**Jan Frídl 4.A**

**Smíchovská Střední Průmyslová Škola a Gymnázium 2021-2022**

Maturitní práce

**Materiály pro budoucí**

**herní projekty**

**Autor:** Jan Frídl **Oponent:** Adam Přibyl **Vedoucí Práce:** Marie Souchová

# Zadání

# Hodnotící list

# Čestné prohlášení

Prohlašuji, že jsem maturitní práci vypracoval samostatně pod vedením Mgr. Marie Souchové. V práci jsem použil informační zdroje uvedené v seznamu použité literatury a internetových stránek.

Praha ……………………

Podpis celým jménem

# Anotace

# Obsah

# Úvod

Když jsem si vybíral téma pro svoji maturitní práci, byl jsem si prakticky jistý oblastí práce, tedy herním vývojem, ale nemohl jsem přijít na to, co přesně bych chtěl udělat. Po nějaké době přemýšlení mne napadlo, že bych si mohl ulehčit svojí budoucnost a vytvořit si nějaké základy pro mé budoucí projekty. Při přesnějším formulování toho, čeho bych chtěl v rámci maturitní práce, se spojila moje dlouholetá záliba v tanky a obrněná vozidla a fascinace fyzickými simulacemi a rozhodl jsem se že vytvořím modulární systém pro tvorbu pásového podvozku pro tank spolu se základním ovládacím systémem. Tento systém by umožnil relativně jednoduché vygenerování podvozku pro téměř jakýkoliv typ tanku či obrněného vozidla a také by umožnil jeho následnou kompletní simulaci v reálném čase, přesněji tedy simulaci jednotlivých článků pásu a jejich poškození, odpružení podvozku a pohon tanku.

Během minulého roku jsem si dokázal osvojit základy herního vývoje a jaké různé nečekané problémy se mohou během práce objevit. S vytvářením simulací jsem ale neměl příliš velké zkušenosti, a tak na mne čekalo prozkoumání obrovského pole a nalezení nespočetně frustrujících a často zdánlivě nepochopitelných problémů. Věřil jsem však, že díky mým dlouhodobým technickým zkušenostem budu schopen svého cíle dosáhnout a otázkou zůstavalo pouze kolik času mi tato práce zabere. Jak se ukázalo, bylo ho hodně.

# Analýza maturitní práce

## Popis úkolu

Mým cílem pro maturitní práci je vytvořit interaktivní 3D prostředí s nádechem muzea ve kterém si hráč bude moci prohlédnout a vyzkoušet mechaniky vyvinuté pro mojí budoucí hru zaměřující se na stavbu a vývoj tanků. Prostředím se hráč bude moct volně pohybovat a u většiny mechanik se mu dostane mého komentáře ohledně důvodu existence určité mechaniky, vývoje a také různých nedostatků. Finální produkt bude tedy cílen spíše na další vývojáře, či na hráče, které zajímá, jak hry fungují za jejich zády.

## Popis stávajícího stavu

Hlavním zaměřením této práce bude realistická simulace tankových pásu v reálném čase a jejich dynamické vytvoření. Přestože se na internetu dá nalézt mnoho příkladů právě těchto simulací, neexistuje příliš mnoho materiálu, který se zabývá jejich tvorbou, většinou se dají nalézt pouze hotové produkty k zakoupení, či videa ukazující výslednou simulaci. K dosažení tohoto cíle budu tudíž muset využít hlavně svých zkušeností s vývojem her kterých jsem nabil minulý rok při tvorbě studentského projektu a také svých programovacích zkušeností, kterých mám díky různému experimentování větší počet. Zároveň také budu využívat různých vývojářských fór či tutoriálů abych pochopil či vymyslel jednotlivé součásti simulace a vyrobil nástroj který mi umožní jednoduše vytvořit funkční pásy dle zadaných požadavků.

## Výběr vhodných prostředků pro řešení úkolu

Pro splnění mého cíle budu potřebovat několik druhů programů. Budu potřebovat software který mi umožní sledovat můj postup. Dále budu potřebovat program který mi umožní vytvářet modely pro mé mechaniky a samozřejmě prostředí ve kterém tyto mechaniky budu moci vytvářet a programovat. Také musím vyřešit problém přenositelnosti souborů, jelikož hodlám pracovat na více zařízení.

Management software

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Program | Dostupnost | Funkce | Znalost | Součet | Pořadí |
| **Trello** | **10** | **7** | **8** | **25** | **1** |
| **Asana** | **10** | **8** | **0** | **18** | **2** |
| **Hubstaff** | **6** | **10** | **0** | **16** | **3** |

Bodování 1-10 (10 nejlepší)

Zde volba nebyla příliš složitá, a to i přesto, že na trhu existuje nespočetně aplikací tohoto druhu. S Trellem mám již delší zkušenosti, a tak nad ostatními funkcionálně lepšími alternativami lehce zvítězilo.

3D Engine

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Engine | Dostupnost | Funkce | Znalost | Komunita | Součet | Pořadí |
| **Unity 3D** | **10** | **8** | **9** | **9** | **36** | **1** |
| **Unreal Engine** | **10** | **9** | **0** | **6** | **25** | **2** |

Bodování 1-10 (10 nejlepší)

Pro svůj herní engine jsem opět vybral Unity, jelikož jsem v s ním pracoval předešlý rok a také jelikož má o mnoho více výukového materiálu a podpory (vyjádřeno jako „Komunita“)

## Výběr řešení a výstupů

Jelikož je můj původní výkonný počítač neopravitelně poškozen a díky čipové krizi nemám k dispozici dostatečné finance na nákupu náhrady, nemám příliš na výběr při řešení práce. Finální produkt bude muset být úsporný a efektivní, jelikož bych na něm jinak nemohl pracovat. Toto bude hlavní limitující faktor. Výstupy poté budou ve formě „buildnuté“ aplikace spolu s dokumentem, ve kterém budou předvedeny vývojářské nástroje nepřístupné hráči v aplikaci a také přehled přidaných věcí.

## Stanovení dílčích úkolů

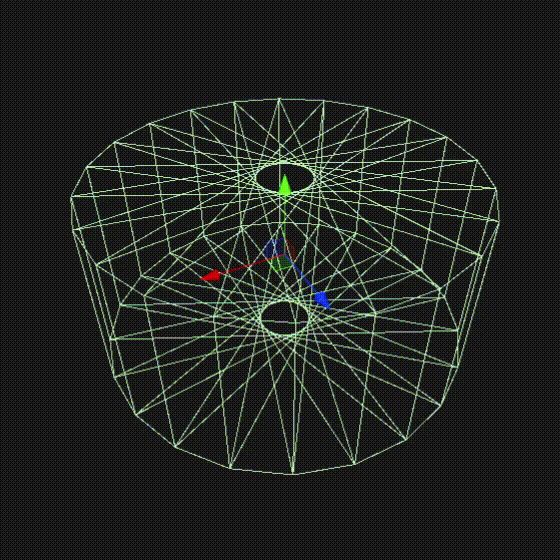
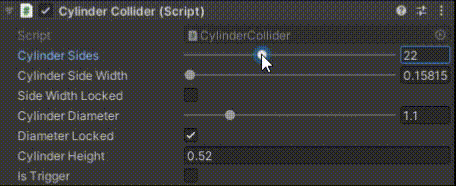
1. Vymyšlení všech realizovatelných mechanik
2. Naprogramování mechanik
3. Návrh a tvorba finálního prostředí
4. Návrh a tvorba grafiky mechanik
5. Tvorba doprovodných komentářů
6. Upravování a vylepšování práce
7. Dokončení práce

# Popis vlastního řešení práce

Když jsem poprvé začal dělat na své maturitní práci, došlo mi že jsem již velmi dlouho nevytvářel žádné 3D modely, a že jsem nikdy předtím nevytvářel modely určené pro hru. Toto byl pro mne problem, jelikož modely určené pro herní prostředí bývají narozdíl od ostatních lépe optimalizované a při jejich tvorbě se dbá na správnou strukturu topologie. Jelikož jsem v té době neměl žádný výkonný hardware, musely být mé modely velmi optimalizované a tak jsem se rozhodl naučit se a využít několik technik které by měly snížit grafickou zátěž mého projektu, což by mi umožnilo využít více výkonu pro část simulační. Základní technikou byl tzv. low-poly styl modelů. V tomto stylu jsou modely záměrně tvořeny vzhledově hrubé a často vypadají jako by pocházely z 20 let starých her. Aby toho nebylo málo rozhodl jsem se take využít absolutní minimum textur a používání jedné textury pro vícero objektů, díky čemuž se do paměti vždy načítalo několik velkých textur které v ní mohly zůstat po delší dobu namísto neustálého načítání a uvolňování mnoho textur pro různé objekty.

Po obnovení a nabití nových modelovacích zkušeností bylo na čase začít pracovat na technické stránce. Ze všeho nejdřív jsem si chtěl vytvořit hráčskou postavu a pohybový systém, pomocí kterého by tato postava šla ovládat. Tato část nebyla sama o sobě příliš důležitá pro vytvoření simulovaných tanků, ale byla důležitá z hlediska vývojového, jelikož jsem pro zprovoznění svého pohybového systému musel vytvořit několik nástrojů které jsem využíval po zbytek své maturitní práce. Také mi práce na tomto pohybovém systému umožnila se znovu zoritentovat v herním programování, jelikož byla více podobná zadáním, které jsem zpracovával v minulosti.

Jeden z velmi důležitých nástrojů který jsem během programování pohybového systému vytvořil byl Cylinder Collider, neboli generátor válcových kolizí. Tento nástroj jsem musel vytvořit, jelikož herní engine který jsem používal ke svojí práci v základu neobsahoval podobný nástroj kvůli složité rovnováze mezi přesností a výpočetní složitostí výsledného produktu. Čirou náhodou jsem základy pro podobný systém vyvinul minulý rok při práci na studentském projektu, a tak jsem se rozhodl tyto základy znovu prostudovat a využít je ve svém novém nástroji.

Původně jsem se pokoušel vymyslet si vlastní vzorce které by mi umožnily rychlejší generování kolizního válce. Jak se ale ukázalo, mé vzorce byly nepřesné při generování zjednodušených válců které měly velmi omezený počet stran pláště, a stávali se přesnými až při vytváření objektu který se blížil svojí přesností k válci. Zde ale nebylo limitujícím faktorem řešení matematických vzorců, ale samotné generování mnoho částí ze kterých se výsledný válec skládal, a tak jsem od svých vzorců upustil a využil jsem opravdových, ozkoušených matematických rovnic. Po tomto rozhodnutí se již žádný velký problém neobjevil a nástroj se mi povedlo dokončit. Tento nástroj mi umožnil stanovit si počet stěn válce a velikost na jeho různých osách a následně dle mých požadavků válec vždy bez problémů vygeneroval. S tímto výsledkem jsem byl velice spokojen a tak nastal čas na to posunout se dál v mojí práci.

Jelikož jsem původně čekal že mi vytvoření takovéhoto systému zabere většinu výstupu, byl jsem velice překvapen když jsem ho dodělal během týdne a tak mi zbylo hodně času na vytvoření čeho chci.

Na mém novém nástroji, a na kolizích celkově, mne iritovala jedna věc – během hraní nešly rychle a jednoduše zobrazit. Abych je viděl, musel jsem se vždy přepnout do editovacího módu, což ale znamenalo že nemohu ovládat svojí postavu. Rozhodl jsem se tedy že vytvořím systém který mi umožní tyto kolize vidět v jakékoliv situaci. Byl to poněkud ambiciózní plán, ale jelikož jsem v tu dobu měl hodně času tak jsem neviděl důvod proč se o něj nepokusit.

Začal jsem tím že jsem si pro každý neviditelný kolizní objekt vygeneroval jeho viditelný protějšek, který sám kolize neměl. Následně jsem musel pouze nastavit identické pozice a všechny nově vytvořené protějšky přidat do seznamu. Toto mi později umožnilo přepínat mezi nově vygenerovanými protějšky a normálními viditelnými objekty.

I když mi fungovalo přepínání objektů zbývalo překonat další problém, který se vyskytnul. Můj generátor válců vytvářel objekty za běhu hry, ale seznam pro objekty s kolizemi se vytvářel pouze při spuštění. To způsobilo nesrovnalosti a nefunkčnost mého nového systému. Musel jsem tedy vytvořit další systém dedikovaný k přijímání a předávání nových událostí. Tento systém přijímal události od mého generátoru válců a předával je dál k mému vizualizačnímu systému, který díky nim mohl aktualizovat seznam když byl nějaký objekt vytvořen či smazán.

# Závěr