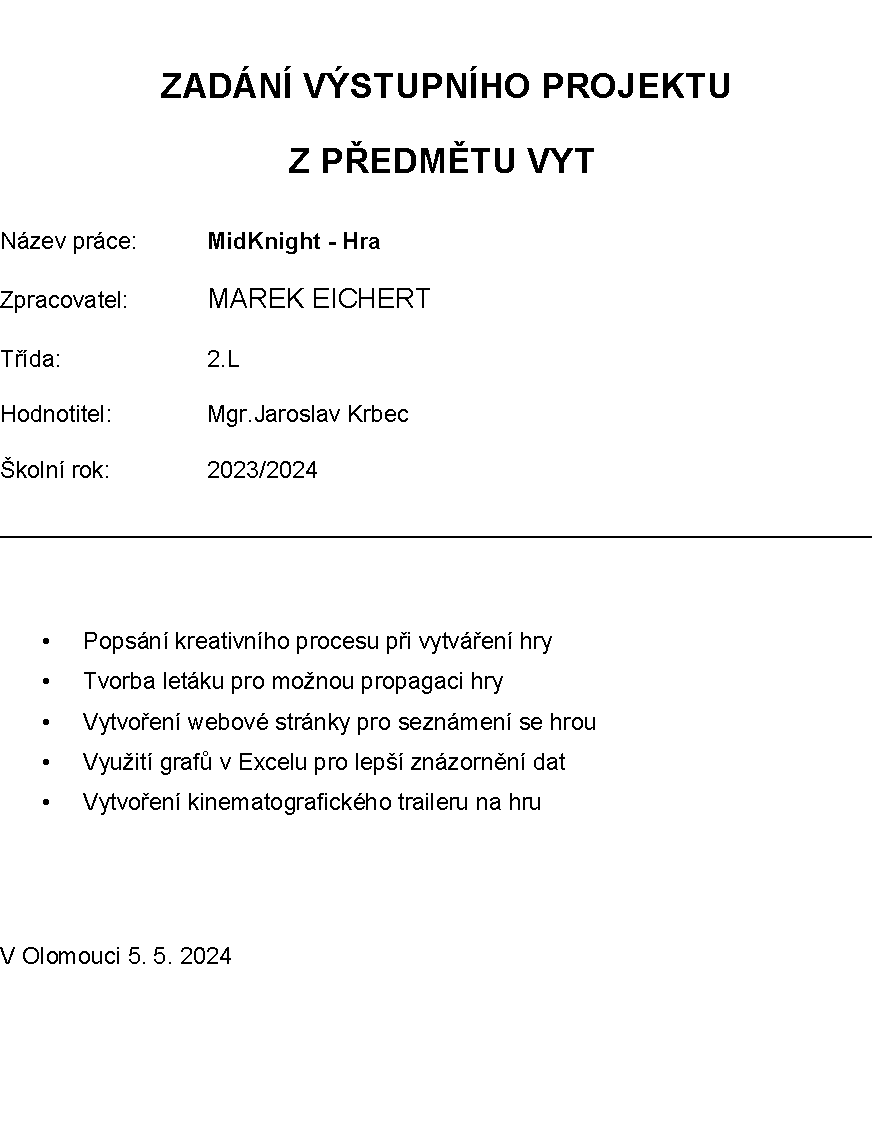
|  |
| --- |
| SPSEOL |
| MidKnight |
| Výstupní projekt z VYT |

|  |
| --- |
| Eichert Marek  01.06.2024 |

2024



# Obsah

[Obsah 2](#_Toc169016835)

[Úvod 3](#_Toc169016836)

[1. Godot 4](#_Toc169016837)

[1.1 Proč právě Godot? 4](#_Toc169016838)

[2. MidKnight 5](#_Toc169016839)

[2.1 Brainstorming hry 5](#_Toc169016840)

[2.2 Vizuální stránka hry 5](#_Toc169016841)

[2.3 Kód hry 7](#_Toc169016842)

[Excel 9](#_Toc169016843)

[Závěr 10](#_Toc169016844)

[Seznam použité literatury 11](#_Toc169016845)

[Seznam obrázků a tabulek 12](#_Toc169016846)

# Úvod

Hlavním cílem bylo vytvořit hru nazvanou MidKnight, kde hráč ovládá rytíře, který se octl na hradě, co není tak opuštěný, jak se zprvu zdálo. Touto dokumentací vám chci přiblížit vývoj indie her v Godot enginu. Co vše vývoj obnáší a s čím jsem se vypořádal. Porovnám výhody a nevýhody Godotu s jeho konkurencí. Popíšu praktické kroky při vývoji hry. Kde jsem hledal inspiraci. Jak jsem vytvářel návrhy a později je realizoval. Popíšu proces tvorby animací v aplikaci Aseprite. Popíšu funkce a způsoby co mi tvorbu animací zjednodušil. Stručně Popíšu, jak funguje programování v Godotu. Jak jsem využil jeho scénový systém a hierarchie. Vysvětlím, jak jsem přidával nové nepřátele, jejich funkce a způsob jakým fungují.

## Godot

Godot je plně zdarma open-source herní engine. Umožňuje tvorbu 2D a 3D her v jednom prostředí. Hlavním skriptovacím jazykem je GDScript, podobný Pythonu, ale podporuje také C#, VisualScript a částečně C++.

Godot využívá scénový systém, kde každá herní entita je považována za scénu, což usnadňuje upravování a zlepšuje přehled. Vývojové prostředí je přívětivé a podobné Unity. Je rozšiřitelné pomocí pluginů.

### Proč právě Godot?

Godot nemá tolik funkcí jako Unity, ale má nižší technické nároky na používání, což mi neumožnilo pohodlně používat Unity. V porovnání s ostatními enginy nabízí tu nejlepší kombinaci funkcí, hardwarových nároků a složitosti právě pro tento projekt. Další podstatný důvod je můj kamarád, který v Godotu již pracoval a s tvorbou hry mi nesmírně pomohl.

#### Výhody

Velkou z výhod je možnost používání více jazyků. Godot jich podporuje několik. GDScript speciálně navržen pro Godot. C#, C++ a VisualScript. To umožňuje vývojářům vybrat si jazyk, který jim nejlépe vyhovuje pro jejich konkrétní potřeby a zkušenosti.

Godot podporuje vývoj her pro různé platformy, včetně Windows, macOS, Linux, iOS, Android a HTML5. Exportování pro různé platformy je velmi lehké a intuitivní.

#### Nevýhody

Problémy s výkonem se začnou ukazovat u velmi rozsáhlých projektů, nebo u větších projektů s 3D prostředím. Multiplayer je podstatně složitější zakomponovat do rozpracovaného projektu. Výukových materiálů a zdrojů je taky méně a podstatně mi to ztížilo vývoj hry. Komunita stále roste, ale nedá se srovnávat třeba s Unity, nebo Unreal Engine.

## MidKnight

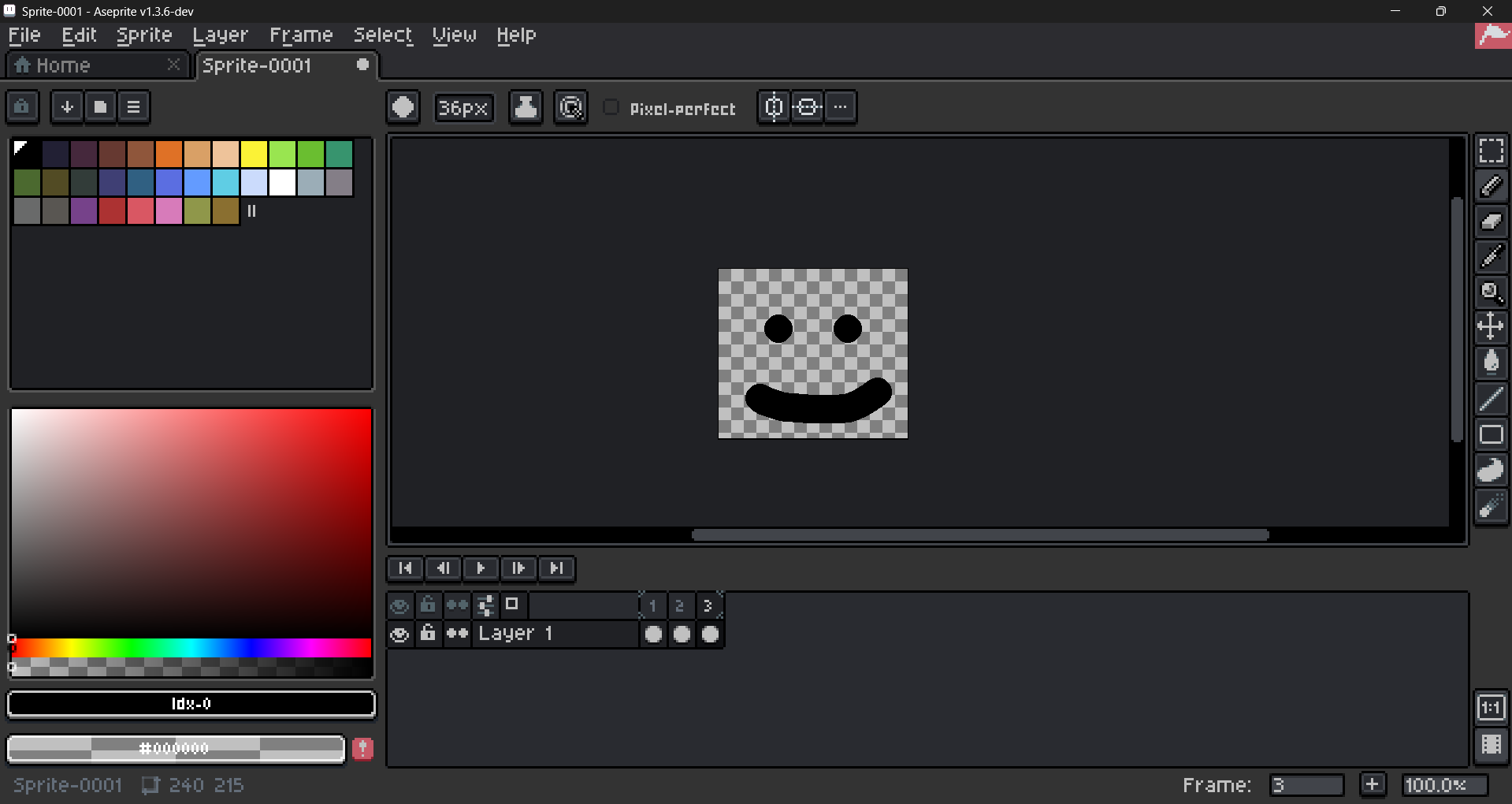
MidKnight je náročná survival hra ve které hrajete za rytíře, co úplně nezapadá do místa, kde se nachází. Čelí zrádným nepřátelům několikrát větší než on sám. Jeho úkolem je přežít a snad se dostat z hradu pryč. Svým silnějším a větších oponentů se rovná pomocí jeho schopnosti se pohybovat. Platformování a taktického využití jeho schopností je nutné k poražení jeho nepřátel. S každou vlnou se zvyšuje počet a síla jeho nepřátel, které musí porazit.

### Brainstorming hry

Základní myšlenku na hru jsem měl už nějakou dobu, ale hra se postupem vývoje se částečně měnila. S novými nápady se z původně pomalejší a taktické hry stávala hra více chaotická a rychlá. Poté jsem se rozhodl zakomponovat nové pohybové prvky, aby tento nápad více rozvinul. Double jump, dash a wall jump přidaly mnohem více možností, jak se s nepřáteli vypořádat. S těmito novými komponenty jsem upravil i samotnou mapu, aby více seděla k rychlejšímu a vertikálnímu stylu hraní.

### Vizuální stránka hry

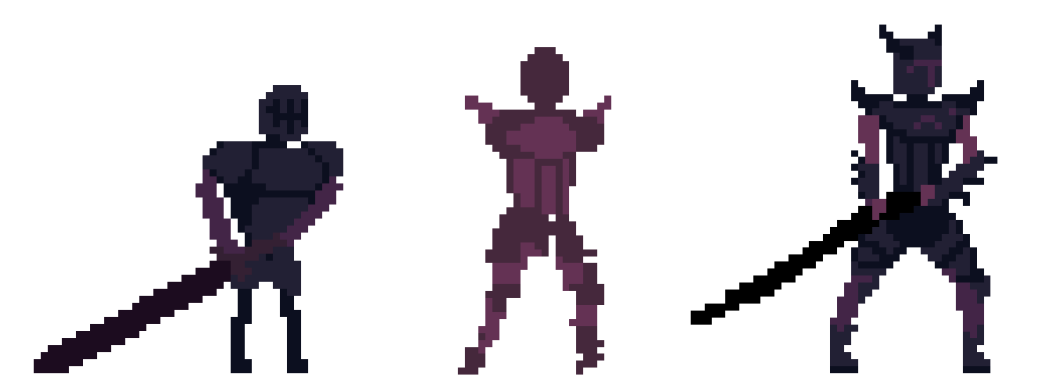
Pro všechny použité assety jsem použil pixel art aplikaci Aseprite. Aseprite nabízí širokou škálu nástrojů pro kreslení, barvení a úpravu pixel artu. Přehledné uživatelské rozhraní usnadňuje práci v editoru. Funkce snímků a časové osy usnadňují tvorbu a úpravy animací. Onion skin je funkce, která umožňuje vidět předchozí snímek v animaci v menší sytosti. Tato funkce zlepšuje kvalitu a plynulost mezi snímky animace.



Obrázek 1 – Aseprite UI

Zdroj: Vlastní výroba

#### Sprite design

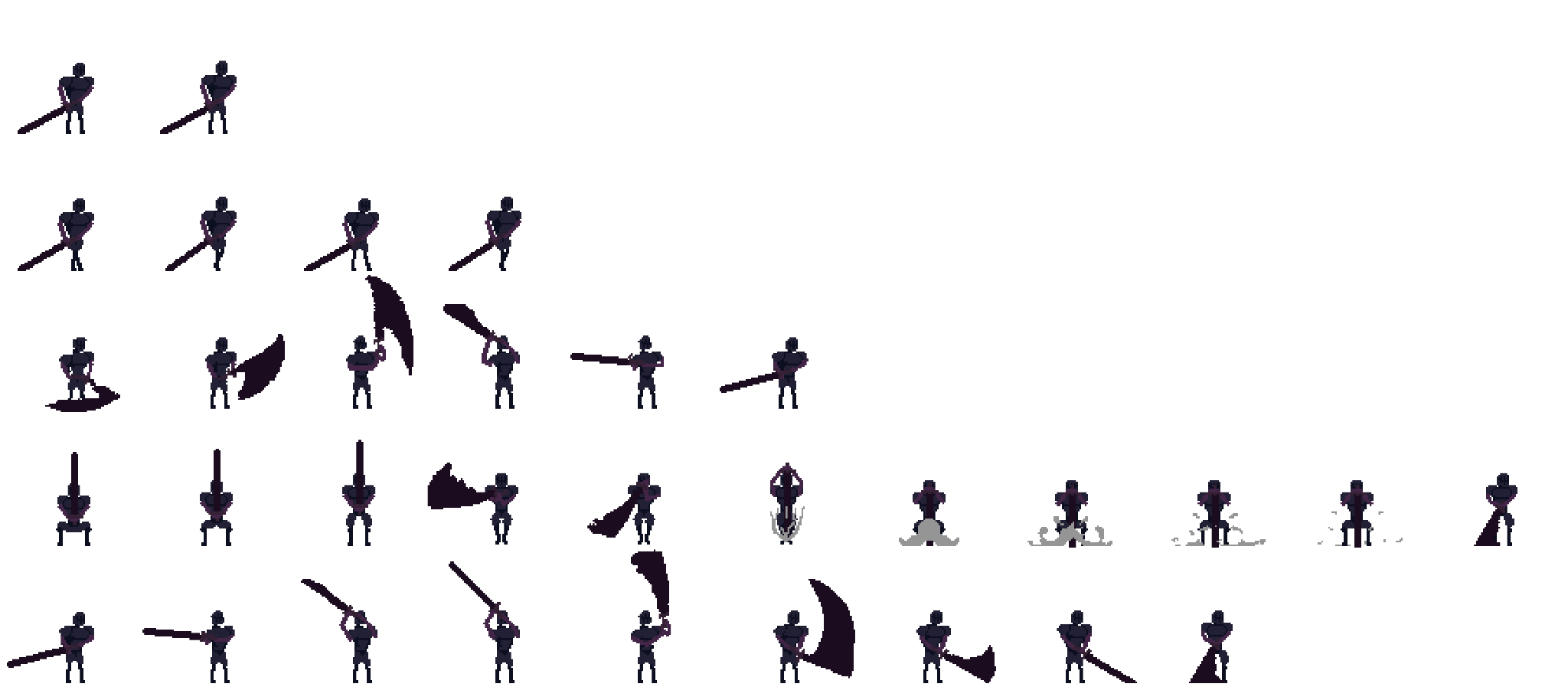
Inspiraci na nepřátele jsem hledal na Pinterestu. Zde jsem si našel hned několik návrhů. Složku s nimi jsem následně procházel a vybíral ty zajímavější designy. Na papír jsem kreslil náčrtky s možnými designy, které jsem následně překreslil do aplikace Aseprite. Vybíral jsem vždy ten design, co vypadal v pixel art stylu nejlépe

Obrázek 2 – koncept na rytíře



#### Animace

Animace určitě patřili k těm, z časově náročným aktivitám. Manuální tvorba animací je poměrně časově náročná a bez pořádné předchozí zkušenosti mi trvalo opravdu dlouho, než jsem byl spokojen s jejich kvalitou. Samotné animace jsem tvořil podle předloh, které jsem na internetu našel. Ne vždy jsem našel vhodné šablony a tak jsem musel improvizovat.

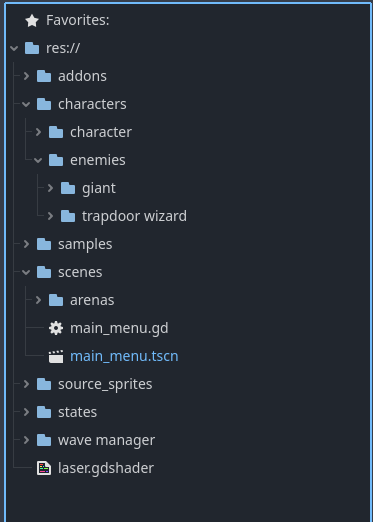


Zdroj: Vlastní výroba

Obrázek 3 – ukázka exportu animace

### Kód hry

Scéna (Scene): je vizuální stránka herního objektu. Scéna může obsahovat různé prvky (nodes), které definují jeho strukturu a chování. Například, scéna pro hráčskou postavu může obsahovat uzly jako sprite (pro zobrazení grafiky), collider (pro detekci kolizí) a další.

Skript (Script): Tento soubor obsahuje kód, který řídí chování herního objektu. Skript může být napsán v jazyce GDScript nebo v jiných podporovaných jazycích jako je C#. Pro zachování správné funkčnosti kódu je důležité jak je uspořádaná hierarchie kódu.

Obrázek 4 – ukázka hierarchie

Zdroj: Vlastní výroba

#### State machine

State machine (stavový automat) je velmi užitečný nástroj, který jsem použil k ovládání stavů herních objektů a postav. Tento koncept je obzvlášť užitečný pro implementaci herní logiky, která vyžaduje přepínání mezi různými stavy. Například, postava ve hře může mít různé stavy jako je "idle" (nečinný), "walking" (chůze), "running" (běh), "jumping" (skákání) a podobně. Právě tento state machine mi dovolil zakomponovat složitější funkce nepřátel, bez kompromitace na čitelnosti. Využil jsem pro Skok rytíře (Knight), nebo projektil od Čaroděje (Wizzard).

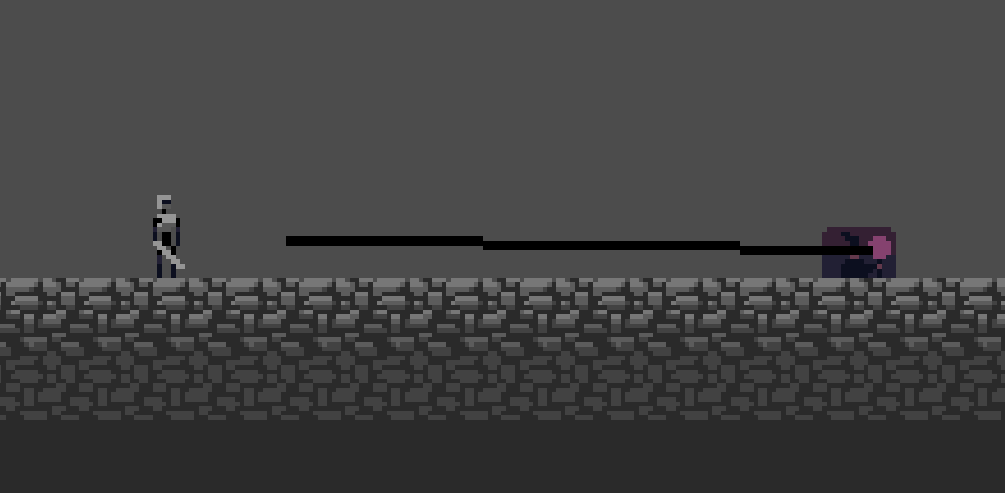
#### Knight

#### Snímek obrazovky 2024-06-08 175819.pngRytíř má dva typy útoků: normální útok, který používá, když je hráč blízko, a výskok, když je hráč daleko. Po výskoku se spustí náhodný časovač, který určuje délku prodlevy před dopadem. Během pádu má rytíř kolem sebe kolizní zónu, která působí hráči poškození. Po dopadu se tato kolizní zóna zvětší, a pokud se v ní hráč nachází, udělí mu poškození.

Zdroj: vlastní výroba

Obrázek 5 – ukázka výskoku

#### Wizzard

Během hry si kontroluje svoji polohu. Pomocí raycastů zjistí, jestli má hráče v LOS, pokud ne, tak se spustí timer. Jestli se hráč nevratí do konce timeru do jeho LOS, tak si hledá novou pozici k přemístění. Svoji orientaci si hlídá pomocí dalších raycastů. Pokud kouzelník utrpí poškození, sníží se jeho zdraví a schová se. Také má schopnost střílet projektily, což je řízeno časovačem, který určuje, kdy může znovu střílet. Pokud jeho zdraví klesne na nulu, je odstraněn ze scény. Celkově kód řídí logiku pohybu, útoku a obrany kouzelníka ve hře.

Obrázek 6 – Ukázka projektilu

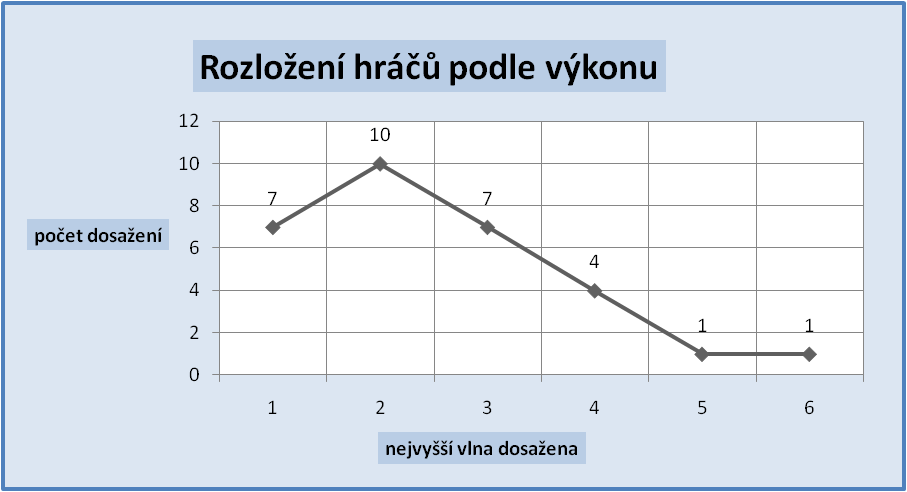
Zdroj: vlastní výroba

# Excel

Tabulka 1 – Výkony hráčů

Zdroj: vlastní výroba

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Výsledky play-testu | | | | |
|  | pokus 1 | pokus 2 | pokus 3 | nejlepší výkon |
| hráč 1 | 1 | 2 | 2 | 2 |
| hráč 2 | 1 | 2 | 4 | 4 |
| hráč 3 | 2 | 4 | 5 | 5 |
| hráč 4 | 2 | 3 | 4 | 4 |
| hráč 5 | 1 | 1 | 2 | 2 |
| hráč 6 | 2 | 2 | 3 | 3 |
| hráč 7 | 3 | 3 | 4 | 4 |
| hráč 8 | 2 | 3 | 6 | 6 |
| hráč 9 | 1 | 1 | 1 | 1 |
| hráč 10 | 2 | 3 | 3 | 3 |



Zdroj: vlastní výroba

Graf 1 – Rozložení hráčů podle výkonu

# Závěr

Vytvoření hry MidKnight pomocí Godot enginu bylo opravdovou výzvou a vyžadovalo obrovské množství času. Godot se na konec ukázal jako spolehlivý nástroj, který obsahoval vše, co jsem potřeboval. I přes některé nedostatky a problémy byl vývoj hry úspěšný. Během projektu jsem získal mnoho zajímavých zkušeností s organizací a časovým rozvržením práce. Výsledky play-testů ukázaly, že hra je příliš náročná a s pozdějšími vlnami téměř nemožná. Tento projekt mě obohatil nejen o technické dovednosti, ale také o zkušenosti s vytvářením a organizací obsáhlých projektům v praxi.

# Seznam použité literatury

LINIETSKY, Juan a MANZUR, Ariel. Godot engine. Online. Dostupné z: https://godotengine.org/features/. [cit. 2024-06-09].

# Seznam obrázků a tabulek

[Obrázek 1 – Aseprite UI 5](file:///C:\Users\eic42401\Downloads\Eichert-Dokumentace.docx#_Toc168988594)

[Obrázek 2 – koncept na rytíře 6](file:///C:\Users\eic42401\Downloads\Eichert-Dokumentace.docx#_Toc168988595)

[Obrázek 3 – ukázka exportu animace 6](file:///C:\Users\eic42401\Downloads\Eichert-Dokumentace.docx#_Toc168988596)

[Obrázek 4 – ukázka hierarchie 7](file:///C:\Users\eic42401\Downloads\Eichert-Dokumentace.docx#_Toc168988597)

[Obrázek 5 – ukázka výskoku 8](file:///C:\Users\eic42401\Downloads\Eichert-Dokumentace.docx#_Toc168988598)

[Obrázek 6 – Ukázka projektilu 8](file:///C:\Users\eic42401\Downloads\Eichert-Dokumentace.docx#_Toc168988599)

[Tabulka 1 – Výkony hráčů 9](file:///C:\Users\eic42401\Downloads\Eichert-Dokumentace.docx#_Toc168988600)

[Graf 1 – Rozložení hráčů podle výkonu 9](file:///C:\Users\eic42401\Downloads\Eichert-Dokumentace.docx#_Toc168988601)