se urychlil nejen vývoj samotných zařízení pro virtuální realitu, ale také programového vybavení užívaného k vývoji virtuálního prostředí a interakci s ním.

Tato diplomová práce se podrobněji zaměřuje na využití virtuální reality jako zábavnější a účinnější formy učení. Hlavní motivací bylo vytvořit nástroj, pomocí kterého bude uživatelům přiblížena podstata sestavování počítačových komponent. V úvodu práce je přiblížen koncept virtuální reality a jeho postupný vývoj. Dále je pak popsáno, za jakými účely je v moderní době využíváno virtuální reality a možná hardwarová omezení. Následující kapitoly se zaměřují na popis a porovnání softwarových nástrojů k vytváření virtuálního prostředí.

Jako praktická část této práce byla pomocí zvolených programů vytvořena virtuální interaktivní scéna, jejímž účelem je přiblížit uživateli koncept skládání počítačových komponent včetně možných omezení. V textu je pak popsán konkrétní účel využití vybraných programů. Na závěr jsou detailně přiblíženy jednotlivé komponenty ve scéně a interakce mezi nimi potřebné k vytvoření finálních počítačových sestav.

2 TEORETICKÁ ČÁST

2.1 Vývoj virtuální reality

V dnešní době je pojem virtuální realita používán v souvislosti s počítačově generovaným, trojrozměrným virtuálním prostředím, které simuluje realitu za využití interaktivních zařízení, jako jsou například brýle, headsety, rukavice, nebo celé obleky. V běžném případě se jedná o uživatele, který má nasazený headset se stereoskopickými obrazovkami pro navození iluze 3D prostoru. Pomocí pohybových senzorů, ať už umístěných po místnosti, nebo přímo v headsetu, jsou odesílána data o pohybech uživatele, a dle těchto dat je upravován i zobrazený obraz pro navození prezence v prostředí. Tímto způsobem se uživatel může pohybovat po virtuálním světě a zažívat různé perspektivy okolí v reálném čase. Toto může být navíc obohaceno například použitím rukavic se silovou zpětnou vazbou, které navíc simulují dotek objektů ve virtuálním prostředí.

Už dříve byli umělci zaujati technikami a způsoby zachycení reality v živé formě. Již v 17. století vznikaly rozsáhlé panoramatické obrazy zobrazující pohled z výšky na města či přístavy. [1] Cílem těchto záběrů bylo ukázat co nejširší možný prostor bez důrazu na opravdové měřítko věcí. V roce 1838 Charles Wheatstone poprvé demonstroval, jak mozek zpracovává dva 2D obrazy z různých perspektiv. [2] Při sledování takto posunutých obrázků pomocí stereoskopu bylo dosaženo vjemu hloubky a vtáhnutí do obrazu. V roce 1960 byl patentován první headset s dvěma obrazovkami a stereo zvukem. [2] Toto zařízení však neposkytovalo žádnou možnost interakce či sledování pohybu. O rok později bylo uvedeno první zařízení se sledováním pohybů hlavy. To ale nebylo navrženo pro aplikaci ve virtuální realitě, nýbrž pouze ke vzdálenému sledování nebezpečných situací v armádě. Pohyby hlavy u tohoto zařízení byly vzdáleně přenášeny do kamery, díky čemuž měl uživatel možnost přirozeně pozorovat své okolí.

Obrázek 1 - *Long View of London from Bankside* [1]

V roce 1987 byl poprvé definován pojem Virtuální Realita. Do této doby neexistoval pojem pro celkové shrnutí tohoto oboru. Firma *VPL Research* vyvinula řadu zařízení pro virtuální realitu, včetně takzvané *Dataglove* a *EyePhone*, což byla první prodávaná zařízení na virtuální realitu. [2] V roce 2017 začal největší rozvoj pro VR i v osobním použití, kde firmy Oculus a HTC vedly trh.

Virtuální realita má velice široké využití. V dnešní době asi nejvýznamnějším sektorem je zábava. Virtuálně generované světy jsou stále více imerzivní a díky tomu jsou lidé schopni strávit spoustu času socializováním se, čerpáním inspirace, nebo prostě čistou zábavou. V tomto ohledu je pro uživatele mnohem zajímavější virtuální prostředí, které se blíží více fantasy než reálnému světu. Od roku 1969 Myron Krueger z univerzity ve Wisconsinu prováděl experimenty s ohledem na lidskou kreativitu ve virtuálním prostředí. [2] Z těchto experimentů vyplynulo, že lidé věnují více pozornosti zpětné vazbě a interaktibilitě s prostředím, než celkovému vzhledu a přesnosti zobrazování.

Obrázek 2 - VR chirurgický simulátor [8]

2.2 Virtuální realita jako nástroj k učení

Mimo zábavu je virtuální realita také využívána například jako školící prostředek. Je dokázáno, že při použití VR je v mozku zachována podstatně větší část informací, což poté vede ke zvýšení produktivity. Při průzkumu firmou *PWC* bylo dokázáno, že zúčastnění se ve virtuální realitě naučili danou věc až čtyřikrát rychleji, byli mnohem více emočně zainteresovaní s obsahem školení, a dle dotazníku byli také téměř třikrát více sebevědomí s aplikací získaných vědomostí v praxi, ve srovnání s klasickým prezentačním školením. [5] Velkou výhodou tohoto způsobu školení je, že lze simulovat i nebezpečné prostředí bez jakéhokoliv ohrožení trénovaného. Tímto je tedy možné připravit trénované i na různé život ohrožující situace. Dále pak poskytují možnost levné náhrady komplexního technického školení, při kterém by možná chyba mohla být velice nákladná. K tomuto je možné použít nejen čistě virtuální realitu, ale případně i augmentovanou realitu.

Vliv školicího média na rychlost učení popisuje například vědecký článek od David Checa & Andres Bustillo z roku 2020. Tento článek se zaměřuje na využití virtuální reality v procesech učení a zkoumá možnosti a limity této technologie na příkladu modelu středověkého města. [6] Konkrétně je kladen důraz na možný potenciál VR technologie zlepšit historické učení a zapojení studentů. Pro účely výzkumu byly určené dva způsoby učení. Prvním z nich bylo třináctiminutové video, které bylo následováno volnou diskuzí se studenty. Druhým způsobem byla přímá prohlídka modelu města ve VR, při které byl v každé lokaci zúčastněným přehrán úryvek ze stejného videa. Tato prohlídka byla rozdělena dále dle druhu pohybu na teleportaci po prostředí, a volný pohyb pomocí ovladače. Po prohlídce byl proveden zkušební test, který porovnával faktuální znalosti popsané ve videu, vizuálně získané znalosti, a schopnost orientace ve městě. [6] Výsledkem bylo, že způsob učení ve formě virtuální prohlídky pomáhá převážně vizuální paměti. Pro zapamatování faktuálních informací lépe posloužila naučná videa. V experimentu také došlo k chybě ve VR prostředí, která umožnila nechtěné přeskočení jedné z naučných lokací, tudíž i ztrátu informací. [6] Velkou výhodou výuky ve virtuálním prostředí byl mnohem větší zájem ze strany testovaných osob. Toto je způsobeno převážně faktem, že VR je pro spoustu lidí novinka, u které neměli příležitost některou z mnoha možností na trhu vyzkoušet. Zároveň je však vyšší zájem ovlivněn i možností přímé interakce s prostředím se zpětnou vazbou. Toto znatelně zlepšuje i předávání informací o reálném měřítku studované věci, ať se jedná o rozložení důležitých budov ve městě, nebo nějaké složitější díly.

Je možné použít pro učení rozdílné scénáře umožňující rozvoj v mnoha ohledech, z nichž mohou být například:

- zlepšení kritického myšlení a schopnosti rozhodování pomocí simulace situací z reálného světa
- eliminace překážek pro studenty se speciálními vzdělávacími potřebami
- zvýšení motivace uživatelů zkoušením jinak těžko přístupných věcí
- možnost přímé interakce s ostatními uživateli odkudkoliv případně i pro lidi s různými omezeními

2.3 Další využití virtuální reality

Mimo velkou výhodu ve vzdělávání má VR i spoustu dalších využití. Pravděpodobně nejrozšířenějším z nich je zábavní a herní průmysl. Pro tato odvětví byl pozorován obrovský růst v posledních letech.

Největší podíl na zábavním trhu má herní průmysl, který zabírá přibližně třetinu celkové hodnoty. V tomto směru se může jednat o hry založené na pohybu, například Beat Saber, kde hlavním cílem je sekat noty plující k uživateli do rytmu hudby, hry s otevřeným světem, které umožňují prakticky neomezené možnosti prozkoumávání virtuálních světů, aplikace sloužící k socializaci a interakci s dalšími lidmi, nebo různé logické hry využívající třetího rozměru k prohloubení imerze uživatele.

Dále zde mají podíl například přímé přenosy sportů a dalších utkání, a streamování videí, která lze sledovat buď v celém rozsahu 360°, nebo i 180° pro vyplnění zorného pole.

Tato technologie bývá mnohdy zakomponována i do některých atrakcí v zábavních parcích, kde je video doplněno navíc o pohyby sedaček a další efekty. Celkově s příchodem cenově dostupných headsetů velice vzrostla obliba VR nejen v použití čistě profesionálním, ale hlavně v osobním.

Příklady dalších odvětví, která využívají virtuální realitu mohou být:

- Armáda
- Zdravotnictví
- Móda
- Podnikání
- Strojírenství
- Vědecká vizualizace
- Telekomunikace
- Stavebnictví

2.4 VR Hardware

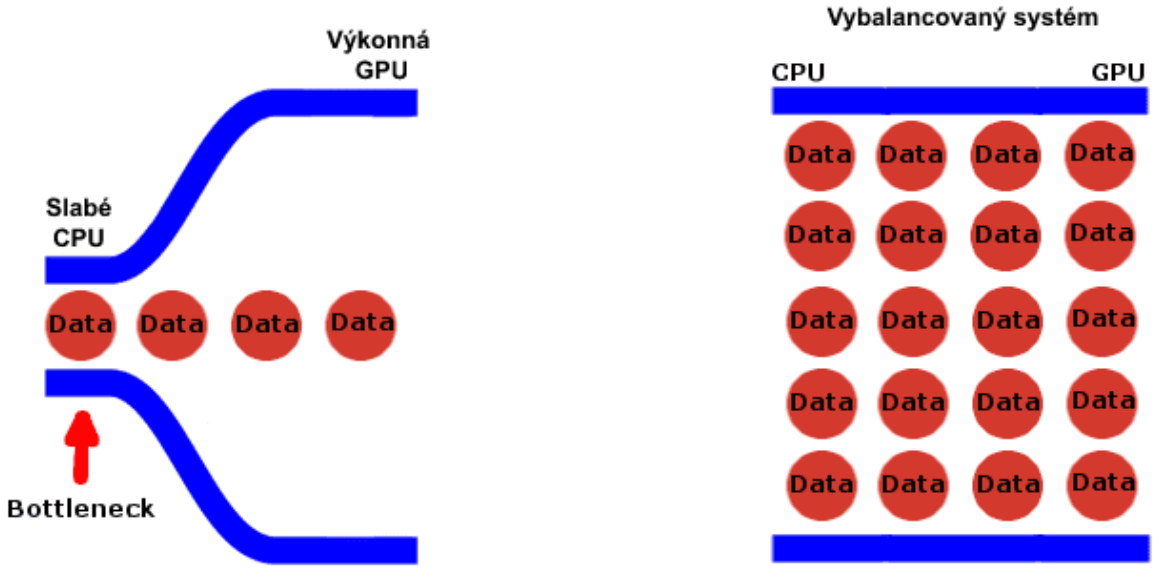
Virtuální realita spočívá v oklamání mozku. Vizuální vjem 3D prostoru je zapříčiněn projekcí perspektivně posunutého obrazu pro každé oko zvlášť. Pro možnost rozhlížení se po scéně je nutné dále pak snímat pohyb hlavy uživatele, podle čeho se poté otáčí i scéna.

První podmínkou pro plynulé a přirozené vnímání scény je potřeba doručení obrazu s minimální obnovovací frekvencí 60 Hz. Hodnoty nižší než 60 Hz mohou způsobovat nevolnost u vyššího procenta lidí a zvýšená odezva velmi ubírá na imerzi uživatele. Většina uživatelských headsetů na trhu se pohybuje mezi hodnotami 72–120 Hz a některé doručují obnovovací frekvenci až 144 Hz, přičemž vyšší číslo obvykle znamená plynulejší vjem.

Druhou podmínkou je nízké zpoždění mezi akcí uživatele a aktualizací scény dle vstupu. Tato latence by pro realistický pocit ze scény neměla přesáhnout kolem 30ms pro pohyby hlavy. Pro snímání pohybu ovladačů je přípustná mnohem vyšší latence, jelikož neovlivňuje přímo sledovaný obraz. Bylo vypozorováno, že do odezvy okolo 200 ms nejsou zaznamenány téměř žádné negativní efekty. Při odezvě do 400 ms dochází jen k mírné ztrátě kvality interakce. Při odezvě ovladačů okolo 800ms už bylo zaznamenáno velké snížení efektivity úkonů vykonávaných v simulovaném prostředí. [9]

Obrázek 3 - *UFO test* – Zobrazení rozdílu mezi snímky při různých FPS [testufo.com]

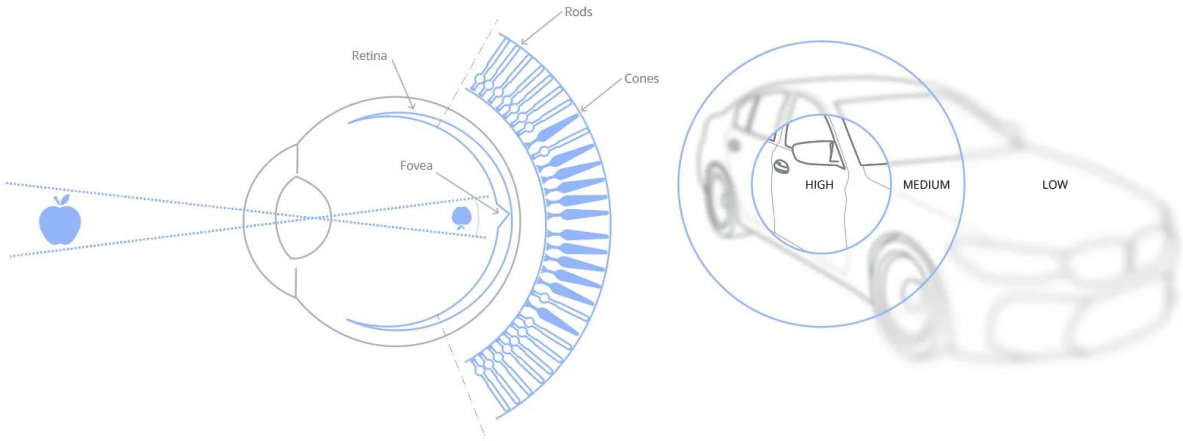
Pro dosažení těchto podmínek je také potřeba dostatečně výkonný výpočetní hardware. Náročnost VR je závislá převážně na software, který musí běžet plynule, a rozlišení konkrétního používaného headsetu. Plynulost software může být závislá na více věcech. Většinou nedostatek plynulosti může způsobovat takzvaný bottleneck neboli úzké místo systému vznikající v některé části hardware. Nejběžnější bottleneck nastává v komunikaci mezi CPU a GPU, kdy slabý procesor nestíhá dostatečně rychle před-zpracovávat a posílat data pro grafickou kartu. Tím je zapříčiněno, že u GPU dochází k nečinnosti, než tato data dostane. Tento problém lze řešit nastavením nižšího počtu věcí, které musí CPU počítat, případně snížit přesnost výpočtů pro některé věci.



Obrázek 4 - Vizualizace CPU bottlenecku [10]

Druhým nejčastějším bottleneckem bývá nedostatečný výpočetní výkon samotné GPU. Tento problém nebývá tak závažný, jelikož plynulost obrazu zůstává v tomto případě přibližně konstantní a nedochází k nepravidelným zásekům obrazu, jak u CPU bottlenecku. Řešením tohoto problému bývá nejčastěji snížení renderovaného rozlišení, nebo omezení výpočetně náročných úkonů, jako jsou stíny a nasvětlení scény. Většina programů podporuje využití takzvaného dynamického rozlišení, které automaticky snižuje počet vykreslovaných pixelů v závislosti na okamžité náročnosti scény. Tím je dosaženo plynulého chodu i při dynamických scénách, během kterých se mohou měnit nároky na výpočetní výkon grafické karty, tudíž frekvence renderování snímků může značně kolísat.

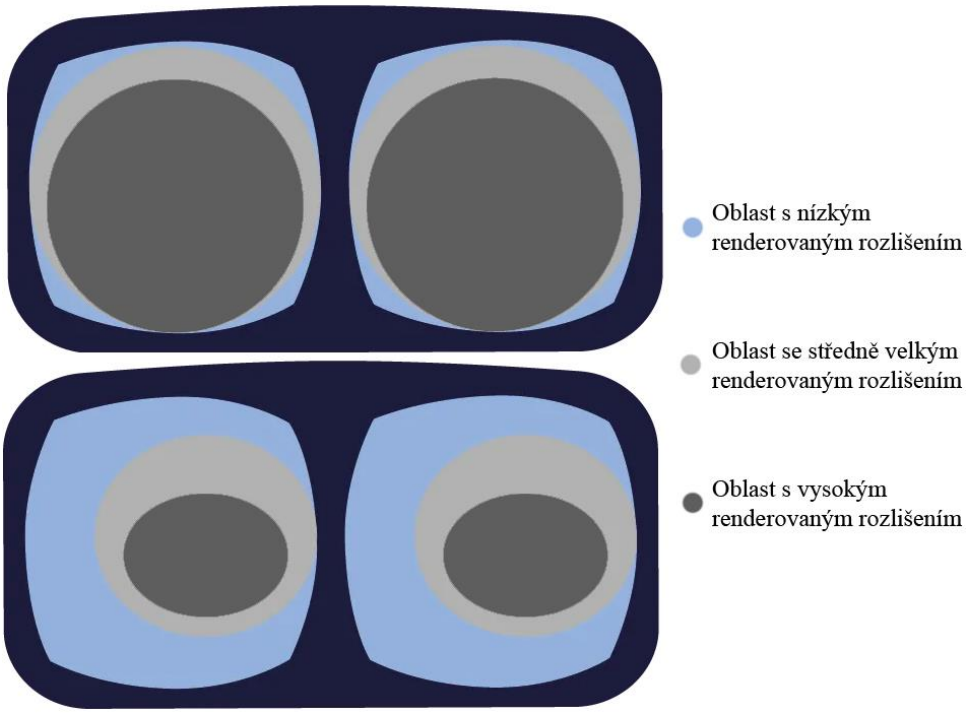
V posledních pár letech stále více programů navíc začíná podporovat takzvané *Foveated Rendering*. [11] Jedná se o podobný princip jak u dynamického rozlišení, avšak tento princip je aplikován pouze na okrajové části obrazovky. Je tím využito vlastnosti oka, které ve středu zorného pole vnímá obraz jako ostrý, a čím více se obraz přibližuje k okraji periferního vidění, tím méně detailů oko vnímá. *Foveated rendering* lze aplikovat na prakticky jakékoliv zařízení, a zároveň tím mírně sníží nároky na výpočetní výkon, avšak ne vždy to přináší optimální uživatelský zážitek.



Obrázek 5 – Vjem obrazu pomocí oka [11]

Statický foveated rendering má větší uplatnění u VR headsetů, jelikož je třeba vykreslovat obraz v poměrně velkém rozlišení a s vysokou frekvencí, tudíž jakákoliv optimalizace potřebných prostředků přijde vhod. Zároveň okraje displejů bývají mnohdy již zkreslené kvůli nepřesnostem na čočkách v zařízení. Avšak s nástupem novějších generací VR headsetů byl tento problém z velké části vyřešen, proto se vrátila potřeba pro vykreslování až do okrajů displeje ve vysokém rozlišení.

U high-end VR headsetů však vznikla nová technologie, která využívá dynamický foveated rendering. Ten využívá sledování pozice očí uživatele s nízkou latencí, z čehož je potom určena lokace malé plochy s plným renderovaným rozlišením ve středu pozornosti uživatele, tudíž nároky na vykreslování každého snímku značně klesají a díky tomu dochází i k výraznému zvýšení stability snímkové frekvence. [11]

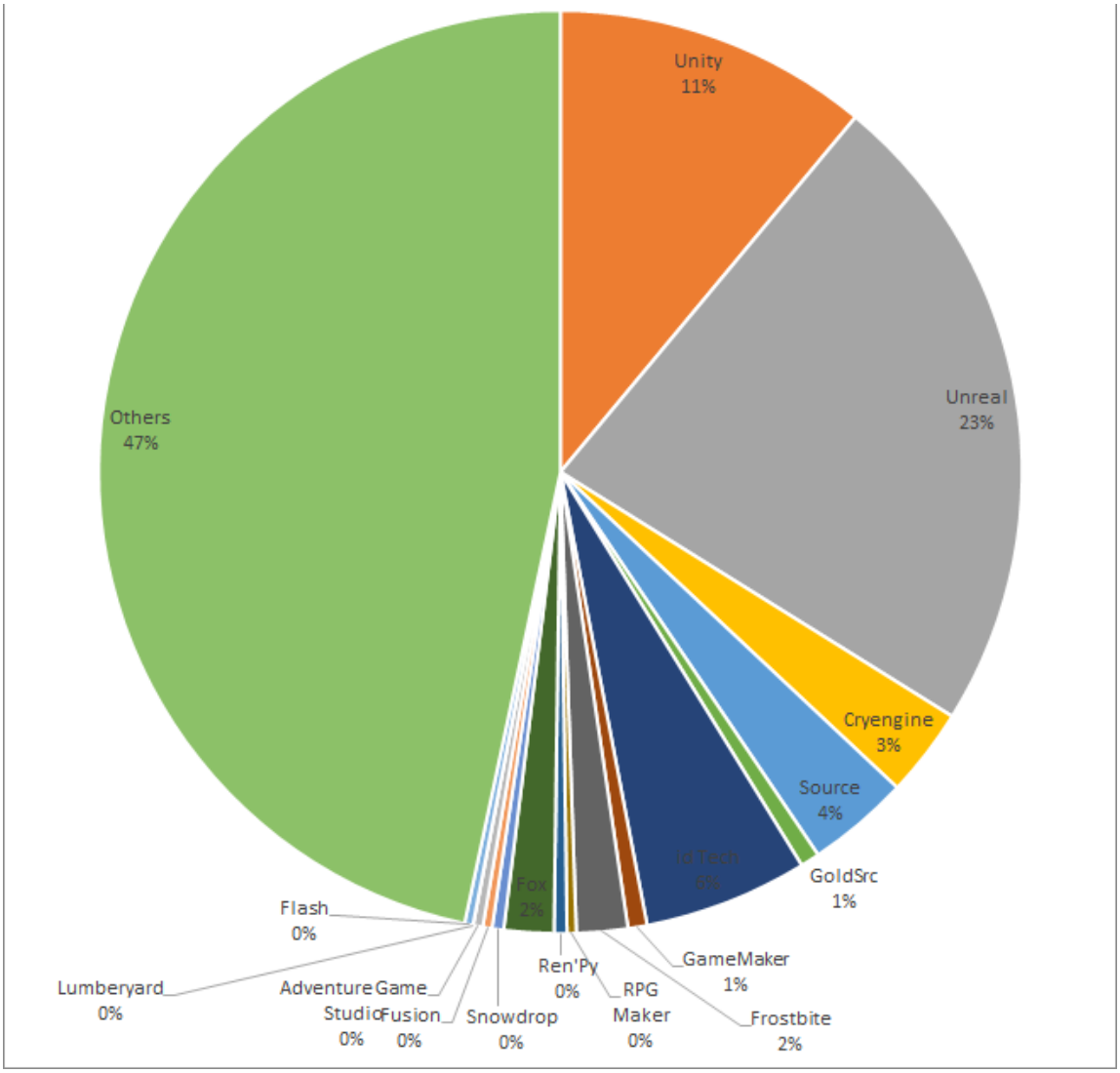


Obrázek 6 - Srovnání statického a dynamického *foveated renderingu* [11]

Dalšími možnými příčinami nestability, nebo nízké plynulosti systému může být například nedostatek paměti RAM, kde minimální paměť RAM v systému by měla být alespoň 8 GB, a v závislosti na aplikaci je doporučováno až 64 GB, nebo dále případně pomalé úložiště. Při použití HDD může docházet k zásekům či pomalému načítání nové scény. Tento problém nastává převážně u velkých, otevřených, či velmi složitých prostředí, která nelze načíst v jednom kuse.

2.5 Nástroje pro tvorbu virtuálního prostředí

Vývojové nástroje pro VR platformy otevírají mnoho možností pro tvorbu imerzivních a interaktivních prostředí. Poskytují vývojářům možnost, jak vytvářet a upravovat trojrozměrné modely a prostředí způsobem, který sedí jejich potřebám. Zároveň umožňují k nim přidávat různé interaktivní a další prvky pro vylepšení celkového výsledného dojmu. Podpora většiny těchto nástrojů zahrnuje vše od jednoduchých VR her po komplexní virtuální simulace. Následující podkapitola obsahuje stručné porovnání nejrozšířenějších editorů.



Obrázek 7 – Tržní podíl vývojových nástrojů ve službě Steam [17]

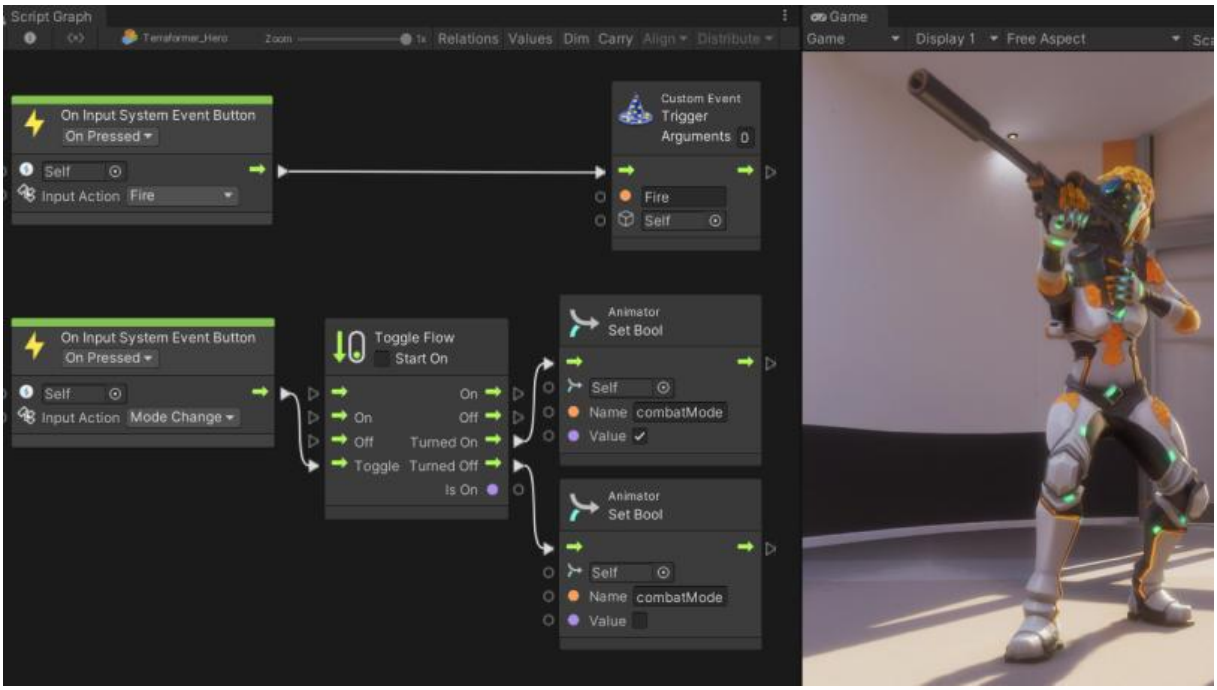
2.5.1 Unity

Jedním z nejrozšířenějších a nejpopulárnějších vývojových softwarů pro VR je *Unity*. V roce 2022 přesáhlo 2,5 milionu registrovaných vývojářů, kteří mezi sebou sdílí nápady a poznatky, od začínajících nadšenců, až po zkušené developery. V posledních letech o tento vývojový nástroj projevily zájem i různá další průmyslová odvětví mimo herní vývojářství. Příkladem lze uvést třeba zdravotnictví, armádu, automobilový nebo filmový průmysl. [13]

Mezi hlavní důvody, proč je Unity tak populární, se řadí třeba široká podpora různých zařízení. Lze vyvíjet aplikace pro mobilní zařízení ať už s *Androidem* nebo *iOS*, přes osobní počítače, až po *Windows Mixed Reality* či *Oculus*. Mimo vytváření 3D prostředí *Unity* umožňuje i tvorbu 2D projektů. Co se týče mobilních her, tak *Unity* vede více jak polovinu trhu a u vývoje VR nebo AR her převažuje s více jak 60% podílem. [13] [14]

Unity poskytuje zdarma licence pro studenty, ale také pro osobní využití s ročním obratem pod $100 000. Díky tomuto je komunita vývojářů velice rozmanitá. Zároveň mají uživatelé přístup k *Asset Store*, odkud lze stáhnout obrovské množství použitelných věcí, jako jsou zvukové efekty, textury, 3D modely, nebo animace. Zároveň tato komunita poskytuje rozšířené návody a fóra pro diskuzi problémů a jejich řešení.

Hlavním programovacím jazykem, který *Unity* používá, je *C#*. Úroveň potřebná pro vytváření základních věcí je nízká a křivka učení velmi pozvolná, tudíž je tento nástroj vhodný i pro úplné začátečníky. Novější verze *Unity* také poskytuje programátorům přístup k funkcím a *API* z *.NET Standardu 2.1*, z čehož vyplývá, že knihovny tvořené pro *.NET* aplikace zde lze použít jako standardní *.dll* plugin. Dále pak *Unity* aspiruje k poskytnutí možnosti tvorby i vývojářům bez zkušeností s programováním. K tomu je využito skriptů a logiky, kde místo psaní kódu uživatel skládá a spojuje dohromady uzly. [14] Jednou z nevýhod, které toto přináší je však poněkud menší výkon oproti čistému *C#*, avšak v budoucnu je plán toto zlepšovat a tím přiblížit výkon k čistě psanému programu.



Obrázek 8 - Příklad *Visual Scripting* v Unity [19]

Z grafické stránky je *Unity* velice versatilní. Používá takzvané *Render Pipelines*, které mají na starost vzít obsah scény a zobrazit je na obrazovce. Na vysoké úrovni pracují ve třech krocích – Redukce, Vykreslování a Post-processing. [18] Při redukci dochází k odstranění polygonů, pixelů, a dalších objektů, které nepřispívají k finálnímu obrazu. Tím dochází ve výsledku ke zlepšení výkonu a plynulejšímu běhu programu. Při vykreslování je scéna převáděna do obrazu, včetně realistických materiálů, textur, nasvícení objektů, nebo například odrazů. Na závěr renderovaný obraz prochází post-processingem. Zde jsou k němu přidány efekty, jako hloubka zorného pole, rozostření obrazu pohybem a dalšími efekty, které napodobují například nedokonalosti kamery pro realističtější podání scény. Unity rozděluje tyto Pipelines na tři hlavní:

- *Built-in*, která je určena jako základní s univerzálním použitím, a je tedy málo přizpůsobitelná.
- *URP* neboli *Universal Render Pipeline*, která poskytuje široké možnosti nastavení, s důrazem na optimalizovanou grafiku pro širokou škálu systémů
- *HDRP*, nebo také *High Definition Render Pipeline*, která se soustředí na vysokou kvalitu detailů a efektů pro výkonné systémy

Obrázek 9 – Srovnání rozdílných *render pipelines* v Unity [18]

2.5.2 Unreal Engine

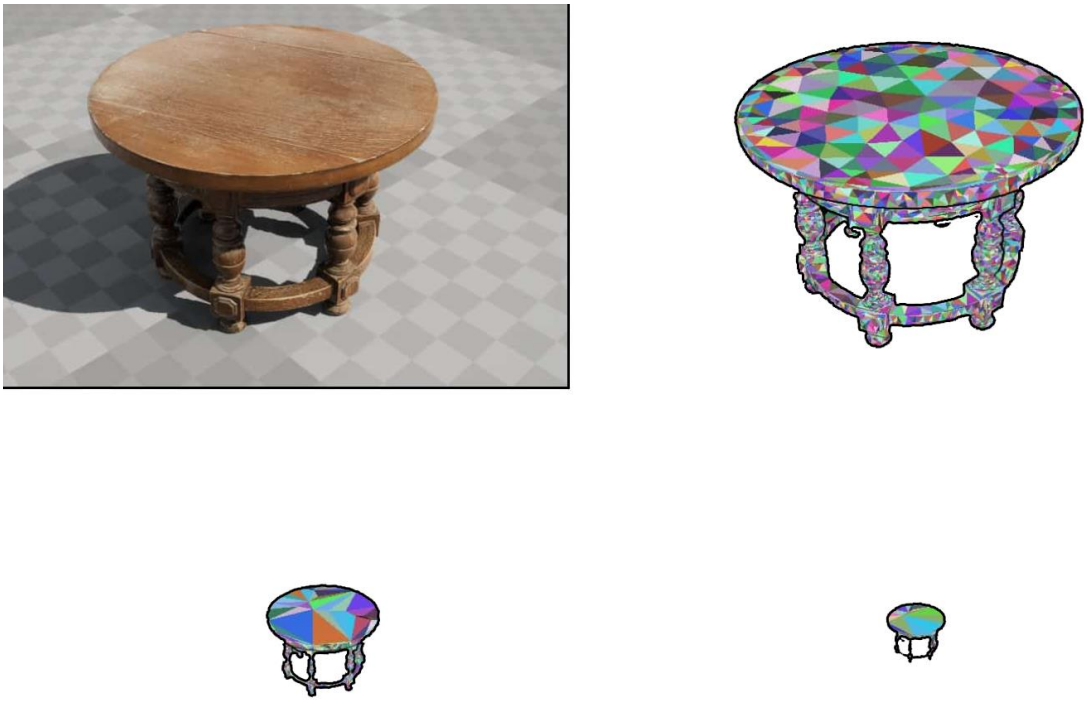
Ve srovnání s *Unity*, které je oblíbené převážně u developerů Indie her a menších studií, *Unreal Engine* je více využíván ke tvorbě AAA her s high-end grafickým zpracováním velkými studii, kdežto *Unity* převládá hlavně na trhu mobilních her. To však neznamená, že se *Unreal* hodí pouze k vývoji velkých her.

Ve srovnání s *Unity* je *Unreal Engine* více zaměřený na realistické zpracování scény. Je označován za jednu z nejvíce otevřených a pokročilých 3D platforem pro tvorbu obsahu v reálném čase. Existuje možnost zde vytvářet i 2D obsah, avšak není ani z daleka tolik využívaná, jelikož je zcela zastíněna potenciálem tvorby velmi kvalitního 3D obsahu. K tomu se řadí například i skvělý výkon při práci s velice komplexními assety či prostředími, což je u velkých projektů důležité.

Jedním z dalších hlavních rozdílů je jazyk, ve kterém je program napsaný. V případě *Unity* se jedná o *C#*, a u *Unreal Engine* o *C++*. Společně se složitějším uživatelským rozhraním toto zapříčiňuje mnohem příkřejší křivku učení pro začínající uživatele. Z toho pak vyplývá využití, kdy *Unreal Engine (C++)* se více hodí pro složité projekty vyžadující vysoký výkon hardwaru a *Unity (C#)* pro rychlejší tvorbu méně složitých a nenáročných desktopových projektů. [15]

Pro tvorbu každého programu či hry jsou potřeba *assety*. Ty si může uživatel buď vytvářet sám, nebo používat již vytvořené z virtuálního obchodu. *Unity* i *Unreal Engine* mají své vlastní obchody s assety. Některé jsou placené, ale spousta z nich lze stáhnout zadarmo. Může se jednat o jednoduché modely použitelné pro vybavení virtuální místnosti, ale i o celé scény včetně nasvícení a dalších efektů. Co se týče kvantity, assetů v *Unity Store* je rozhodně více, jelikož existuje delší dobu. Přestože je *Unreal Engine* podobně starý jak *Unity*, ve srovnání byl k němu obchod s obsahem vytvořen relativně nedávno. Ke květnu 2023 *Unreal Marketplace* obsahoval okolo 33 000 objektů od zvukových efektů, přes modely postav, až po vysoce detailní scény interiéru místnosti. Pro srovnání, *Unity Asset Store* obsahovalo přes 80 000 různých assetů, z nichž přibližně 47 000 bylo označeno jen jako 3D modely. V ohledu kvality assetů závisí na plánovaném využití. Co se týče běžných modelů, jako jsou části architektury a skripty, tak *Unity Store* stále převažuje. V případě environmentálních objektů, jako jsou například skály, tráva, nebo stromy, nabízí však mnohem lepší nabídku *Unreal Engine*. Velkou výhodou také je, že spousta těchto assetů je nabízena zcela zdarma. [21] [22]

Novým přídavkem k nástrojům pro tvorbu obsahu je *Unreal Engine 5*, který byl oficiálně vydán v první polovině roku 2022. Jedním z největších rozdílů od *UE4* je možnost práce s nesrovnatelně větším množstvím polygonů ve scéně. Toho je docíleno nově uvedenou technologií zvanou *Nanite*. V základu se jedná o nahrazení klasické pevné mřížky objektu za dynamickou, která se mění v závislosti na pohledu kamery tak, že vykresluje jen potřebné množství polygonů. [20] [23] Touto optimalizací je umožněno vytvářet velice komplexní scény v reálném čase, jelikož vzdálené objekty vyžadují jen malý výpočetní výkon. Další novou technologií představenou v *UE5* je takzvaný *Lumen* systém, který v reálném čase počítá trajektorii světelných paprsků včetně odrazů pro simulaci velice přesvědčivého nasvícení scény.



Obrázek 10 - Vizualizace technologie *Nanite* v *Unreal Engine 5* [23]

2.5.3 CryEngine

Méně používaným, ale stále velice silným nástrojem je například *CryEngine*. Co se týče grafické stránky i fyzikálních simulací, *CryEngine* je srovnatelný s *Unreal Engine*, ale exceluje hlavně ve venkovních scénách, při simulaci lesnatých oblastí a přírody. Naopak potom ve srovnání upadá jeho podání interiérů. [24] Mezi další nevýhody lze zařadit například nedostatek dokumentace, tudíž potenciální novější developer by mohl být odkázán na tutoriály či příspěvky na forech přímo od komunity uživatelů. Zároveň vzhledem k užší komunitě a dalším nárokům může být křivka učení vývoje v *CryEngine* poněkud příkrá.

Existují další nástroje pro vývoj VR obsahu, jako například *Source 2* od společnosti *Valve*, *Blender*, nebo *WebVR* od *Google*, a mnoho dalších. Ty však většinou nabízejí specifické funkce nebo omezení, a komunity okolo nich nebývají tak rozsáhlé.

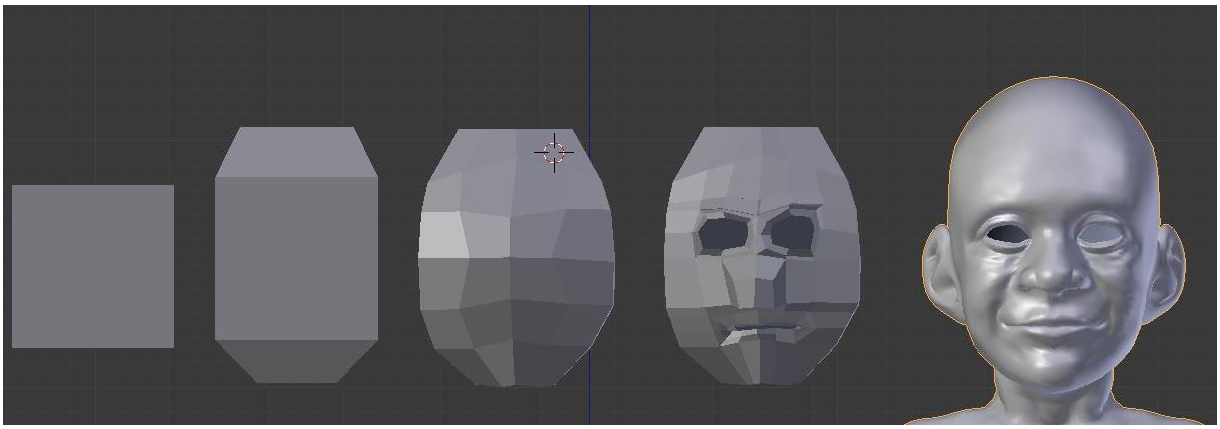
2.6 Pomocný software a tvorba assetů

Většina z výše zmíněných vývojových softwarů pro virtuální realitu, nebo celkově 3D obsah, nabízí krom možnosti vytváření scény a skládání základních tvarů, také nástroje k tvorbě komplexních modelů. Ačkoliv lze mnohdy pomocí těchto nástrojů vytvořit plně imerzní prostředí, většina vývojářů upřednostňuje využití kombinace dalších nástrojů. Důvodů může být mnoho, od rozšířených možností, přes zjednodušenou práci a lepší interface, až po prostou osobní preferenci. V této podkapitole je uvedeno pár alternativ, popřípadě dalších nástrojů pro tvorbu assetů do virtuálního prostředí.

2.6.1 Způsoby tvorby 3D modelů

Vytváření kvalitních a detailních modelů bývá složitý a zdlouhavý proces, který lze rozdělit do několika kroků. Prvním, a zároveň nejdůležitějším krokem je samotné modelování. Jedná se o proces tvorby vizuální a digitální reprezentace objektu. Tento postup není pevně daný, ale pro určité skupiny objektů lze vytypovat nejčastěji používané techniky.

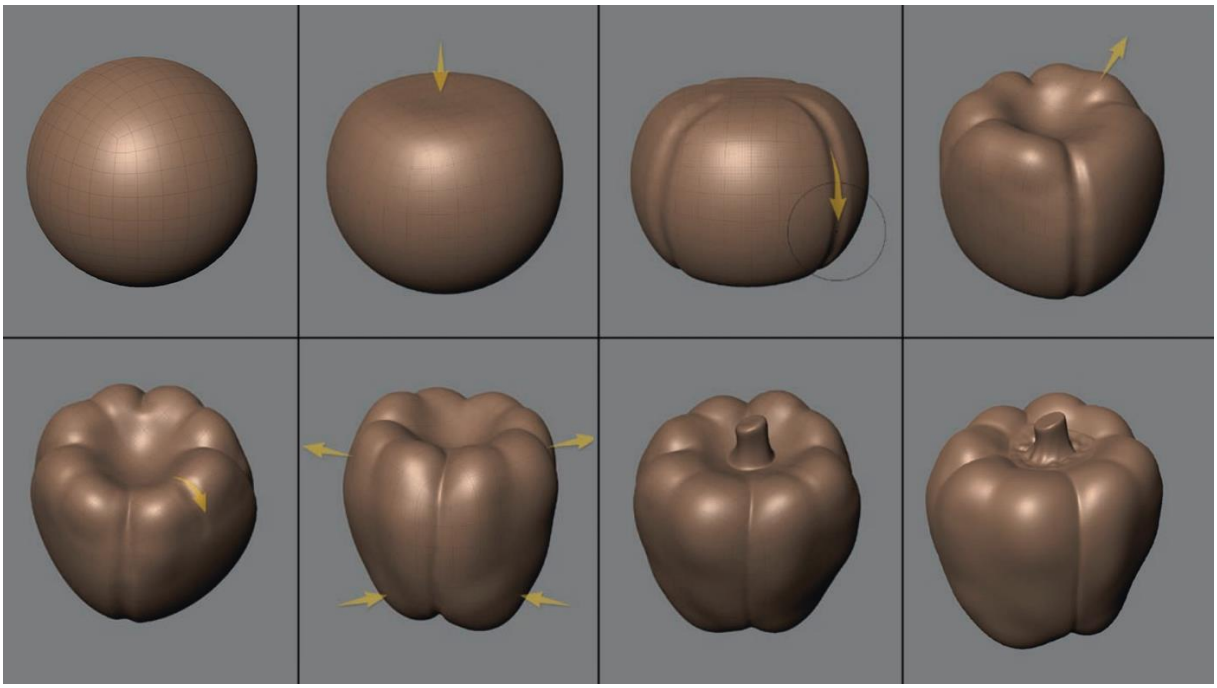
Nejznámější z těchto technik je *modelování ze základního tvaru*, z čehož vyplývá, že osoba vytvářející model začíná scénu s jednoduchým objektem, jako je například krychle, koule, nebo válec, který poté pomocí dostupných nástrojů tvaruje do výsledné podoby. [25] [26] Výhodou této techniky je relativní jednoduchost a přímočarost postupu, což je zejména vhodné pro rychlé prototypování a zkoušení nápadů. [27] Tímto způsobem je nejlepší modelovat objekty s pevným povrchem, jako je například různý nábytek, ale je možné takto vytvářet i komplexní tvary, jako jsou třeba postavy lidí nebo zvířat, i když k tomu nemusí být použití této metody tolik efektivní.



Obrázek 11 - *Box modeling* – příklad možného postupu [27]

Pro tvorbu organických tvarů je však nejvíce využívána další technika modelování zvaná *sculpting* neboli *digitální sochání*. Jak už název napovídá, při této technice je s modelem zacházeno stejně jako s hliněnou sochou. Celkový tvar je upravován a hněten do výsledné podoby pomocí skupiny virtuálních nástrojů, které zabezpečují funkci přidávání či odebírání materiálu, zdrsňování nebo uhlazování povrchu, či přidávání různých záhybů a posunutí. [28]

Sculpting je velmi využívaný pro tvorbu modelů pro filmy, reklamy, pokročilé hry a další scény, kde je požadována vysoká kvalita vizuálních efektů a 3D animace. Pro začátečníky může být tato technika poněkud obtížná, jelikož bývá těžké odhadnout, jak komplexní musí být model pro dané využití, případně jak má vypadat jeho topologie. [28] Mimo tato úskalí je však digitální sochání velmi oblíbená technika, jelikož jde převážně o kreativní činnost, tudíž po zvládnutí několika základních instrumentů pro tváření je digitální modelář omezen prakticky jen vlastní představivostí a kreativitou. [29]



Obrázek 12 - Příklad postupu u *digitálního sochání* [29]

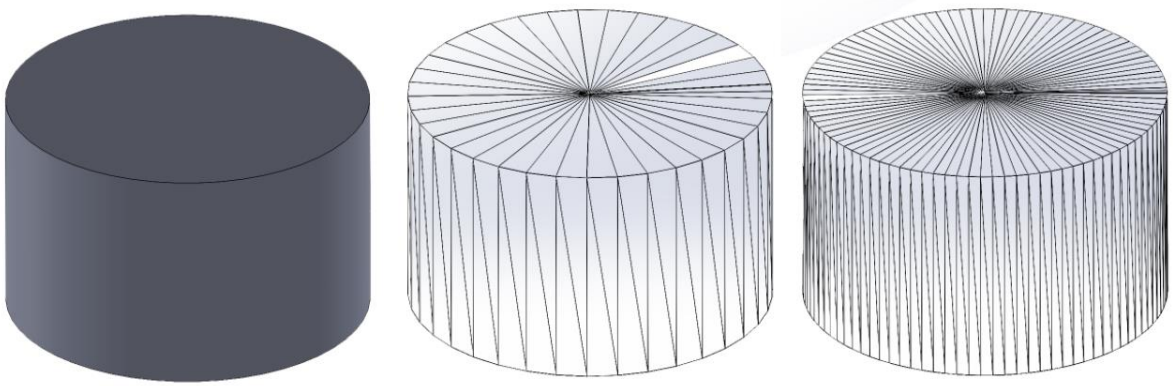
Procedurální generování, nebo také simulace objektu je metodou modelování, při které je využito matematických vzorců, algoritmů a přesně určených pravidel k vytvoření dynamických trojrozměrných objektů a prostředí. V základu jde o techniku, která umožňuje tvorbu velkého množství unikátních assetů bez potřeby modelovat každý z nich individuálně. Pro procedurální generaci objektů je počáteční náročnost vyšší, jelikož je potřeba při tvorbě provádět velké množství kroků navíc. Tato iniciální investice času a prostředků začne být však výhodná v případě velkých projektů, kdy možnost automatické generace mnoha assetů efektivitou rychle překoná jejich individuální tvorbu.

Jedním odvětvím této techniky je takzvané *modulární modelování*, kdy je objekt rozdělen na pevně dané části. Tyto díly je potom možné dodatečně dynamicky upravovat: měnit jejich rozměry, zakřivení, a do určité míry i pozici. Na následujícím obrázku je zobrazen model stolu složen ze tří hlavních komponent, označených různými barvami. Za pomoci dodatečné úpravy těchto částí bylo dosaženo nových konfigurací s unikátními tvary. [30]

Obrázek 13 – Zobrazení příkladu *procedurálního generování* na modelu stolu [30]

Podobným způsobem tvorby většího množství assetů je takzvané *auto-variační modelování*. Při tvorbě assetu se nespoléhá čistě na změnu tvarů objektu, ale spíše je pohlíženo na věc jako na skládanku. Jednotlivé součásti jsou na sebe navázány v určitých bodech a podle daných pravidel je možné je obměňovat. Mimo obyčejné ohyby a změny délek je možné takto na sebe napasovat i díly s úplně rozdílným tvarem a materiály. Tímto způsobem lze z knihovny obsahující pár komponent opět poskládat obrovské množství různých variací. [30] Procedurální modelování je také často využíváno při tvorbě randomizovaných objektů, jako jsou stromy, krajina, a simulované materiály jako různé tekutiny nebo oheň. [25]

Pro tvorbu více rozměrově přesných a technických objektů se využívá nejčastěji *CAD modelování*. Na rozdíl od předešlých metod, které se spoléhaly převážně na úpravu topologie a přetváření polygonů objektu, s čímž jsou spojené problémy jako vznik nechtěných přehybů, nepřesností, a podobně, spočívá tento způsob modelování ve vytváření pevně daných vektorových modelů. Jelikož má uživatel pevnou kontrolu nad každou hranou a plochou objektu, vytváření organicky vypadajících modelů může být poněkud obtížné. [28] Navíc je třeba do většiny programů pro tvorbu 3D scén tyto modely převádět do podoby geometrie s mřížkou, jelikož nepodporují práci s vektorovým formátem objektu. V závislosti na zvolené úrovni přesnosti detailů je pak vygenerován model složený z trojúhelníkových ploch.

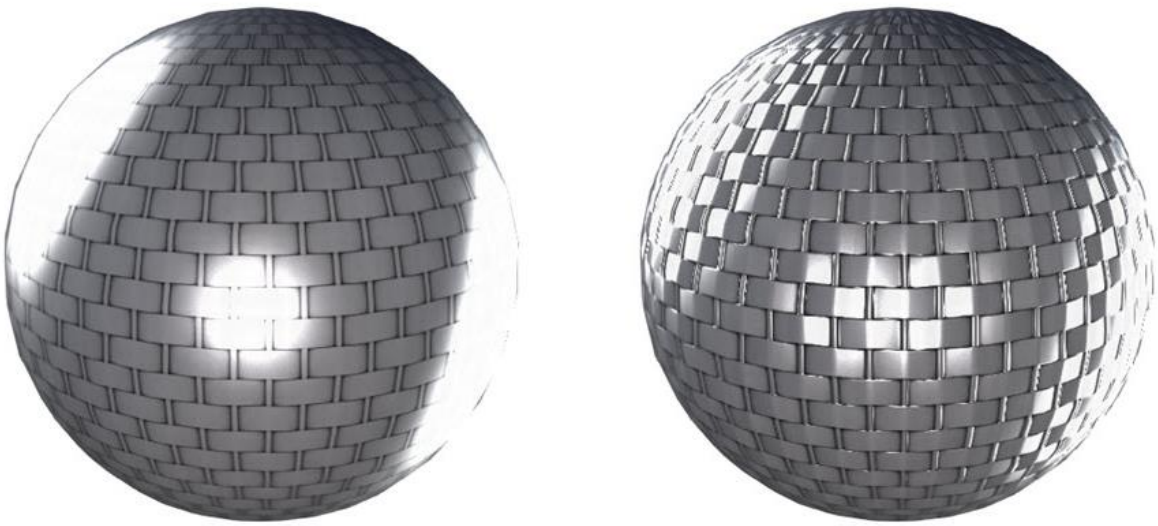


Obrázek 14 - Převedení *CAD* modelu na objekt s mřížkou (formát *STL*) s různou úrovní přesnosti

2.6.2 Dodatečná úprava 3D modelů

Po vytvoření trojrozměrného modelu je mnohdy potřebná jeho optimalizace. Hlavně při vytváření objektů pomocí sochání může vznikat až zbytečně detailní model, který poté může negativně ovlivnit plynulost chodu aplikace. K tomu je využívána takzvaná *retopologie*, pomocí které je snížen počet polygonů objektu. Toto lze učinit buď ručním vytvořením nové mřížky přes existující model, nebo použití některého z nástrojů pro její automatické vygenerování. [25]

Dalším krokem je *UV mapování*. To spočívá ve vytvoření rozvinutého tvaru modelu na dvojrozměrnou plochu, na kterou je poté aplikována textura. Tento krok je stěžejní při tvorbě texturovaných povrchů. [25] [35] Finálním krokem je poté nanesení samotné textury. Ta se může skládat z více vrstev. Základem je samotná *barevná vrstva*, která dodává modelu požadovaný vzhled. Na ni lze poté nanést *normálovou texturu*, která dodává modelu iluzi trojrozměrné geometrie. Toto je jeden z nejčastěji využívaných způsobů optimalizace komplexní virtuální scény, jelikož nejsou vytvářeny žádné další geometrické prvky, ale na povrch objektu lze tímto přidat drobné detaily, které většinu času vypadají totožně se skutečným modelem. [36] Dále je například možné přidat *spekulární*, nebo také *odrazovou texturu*, která obsahuje informace o odrazu světla od objektu, čímž lze třeba oddělit dva různé materiály na stejném objektu, či ještě zvýraznit vjem třetího rozměru u normálové mapy. [25] [37]



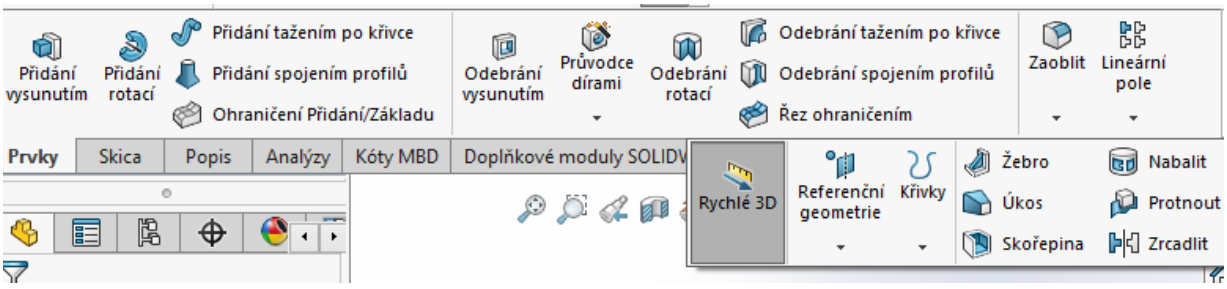
Obrázek 15 - Příklad objektu před a po aplikování *normálové textury* [37]

2.6.3 Příklady software k úpravě 3D modelů

Programů, které lze použít pro vytváření 3D modelů je spousta. Většina profesionálního softwaru je však placená. Mezi nejznámější z nich v oblasti tvorby pokročilých animaci a vizuálních efektů se řadí *Autodesk Maya*, nebo třeba *Cinema 4D* od firmy *MAXON Computer*. Dále pak stojí za zmínku program *Houdini*, který využívá hardwarové akcelerace pomocí grafické karty pro komplexní simulace zvířecí srsti, tekutin, explozí, a další. *Houdini* dále, místo klasického pracovního postupu, využívá procedurálního generování uzlů, ke kterým se lze zpětně vracet a prozkoumávat tak různé iterace projektu, na rozdíl od běžného postupu, kdy se změny provedené uživatelem ukládají postupně do historie. [32] [33]

Co se týče neplaceného software pro vytváření 3D obsahu, z daleka nejvíce možností poskytuje program *Blender*. Nabízí nepřeberné množství nástrojů, od tvorby modelů včetně jejich texturování a editace rastrové grafiky, přes úpravu videí, tvorbu aplikací a videoher, až po komplexní simulace tekutin, částic a jejich animace. Zároveň je kompletně *open-source*, což umožňuje uživatelům se podílet na změnách, opravách chyb a aplikování nových funkcí. [34]

Softwarových nástrojů vhodných převážně pro tvorbu přesných technických částí je spousta. Mezi těmi nejznámějšími se dá uvést například *Autodesk Fusion 360*, který se soustředí hlavně na jednoduchost použití a intuitivní uživatelské rozhraní. Dále pak komplexnější software, který poskytuje mnohem více nástrojů nejen pro modelování, ale také pro různé analýzy objektů a sestav, jako například *SOLIDWORKS* či *PTC Creo*. [39]



Obrázek 16 - Základní nástroje pro úpravu modelů v *SOLIDWORKS*

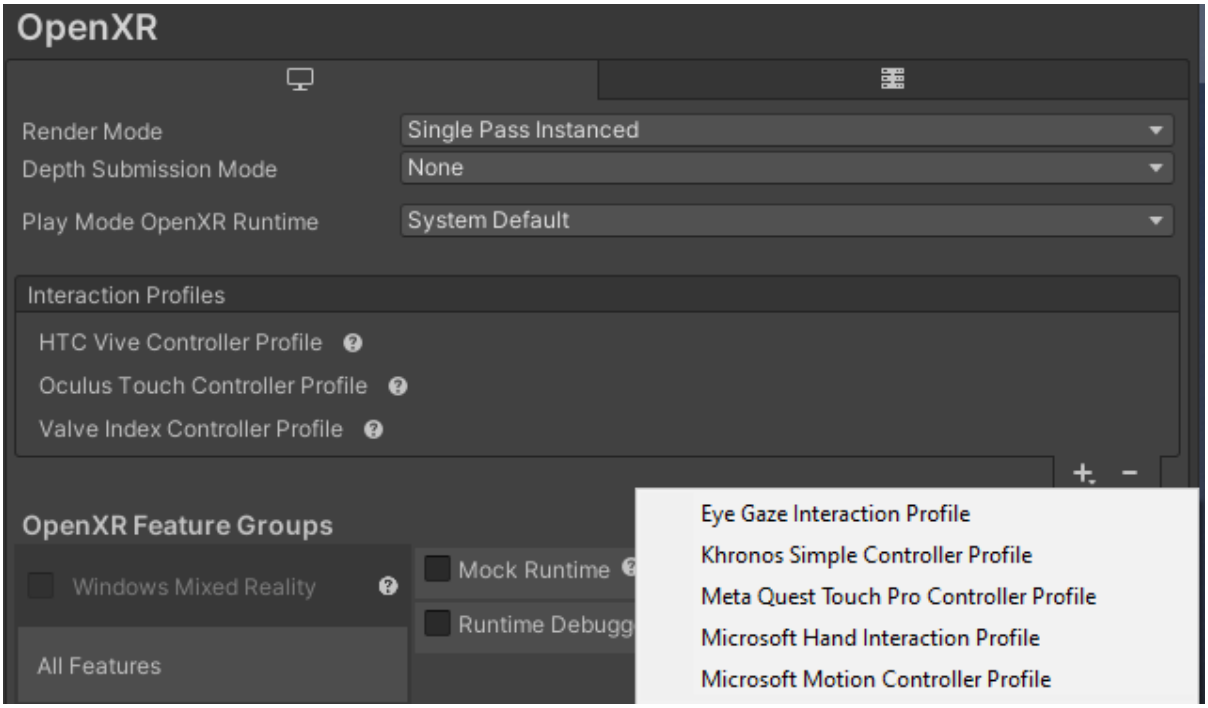
3 PRAKTICKÁ ČÁST

Tato kapitola se věnuje vypracování simulace sestavování komponent v multiplatformním herním engine *Unity*. Pro účely této diplomové práce byla vybrána simulace sestavování počítačových komponent. V následujících podkapitolách jsou uvedeny jednotlivé části postupu. Jako první bylo nutné zvolit vhodný software pro tvorbu virtuálního prostředí a jeho součástí. Mimo výběr herního enginu a využití některých volně dostupných assetů bylo třeba vytvořit většinu modelů pro komponenty. Dalším krokem bylo nastavení uživatelsky přívětivého rozhraní pro virtuální realitu. Hlavní částí je vytvoření pravidel pro jednotlivé interakce mezi komponenty a uživatelem.

3.1 Použitý software

3.1.1 Vývojový engine – Unity

Základním stavebním kamenem projektu pro virtuální realitu je vývojové prostředí. Z několika programů uvedených v teoretické části vyšlo *Unity* jako nejvhodnější engine pro daný účel. Samotný program je volně dostupný pro vývojáře s ročním obratem pod $100 000. Má velice rozšířenou komunitu a obrovské množství dostupných assetů a podpůrných doplňků. Velkou výhodou jsou sady nástrojů s názvem *XR Interaction Toolkit* a *XR Plugin Management* zaměřující se na vývoj VR obsahu, které velice ulehčují nastavení a tvorbu interakcí mezi uživatelem a prostředím ve virtuální realitě. Umožňují přijímat vstupy ze široké škály VR zařízení, přidávají základní interakce s prostředím, jako úchop a přenášení objektů, a také vracení zpětné vazby k uživateli, ať už ve hmatové nebo vizuální formě.



Obrázek 17 - Zvolení podpory různých VR ovladačů pro projekt v *Unity*

Při vytváření nového projektu bylo třeba vybrat základní šablonu, která usnadní pozdější nastavování projektu ke konkrétnímu účelu. Pro tvorbu simulace ve virtuální realitě byla vybrána šablona přednastavená pro 3D scénu s *Universal Render Pipeline*, která se nezaměřuje tolik na realistické, graficky náročné zpracování scény, ale převážně na její plynulý chod, což je pro VR podstatné.

Jak již bylo zmíněno, *Unity* také poskytuje vlastní obchod s assety, odkud lze stáhnout a importovat přímo do projektu již přednastavené modely, textury, skripty, a další obsah. Velkou část assetů v obchodě je nejdříve nutné zakoupit. Některé assety jsou však poskytovány i zadarmo. Sice je tím poměrně omezen možný výběr, ale lze mezi nimi najít i velice kvalitně zpracované modely. Pro účely tohoto projektu byl využit hlavně detailní model pro VR ovladače. Dále pak, spíše pro vizuální ozvláštnění scény, byla zvolena kontrastní textura na podlahu a stěny. Většina modelů byla vytvořena ve zvoleném CAD programu přiblíženém v následující kapitole, a některé další vznikly poskládáním více základních geometrických těles dohromady přímo v *Unity*.

3.1.2 Software ke tvorbě 3D modelů – SOLIDWORKS

Jelikož pro účely tohoto projektu byly vybrány konkrétní počítačové komponenty, ke kterým je poté přiřazen i jejich popis, bylo nutné jejich modely vytvořit. Protože se jedná o velice přesné objekty s drobnými detaily, vhodným řešením pro jejich tvorbu byl výběr nějakého CAD softwaru. Z osobní zkušenosti byl zvolen program *SOLIDWORKS*, který umožňuje rychlou tvorbu 3D modelů za pomoci základních operací, jako je vytažení plochy či 2D náčrtu, odebrání materiálu, zaoblení hran a další. Z modelů vytvořených pomocí nástrojů v *SOLIDWORKS* bylo poté nutné vygenerovat soubor ve formátu kompatibilním s Unity, tedy *FBX* nebo *OBJ*. Ani jeden z těchto formátů však není v možnosti exportu ze *SOLIDWORKS*, tudíž bylo nutné nejdříve z modelu vygenerovat objekt s geometrickou mřížkou složenou z trojúhelníků ve formátu *STL*, a poté použit další software pro převod do podporovaného formátu.

3.1.3 Úprava 3D modelů – Blender

K tomuto převodu byl použit program *Blender*, který je *open-source* a volně dostupný. Zároveň podporuje široké množství formátů a bylo možné tedy vybrat ten správný. Dále by bylo možné *Blender* použít i k vytvoření *UV mapy* objektu a následnému nanesení textur. Tento proces je však náročný na grafický design, tudíž jsou modely ve výsledném projektu odlišitelné převážně tvarem a poskytnutými popisky. Jednou z uvažovaných možností bylo také využít *Blender* pro vytvoření celé virtuální scény a vyhnout se tak jednomu kroku v procesu. Ve srovnání s *Unity* však není tvorba VR obsahu tak rozšířená, nedostává se jí ani takové podpory od komunity a celkový pracovní proces vytváření aplikací pro virtuální realitu je méně uživatelsky přívětivý.

3.1.4 Úprava textur a obrázků – Photopea

Za zmínku určitě stojí i tento online program, který je možné použít jako neplacenou náhradu za *Adobe Photoshop* přímo v prohlížeči. Jeho hlavní využití spočívalo ve tvorbě textur pro popisky objektů ve výsledném projektu a drobné úpravy obrázků nebo získávání údajů o některých rozměrech komponent, co nebylo možné přesně dohledat.

3.2 Použitý hardware

Pro testování virtuálního prostředí byl použit headset *HTC Vive*. Jedná se o jeden z prvních profesionálních headsetů se snímáním pohybů po místnosti, v prostoru o rozměrech až 3.5m x 3.5m, pomocí sledovacích základen. Samotný headset obsahuje dva *AMOLED* displeje s rozlišením 1080 x 1200 pixelů na oko s obnovovací frekvencí 90 Hz a zorným úhlem 110°. Mimo headset a základny jsou součástí i dva ovladače pro interakci s prostředím, na kterých je multifunkční dotyková plocha, boční tlačítko pro úchop, dvě menu tlačítka a spoušť s analogovým přenosem hloubky stisknutí.

Základem pro vývoj virtuálního prostředí, zejména pro VR je dostatečně výkonný hardware. Jelikož se v posledních letech stal virtuální obsah velice populárním, nový hardware bývá těmto nárokům přizpůsoben i v nižší cenové třídě. Jako minimum pro plynulý chod VR na HTC Vive jsou doporučeny následující komponenty [41]:

- Procesor: *Intel Core i5-4590/AMD FX 8350* nebo lepší
- GPU: *NVIDIA GeForce GTX970/AMD Radeon R9 290* nebo lepší
- Paměť: *4GB RAM* nebo více
- Video výstup: *HDMI 1.4/DisplayPort 1.2* nebo novější
- USB výstup: *USB 2.0* nebo novější
- Operační systém: *Windows 7*, *Windows 8.1* nebo novější a *Windows 10*

Těmto nárokům by měla v dnešní době dostát již většina osobních počítačů. Zároveň scéna vytvořená pro tuto diplomovou práci není velmi rozsáhlá, ani nevyužívá graficky náročných úkonů, jako je například raytracing, jelikož jde hlavně o prezentaci konceptu skládání komponent, a ne grafických možností enginu. Zároveň scéna obsahuje pouze samostatnou místnost s pár modely potřebnými pro tyto účely.

Obrázek 18 - *HTC Vive* – Headset, ovladače a sledovací základny

3.3 Vytváření modelů pro komponenty

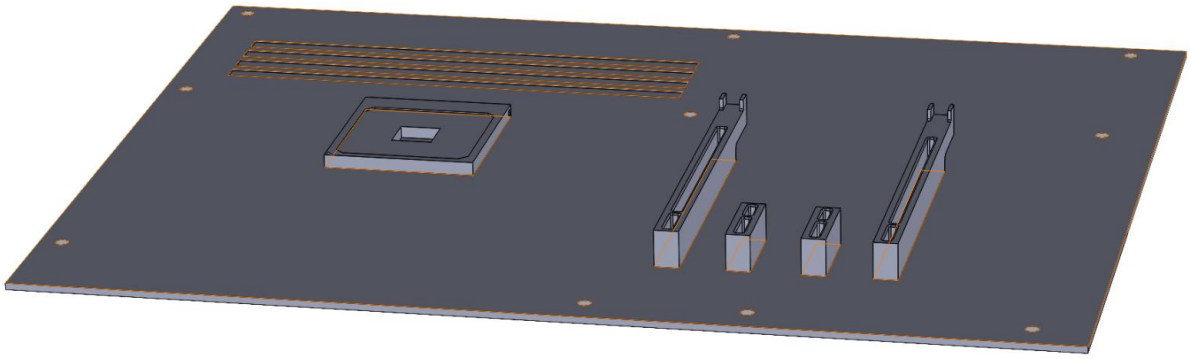
Jak bylo již zmíněno, tvorba všech assetů pro počítačové komponenty probíhala v programu *SOLIDWORKS* s následným převodem přes *Blender* do formátu podporovaného v *Unity*. Většina součástí prošla zjednodušením. Pro účely simulace skládání komponent byly opomenuty některé nepodstatné detaily. Hlavními charakteristikami komponent tedy zůstal jejich celkový rozměr a tvar, umístění a měřítko míst pro připojení dalších dílů, a vizuálně nejvýraznější rysy při pohledu na model, aby bylo možné je mezi sebou jednoznačně rozlišit

3.3.1 Základní deska

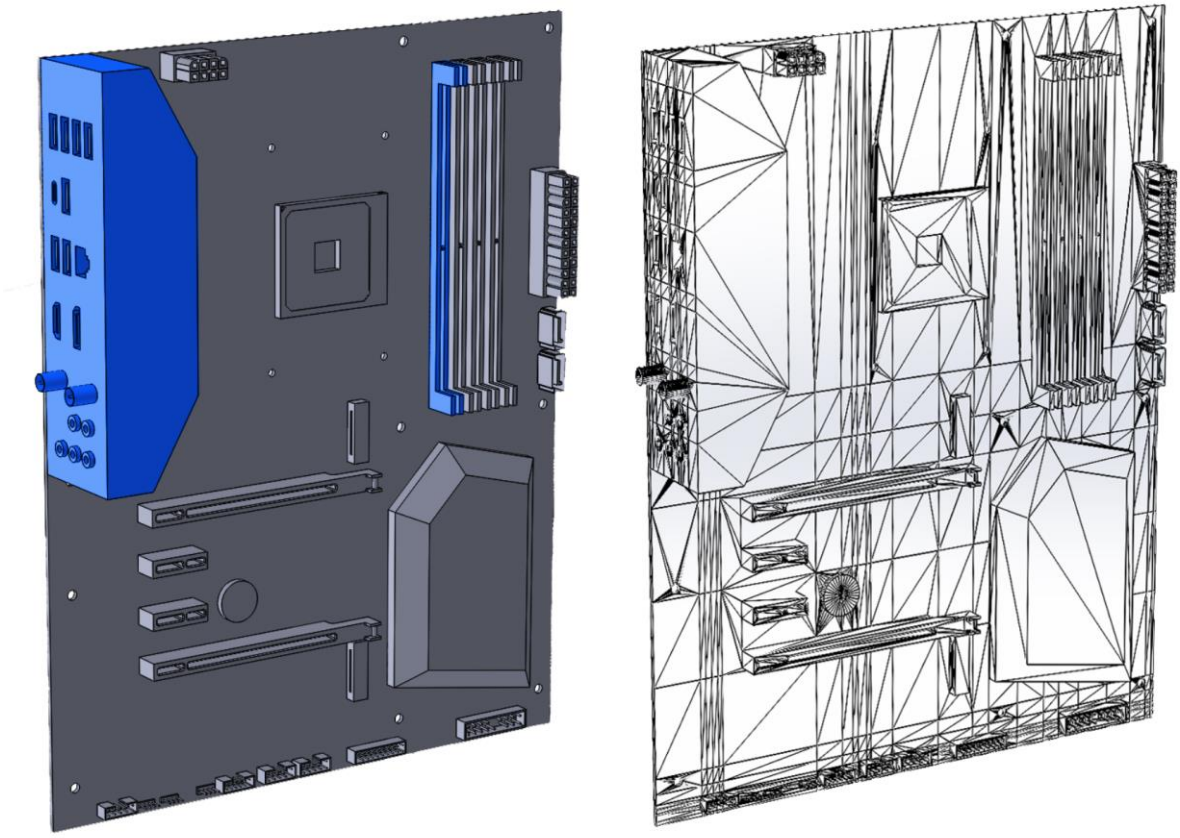
V projektu byly použity dva rozdílné modely základních desek. První z nich byla určená pro procesory s paticí *AM4*. Nejznámější řada procesorů pro tuto patici se nazývá AMD Ryzen. Pro účely simulace byla vybrána řada 5000, přesněji *CPU Ryzen 5 5600* s 6 fyzickými jádry a 12 vlákny reprezentující střední třídu a *Ryzen 9 5900X* s 12 fyzickými jádry a 24 vlákny jako příklad high-end CPU. Dále deska podporuje typ operační paměti *DDR4*. Tento konkrétní model je vytvořen pro základní desku ve formátu *ATX*, což je nejčastěji používaná velikost ve stolních počítačích.

Vytváření modelu začalo spodní PCB deskou s dírami na uchycení podle reálné předlohy základní desky s čipsetem *AMD B550*. Na ni pak byly domodelované PCI sloty, které slouží k připojení grafické karty, případně i dalších rozhraní, jako zvuková či síťová karta. Poté následoval slot na CPU, který prošel značným zjednodušením, jelikož software měl problém vygenerovat velké množství děr pro piny na samotném procesoru. Zjednodušená geometrie ve výsledku snižuje i konečné nároky na hardware při samotné simulaci ve virtuálním prostředí. Jedním ze způsobů, jak by toto šlo obejít, ale stále zachovat jednoduchou geometrii, je aplikace textury s normálovou mapou, což by vytvořilo iluzi složitého tvaru.

Jako další krok následovalo vytvoření drážek v desce, do kterých byly následně vsazeny některé komponenty, zvýrazněné modře v obrázku 20. K tomuto byla v *SOLIDWORKS* vytvořena sestava, kde bylo jen třeba vložit jednotlivé modely a nadefinovat vztahy propojení mezi nimi. Například pro sloty RAM byl tento způsob rychlejší, jelikož stačilo k nim vytvořit pouze jeden model, který byl dále rozkopírován do jednotlivých drážek. Podobným způsobem byl vložen boční panel se vstupy a výstupy, který však byl uchycen přímo k plochám základní desky. Na závěr byly vytaženy sloty pro napájení a další kabely.

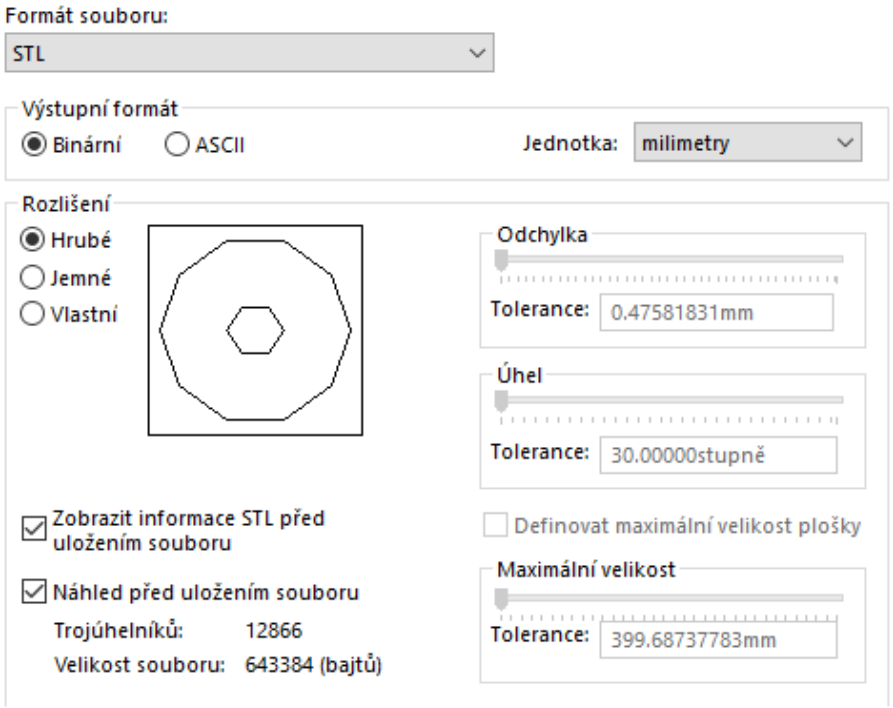


Obrázek 19 - Model základní desky v *SOLIDOWORKS* po vytvoření některých prvků

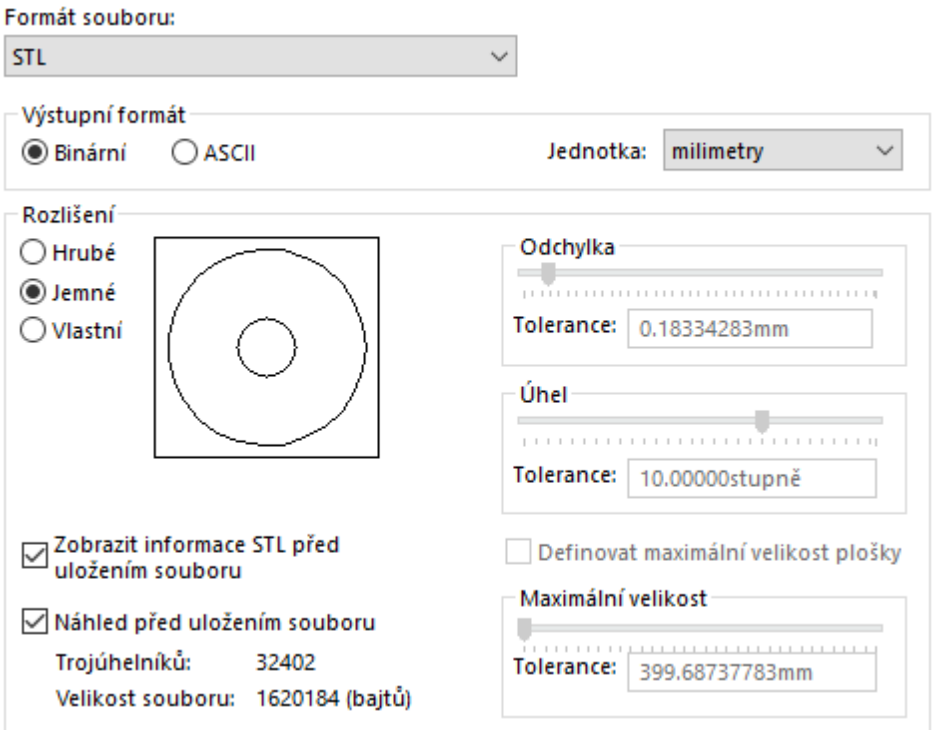


Obrázek 20 - Model základní desky v *SOLIDWORKS* a jeho následný převod do formátu *STL*

Po vytvoření každého z modelů bylo nutné vygenerovat z nich mřížku ve formátu *STL*. K tomu *SOLIDWORKS* nabízí možnost základního nastavení přesnosti geometrie v podobě tolerancí odchylky a úhlu. Lze k tomu použít automaticky přednastavené možnosti pro jemné či hrubé rozlišení nebo nastavit vlastní tolerance.



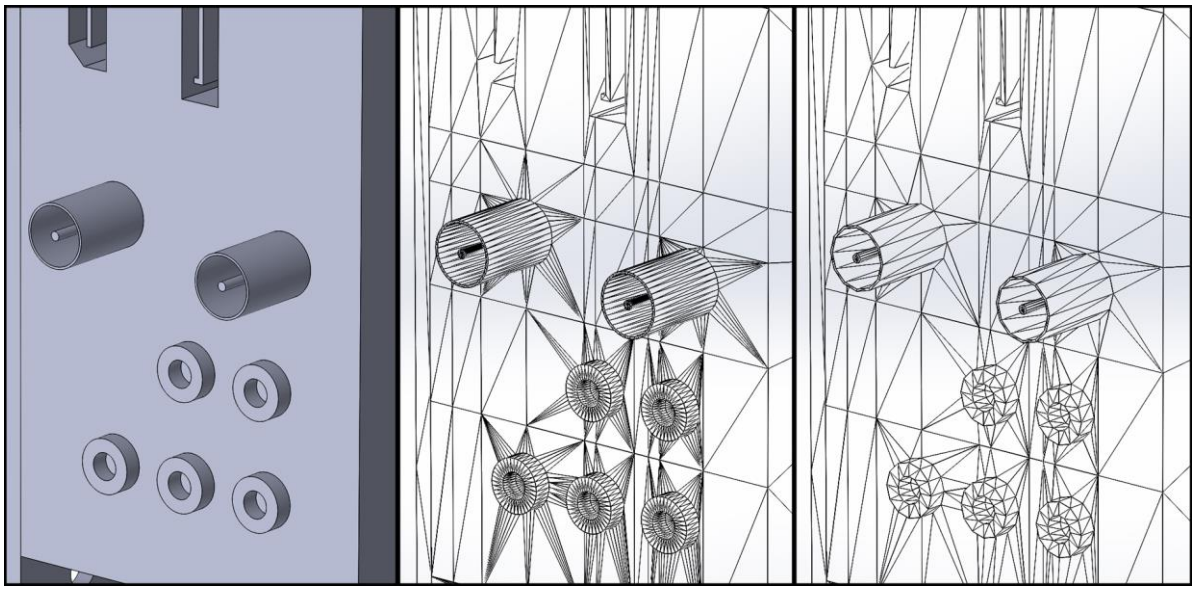
Obrázek 21 - Příklad nastavení přesnosti mřížky při převodu do formátu *STL – Hrubé*



Obrázek 22 - Příklad nastavení přesnosti mřížky při převodu do formátu *STL – Jemné*

Toto nastavení samozřejmě ovlivňuje nejen výslednou kvalitu modelu, ale i jeho velikost. *SOLIDWORKS* zobrazuje přímo v okně nastavení i velikost souboru a počet trojúhelníků nacházejících se na modelu, včetně živého náhledu na převáděnou součást. To umožňuje rychlou optimalizaci pro potřebu uživatele, jelikož jsou případné změny ihned viditelné.

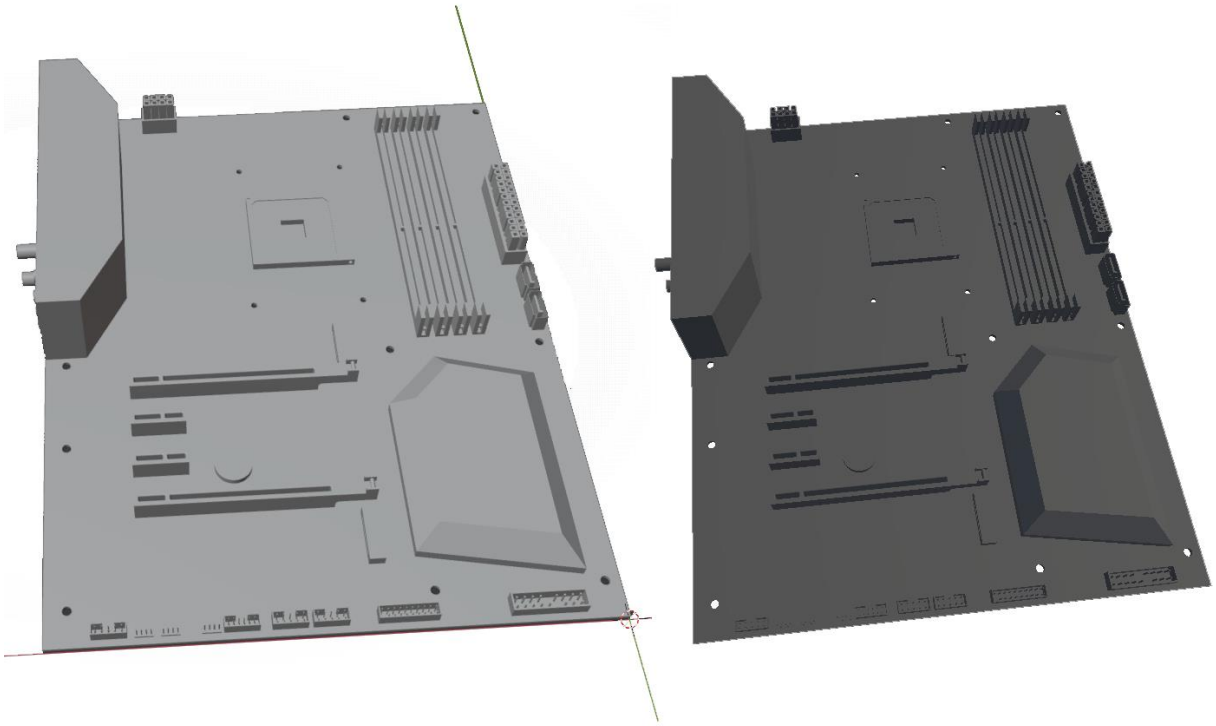
Při generaci mřížky v *SOLIDWORKS* dochází k optimalizaci dané mřížky tak, aby výsledný soubor nebyl příliš velký. Toto lze vidět hlavně na rovných plochách modelu, kde není téměř žádný rozdíl v hustotě mřížky, jelikož by tím nebylo docíleno vyšší úrovně detailů. Změna jemnosti je naopak nejvíce patrná na zakřivených místech, kde není možné reprezentovat plochu součásti pomocí trojúhelníků bez degradace výsledného vzhledu. V následujícím obrázku jsou porovnány modely vygenerované pomocí dvou různých nastavení.



Obrázek 23 - Porovnání původního CAD modelu s převedeným *STL* s nastavením *Jemné a Hrubé*

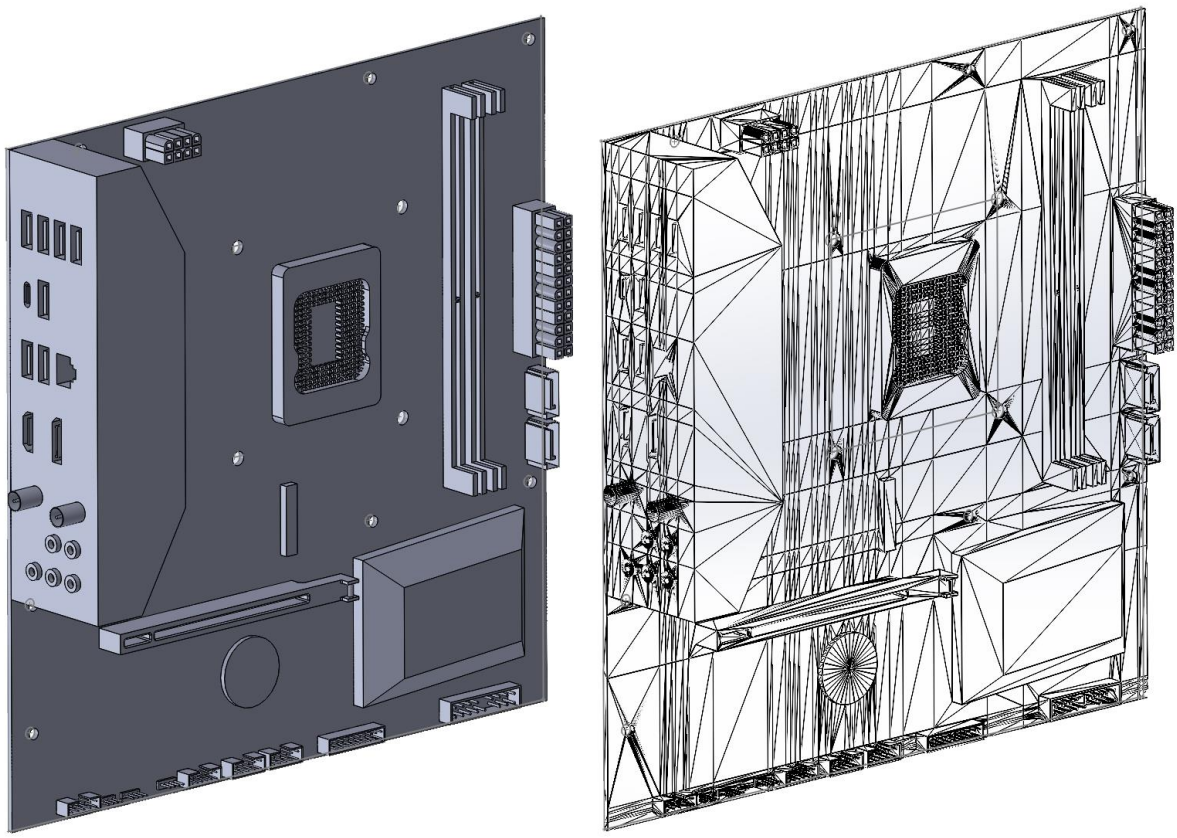
Pro modely bylo ve výsledku zvoleno vlastní nastavení přesnosti, které zasahovalo mezi obě přednastavené hodnoty. Tím bylo docíleno relativně hladké reprezentace zakřivených součástí a zároveň malé velikosti souboru. Pro tento projekt nebyla nutná velká optimalizace geometrie modelů, jelikož se jedná o malou a omezenou scénu. Kdyby se jednalo o rozsáhlé virtuální prostředí s velkým množstvím modelů na obrazovce, bylo by pravděpodobně nutné snížit nároky na hardware.

Po vytvoření *STL* souborů bylo nutné převést je na formát podporovaný programem pro tvorbu virtuálního prostředí, v tomto případě *Unity*. Dle rad a zkušeností rozsáhlé komunity byl vybrán formát *FBX*, což je jeden z nejrozšířenějších formátů souborů pro 3D modely. Při vkládání do *Blenderu* bylo třeba dát pozor na automaticky vygenerované objekty při založení nové scény. Jedná se o kameru, zdroj světla a základní krychli. Pokud nebyly tyto objekty odstraněny před exportem do formátu *FBX*, došlo k jejich vmísení do výsledného souboru a při importu do scény Unity rozpoznalo jednotlivé objekty a docházelo k neočekávanému chování.



Obrázek 24 - Model ve formátu *STL v Blenderu* (vlevo) a výsledný asset import do *Unity* (vpravo)

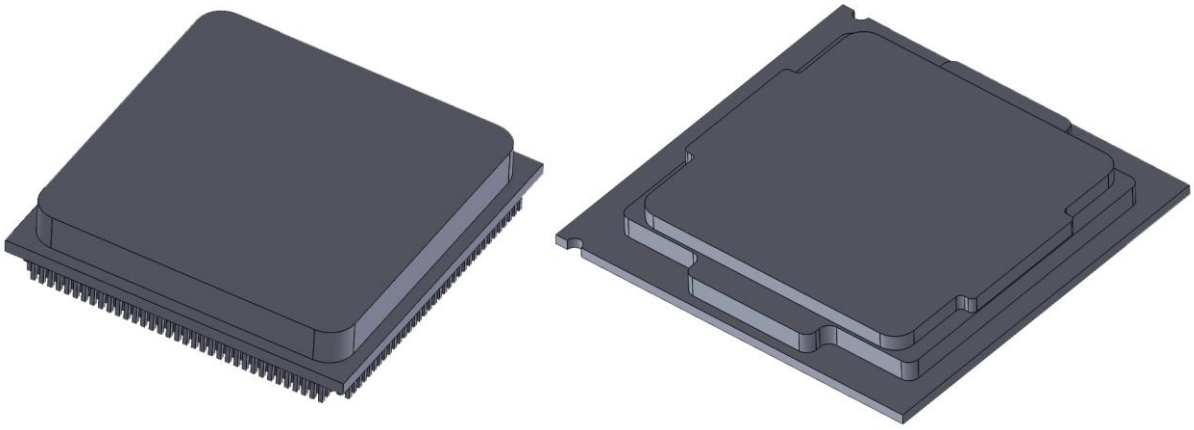
Jako příklad základní desky pro Intel procesor byl zvolen model s čipsetem *H510M*, která obsahuje socket *LGA 1200*. Pro jednodušší odlišitelnost byl zvolen velikostní formát mATX. Jedná se o menší verzi *ATX*, která může být osekaná o určité části, například některé RAM či PCI sloty. Hlavní využití spočívá v úspoře místa, kupříkladu v kompaktních počítačových sestavách. V případě assetu pro tuto základní desku byly vymodelovány i piny pod CPU sloužící k jednoduššímu odlišení.



Obrázek 25 - Základní deska pro *Intel* CPU – model a vygenerovaná mřížka ve formátu *STL*

3.3.2 CPU

Model procesoru je relativně jednoduchý. Jde o drobnou součást, jejíž detaily nemají pro simulaci sestavování velkou prioritu. Hlavním rozpoznávacím znakem mezi druhy procesorů je jejich základní tvar a umístění pinů, kde u *AMD* se piny nachází přímo na CPU, kdežto pro Intel jsou piny umístěné v základní desce, tudíž spodní část procesoru je rovná. Mezi dvěma použitými procesory od *AMD* není viditelný rozdíl. Jako řešení tohoto problému bylo využito informační tabulky, která se zobrazuje pouze při zvednutí komponenty uživatelem.



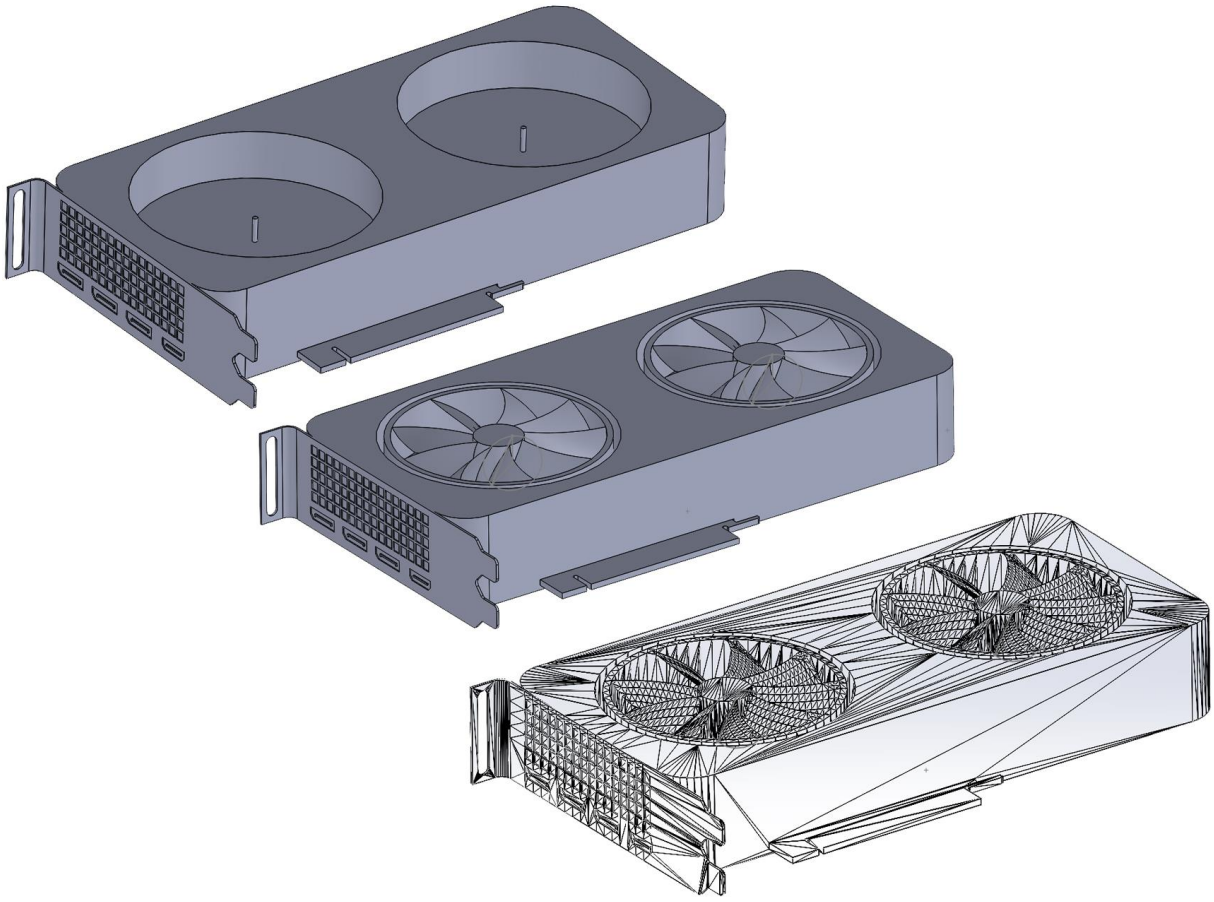
Obrázek 26 - Srovnání modelů pro *AMD* CPU (vlevo) a *Intel* CPU (vpravo)

3.3.3 Grafická karta

Grafický procesor v počítači zajišťuje grafické výpočty i generaci výsledného obrazu vykreslovaného na monitoru. GPU lze v základu rozdělit na integrované a dedikované. V případě integrované grafické karty se jedná o její zakomponování přímo do CPU. Tyto karty nedosahují takového výkonu jako dedikované. Pro účely simulace sestavování komponent byly vybrány pouze dedikované GPU, jelikož jsou vhodnější pro vizualizaci všech důležitých částí ve výsledné sestavě. Vybrány byly grafické karty od společnosti *NVidia*, konkrétně karty GeForce RTX řady 30. Ve výsledném virtuálním prostředí lze najít 3 konkrétní modely:

- *NVidia GeForce RTX 3050* jako základní model pro levné systémy
- *NVidia GeForce RTX 3060 Ti* jako GPU střední třídy
- *NVidia GeForce RTX 3090* reprezentující high-end komponentu

Výsledné assety jsou opět zredukovány na základní tvar o přesných vnějších rozměrech. Důležité části jako výstupy pro obraz, panel k uchycení a PCI konektor jsou přítomny. Jejich modelování začalo opět deskou o základním tvaru, ze kterého byly vyříznuty sloty pro ventilátory se středovým válcem sloužícím k jednoduššímu sestavení v dalším kroku. Následně byly vymodelovány potřebné drobnější detaily. Ve druhém souboru byl vytvořen model ventilátoru, který se následně v sestavě propojil s danou GPU.

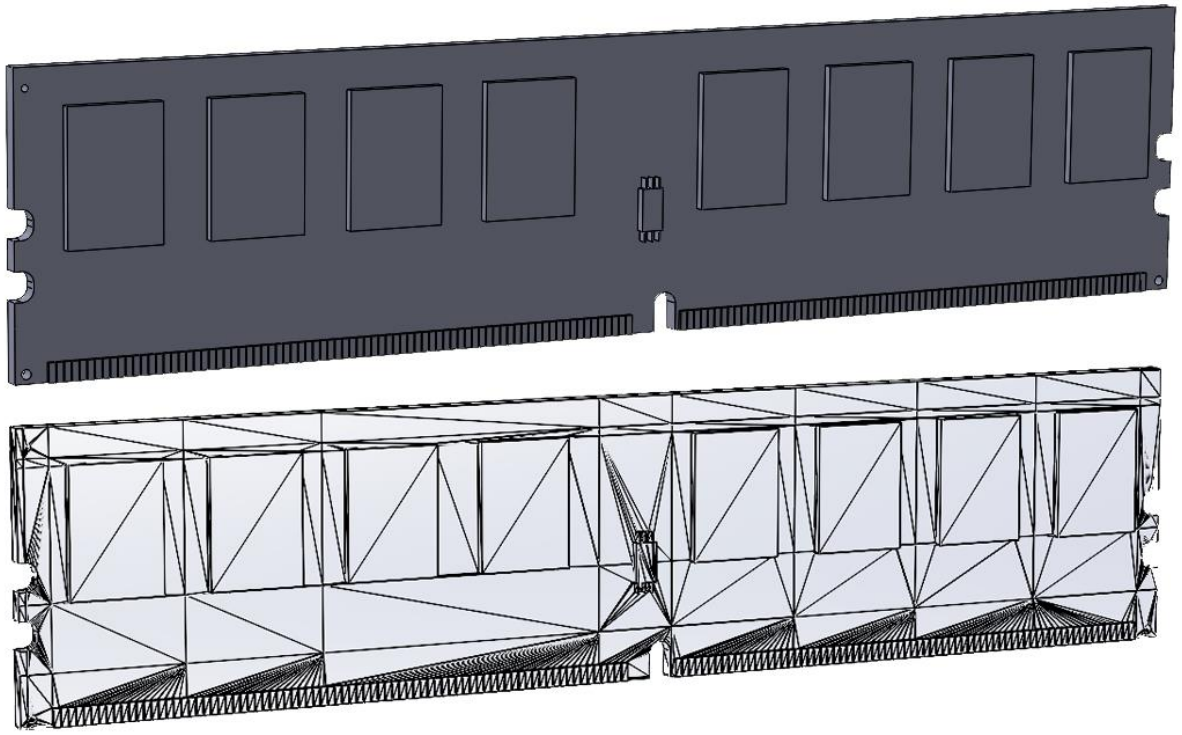


Obrázek 27 – Společný model pro grafické karty *NVidia RTX 3060 Ti* a *3050 Founders Edition*

Pro model *GeForce RTX 3090* byl postup podobný, jen byly upraveny některé prvky a rozměry, aby přibližně odpovídaly skutečnosti.

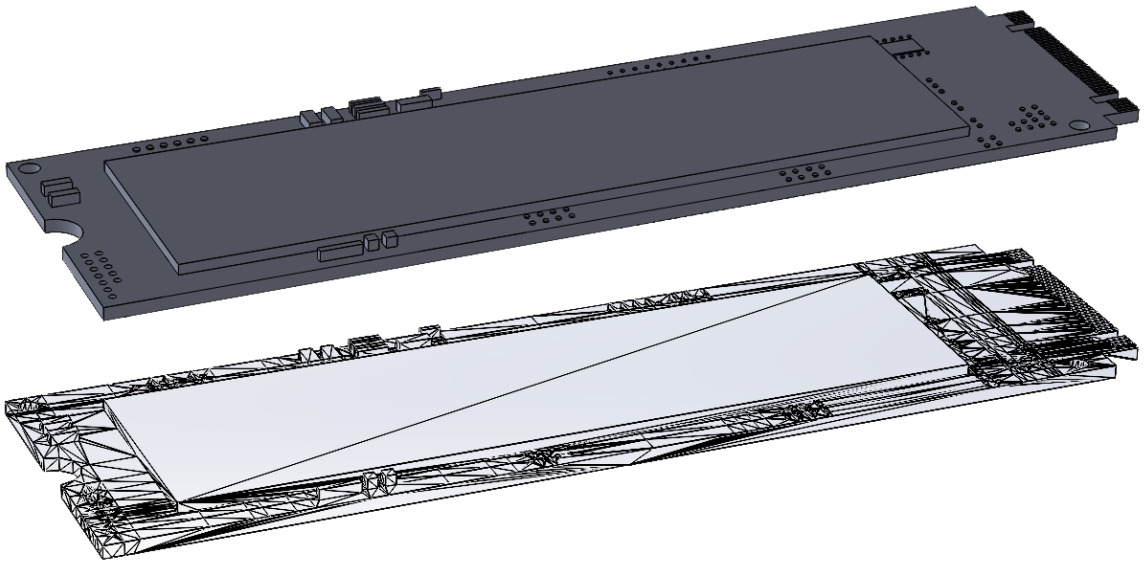
3.3.4 Další komponenty

Mezi další díly potřebné k sestavení počítače se řadí paměti RAM. V tomto projektu jsou použity moduly DDR4. Existuje mnoho výrobců, z nichž každý má vlastní unikátní design. Pro účely jednoduchého rozpoznání i nízkou komplexnost modelu byly zvoleny díly, kde je přímo viditelné PCB s připájenými paměťovými čipy.



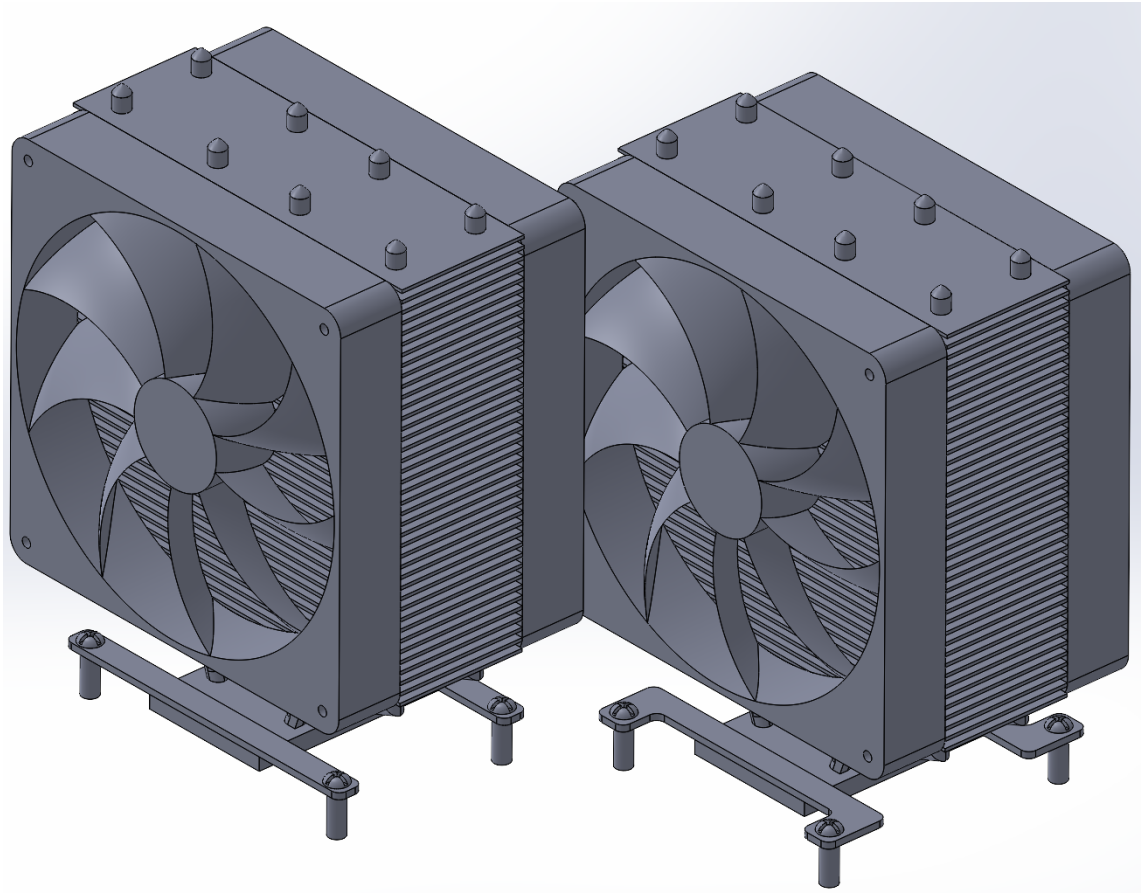
Obrázek 28 - Modul paměti RAM

Následující důležitou komponentou je interní pevný disk, který je používán pro ukládání dat v počítači. Jelikož plotnové harddisky jsou již na ústupu a nahrazují je stále častěji SSD disky, byl pro účely virtuální sestavy vybrán M.2 NVMe SSD, který má oproti klasickému SATA propojení mnohem vyšší rychlosti čtení i zápisu, ale cenově se již velmi neodlišují. První základní deska ve formátu *ATX* obsahuje dva sloty pro disk, kdežto deska formátu *mATX* poskytuje místo pro zapojení pouze jednoho. Při vytváření modelu pro SSD byly přidány podrobnější detaily jednotlivých elektronických součástí na PCB. Tento způsob tvorby zvyšuje počet geometrických tvarů, což se poté přenese i do výsledného převádění na mřížku. V tomto případě měl výsledný model po uložení do *STL* na *Hrubé* nastavení více jak polovinu počtu trojúhelníků modelu základní desky. U větších projektů by tedy mohlo být důležité dát pozor hlavně na rozměrově malé modely, jelikož tato úroveň detailu je na nich mnohdy ztracena právě kvůli jejich velikosti, ale stále je jimi zbytečně zabírána část výpočetního výkonu, tudíž tím může být zapříčiněn neplynulý běh simulace.



Obrázek 29 - Model M.2 NVMe SSD

Posledním důležitou komponentou při skládání virtuální sestavy počítače je chladič procesoru. Ty je možné rozdělit podle několika hlavních charakteristik. První z nich je způsob chlazení – jestli je teplo od procesoru odváděno vodou či vzduchem. Dále pak dle patice, na kterou je daný chladič vyroben, díky čemuž se může lišit rozteč jednotlivých děr pro uchycení, nebo podle TDP, což je maximální termální výstup, který je možné tímto chladičem odvést od komponenty.

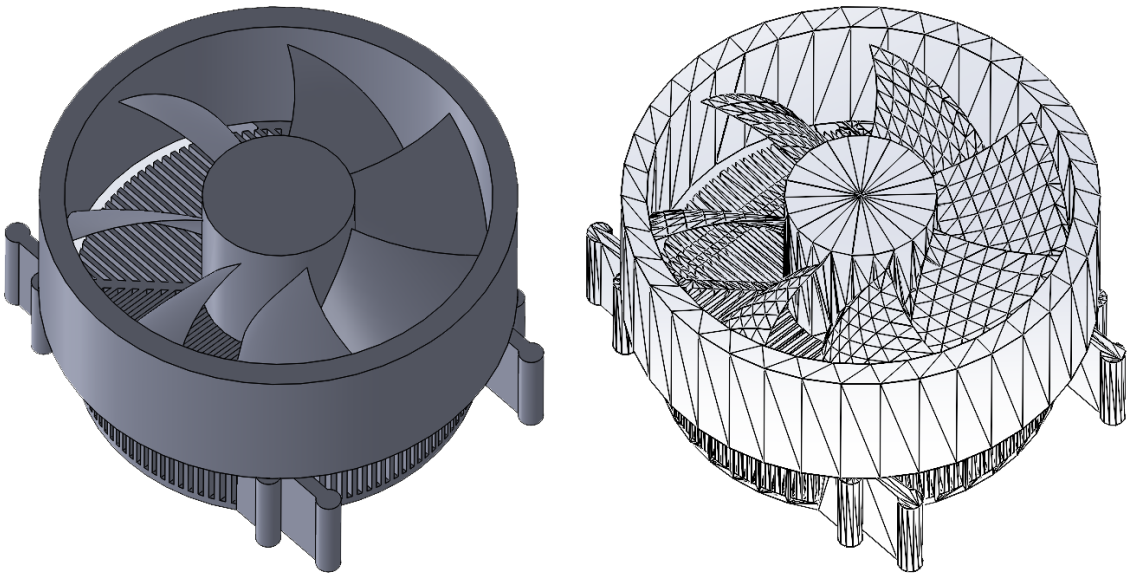


Obrázek 30 - Srovnání chladiče pro patici *AM4* (vlevo) a pro patici *LGA1200* (vpravo)

Výhodou vodních chladičů oproti vzduchovým je rychlejší a efektivnější odvod tepla od komponenty. To pak umožňuje jeho chod za nižší teploty, a zároveň menší hlasitost samotného chladiče. Tato vlastnost je využívána převážně u high-end sestav nebo při taktování, kdy je potřeba odvádět velké množství tepla, na které nemusí obyčejný vzduchový chladič vždy dostačovat. Zároveň je instalace a případná údržba vodního chlazení mnohdy náročnější ve srovnání se vzduchovým. Pro účel této práce byly vytvořeny tři modely vzduchových chladičů:

- Velký chladič pro socket *LGA 1200 (Intel)* s roztečí šroubů 75 x 75 mm a s maximálním TDP 210W
- Velký chladič pro socket *AM4 (AMD)* s roztečí 54 x 90 mm a TDP 210W
- Malý chladič pro socket *AM4* se vstupem vzduchu shora s max. TDP 95 W

Velké chladiče používají stejný model, ale obsahují modulární způsob uchycení k základní desce, který poskytuje možnost aplikace na obě zvolené patice.



Obrázek 31 - Model malého chladiče pro patici *AM4*

Za zmínku pak stojí ještě PSU neboli počítačový zdroj. Ten slouží ke zpracování vstupu z elektrické sítě a následné distribuci do jednotlivých komponent. Jelikož se rozměrově jedná o obyčejný kvádr s jedním vstupem a výstupy na kabely, byl model nahrazen přímo v unity jednoduchým tvarem, odlišeným pouze rozdílnou barvou. Zdroje použité v sestavách mají výkon 450 W pro slabší komponenty a 850 W zamýšlený pro sestavu s *GeForce RTX 3090*.

3.4 Tvorba virtuálního prostředí v UNITY

3.4.1 Iniciální nastavení virtuálního prostředí

Prvním krokem po založení projektu bylo vytvoření scény, ve které se uživatel bude pohybovat. Jako podlaha posloužil dvourozměrný objekt plochy. U něj se později ukázalo, že při jistých podmínkách je možné skrze něj protlačit některé objekty, tudíž byl vyměněn za krychli transformovanou do požadovaných rozměrů. Prvotní nastavování scény značně ulehčuje již zmíněný *XR Interaction Toolkit*, který zabezpečuje nejen podporu mnoha různých VR zařízení, ale obsahuje i velké množství nástrojů ke zjednodušení nejčastěji používaných funkcí. Mezi hlavní z nich se řadí objekt zvaný *XR Origin*, jenž usnadňuje nastavení snímání pohybu samotného VR headsetu a ovladačů uživatele.

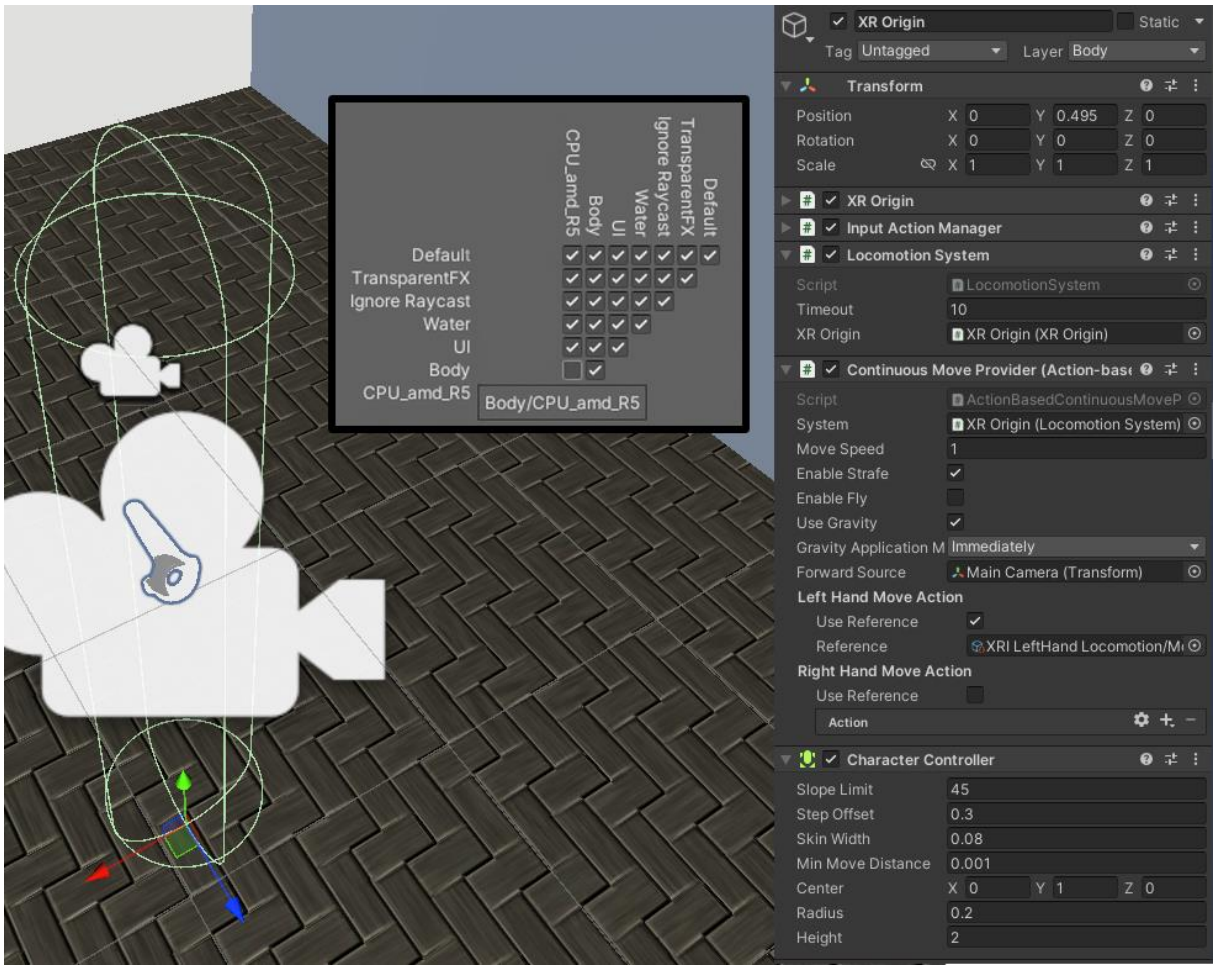
Jelikož k testování projektu se používala sestava *HTC Vive*, byl do scény vložen také model představující ovladače z této sestavy. Jeho původní pozice však neodpovídala orientaci ovladače v reálném čase, tudíž bylo dodatečně nutné upravit polohu modelu. Pro předvolený *XR Origin* je provedeno veškeré základní nastavení pro headset i ovladače, včetně prvků potřebných k interakci s prostředím, jako je úchop objektů.

Obrázek 32 - *HTC Vive* ovladače s prvky k interakci

3.4.2 Základní interakce s prostředím

Důležitým prvkem interakce s virtuálním prostředím je pohyb po scéně. Nejčastěji využívanými způsoby transportu ve virtuální realitě, mimo samotnou chůzi uživatele po místnosti, je teleportace buď pomocí pevně daných záchytných bodů, nebo dynamickým označováním místa pro teleport a dále kontinuální pohyb pomocí touchpadu či joysticku. Nejvíce uživatelsky přívětivá je teleportace, zvláště v případě rozsáhlejších scén, kdy kontinuální pohyb může způsobovat nejen nevolnost, ale zároveň je znatelně pomalejší.

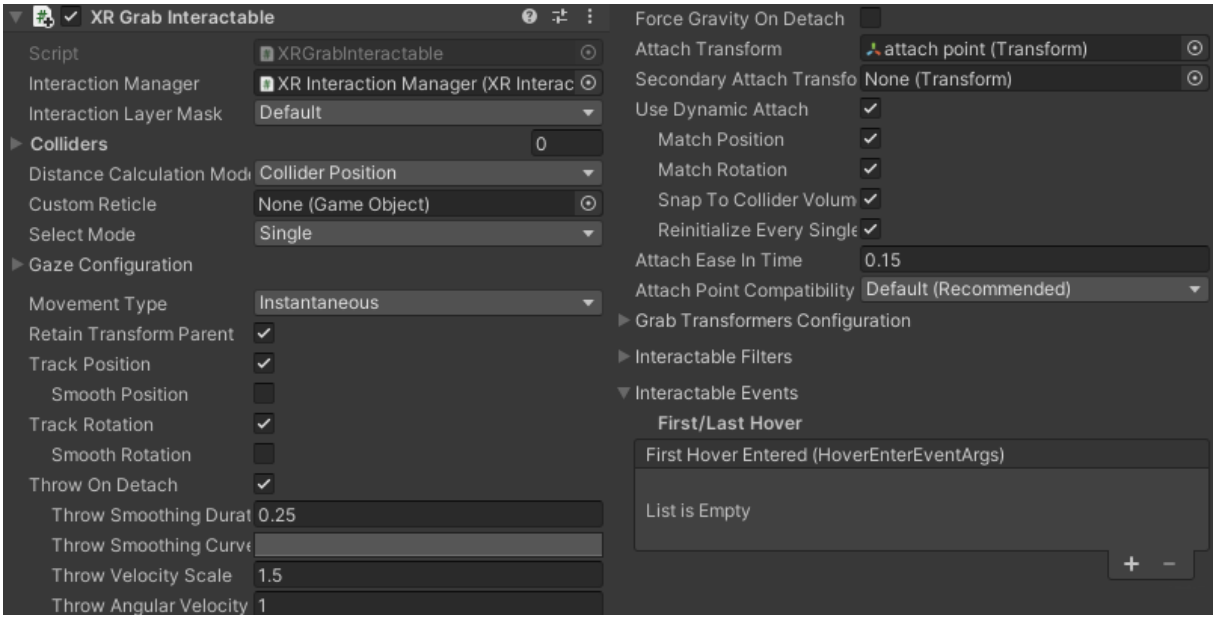
V případě tohoto projektu však bylo přistoupeno právě k volnému hýbání se po scéně, jelikož je tím zajištěna vyšší imerze do virtuální reality a výsledně vytvořená plocha má vnitřní rozměr pouze 4 x 5 m, což umožňuje rychlý přesuv i bez využití teleportace. Do *XR Originu* byl dále přidán kolizní objekt představující postavu, který zajišťoval, aby uživatel nemohl procházet objekty ve scéně. Při následujícím testování se však ukázalo, že kolize může nastat i s objekty drženými uživatelem, přičemž došlo k „vystřelení“ uživatele do vzduchu. Kvůli tomu bylo nutné navíc nastavit vrstvy kolizí mezi tělem uživatele a uchopitelnými objekty.



Obrázek 33 - Nastavení kolizí mezi jednotlivými objekty ve scéně

Pro interakce uživatele s konkrétními objekty ve scéně bylo potřeba definovat dvě vlastnosti. První z nich je objekt, pomocí kterého je nastavována požadovaná akce, zvaný *interactor*, a druhý protikus, u kterého je definována odpověď na tuto akci, označený jako *interactable*. Asset použitý pro *XR Origin* má již přednastavený *interactor* pro každý ovladač zvlášť, tudíž pro možnost uchopení jednotlivých objektů zbývá k nim nastavit *interactable*.

K uchopení objektu byla využita předdefinovaná funkce *XR Grab Interactable*. Umožňuje nastavení typu pohybu mezi *kinematickým*, *sledováním rychlosti* a *okamžitým*. Hlavní rozdíl mezi těmito třemi módy je, že pomocí prvních dvou je zajištěna kolize drženého objektu s ostatními modely ve scéně. K účelům simulace sestavování komponent byl použit okamžitý mód sledování, jelikož nabízí nižší náročnost simulace bez přidaných kolizí a zároveň zabraňuje například nechtěnému převrácení komponent uživatelem při jejich vzájemném propojování.



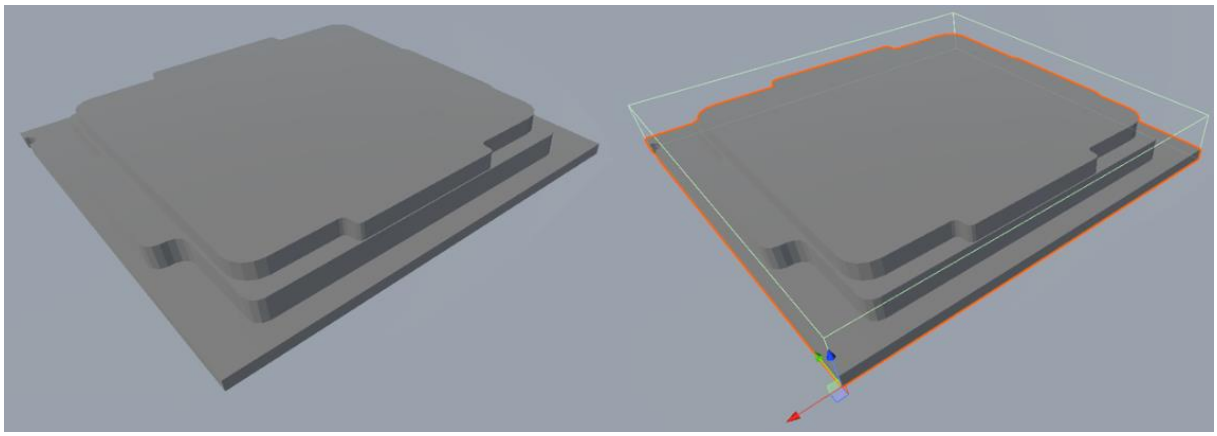
Obrázek 34 - Nastavení *interactable* pro uchopení objektu

Důležitým nastavením je také transformace objektu při jeho uchycení. To definuje, v jaké vzdálenosti a v jakém natočení bude daný objekt při zvednutí uživatelem. Pro všechny komponenty bylo povoleno dynamické uchycení, což umožňuje libovolné napolohování úchopu v simulaci pomocí ovladačů. Dále tato funkce dává možnost nastavit například sledování pozice a rotace objektu, vyhlazení jeho pohybů v prostoru a rychlost při hodu.

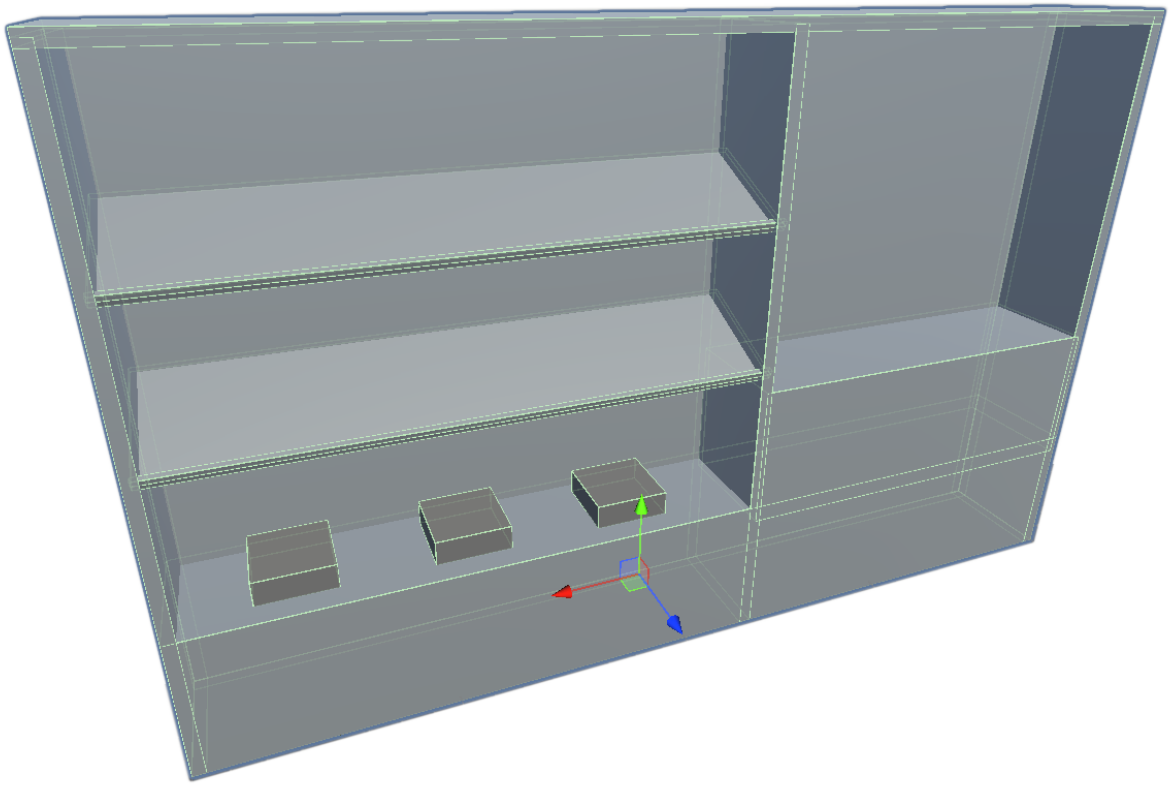
3.4.3 Sestavování komponent

Po nastavení *Grab Interactable* pro všechny komponenty bylo možné začít přiřazovat k jednotlivým komponentám další atributy. Aby se objekt choval jako fyzické těleso, bylo nutné u něj nastavit vlastnosti tuhého tělesa. K tomu slouží funkce zabudovaná přímo v Unity zvaná *Rigidbody*. Pak bylo třeba definovat kolizní model tělesa stejně jako u postavy. Bez něj by daná věc nedetekovala žádný dotek s prostředím a působením gravitace by se pouze propadla skrze podlahu. Pro objekty je možné vytvořit kolizní model buď poskládáním ze základních geometrických tvarů, nebo jej nechat vygenerovat dle vnějších hran geometrické mřížky.

U jednodušších modelů, jako CPU nebo RAM byl použit automaticky model kvádru, jelikož rozdíl v kolizích není v tomto měřítku téměř znatelný. Pro složitější modely byla mřížka automaticky vygenerována. Tento způsob není z daleka perfektní, jelikož využívá pouze vnějších hran modelu. Nelze jej tedy využít pro duté objekty, jako je například skříň na komponenty, která byla vytvořena poskládáním z jednotlivých kvádrů následně sloučených do jedné skupiny přímo v *Unity*. Skládání složitějších modelů tímto způsobem má velkou výhodu v tom, že kolizní modely přesně odpovídají těm vizuálním, tudíž nedochází k viditelnému prolínání geometrií objektů při fyzikální kolizi, ani naopak k jejich zdánlivé levitaci.

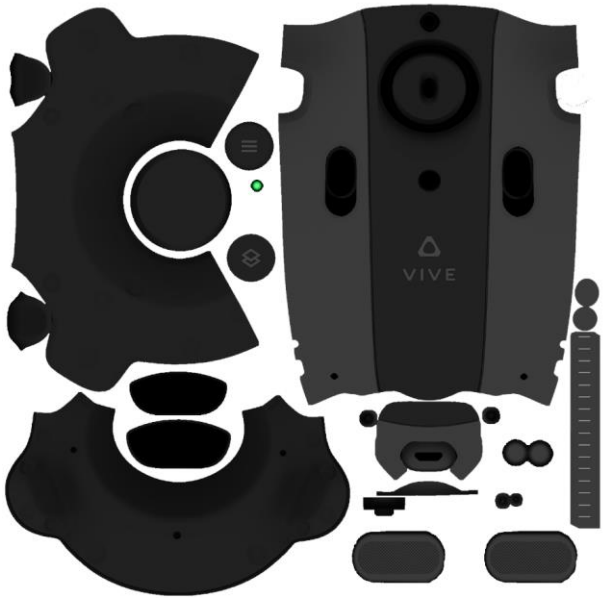


Obrázek 35 - CPU s napasovaným kolizním modelem krychle



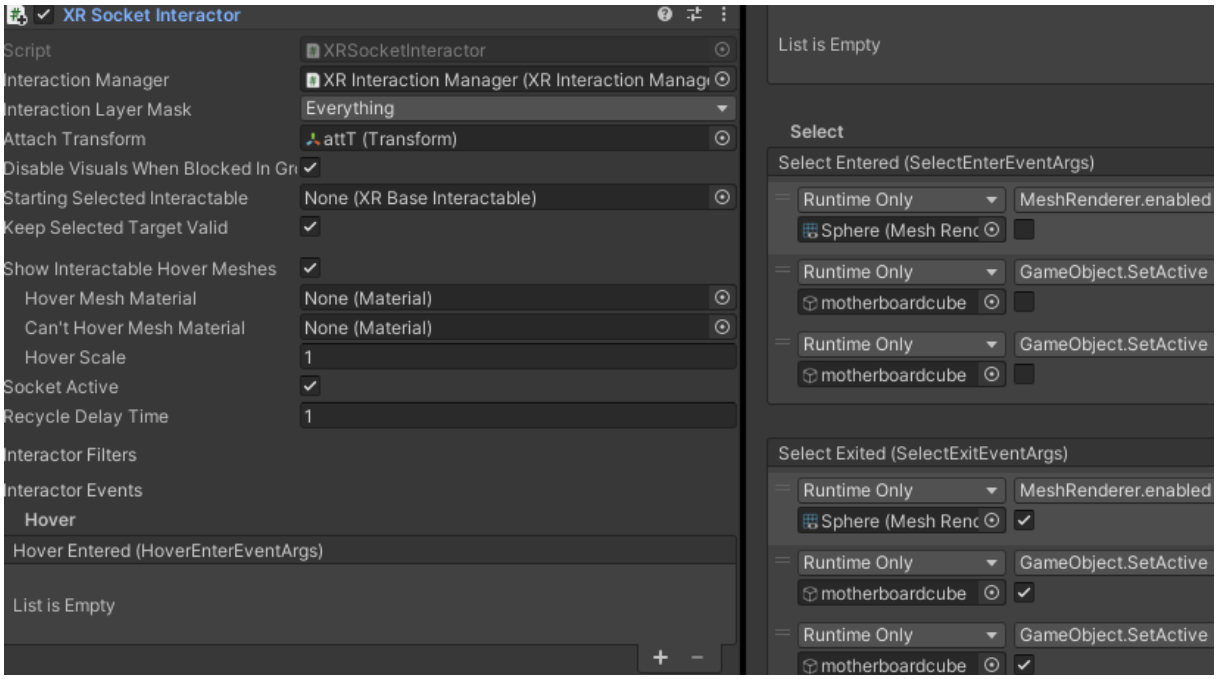
Obrázek 36 - Skříň pro uložení některých komponent – zobrazení modelu včetně obrysů pro kolize

Na každý vytvořený model byla nanesena jednoduchá textura pro zvýšení kontrastu s okolím. Jako příklad assetu s již aplikovanou texturou lze uvést importovaný *HTC Vive* ovladač viditelný na *obrázku 32*. Možným místem pro zlepšení by bylo využití podobně komplexních textur na počítačových komponentách ke zvýraznění detailů a celkovému vylepšení vizuálního dojmu scény.



Obrázek 37 - Rozvinutá textura pro model ovladače *HTC Vive*

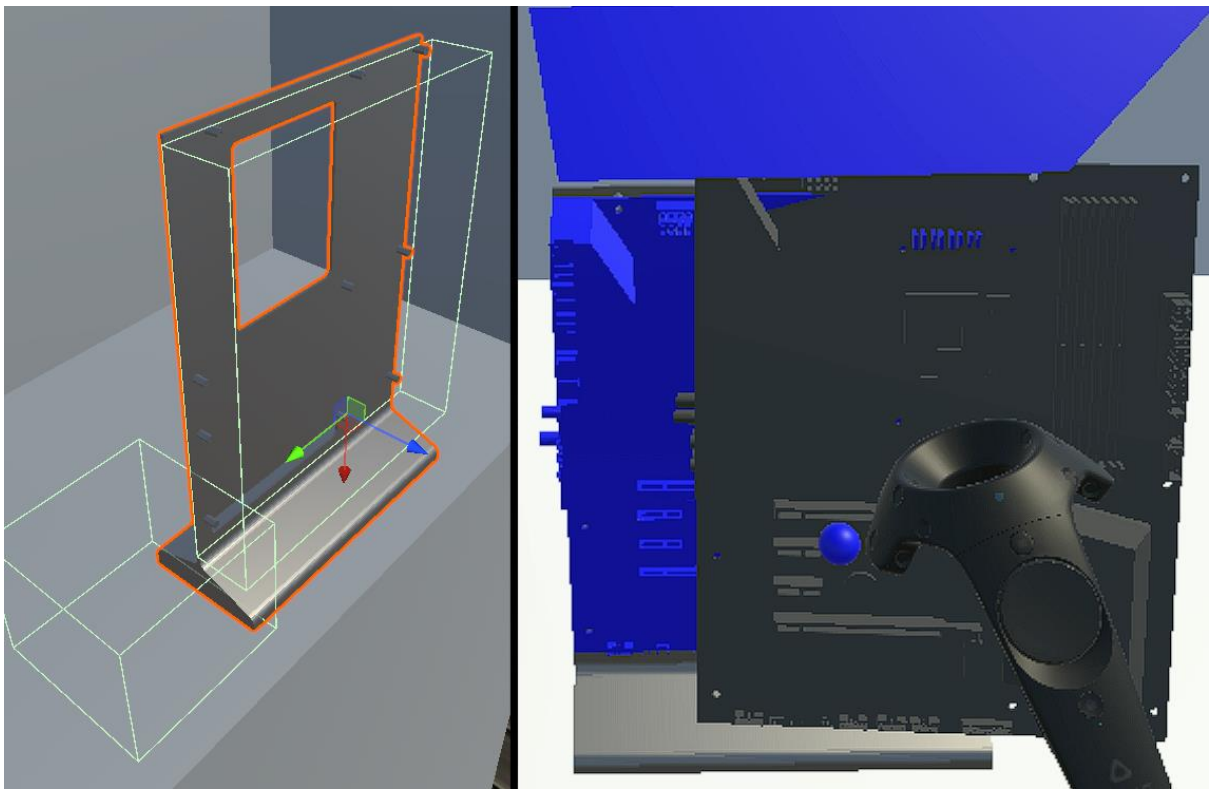
Nejdůležitější částí je samotné sestavování vložených komponent. To bylo realizováno pomocí funkce *XR Socket Interactor*, díky které šlo nastavit vzájemné propojení komponent v závislosti na přiložení objektu uživatelem pomocí ovladače k definovanému slotu. Pro každý unikátní objekt bylo nutné vytvořit vlastní socket kvůli aplikaci vlastního natočení po připojení.



Obrázek 38 - Nastavení pro *XR Socket Interactor*

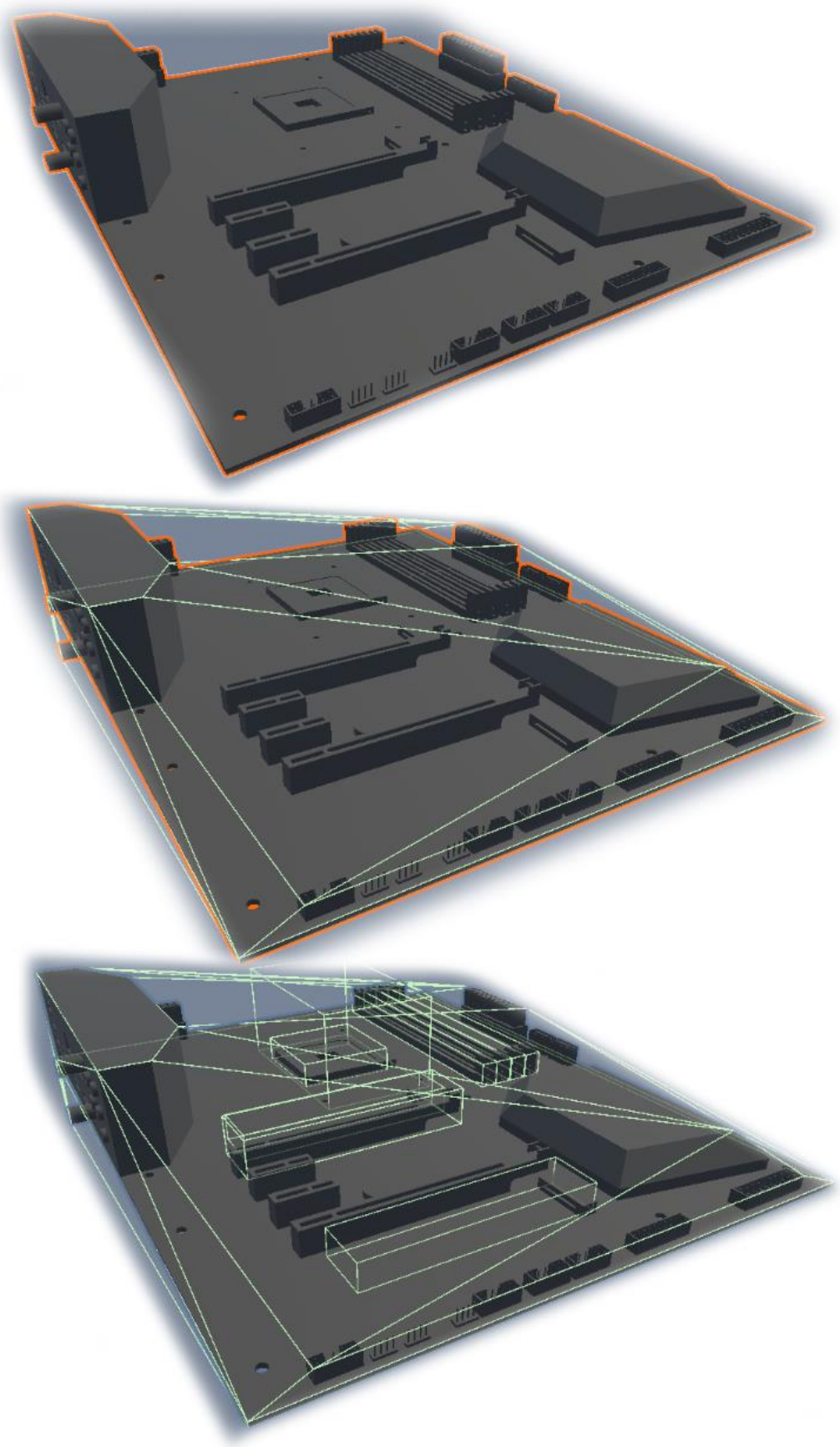
Aby bylo slot možné používat, musel se k němu přiřadit kolizní model. K tomu posloužil stejný *Box Collider*, jako je například u CPU. Jediným rozdílem bylo dodatečné zapnutí spouštěcího eventu při vniknutí tělesa do oblasti, kde dochází k detekci kolize. Když uživatel vloží správný objekt do daného prostoru, provedou se úkony umístěné v poli *Hover Entered*. Vhodné by bylo v tomto momentu zobrazit, jak by vypadalo umístění kostry modelu po připojení do daného slotu pro držený objekt. Jelikož se však jedná o často využívanou funkci, je její použití zjednodušeno automatickým nastavením již přímo v *XR Socket Interactor*.

Poté, co uživatel pustí držený objekt v blízkosti správného socketu, je model automaticky natočen podle předchozí vizualizace kostry a přesně připojen na místo. Dále jsou provedeny úkony nadefinované v poli *Select Entered*. Na *obrázku 38* je uvedeno nastavení socketu pro základní desku. Zde bylo třeba, po připojení desky, vypnout možnost vkládání dalších objektů do místa představujícího stejný slot. Dále je vložením zrušený i panel zobrazující základní informace o držené komponentě. Naopak při vyjmutí objektu ze slotu proběhne opětovné zapnutí těchto položek.



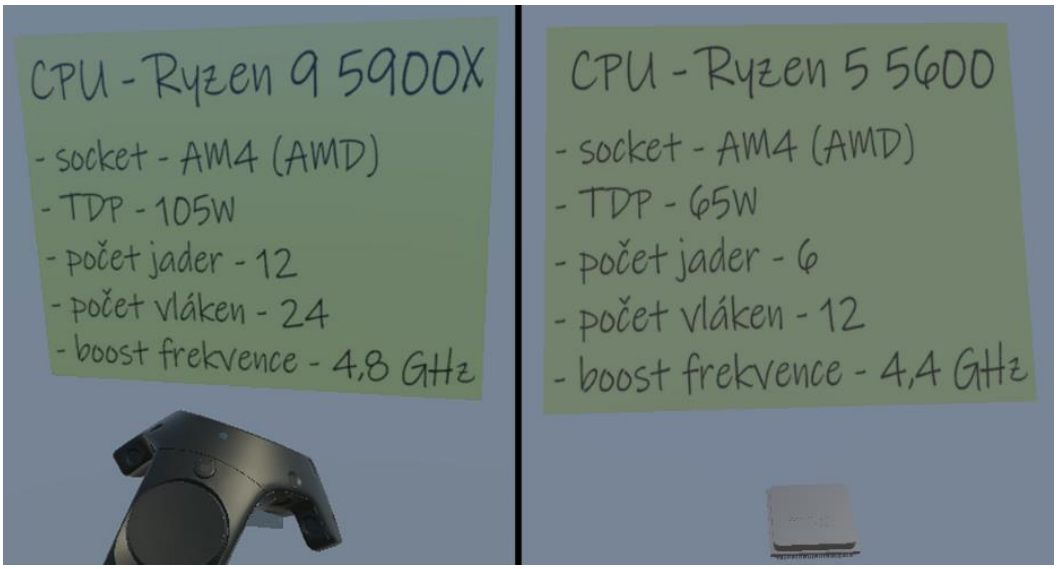
Obrázek 39 – Socket pro připojení základní desky a vizualizace objektu po vložení

Pro objekt základní desky byla obdobným způsobem vytvořena místa pro připojení ostatních komponent. Sloty pro RAM ani SSD nemají žádné speciální vlastnosti mimo vypínání již zmíněné popisové tabulky. Složitější vztahy bylo nutné nastavit pro GPU a hlavně CPU, jelikož to je použito zároveň v kombinaci s chladičem. Na obrázku 40 je náhled na objekt základní desky ve třech stádiích vývoje – nejdříve byl vložen čistý model, podle kterého byla poté vygenerována kolizní mřížka, a na závěr došlo k přidání socketů pro jednotlivé komponenty.



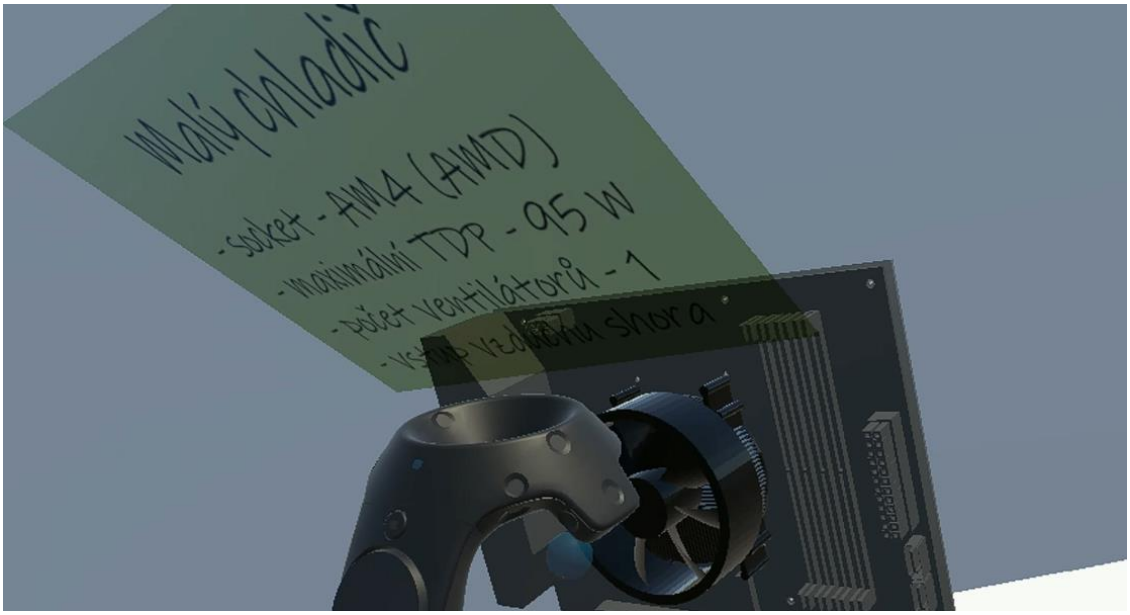
Obrázek 40 – Postup tvorby objektu základní desky

Jak již bylo zmíněno, ke každému objektu byl přidán panel s popiskem základních specifikací komponenty, umožňující jednoduchou identifikaci uživatelem, jelikož u některých komponent jsou jen nepatrné rozdíly. Jako příklad lze uvést CPU, kde použitý *AMD Ryzen 5* i *Ryzen 9* mají totožné fyzické atributy a nelze je ve virtuálním prostředí přímo z modelu rozeznat. Tato informace je zobrazena pouze během doby držení objektu uživatelem, aby bylo zamezeno nepřehlednosti ve scéně.



Obrázek 41 - Popisky pro CPU

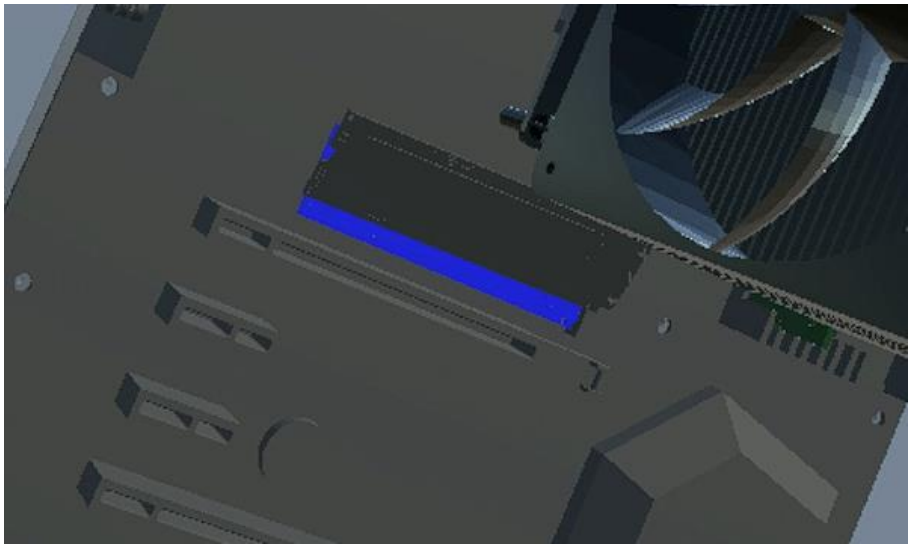
Jelikož výkonnější CPU má TDP 105 W a u malého chladiče je uvedeno maximální TDP pouze 95 W, není možné pro uživatele v simulaci použít tuto kombinaci komponent. Zároveň není možné připojit k základní desce chladič v případě, že zde není vloženo žádné CPU. Toto je patrné na *obrázku 42* na další straně, kde se při přiblížení objektu k socketu nezobrazí typický modrý obrys a zároveň při puštění komponenta spadne vlivem gravitace dolů, místo uchycení do socketu. Stejná logika je na součást aplikována v případě, že se uživatel pokusí použít chladič s jinou roztečí šroubů, nebo vložit do desky procesor s rozdílnou paticí.



Obrázek 42 - Při nesprávném použití komponenty není umožněno jeho sestavení

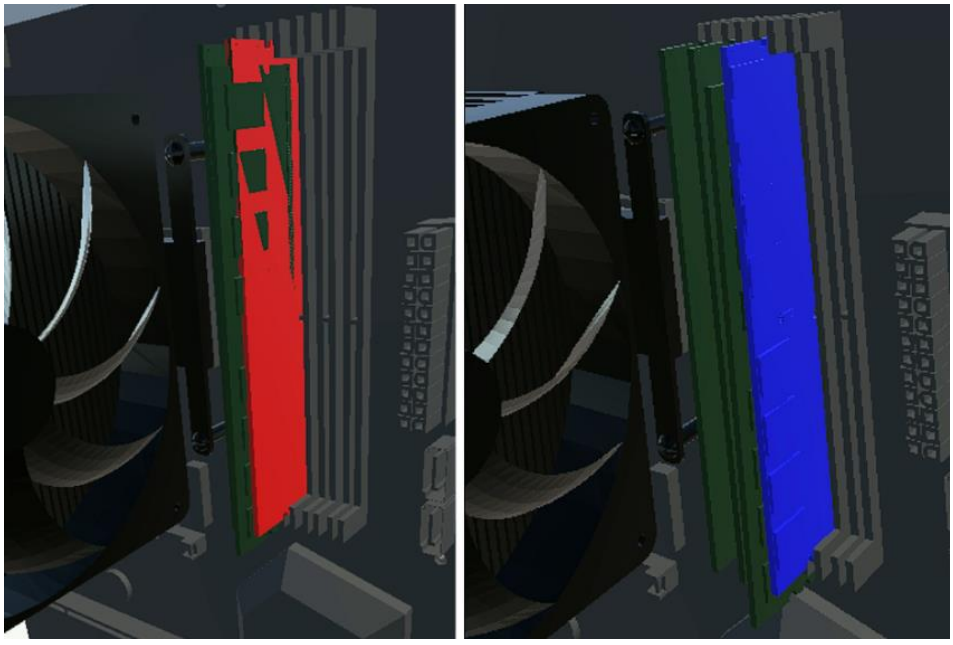
V reálné sestavě by sice použití nedostačujícího chladiče možné bylo, systém by se ale při zátěži přehříval, docházelo by k termálnímu omezení výkonu CPU a při delším provozu za těchto podmínek i ke zkrácení jeho životnosti.

Naopak může být výhodné použití velkého chladiče pro slabší procesor, což je demonstrováno u sestavy s procesorem *Intel i3-10100F* a lze realizovat i pro *Ryzen 5600*. Lepší chladič totiž umožňuje odvádět stejné množství tepla od CPU při nižších otáčkách na něm umístěného ventilátoru, čímž je zajištěn znatelně tišší chod.



Obrázek 43 - Umístění SSD do slotu

Pro SSD jsou u *ATX* formátu základní desky dostupné dva sloty a u *mATX* pouze jeden. V případě RAM se jedná o rozdíl čtyř oproti dvěma slotům. Při skládání RAM do socketů je pro uživatele zvýrazněno, zda slot, nad kterým má komponentu umístěnou, je již obsazen. Na *obrázku 44* jsou zobrazeny tyto dva stavy pro označení obsazeného a volného slotu.

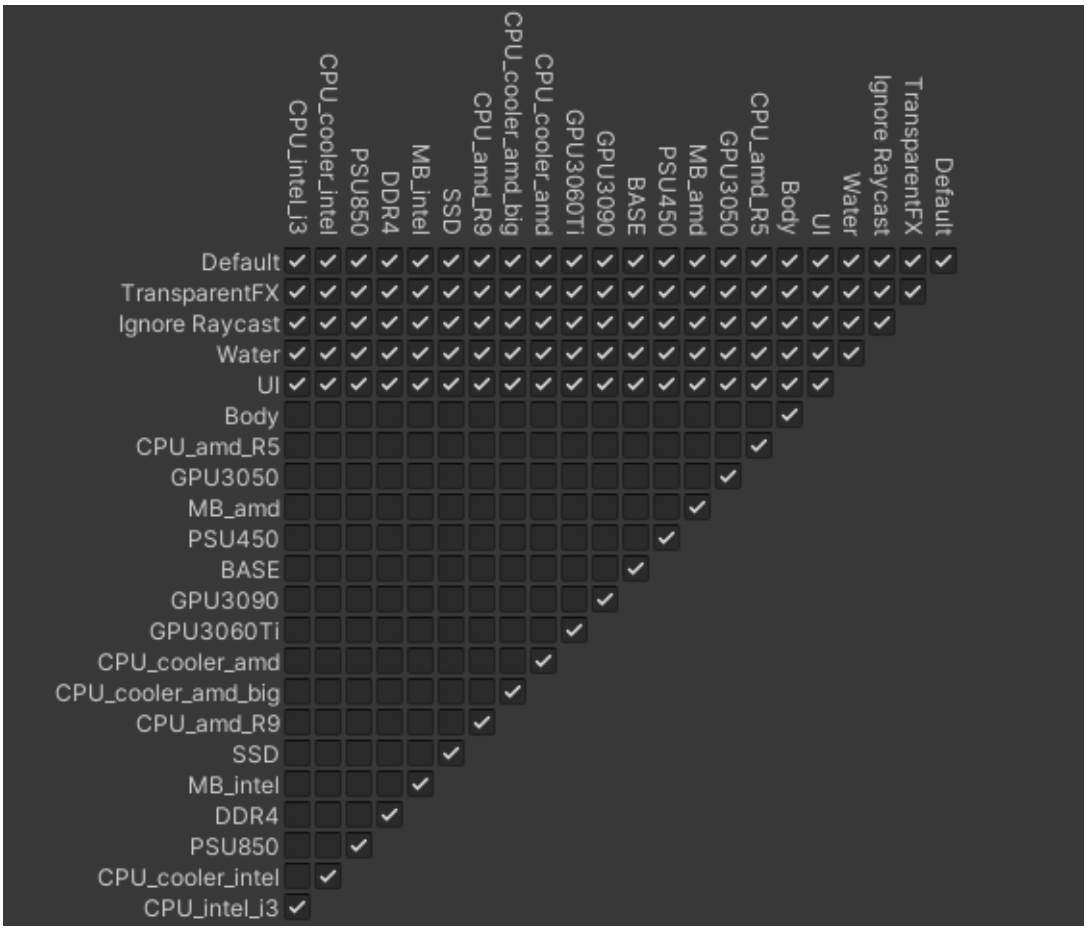


Obrázek 44 - Umístění RAM do obsazeného a volného slotu

Grafická karta bývá ve většině sestav komponentou s největším odběrem ze zdroje. V případě použité *GeForce RTX 3090* se jedná o 350 W v základu, což může být ještě zvýšeno pomocí přetaktování. Z tohoto důvodu je v simulaci také hlídáno, že v případě použití této GPU nemůže být zároveň použit slabší zdroj dimenzovaný na maximální odběr celého systému do 450 W. V reálné sestavě by bylo možné takto slabý zdroj zapojit, ale uživatel by se poté potýkal s problémy nestability systému, popřípadě by se daný počítač vůbec nezapnul, jelikož dochází k přetížení zdroje, proti čemuž je většina zdrojů ochráněna. V případě použití levného zdroje od pochybného výrobce se pak uživatel při jeho přetížení vystavuje nebezpečí vyhoření nejen samotného zdroje, ale dost často i některých připojených komponent.

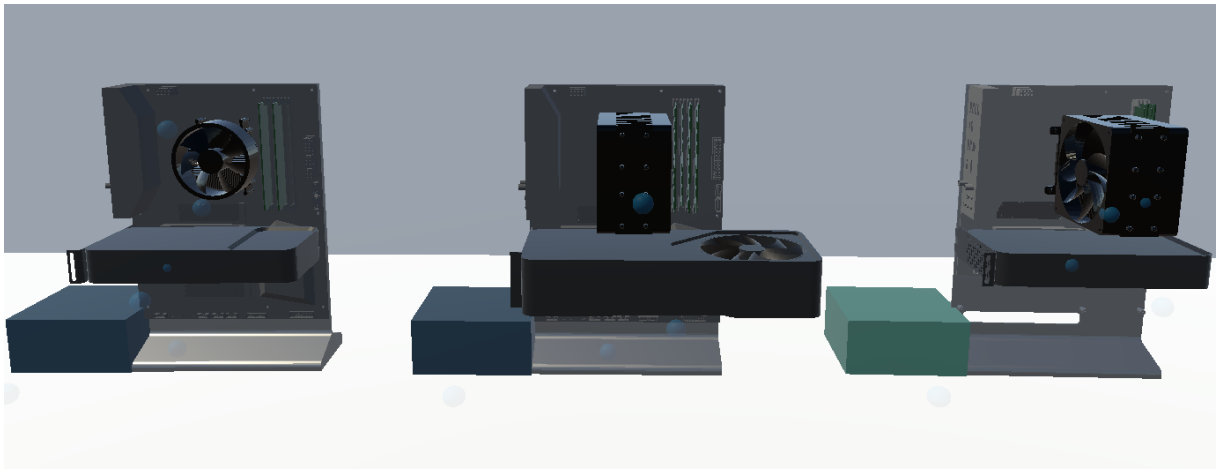
Obrázek 45 - Modely GPU s popisky

Pro rozlišení jednotlivých komponent je použita matice kolizí mezi vrstvami, která je aplikována nejen na samotné objekty, ale i na přiřazené sockety. Díky tomu je možné vložit konkrétní komponentu pouze do socketu ve stejné vrstvě.



Obrázek 46 - Matice kolizí mezi vrstvami

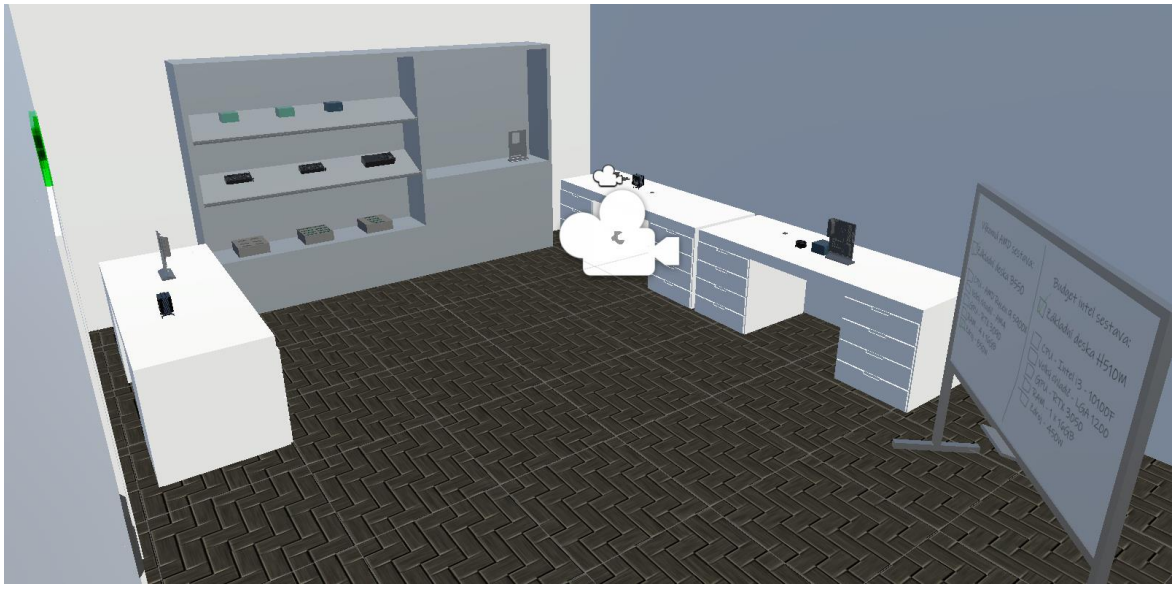
Díly ve virtuální scéně jsou v přesném počtu, což znamená, že při použití všech komponent je uživatel schopen sestavit tři počítačové sestavy se všemi potřebnými částmi. Všechny tyto modely jsou statické a lze je znovu odpojit a podle daných pravidel mezi sestavami vyměňovat.



Obrázek 47 - Jedna z možných finálních kombinací komponent

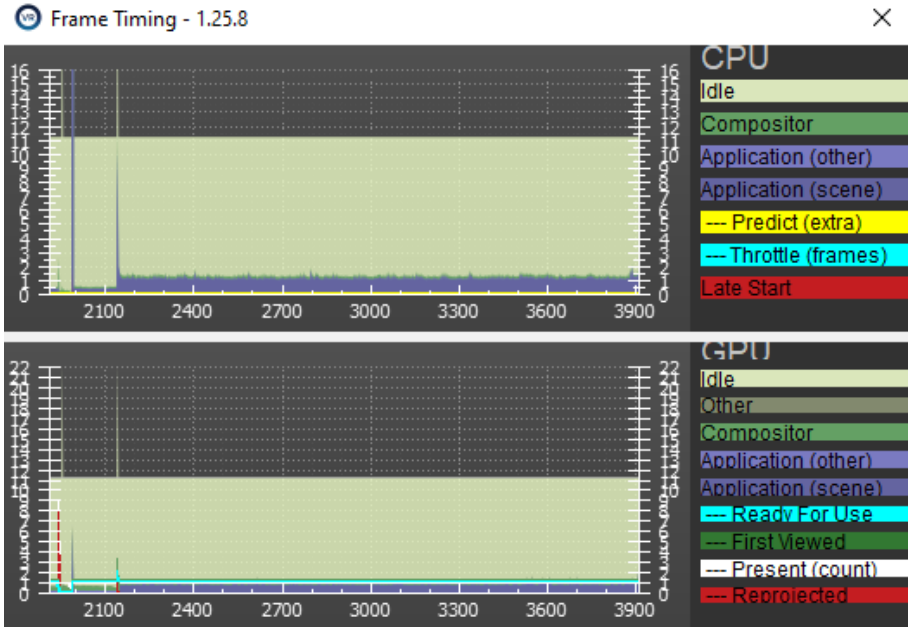
Pro uživatele je ve scéně připravena také tabule s listem komponent doporučených pro dvě možné sestavy, které jsou i na *obrázku 47*. List komponent ke všem sestavám:

- Základní deska B550	*- Základní deska B550*	*- Základní deska H510M*
- CPU AMD Ryzen 5 5900X	*- CPU AMD Ryzen 5 5600*	*- CPU Intel i3 10100F*
- CPU Tower Cooler	*- CPU Basic Cooler*	*- CPU Tower Cooler*
- GPU NVidia RTX 3090	*- GPU NVidia RTX 3060 Ti*	*- GPU NVidia RTX 3050*
- RAM 4x 16 GB DDR4	*- RAM 2x 16 GB DDR4*	*- RAM 1x 16 GB DDR4*
- SSD 2x NVMe M.2 2TB	*- SSD 1x NVMe M.2 2TB*	*- SSD 1x NVMe M.2 1TB*



Obrázek 48 - Finální virtuální scéna

V obrázku 49 je vidět čas mezi snímky při běhu scény. Světlá oblast značí základní linii 11,1 ms pro obnovovací frekvenci headsetu 90 Hz. Ostatní plochy zobrazují využití CPU a GPU, což mimo prvotní načítání scény, viditelné na začátku grafu, nepřesahuje v žádné části, značící plynulý chod programu.



Obrázek 49 - Čas mezi snímky ve scéně

4 ZÁVĚR

Cílem této diplomové práce bylo vytvořit a otestovat scénu pro skládání komponent ve virtuální realitě. Pro tento účel byly vybrány PC komponenty, jelikož mají pevně daná pravidla sestavování a koncept je jednoduše pochopitelný, díky čemuž posloužil jako vhodný pro demonstraci.

Začátek práce byl věnován vývoji virtuální reality od prvotních pokusů oklamat vnímání scény člověkem až po moderní dobu. Dále byly uvedeny různé možnosti využití VR jako výukového prostředku, včetně možných limitací. Jelikož se jedná o velice rychle se rozvíjející odvětví, některé nástroje teprve dohání dobu, co se týče nabízených možností pro tvorbu a práci s virtuálním prostředím.

V další části byly popsány jednotlivé nástroje pro tvorbu samotného virtuálního prostředí a jeho prezentace uživateli, a byly uvedeny i jejich vzájemné výhody a nevýhody. Poté byla část práce věnovaná samotné tvorbě assetů pro virtuální prostředí, od možných způsobů vytváření 3D modelů a jejich dodatečné úpravy pro scénu, včetně vypsání nejvhodnějších nástrojů pro každou část této tvorby.

V praktické části diplomové práce byl podrobně popsán způsob tvorby jednotlivých komponent pro výslednou scénu, včetně nastavení vztahů a pravidel propojování všech dílů. K tomu všemu byly popisy doprovázeny obrazovými ilustracemi ve všech použitých editorech i v různých stádiích jejich vytváření. Na konci je ukázán pohled na celou virtuální scénu s uvedením příkladu konkrétních sestav.

Počítačové sestavy uvedené v poslední části práce jsou poskládány za pomoci ovládacích prvků zařízení *HTC Vive* použitého k testování celého programu. Jelikož je celá scéna relativně malá, tak nedochází k žádným zásekům. Podle vygenerovaného grafu času mezi snímky lze určit, že pro hardware použitý při testování scény existuje ještě velká rezerva, co se týče dostupného výkonu.

Možným prostorem pro rozšíření práce by mohla být například simulace zprovoznění počítače po jeho složení, následovaná ohodnocením výkonu použitých součástí a vhodnost dané sestavy (zda například uživatel nepoužil nejsilnější GPU s nejslabším CPU, kde by ve většině případů docházelo k bottlenecku systému). Dalším možným vylepšením je grafická stránka celé scény. Bylo by možné přidat dynamické osvětlení pro všechny objekty, vytvořit detailní textury pro jednotlivé modely, nebo například rozšířit scénu o další místnost s rozdílným účelem, a tím rozšířit možnosti využití simulace.

SEZNAM POUŽITÉ LITERATURY

[1] FORD, Lily. Unlimiting the Bounds: the Panorama and the Balloon View. *Public Domain Review* [online]. 2016 [cit. 2023-05-25]. Dostupné z: https://publicdomainreview.org/essay/unlimiting-the-bounds-the-panorama-and-the-balloon-view

[2] History Of Virtual Reality. *Virtual Reality Society* [online]. 2017 [cit. 2023-05-25]. Dostupné z: https://www.vrs.org.uk/virtual-reality/history.html

[3] LOWOOD, Henry. Virtual reality: computer science. *Britannica* [online]. Updated 9 May 2023 [cit. 2023-05-25]. Dostupné z: https://www.britannica.com/technology/virtual-reality

[4] VR Training: Examples of how it's Helping Businesses in 2023. *Future Visual* [online]. Brighton, 2023 [cit. 2023-05-25]. Dostupné z: https://www.futurevisual.com/blog/vr-training/

[5] THOMPSON, Sophie. VR Applications: 23 Industries using Virtual Reality. *Virtual Speech* [online]. 1 March 2022 [cit. 2023-05-25]. Dostupné z: https://virtualspeech.com/blog/vr-applications

[6] CHECA, David a Andres BUSTILLO. Advantages and limits of virtual reality in learning processes: Briviesca in the fifteenth century [online]. B.m.: Springer Science and Business Media LLC. 15. červenec 2019. Dostupné z: https://doi.org/10.1007/s10055-019-00389-7

[7] PARTIDA, Devin. 9 Ways VR Elevates Learning & Education. *AR insider* [online]. 23 February 2023 [cit. 2023-05-25]. Dostupné z: https://arinsider.co/2023/02/23/9-ways-vr-elevates-learning-education/

[8] DOWNEY, Andrea. VR surgical simulator first to receive Royal College accreditation. *Digital Health* [online]. 24 April 2019 [cit. 2023-05-25]. Dostupné z: https://www.digitalhealth.net/2019/04/virtual-reality-training-simulator-royal-college-accreditation/

[9] BRUNNSTRÖM, Kjell, Elijs DIMA, Tahir QURESHI, Mathias JOHANSON, Mattias ANDERSSON a Mårten SJÖSTRÖM. Latency impact on Quality of Experience in a virtual reality simulator for remote control of machines [online]. B.m.: Elsevier BV. listopad 2020. Dostupné z: https://doi.org/10.1016/j.image.2020.116005

[10] BALAS, Fotis. What Is CPU Bottleneck In PC Gaming And How To Avoid It. *PC Steps* [online]. 30 November 2018 [cit. 2023-05-25]. Dostupné z: https://www.pcsteps.com/21732-what-is-cpu-bottleneck-in-pc-gaming-and-how-to-avoid-it/

[11] EGGERT, Doug. What is foveated rendering. *Tobii* [online]. 2022 [cit. 2023-05-25]. Dostupné z: https://www.tobii.com/blog/what-is-foveated-rendering

[12] GAUTAM, Skashi. Virtual Reality Development: Strategies for Effective Planning and Execution. *Idea Usher: Ushering in innovation* [online]. USA, 2023 [cit. 2023-05-25]. Dostupné z: https://ideausher.com/blog/virtual-reality-software-development/

[13] What Makes Unity So Popular in Game Development? *Arnia Software* [online]. Bucuresti, 14 Sep 2022 [cit. 2023-05-25]. Dostupné z: https://www.arnia.com/what-makes-unity-so-popular-in-game-development/

[14] TITOV, Andrii. Why is the Unity Engine considered the best: Pros and cons. *Stepico: Step your game up* [online]. Limassol (Cyprus), 1 Jul 2022 [cit. 2023-05-25]. Dostupné z: https://stepico.com/blog/why-is-unity-the-best-game-engine-pros-and-cons/

[15] MOZOLEVSKA, Victoria. Best game engines of 2022: Pros, cons, and top picks for different types of games. *Kevuru games* [online]. Delaware, 28 Jul 2022 [cit. 2023-05-25]. Dostupné z: https://kevurugames.com/blog/best-game-engines-2022-pros-cons-and-top-picks-for-different-types-of-games/

[16] L. GARBETT, Samuel. Unreal vs. Unity for VR Game Development: Which Is Best? *Make Use Of* [online]. 4 Jun 2022 [cit. 2023-05-25]. Dostupné z: https://www.makeuseof.com/unreal-vs-unity-vr-best-game-development/

[17] GAJSEK, Dejan. Unity vs Unreal Engine for XR Development: Which One Is Better. *Circuit Stream* [online]. 12 Aug 2022 [cit. 2023-05-25]. Dostupné z: https://circuitstream.com/blog/unity-vs-unreal

[18] *Unity Documentation* [online]. Unity Technologies, 2021 [cit. 2023-05-25]. Dostupné z: https://docs.unity3d.com/

[19] *Unity: Real-time tools and more* [online]. Unity Technologies, 2021 [cit. 2023-05-25]. Dostupné z: https://unity.com/

[20] *Unreal Engine Documentation: Real-time tools and more* [online]. Epic Games, 2023 [cit. 2023-05-25]. Dostupné z: https://docs.unrealengine.com/

[21] *Unity Asset Store* [online]. Unity Technologies, 2023 [cit. 2023-05-25]. Dostupné z: https://assetstore.unity.com/

[22] *Unreal Engine: Products* [online]. Epic Games, 2023 [cit. 2023-05-25]. Dostupné z: https://www.unrealengine.com/marketplace

[23] MAYWELL, Wayne. Does Unreal Engines 5 Nanite Make Poly Count Irrelevant. *CG Obsession* [online]. 29 Jun 2021 [cit. 2023-05-25]. Dostupné z: https://cgobsession.com/does-unreal-engines-5-nanite-make-poly-count-irrelevant/

[24] WIRTZ, Bryan. Unreal vs CryEngine: Which Engine Should You Use? Pros, Cons, and the Final Verdict. *Game Designing* [online]. 7 May 2023 [cit. 2023-05-25]. Dostupné z: https://www.gamedesigning.org/engines/cry-vs-unreal/

[25] How to Perform 3D Modeling in Unity with No Drop in Quality. *Game Ace: Creative Studio* [online]. 6 Oct 2022 [cit. 2023-05-25]. Dostupné z: https://game-ace.com/blog/3d-modeling-in-unity/

[26] RATNER, Peter. *3-D Human Modeling and Animation*. Second Edition. United States of America: Wiley, 2003. ISBN 0-471-21548-1.

[27] THILAKANATHAN, Danan. Box Modeling: The 3D Modeling Technique. *Thilakanathan Studios* [online]. 5 Oct 2016 [cit. 2023-05-25]. Dostupné z: https://thilakanathanstudios.com/2016/10/box-modeling-the-3d-modeling-technique/

[28] ZUZA, Mikolas. 3D sculpting: Modeling characters and organic shapes for 3D printing. *Prusa Research: by Josef Prusa* [online]. 30 Apr 2020 [cit. 2023-05-25]. Dostupné z: https://blog.prusa3d.com/3d-sculpting-modeling-characters-and-organic-shapes_31705/

[28] DE LA FLOR, Mike and Bridgette MONGEON. Digital sculpting with mudbox: Essential tools and techniques for artists. *Burlington: Focal Press*, c2010. ISBN 978-0-240-81203-8.

[30] Procedural Modeling: a brief introduction. *Datagen* [online]. New York, 11 Sep 2020 [cit. 2023-05-25]. Dostupné z: https://datagen.tech/blog/procedural-modeling/

[31] GETTO, Roman, Arjan KUIJPER a Dieter W. FELLNER. *Automatic procedural model generation for 3D object variation* [online]. B.m.: *Springer Science and Business Media LLC*. 20. srpen 2018. Dostupné z: doi:10.1007/s00371-018-1589-4

[32] PETTY, Josh. What is Houdini & What Does It Do? *Concept Art Empire* [fd]. dsaffs, 2023 [cit. 2023-05-25]. Dostupné z: https://datagen.tech/blog/procedural-modeling/

[33] ZHOU, Ryle. Maya vs 3ds Max vs Houdini vs Cinema 4D vs Blender. *Medium* [online]. 18 Oct 2018 [cit. 2023-05-25]. Dostupné z: https://medium.com/codex/maya-vs-3ds-max-vs-houdini-vs-cinema-4d-vs-blender-76d173696f2a

[34] *Blender* [online]. Blender Foundation, 2023 [cit. 2023-05-25]. Dostupné z: https://www.blender.org/

[35] CALVELLO, Mara. What Is UV Mapping: How It Makes 3D Models Come to Life. *G2* [online]. Chicago, 27 Apr 2022 [cit. 2023-05-25]. Dostupné z: https://www.g2.com/articles/uv-mapping

[36] Differences between Displacement, Bump and Normal Maps. *Pluralsight* [online]. 2 Nov 2022 [cit. 2023-05-25]. Dostupné z: https://www.pluralsight.com/blog/film-games/bump-normal-and-displacement-maps

[37] *DriveWorks Pro 20* [online]. 2023 [cit. 2023-05-25]. Dostupné z: https://docs.driveworkspro.com/topic/DriveWorks3DAppearances

[38] SANTOS, Bruno, Nelson RODRIGUES, Pedro COSTA a António COELHO. Integration of CAD Models into Game Engines [online]. B.m.: SCITEPRESS – Science and Technology Publications. 2021. Dostupné z: doi:10.5220/0010201701530160

[39] GRIFFIN, Melanie. Creo vs SolidWorks: The Differences. *All3DP: All About 3D Printing & Additive Manufacturing* [online]. 31 May 2020 [cit. 2023-05-25]. Dostupné z: https://all3dp.com/2/solidworks-vs-creo-cad-software-compared/

[40] TOMMASO DE PAOLIS, Lucio a Patrick BOURDOT. Augmented Reality, Virtual Reality, and Computer Graphics: Proceedings, Part I [online]. 5th International Conference. Otranto, Italy: Springer Cham, 2018 [cit. 2023-05-22]. ISBN 978-3-319-95270-3. Dostupné z: https://doi.org/10.1007/978-3-319-95270-3

[41] *Vive* [online]. HTC Corporation, 2023 [cit. 2023-05-25]. Dostupné z: https://www.vive.com/

SEZNAM ZKRATEK

VR – Virtual Reality

2D – Two-Dimensional

3D – Three-Dimensional

GPU – Graphics Processing Unit

CPU – Central Processing Unit

RAM – Random Access Memory

HDD – Hard Disk Drive

SSD – Solid-State Drive

PCB – Printed Circuit Board

PCI – Peripheral Component Interconnect

ATX – Advanced Technology Extended

mATX – Micro ATX

CAD – Computer-Aided Design

TDP – Thermal Design Power