

**STREDNÁ PRIEMYSELNÁ ŠKOLA ELEKTROTECHNICKÁ
ZOCHOVA 9, BRATISLAVA**



2D počítačová hra – Skákačka

projekt z praktickej časti odbornej zložky maturitnej skúšky

Meno kandidáta: Martin Klčo

Trieda: 4.D

Vedúci práce: Bc. Lenka Ráčková

Šk.rok: 2024/2025

**STREDNÁ PRIEMYSELNÁ ŠKOLA ELEKTROTECHNICKÁ
ZOCHOVA 9, BRATISLAVA**

2D počítačová hra – Skákačka

Meno kandidáta: Martin Klčo

Trieda: 4.D

Vedúci práce: Bc. Lenka Ráčková

Rozsah: 28 strán, 24 obrázkov, 0 tabuliek, 0 príloh

Odovzdané:

Podpis vedúceho práce:

Čestné prehlásenie

Čestne prehlasujem, že problematiku týkajúcu sa náplne zadaného projektu v rámci praktickej časti odbornej zložky maturitnej skúšky formou obhajoby som spracoval sám za pomoci uvedenej použitej literatúry. Inú, ako uvedenú literatúru som nepoužil.

V Bratislave dňa:

Martin Klčo

Pod'akovanie

Chcel by som sa poďakovať svojej vedúcej práce nie len za pomoc pri tvorbe maturitného projektu ale aj za celé štyri roky, ktoré vo mňa verila a naučila ma mnoho do budúcnosti. Taktiež ďakujem mojim rodičom za ich neustálu podporu. Nakoniec ďakujem spolužiakom. Za to, že mi pomohli pri testovaní.

OBSAH

Úvod	1
1 Využitie technológie	2
1.1 GML Visual	2
1.2 GML Code	2
1.3 GameMaker	2
1.3.1 Dôvod výberu GameMakeru	3
1.3.2 GameMaker Image Editor	3
1.3.3 GameMaker Sound Editor	4
1.3.4 GameMaker Room Editor.....	4
1.3.5 GameMaker Asset Browser	5
2 Výber žánru	6
3 Funkčnosť hry	7
3.1 Hráč	7
3.1.1 Pohyb.....	8
3.1.2 Skok	8
3.1.3 Skrčenie	9
3.1.4 Útok.....	9
3.1.5 Zdravie.....	11
3.1.6 Smrť hráča	12
3.1.7 Predmety	12
3.2 Kolízie	12
3.2.1 Testovanie kolízií	13
3.2.2 Kontrola, či je hráč na zemi	14
3.3 Nepriatel ia	14
3.3.1 Žaba	14
3.4 Pasce	16
3.4.1 Pasca so šípami (Arrow Trap)	16
3.4.2 Hroty (Spikes)	17
3.4.3 Prázdnota (Void).....	17
3.4.4 Vodopád	18
3.4.5 Falošné steny.....	18
3.5 Prechod medzi úrovňami	19
3.6 Vývojárske pomôcky (Cheaty)	20
4 Popis častí hry	21
4.1 Hlavné menu	21
4.2 Levely – Úrovne	21
4.2.1 Tutoriálové (náučné) levely	22

4.2.2	Stredné levely.....	22
4.2.3	Posledný level.....	22
4.3	Koniec hry	23
4.3.1	Prehra	24
4.3.2	Výhra	24
5	Postup tvorenia hry	25
Záver	28

ZOZNAM OBRÁZKOV

Obrázok 1 - Image Editor	3
Obrázok 2- Room Editor [2]	4
Obrázok 3- Asset Browser [3]	5
Obrázok 4- Hollow Knight	6
Obrázok 5- Postava Hráča	7
Obrázok 6- Get_input().....	8
Obrázok 7- Efekt pádu na zem	8
Obrázok 8- Rozdiel kolíznych másk v nečinnom stave a v skrčení	9
Obrázok 9- Útok hráča.....	10
Obrázok 10- HP bar	11
Obrázok 11- Magické čižmy.....	12
Obrázok 12- Kľúč	12
Obrázok 13- Collisions tile set.....	13
Obrázok 14- On_ground()	14
Obrázok 15- Žaba	15
Obrázok 16- Pasca so šípami.....	16
Obrázok 17- Hroty	17
Obrázok 18- Vodopád.....	18
Obrázok 19- Hráč za falošnou stenou.....	19
Obrázok 20- Hlavné menu.....	21
Obrázok 21- Tutoriálová správa	22
Obrázok 22- Otvorená brána v poslednom leveli	23
Obrázok 23- "Congratulations" okno.....	24
Obrázok 24- "Game Over" okno.....	24

ZOZNAM SKRATIEK

Skratka	Popis
hp	Body života (health points)
hsp	Horizontálna rýchlosť (horizontal speed)
vsp	Vertikálna rýchlosť (vertical speed)

Úvod

Platformové hry sú dnes jedným z najpopulárnejších žánrov v hernom priemysle. Jednoduchosť ovládania, kreatívny dizajn levelov a zábavné mechaniky robia z platformoviek obľúbenú voľbu pre hráčov všetkých vekových kategórií.

Od detstva nás fascinoval svet videohier. Skúšali sme rôzne herné žánre a vždy nás zaujímal, čo stojí za ich vývojom. Postupne sme začali premýšľať nad tým, ako hry vznikajú, a túžili sme vytvoriť vlastný herný projekt podľa našich predstáv. Chceli sme vytvoriť hru, ktorá bude unikátna a zároveň prinesie hráčom výzvu a zábavu. Na trhu existuje množstvo platformových hier, no často sa opakujú rovnaké mechaniky bez výraznej inovácie. Naším cieľom bolo vytvoriť hru, ktorá skombinuje precízne navrhnuté úrovne a premyslené herné mechaniky, ktoré od hráča vyžadujú nielen rýchle reakcie, ale aj strategické myslenie. Pri výbere maturitného projektu sme sa rozhodli túto víziu zrealizovať a vyskúšať si vývoj vlastnej hry.

Cieľom tohto projektu je vytvoriť 2D platformovú hru, ktorá bude zábavná nielen pre vášnivých hráčov počítačových hier, ale aj pre tých, ktorí hry bežne nehrávajú. Hra bude postavená na jednoduchých, no precízne navrhnutých herných mechanikách, ktoré budú jednoduché na pochopenie, no zároveň dostatočne náročné. Hráči tak budú motivovaní zdokonaľovať svoje schopnosti a získať zručnosti potrebné na úspešné zdolanie všetkých levelov. Zámerom je poskytnúť zábavný a naplňujúci herný zážitok, ktorý bude spájať kreativitu, stratégiu a zručnosť.

Celá hra bude ovládaná výhradne klávesnicou bez potreby myši alebo iného hardvéru s výnimkou vývojárskych pomôcok. Hráč sa vžije do úlohy rytiera vyzbrojeného mečom a štítom, ktorý bude preskúmať hrad a jeho blízke okolie. Hra bude obsahovať minimálne 10 unikátnych levelov, pričom každý z nich bude vyžadovať využitie rôznych schopností na prekonanie prekážok a nepriateľov.

V projekte využijeme GML Visual, GML Code, alebo ich kombináciu na písanie kódu. Grafické prvky budú vytvárané v zabudovanom GameMaker Image Editore, s dôrazom na konzistentný vizuálny štýl a plynulé animácie.

TEORETICKÁ ČASŤ

1 Využité technológie

1.1 GML Visual

GML Visual je vizuálny skriptovací nástroj v prostredí GameMaker, ktorý je navrhnutý predovšetkým pre začiatočníkov alebo pre vývojárov, ktorí uprednostňujú intuitívne grafické rozhranie pred písaním kódu. Tento nástroj umožňuje vytvárať herné mechaniky pomocou akcií, ktoré sú zobrazené ako bloky. Tieto bloky sa dajú jednoducho prepojiť a tvoria tak logickú štruktúru hry. Vizuálne programovanie v GML Visual uľahčuje návrh základných herných prvkov a umožňuje rýchly vývoj hier, pričom poskytuje prehľadný pohľad na pracovný postup.

1.2 GML Code

GML Code je textový skriptovací jazyk v GameMakeri, ktorý poskytuje flexibilitu a kontrolu pri tvorbe hier. Umožňuje manipuláciu s objektmi, spritmi (vizuálna reprezentácia objektov), vykresľovanie 2D grafiky a implementáciu pokročilých mechaník. Kombinovanie GML Code a GML Visual umožňuje efektívne spojiť vizuálny dizajn s pokročilým kódovaním, čím sa dosahuje flexibilita pri vývoji hier a zároveň sa zachováva jednoduchosť pri práci s hernými mechanikami.

1.3 GameMaker

Je určený na tvorbu hier a prototypov. Ponúka možnosť vyvíjať hry na viaceré platformy, pričom umožňuje používať vizuálne programovanie prostredníctvom GML Visual alebo tradičné kódovanie v jazyku GML Code. Pre flexibilitu je možné tieto dva spôsoby kombinovať.

GameMaker obsahuje množstvo integrovaných nástrojov, ktoré uľahčujú vývoj hier a rozširujú kreatívne možnosti tvorby. Medzi ne patrí editor obrázkov na tvorbu spritov, Object Editor na pridávanie logiky objektom, Room Editor na návrh herných miestností, a nástroje ako Paths, Sequences a Shaders na rozšírenie možností dizajnu a implementácie herných mechaník.

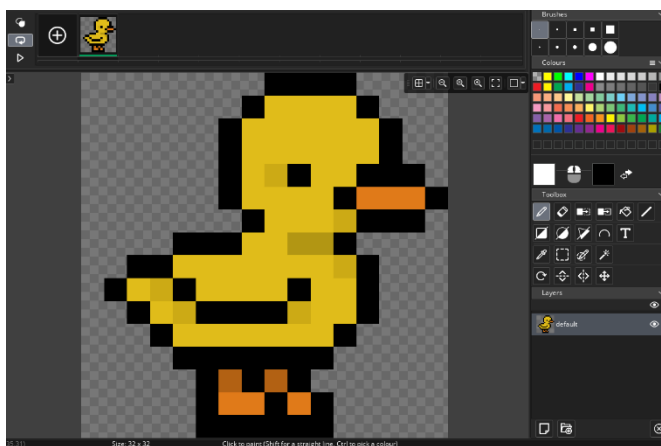
1.3.1 Dôvod výberu GameMakeru

Dôvodov, prečo som si na tento projekt vybral GameMaker je viacero. V prvom rade ponúka jednoduchosť používania vďaka intuitívnemu rozhraniu a drag-and-drop mechanike, čo výrazne uľahčuje tvorbu hier. Je ideálny pre začiatočníkov, keďže umožňuje začať bez potreby rozsiahlych znalostí programovania, ale zároveň má nástroje potrebné pre profesionálov, ktorí môžu využiť skriptovací jazyk GML na komplexnejšie projekty.

Ďalším významným dôvodom je jeho schopnosť exportovať hry na viaceré platformy vrátane počítačov, mobilných zariadení a dokonca herných konzol, čo otvára možnosti osloviť širšie publikum. GameMaker je zároveň špecialista na 2D hry, pričom poskytuje množstvo nástrojov na animácie, prácu s pixel-artom a vývoj mechaník špecifických pre platformovky či RPG žánery. Tieto vlastnosti z neho robia ideálny nástroj pre môj projekt.

1.3.2 GameMaker Image Editor

GameMaker Image Editor je nástroj v programe GameMaker určený na tvorbu a úpravu grafiky pre hru. Tento editor umožňuje vytváranie pixel-artu a animácií s využitím širokej škály nástrojov, ako sú ceruzka, guma, výplň alebo farebný výber. Jeho používateľské rozhranie je jednoduché a intuitívne, čo umožňuje rýchlu tvorbu a úpravy obrázkov. Na obrázku 1 môžete vidieť rozhranie Image Editoru.



Obrázok 1 - Image Editor

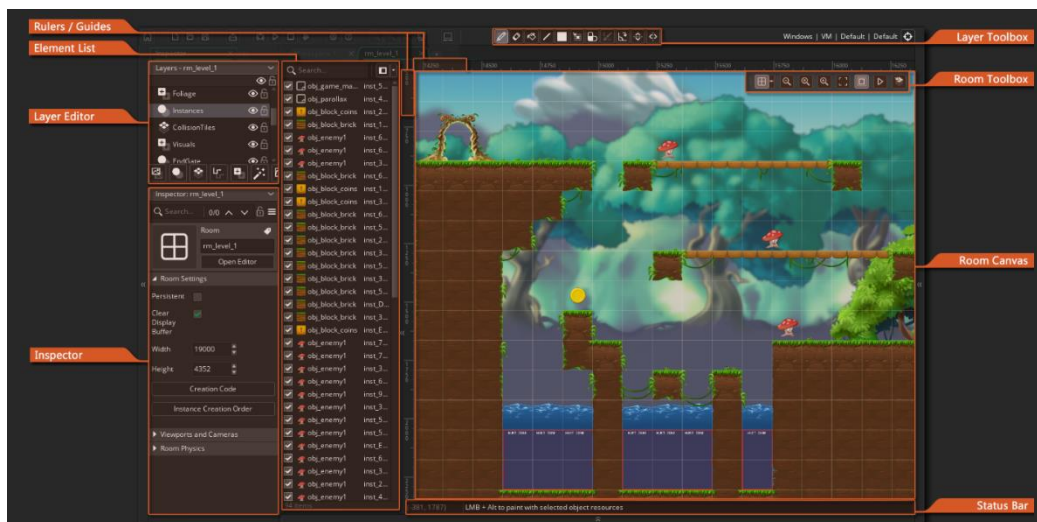
1.3.3 GameMaker Sound Editor

GameMaker Sound Editor je nástroj na tvorbu a úpravu zvukových efektov a hudby pre hru. Tento editor umožňuje pridať zvukové súbory do hry, upravovať ich (napríklad zmena hlasitosti, rýchlosti prehrávania alebo aplikácia rôznych efektov) a implementovať ich do herného prostredia.

1.3.4 GameMaker Room Editor

GameMaker Room Editor slúži na tvorbu herných miestností (úrovní), kde môžete usporiadať objekty ako sprity, dlaždice, pozadia a ďalšie herné prvky. Umožňuje definovať vrstvy pre každý objekt, ktoré sa dajú upravovať a organizovať pomocou Layer Editoru. Okrem toho poskytuje nástroje na nastavenie hernej logiky a interakcií medzi objektmi.

Editor miestností umožňuje jednoduchú manipuláciu s rôznymi hernými prvkami a ponúka nástroje ako Tile Editor na prácu s dlaždicami a "Convert Image to Tile Map" na import obrázkov a ich konverziu na tile mapy. Tento nástroj poskytuje flexibilitu pri tvorbe a úprave herných prostredí a mechaník. Príklad Room Editoru môžete vidieť na obrázku číslo 2.



Obrázok 2- Room Editor [2]

1.3.5 GameMaker Asset Browser

Jadro každej hry v GameMakeri sa tvorí z assetov, ktoré sú spravované v Asset Browseri, nachádzajúcom sa štandardne na pravej strane rozhrania IDE. Tento prehľadný nástroj slúži na pridávanie všetkých prvkov potrebných pre fungovanie hry, ako sú miestnosti (rooms), grafické prvky (sprites), objekty a množstvo ďalších typov assetov.



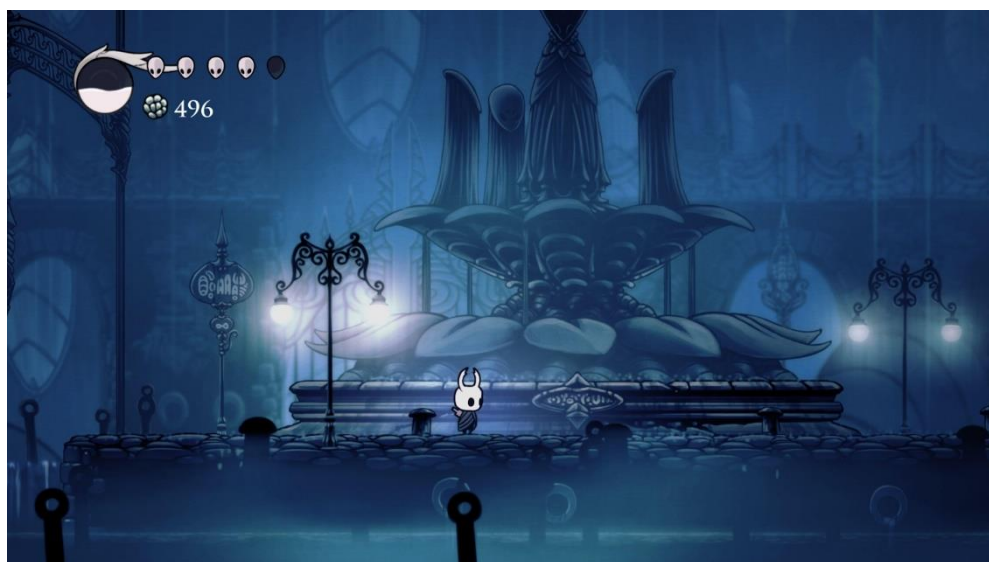
Obrázok 3- Asset Browser [3]

2 Výber žánru

Platformové hry, známe aj ako platformovky, patria medzi najstaršie a najpopulárnejšie žánre videohier. Ich herná mechanika je založená na ovládaní postavy, ktorá musí skákať medzi plošinami, prekonávať prekážky a riešiť rôzne výzvy. Tento žáner vznikol začiatkom 80. rokov s titulmi ako Donkey Kong a Mario Bros, pričom neskôr hry ako Super Mario Bros zaviedli plynulý posun obrazovky a komplexnejší dizajn úrovni[5].

Aj keď popularita platformových hier v 90. rokoch mierne poklesla, dnes zažívajú obrovský návrat vďaka indie hrám (hry tvorené nezávislými vývojármi) a moderným remakom[5]. Tituly ako Hollow Knight ukázali, že platformovky môžu mať nielen skvelú hrateľnosť, ale aj bohatý umelecký štýl a atmosféru.

V našej hre budú úrovne navrhnuté s dôrazom na plynulý pohyb, precízne načasované skoky a skryté oblasti na objavovanie. Rovnako ako Hollow Knight, aj naša hra umožní hráčom preskúmať svet vlastným tempom, objavovať alternatívne trasy a odhaľovať tajomstvá, ktoré odmenia ich zvedavosť. Zároveň si ponecháme klasickú platformovú mechaniku, pričom integrujeme moderné prvky, ako sú plynulé animácie



Obrázok 4- Hollow Knight

PRAKTICKÁ ČASŤ

3 Funkčnosť hry

3.1 Hráč

Hráč je najdôležitejší objekt tejto hry, bez ktorého by hra nemala zmysel. Za jednoducho vyzerajúcou postavou sa skrýva viacero mechaník. Patria sem pohyb, skok, skrčenie a útok. Hráč využíva stavový systém (state machine) na efektívne spravovanie správania, čo umožňuje hladké prepínanie medzi jednotlivými akciami. Takmer každý stav má priradený svoj vlastný animovaný sprite. Jednotlivé stavy hráča sú: IDLE (nečinnosť), WALK (chôdza), JUMP (skok), ATTACK (útok), CROUCH (skrčenie), HURTING (zranenie), DIE (smrť).



Obrázok 5- Postava Hráča

3.1.1 Pohyb

Pohyb hráča v hre je ovládaný klávesami A, S, D, pričom skákanie je priradené klávese medzerník (space). Pohyb funguje na princípe získania vstupu (inputu) od hráča, čo sa rieši pomocou skriptu `get_input()`, ktorý zisťuje, ktoré klávesy boli stlačené. Tento skript sa volá v každom stave hráča. Mechanika pohybu zahŕňa horizontálnu rýchlosť (*hsp*), maximálnu rýchlosť (*max_hsp*), a trenie (*drag*), ktoré postupne spomaľuje pohyb, ak hráč prestane držať klávesy. Na obrázku 6 môžete vidieť skript `get_input()`

```
function get_input(){
  //keyboard input
  left = keyboard_check(ord("A"));
  right = keyboard_check(ord("D"));
  up = keyboard_check(ord("W"));
  down = keyboard_check(ord("S"));
  attack = keyboard_check_pressed(ord("J"));
  jump = keyboard_check_pressed(vk_space);
  jump_held = keyboard_check(vk_space);
}
```

Obrázok 6- Get_input()

3.1.2 Skok

Skok je spustený stlačením medzerníka. Hráč môže použiť dvojité skok, čo poskytuje jeden skok vo vzduchu. Dvojité skok funguje veľmi jednoducho, a to len použitím premennej *jumps*, ktorá sa resetuje na 2 pri dotyku hráča so zemou. Na skoky je použitý skript `jumped()`, ktorý znižuje hodnotu *jumps* a nastaví vertikálnu rýchlosť (*vsp*) na hodnotu *jump_spd*. Ak hráč počas skoku drží medzerník, umožňuje to vyšší skok. Pre tento účel je implementovaný mechanizmus "dampened jump", ktorý redukuje rýchlosť skoku, ak hráč pustí klávesu. Toto umožňuje hráča kontrolovať výšku skoku. Ako malý detail sa pri každom skoku hráča vytvára vizuálny efekt rozvíreného prachu, ktorý po krátkej chvíli zmizne. Podobný efekt sa aplikuje aj pri dopade hráča na pevnú zem alebo platformu.



Obrázok 7- Efekt pádu na zem

3.1.3 Skrčenie

Skrčenie sa používa držaním klávesy S. Skrčenie znižuje kolíznu masku (collision mask) hráča a tým pádom sa dá vyhnúť útokom miereným na úroveň hlavy postavy. V stave skrčenia sa horizontálna rýchlosť (*hsp*) nastaví na 0, čím je znemožnený pohyb do strán, a jedinou možnou interakciou je otáčanie v smere pohľadu. Hráč môže zo stavu skrčenia vykonať len niekoľko akcií, ako napríklad útok, alebo skok. V prípade stlačenia skoku sa hráč automaticky vráti do normálnej výšky a vykoná skok, pričom sa resetuje kolízna maska na pôvodnú veľkosť. Na obrázku 8 môžete vidieť rozdiel medzi kolíznou maskou v stave nečinnosti a skrčení.



Obrázok 8- Rozdiel kolíznych másk v nečinnom stave a v skrčení

3.1.4 Útok

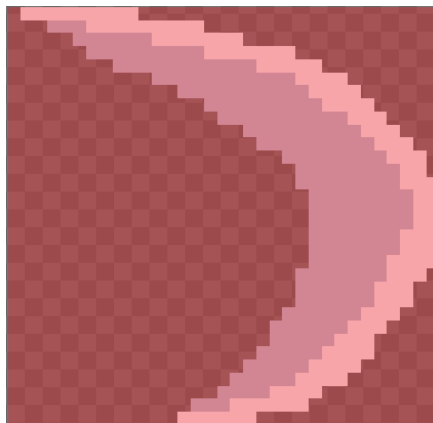
Útok hráča je jednou z kľúčových mechaník hry, ktorá umožňuje interakciu s nepriateľmi. Spustený je stlačením klávesy J. Hráč pri útoku využíva animáciu, počas ktorej sa vytvorí hitbox. Tento hitbox je dočasný a jeho pozícia závisí od pozície hráča a stavu útoku. Je vytváraný len počas určitej časti animácie.

Útok zasiahne len nepriateľov, ktorí sa nachádzajú v blízkosti hráča (napr. do vzdialenosti 60 pixelov). Toto obmedzenie zabezpečuje, že hra neplytvá zdrojmi na zisťovanie kolízií s nepriateľmi, ktorí sú od hráča príliš ďaleko.

Pri kolízii hitboxu s nepriateľským objektom je vykonaných viacero akcií:

- **Poškodenie:** Po zásahu sa nepriateľovi zníži hodnota životov (*hp*) o určitú hodnotu, ktorá môže byť špecifikovaná alebo ovplyvnená inými mechanikami hry. Základná hodnota poškodenia je 1.
- **Imunita:** Ak je nepriateľ zasiahnutý, spustí sa časovač imunity (*hurt_timer*), počas ktorého nepriateľ nemôže byť zasiahnutý ďalším útokom. V tomto stave je nepriateľ zvýraznený červenou blikajúcou farbou. Tento mechanizmus zabezpečuje, že útoky hráča nie sú príliš silné.
- **Otočenie nepriateľa:** Na základe pozície hitboxu voči nepriateľovi sa vypočíta smer zásahu (pomocou funkcie *sign()*), čo obráti nepriateľa smerom k hráčovi.

Na obrázku 9 môžete vidieť útok hráča s vyznačeným hitboxom.



Obrázok 9- Útok hráča

3.1.5 Zdravie

Systém zdravia je implementovaný formou grafického ukazovateľa (HP bar), ktorý vizuálne zobrazuje aktuálny stav zdravia hráča na ľavej hornej strane obrazovky. Tento ukazovateľ sa vykresľuje na obrazovke pomocou funkcie Draw GUI, ktorá sa vykonáva nezávisle od hernej kamery, aby bol ukazovateľ vždy viditeľný.

Zdravie hráča je reprezentované dvoma premennými: *hp* (aktuálna hodnota zdravia hráča) a *max_hp* (maximálna hodnota zdravia hráča). HP bar sa skladá z troch vrstiev:

- **Pozadie:** Dekoratívny rámik HP baru
- **Dynamický ukazovateľ:** Ukazuje aktuálne zdravie hráča ako proporciu z jeho maximálneho zdravia. Dĺžka tejto vrstvy sa prispôsobuje na základe pomeru $o_player.hp / o_player.max_hp$.
- **Zadná vrstva:** Zadná časť HP baru, ktorá tvorí červené pozadie, zobrazujúce stratené body života.

Rovnako ako nepriatelia, aj hráč má tzv. "invulnerability frames" (obdobie dočasnej nezraniteľnosti), ktoré sa aktivujú krátko po tom, ako utrpí zásah od nepriateľa alebo iného zdroja poškodenia. Na rozdiel od nepriateľov hráč v tomto období bliká na bielo, čím jasne signalizuje, že bol zasiahnutý, a zároveň sa tým odlišuje od ostatných objektov.



Obrázok 10- HP bar

3.1.6 Smrť hráča

Smrť hráča nastáva, keď jeho aktuálne zdravie (*hp*) klesne na nulu. Ak sa hodnota hráčovho zdravia (*hp*) dostane na nulu alebo pod nulu, hráč prejde do stavu smrti, kde sa spustí animácia smrti a zastaví sa pohyb hráča. V tomto stave je hráč neaktívny a prestane reagovať na akékoľvek vstupy. Po dokončení animácie smrti sa hra vyhodnotí ako prehraná (*game_over_lose = true*) a zobrazí sa okno “Game Over”.

3.1.7 Predmety

Hráč sa v určitých úrovniach môže stretnúť s predmetmi, ktoré môže zdvihnúť. Po dotknutí predmetu sa hráčovi umiestni predmet do inventára. V pravom hornom rohu je vždy vidieť, aký predmet hráč nesie. Po prechode do inej miestnosti hráč predmet stráca. Medzi použiteľné predmety patria:

- **Magické čižmy** - Magické čižmy hráč môže nájsť na ôsmej úrovni. Sú skryté za falošnou stenou. Umožňujú hráčovi urobiť tri skoky bez dotknutia zeme. Magické čižmy sú použiteľné, až kým hráč odíde z úrovne, na ktorej ich získal. Nie sú potrebné na prekonanie ôsmej úrovne, ale dokážu hráčovi značne pomôcť.



Obrázok 11-
Magické čižmy

- **Kľúč** - Kľúč sa nachádza v poslednej miestnosti. Je použitý na odomknutie dverí na konci úrovne, aby hráč mohol vyhrať hru. Bez kľúča je nemožné prejsť poslednú úroveň.



Obrázok 12- Kľúč

3.2 Kolízie

V platformovej hre sú kolízie základnou mechanikou, ktorá určuje interakcie medzi objektmi v hre. Kolízie sú najmä zodpovedné za detekciu kontaktu medzi hráčom a prostredím. Pre detekciu kolízií používame tzv. bounding box (obvodový box), ktorý obklopuje hráčovú postavu. Tento box je využívaný na detekciu, či sa objekt hráča nachádza v kontakte s inými objektmi v hre, ako sú steny, platformy alebo nepriatelia.

Kolízie v hre sú spravované prostredníctvom "Collisions" vrstvy v miestnosti. Táto vrstva nie je viditeľná pre hráča, ale zohráva kľúčovú úlohu pri interakcii objektov s prostredím. Na tejto vrstve je použitý špecifický tile set (dlaždicová sada), ktorý obsahuje tri základné typy dlaždíc:

- **Void (Prázdne miesto):** Oblasti, ktoré nemajú žiadne kolízne vlastnosti, čo znamená, že objekty nimi môžu voľne prechádzať.
- **Solid (Pevné):** Pevné bloky, ktoré úplne zastavujú pohyb objektov a fungujú ako steny alebo podlahy.
- **Platform (Platforma):** Polopevné povrchy, ktoré umožňujú objektom pohybovať sa po ich vrchnej strane, ale umožňujú prechod zospodu, čím podporujú mechaniky ako skákanie cez platformy.
- **Death (Smrť):** Po dotyku hráč okamžite stratí všetko svoje zdravie a prejde do stavu smrti.



Obrázok 13- Collisions tile set

3.2.1 Testovanie kolízií

Kolízie sú kontrolované pomocou funkcie *collision()*, ktorá kontroluje pozíciu hráča voči okolitému prostrediu. Táto funkcia najprv aplikuje priradené horizontálne a vertikálne rýchlosti (*hsp*, *vsp*) a následne testuje každú stranu hráčovho bounding boxu na prítomnosť pevných objektov v prostredí, ako sú steny a platformy.

- **Horizontálna detekcia:** Funkcia zisťuje, či je pozícia na pravej alebo ľavej strane objektu v kontakte s pevným povrchom (napr. stena alebo platforma). V prípade zrážky sa rýchlosť zastaví a pozícia sa upraví tak, aby sa hráč neprepadol do objektu.

- **Vertikálna detekcia:** Funkcia testuje kolíziu so zemou alebo inými platformami pod alebo nad postavou. Ak detekuje pevnú prekážku, vertikálna rýchlosť sa zastaví a pozícia sa upraví.

3.2.2 Kontrola, či je hráč na zemi

Funkcia *on_ground()* slúži na zisťovanie, či je hráč na zemi. Funkcia kontroluje, či sú pod hráčom pevné objekty, ktoré mu umožňujú vykonávať akcie ako skoky alebo chôdzu. Ak nie sú detekované žiadne pevné objekty pod hráčom, funkcia vráti hodnotu *false*, čo naznačuje, že hráč nie je na zemi. Táto funkcia sa používa napríklad pri skrčení, alebo skoku. Pre skrčenie je nutné, aby funkcia *on_ground()* vrátila hodnotu *true*, pretože vo vzduchu sa hráč skrčiť nemôže. Zároveň pre skok je táto funkcia potrebná najmä kvôli obnovovaniu skokov pri dotyku pevnej plochy. Na obrázku 14 môžete vidieť funkciu *on_ground()*.

```
function on_ground(){
    var side = bbox_bottom;
    var t1 = tilemap_get_at_pixel(global.map, bbox_left, side + 1);
    var t2 = tilemap_get_at_pixel(global.map, bbox_right, side + 1);
    var t3 = tilemap_get_at_pixel(global.map, bbox_left, side);
    var t4 = tilemap_get_at_pixel(global.map, bbox_right, side)

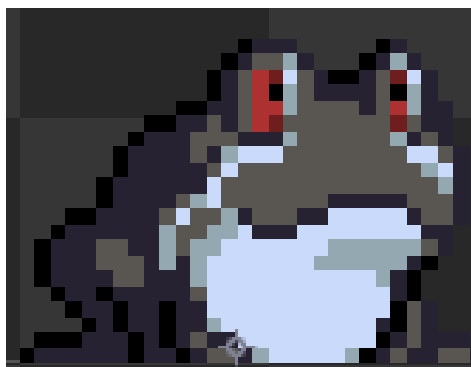
    if ((t1 == SOLID or t1 == PLATFORM) and (t3 != SOLID and t3 != PLATFORM) or (t1 == SOLID and t3 == PLATFORM)
        (t2 == SOLID or t2 == PLATFORM) and (t4 != SOLID and t4 != PLATFORM) or (t2 == SOLID and t4 == PLATFORM))
        return true else return false;
}
```

Obrázok 14- On_ground()

3.3 Nepriatelia

3.3.1 Žaba

Žaba (*o_frog*) je nepriateľ, ktorý obsahuje komplexné správanie. Správanie žaby je založené, tak ako hráč, na stavovom systéme (state machine). Tento systém určuje, v akom stave sa žaba nachádza, ako napríklad: IDLE (nečinnosť), JUMP_START (začiatok skoku), JUMP (skok) a JUMP_LAND (dopad po skoku).



Obrázok 15- Žaba

Mechaniky žaby zahŕňajú:

- **Pohyb a skákanie:** Žaba vykonáva náhodné skoky s pravdepodobnosťou určenou špecifickým intervalom. Skoky môžu byť spustené iba vtedy, keď je žaba v stave nečinnosti. Pravdepodobnosť skoku je riadená náhodným generátorom, kde ak hodnota prejde nastavenú šancu, žaba začne skákať.
- **Animácie:** Funkcia *frog_anim()* priradzuje sprity podľa aktuálneho stavu. Napríklad v stave JUMP žaba prepne animáciu, ak vertikálna rýchlosť dosiahne zápornú hodnotu (pád).
- **Kolízie:** Žaba používa funkciu *collision()* na detekciu kolízií s prostredím. Kolízie určujú, kedy má žaba pristáť, zastaviť pohyb alebo zmeniť smer.
- **Dýchanie:** Funkcia dýchania žaby riadi animácie počas nečinnosti. Keď žaba dýcha, zobrazuje sa animácia, ktorá je riadená časovačom (*breath_timer*). Po skončení animácie sa časovač resetuje a žaba začne od začiatku.
- **Detekcia platforiem:** Žaba môže skákať iba na stabilné platformy, pričom jej pohyb je riadený detekciou dlaždíc pred ňou. Ak sa žaba nachádza na okraji platforiem alebo pri voľnom priestore, automaticky sa otočí, aby zabránila skákaniu z okrajov.

Ak hráč zasiahne žabu, spustí sa funkcia, ktorá znižuje životy nepriateľa (*hp*) o 1, spustí *imunity frames* a nastaví stav HURTING. Počas tejto fázy žaba bliká červenou farbou ako indikácia zranenia a zároveň imunity proti útokom. Ak žaba zasiahne hráča, otočí sa k nemu pomocou hodnoty *facing*, poškodí ho, spôsobí *knockback* (odraz) a nastaví stav HURTING pre hráča.

3.4 Pasce

V hre sa nachádzajú rôzne typy pascí, ktoré predstavujú pre hráča nebezpečenstvo. Pasce hráčovi môžu ublížiť dvoma spôsobmi:

Uberanie bodov života - uberie hráčovi určité množstvo bodov života

Okamžitá smrť - po kontakte s pascou hráčove body života klesajú na 0 a zomiera

3.4.1 Pasca so šípmi (Arrow Trap)

Pasca so šípmi testuje reflexy, pohybové schopnosti a časovanie hráča. Je to pasca, ktorá v pravidelných intervaloch strieľa šípy v určenom smere. Umiestnená môže byť na stenách, od ktorých sa líši len dvomi malými dierami na strieľanie šípov. Nenápadnosť tejto pasce pridáva k obtiažnosti hry a núti hráča byť neustále v pozore. S touto pascou sa hráč stretáva už v skorých úrovniach hry.

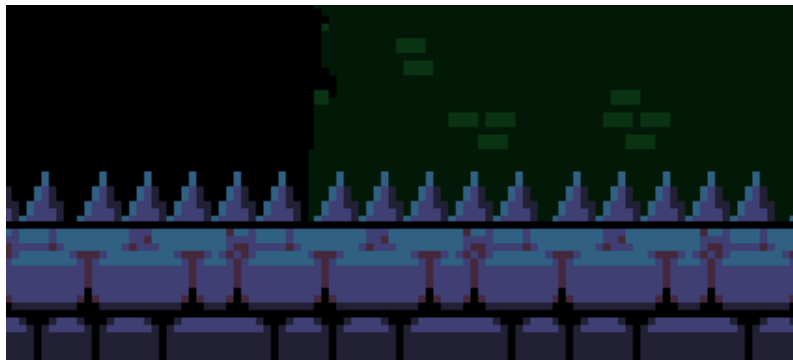
Mechanika pasce - Pasca sa aktivuje, keď sa zobrazí na obrazovke hráča. Kým je pasca mimo hráčovej obrazovky, je nečinná. Každá pasca má dve možné miesta, z ktorých môže strieľať. Pozícia prvej strely pasce je náhodne vybraná pre každú pascu individuálne. Následne sa pozícia mení z hornej na spodnú a naopak. Šípu z pasce sa dá vyhnúť preskočením, alebo skrčením podľa pozície, z ktorej je šíp vystrelený. Po aktivácii pasca vystrelí šíp v smere, v ktorom je otočená. Rýchlosť všetkých šípov je rovnaká. Každý výstrel šípu je nasledovaný časovou prestávkou, ktorá je pre každú pascu nastavená náhodne, aby sa nestalo, že všetky pasce budú strieľať naraz. Vystrelený šíp sa zničí, keď opustí hernú miestnosť, po kontakte so stenou alebo po kontakte s hráčom. Zasiahnutý hráč stráca 1 bod života. V prípade, že hráča zasiahne šíp počas imunity, sa šíp ničí, ale nespôsobuje žiadne škody.



Obrázok 16- Pasca so šípmi

3.4.2 Hroty (Spikes)

Pasca s hrotmi predstavuje jednu z najnebezpečnejších prekážok v hre, ktorá okamžite uberá všetky body života hráča pri kontakte. Hroty sú umiestnené na rôznych miestach úrovne, čím nútia hráča byť opatrný pri pohybe a plánovaní skokov. Často sú využívané v kombinácii s ostatnými pascami. Pri každom kroku hry sa kontroluje kolízia hráča s dlaždicami. Ak hráč vstúpi do oblasti, kde sa nachádza dlaždica typu DEATH(napr. hroty), okamžite stratí všetky body života. Mechanizmus detekcie kolízie s hrotmi je implementovaný v hlavnej kolíznej funkcii *collision()*



Obrázok 17- Hroty

3.4.3 Prázdnota (Void)

Prázdnota je herný prvok, ktorý simuluje nekonečný pád mimo hraciu plochu. Ak hráč spadne do prázdnoty, okamžite zomrie a musí reštartovať úroveň. Tento mechanizmus zvyšuje náročnosť hry a trestá nepresné skoky alebo zlé načasovanie pohybu. V kolíznej funkcii *collision()* sa kontroluje, či sa hráč nachádza pod hernou úrovňou. Ak áno, okamžite sa mu nastaví hodnota života (*hp*) na nulu.

Prázdnota je často využívaná na vytvorenie rizikových oblastí v úrovni, kde hráč musí presne skákať medzi plošinami alebo sa vyhýbať nepriateľom. Vizualne je prázdnota obvykle znázornená ako tmavá oblasť bez akýchkoľvek interaktívnych prvkov, aby bolo hráčovi jasné, že pád do nej je fatálny.

3.4.4 Vodopád

Vodopád v hre predstavuje vizuálny a herný prvok, ktorý funguje ako oblasť bez pevných dlaždíc. Keď hráč prechádza cez vodopád, prepadne cez neho smerom nadol, pretože nie je považovaný za pevnú prekážku. Hráč neprepadá v prípade, že nad vodopádom je most. Vodopád je často využívaný v spojení s prázdnotou (void). Na rozdiel od prázdnoty však vizuálne zapadá do prostredia bežných dlaždíc, a tak vytvára ilúziu dynamického prostredia. Okrem toho môže slúžiť aj ako súčasť skrytých priechodov, ktoré umožňujú hráčovi objavovať nové časti úrovne, ktoré poskytujú rýchlejšiu, alebo jednoduchšiu cestu do ďalšej úrovne.

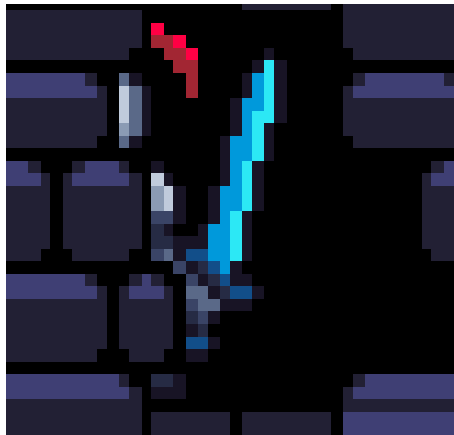


Obrázok 18- Vodopád

3.4.5 Falošné steny

Falošné steny sú špeciálnym typom dlaždíc, ktoré môžu pôsobiť ako bežné pevné steny, avšak hráč cez ne môže prechádzať alebo nimi môže prepadnúť. Od bežných stien sa líšia prasklinami. Tieto steny sa často používajú na vytváranie tajných priechodov, skratiek alebo pascí, čím pridávajú do hry prvok objavovania a nečakaných prekvapení. Na rozdiel od vodopádov, ktoré umožňujú hráčovi zostať viditeľný, falošné steny sa nachádzajú v popredí a zakrývajú hráča spolu s ostatnými objektmi v úrovni. Tým výrazne obmedzujú viditeľnosť a sťažujú orientáciu v priestore.

Na obrázku 19 môžete vidieť hráča skrytého za falošnou stenou.



Obrázok 19- Hráč za falošnou stenou

3.5 Prechod medzi úrovňami

Prechod medzi úrovňami v hre je spravený pomocou efektu *fade* (postupné vyblednutie/stmavnutie obrazovky), ktorý zabezpečuje plynulý a vizuálne príjemný prechod medzi rôznymi miestnosťami. Tento efekt sa používa pri presune hráča z jednej úrovne do druhej alebo pri reštarte úrovne, napríklad po smrti hráča.

Prechod medzi úrovňami začína postupným zatemnením obrazovky, ktorý je riadený premennou *alpha*. Táto hodnota sa postupne zvyšuje, čím sa obrazovka stáva čoraz tmavšou. Po dosiahnutí plnej nepriehľadnosti ($alpha \geq 1$) sa zmení miestnosť a pozícia hráča. Funkcia *fade_to_room()* nastaví cieľovú miestnosť a pozíciu hráča, pričom sa resetujú hodnoty pohybu a obnoví zdravie, ak je hráč v stave smrti. Po presune sa kamera rýchlo prispôsobí novej pozícii hráča. Následne začne obrazovka blednúť, pričom hodnota *alpha* sa znižuje, až kým sa obrazovka nevráti do normálneho stavu. Tento efekt je riadený rýchlosťou vyblednutia *fade_spd*.

Prechod do ďalšej úrovne je spustený pri dotyku prechodného objektu *o_transition*, ktorý je na začiatku a konci každej úrovne. *o_transition* na konci úrovne presunie hráča do ďalšej miestnosti, zatiaľ čo prechod na začiatku úrovne presunie hráča do predošlej miestnosti.

3.6 Vývojárske pomôcky (Cheaty)

Vývojárske pomôcky slúžia primárne na testovanie, ladenie kódu a kontrolu herných mechaník počas vývoja. Umožňujú vývojárom rýchlo overiť správne fungovanie jednotlivých prvkov bez nutnosti prechádzať hrou bežným spôsobom. Bežnému hráčovi neodporúčame používať cheaty, ak chce mať z hry plnohodnotný zážitok.

Cheaty sú v hre implementované ako jednoduché klávesové skratky, ktoré vykonávajú určité funkcie po ich stlačení. Tieto skratky sú zaznamenané v hernom skripte a aktivujú špecifické úpravy herného stavu, ako napríklad nekonečné skoky, obnovenie zdravia alebo okamžitú smrť.

Zoznam dostupných cheatov:

- **Klávesa 8 („á“ na slovenskej klávesnici)** – Aktivuje možnosť nekonečných skokov, čím hráč môže vykonávať neobmedzené množstvo skokov vo vzduchu. Tento cheat je užitočný na testovanie úrovni a zisťovanie, či hráč dokáže dosiahnuť určité oblasti mapy.
- **Klávesa 9 („í“ na slovenskej klávesnici)** – Obnoví hodnotu života na maximum, čo umožňuje vývojárom testovať rôzne bojové situácie alebo prechádzať náročné pasáže bez rizika smrti.
- **Klávesa 0 („é“ na slovenskej klávesnici)** – Okamžite zabije hráča. Tento cheat je určený predovšetkým na ladenie herných mechaník spojených so smrťou a respawnom.
- **Ľavé tlačidlo myši** - Po kliknutí ľavým tlačidlom myši sa hráč okamžite teleportuje na pozíciu kurzora.
- **Číselné klávesy 1 - 9** - Teleportuje hráča na začiatok úrovne zodpovedajúcej stlačenému číslu (1 až 9).
- **Číselná klávesa 10** - Teleportuje hráča na začiatok poslednej úrovne.

4 Popis častí hry

Naša hra sa skladá z troch hlavných častí: Hlavné menu, Levely a Koniec hry. Každá z týchto častí má svoju špecifickú úlohu a spoločne zabezpečujú plynulý priebeh hry.

4.1 Hlavné menu

Hlavné menu je prvé okno, ktoré sa otvorí po spustení hry. Jeho pozadie sa objaví postupnou animáciou pádu, počas ktorej nie je možné hru spustiť. Hlavné menu sme obohatili o vlastný grafický návrh, ktorý sme následne upravili s asistenciou umelej inteligencie. Hru možno začať až po úplnom zobrazení hlavného menu. Po stlačení klávesy medzerník (space) menu pomaly mizne a hráč je presunutý do prvej úrovne. Následne sa aktivuje ovládanie hráča a hra začína. Spolu s ovládaním hráča sa aktivujú cheaty, ktoré hráč môže použiť pre testovanie hry. Na obrázku 20 môžete vidieť hlavné menu.



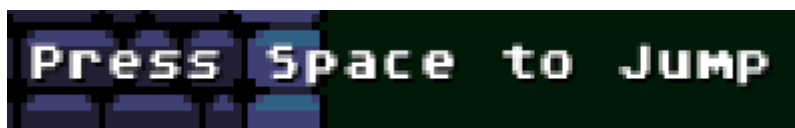
Obrázok 20- Hlavné menu

4.2 Levely – Úrovne

Hra obsahuje rôzne úrovne, ktoré sú rozdelené do troch kategórií: tutoriálové (náučné) levely, stredné levely a posledný level. Každá kategória sa odlišuje svojou funkcionalitou a stupňom náročnosti.

4.2.1 Tutoriálové (náučné) levely

Levely 1 - 3 - Tutoriálové (náučné) levely slúžia na oboznámenie hráča so základnými hernými mechanikami. Každá mechanika je predstavená v momente, keď ju hráč potrebuje použiť alebo keď mu môže pomôcť. To však neznamená, že ich nemôže využiť už od začiatku hry. Počas týchto levelov sa hráč postupne stretáva aj s novými nepriateľmi a pascami. Cieľom prvých troch úrovní je pomôcť hráčovi osvojiť si ovládanie a mechaniky hry, čím ho pripravujú na ďalšie, náročnejšie levely.. Na obrázku 20 je znázornená prvá správa, ktorá sa zobrazí v tutoriálových leveloch.



Obrázok 21- Tutoriálová správa

4.2.2 Stredné levely

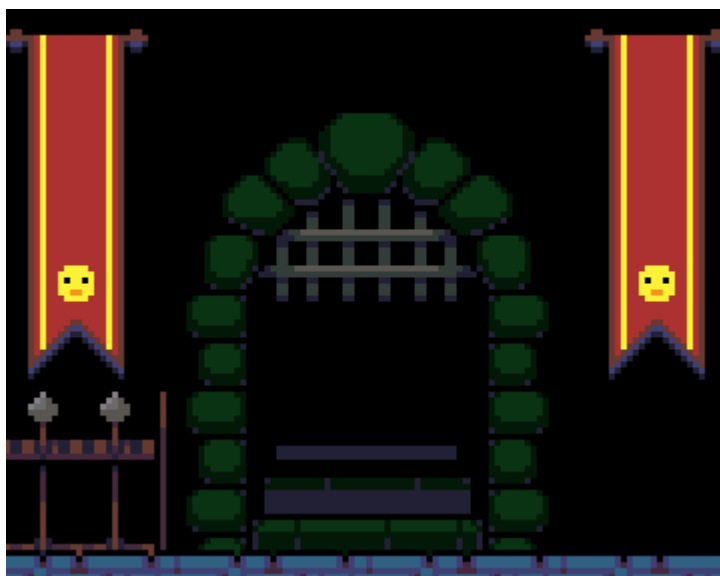
Levely 4 - 9 - Stredné levely sú navrhnuté tak, aby preverili hráčove schopnosti. V stredných leveloch sa nachádzajú rôzne kombinácie pascí a nepriateľov spolu s prostredím, ktoré hráč musí prekonať. V týchto leveloch sa tiež môže stretnúť so situáciami, kde si musí vybrať správnu cestu – nesprávna voľba často vedie k smrti. Oproti tutorialovým levelom sú výrazne náročnejšie a ich náročnosť sa postupne zvyšuje s každou ďalšou úrovňou.

4.2.3 Posledný level

Level 10 - Posledná úroveň v našej hre je zároveň posledná prekážka stojaca medzi hráčom a výhrou. Ide o najnáročnejší level, v ktorom sa kombinujú všetky predchádzajúce prekážky, nepriatelia a pasce v jednej miestnosti. Nepriatelia sú v tomto leveli skrytí a objavujú sa až po prekročení určitých úsekov.

Hlavným cieľom hráča je dostať sa k bráne označenej špeciálnymi značkami. Ak však nemá v inventári kľúč, brána sa automaticky zatvorí a hráč nemôže pokračovať. Po získaní kľúča sa brána opäť otvorí, čím hráč získa prístup k cieľu.

Po úspešnom prejdení bránou sa hra skončí a vyhodnotí stav ako vyhratý (*game_over_won = true*). Na obrazovke sa zobrazí víťazné okno "Congratulations", v ktorom hráč uvidí štatistiku svojich úmrtí. Následne sa môže vrátiť do hlavného menu a začať hru odznova.



Obrázok 22- Otvorená brána v poslednom leveli

4.3 Koniec hry

Koniec hry nastáva po splnení podmienok pre výhru alebo prehru. Hráč môže hru úspešne dokončiť, alebo naopak, neuspieť a začať úroveň odznova. Obe možnosti sú sprevádzané vizuálnymi a hernými prvkami, ktoré dodávajú hre plynulosť a atmosféru.

4.3.1 Prehra

K prehre dochádza v prípade, že hráčova postava umrie. Po smrti hráča sa spustí animácia smrti, ktorá sa postupne dostane do konca. Keď je animácia dokončená a obrazovka sa úplne zatmaví, hra vyhodnotí stav ako prehratý (*game_over_lose = true*). V tomto momente sa zobrazí okno "Game Over", na ktorom môže hráč začať úroveň znova stlačením medzerníka. Okno „Game Over“ môžete vidieť na obrázku 24.



Obrázok 24- "Game Over" okno



Obrázok 23- "Congratulations" okno

4.3.2 Výhra

Hráč vyhrá po prejdení cez bránu v poslednej úrovni. Túto bránu otvorí kľúčom, ktorý nájde na opačnej strane úrovne. Po prejdení cez bránu aj s kľúčom hra vyhodnocuje stav ako vyhratý (*game_over_won = true*), presúva hráča do finálnej miestnosti, kde sa zobrazuje okno podobné "Game Over" oknu, ktoré oznamuje hráčovi, že vyhral a vidí svoje úmrtia. V "Congratulations" okne má hráč po úplnom načítaní možnosť začať hru odznova. Na obrázku 23 je okno, ktoré sa zobrazí po vyhratí.

5 Postup tvorenia hry

GameMaker ponúka dve možnosti vývoja – vizuálne programovanie pomocou blokov alebo písanie kódu. Zvolili sme druhú možnosť, pretože nám poskytovala väčšiu flexibilitu a kontrolu nad hernými mechanikami.

Pri návrhu grafiky sme zvažovali dve herné prostredia: hustý les s assassinom alebo hrad s rytierom. Nakoniec sme sa rozhodli pre hrad s rytierom a vytvorili vlastný tile set, inšpirovaný jednoduchými pixel-artovými grafikami. Tento tile set sme prispôbili tak, aby vizuálne ladil s atmosférou hry.

Každú miestnosť v hre sme rozdelili do viacerých vrstiev pre lepší vizuálny efekt a interaktivitu. Na najnižších vrstvách sme umiestnili pozadie a dekoratívne prvky, stredné vrstvy obsahujú interaktívne steny, pasce a objekty, s ktorými môže hráč interagovať. Kolízna vrstva definuje pohybové možnosti hráča, ako skákanie či zastavenie pri stenách. Do najvyššej vrstvy sme pridali falošné steny, ktoré vytvárajú ilúziu hĺbky a umožňujú hráčovi vizuálne prechádzať za objektmi.

Pred dokončením prvej miestnosti sme najskôr vytvorili hráča a jeho interakciu so stenami. Nadizajnovali sme sprite pre každú akciu, ktorú hráč môže vykonať. Hráča sme najskôr začali programovať bez prípravy, avšak tento spôsob nám nevychádzal a museli sme začať odznova. Oboznámili sme sa s pojmom state machine (stavový systém). Vytvorili sme rôzne stavy, ako nečinnosť, skok či útok, medzi ktorými hráč plynule prechádza. Pôvodne sme plánovali pridať aj mechaniku blokovania útokov štítom, no nakoniec sme ju odstránili, pretože príliš zjednodušovala hru. Hneď po vytvorení hráčovej postavy v hre priradíme hráčovi stav nečinnosti, z ktorého môže prejsť do ostatných stavov. Nemôže sa stať, že hráč nebude v žiadnom stave.

Kolízie sme naprogramovali pomocou skriptu *collision()*. Vytvorili sme si tile set obsahujúci tri základné typy dlaždíc – červené (Solid), modré (Platform) a čierne (Death). V skripte *macros* sme im priradili číselné hodnoty, aby sme zjednodušili písanie kódu a zvýšili jeho prehľadnosť. Na uľahčenie programovania sme vytvorili funkciu *on_ground()*, ktorá vracia hodnotu *true* alebo *false* podľa toho, či hráč stojí na dlaždici.

V polovici vývoja sme sa rozhodli dlaždice smrti dočasne odstrániť, pretože nefungovali podľa našich predstáv. Neskôr sme ich opravili a vrátili do hry, pričom sme ich využili pri tvorbe pascí.

Začali sme navrhovať základné rozloženie levelov. Po dokončení prvého levelu sme sa pustili do kreslenia všetkých animácií nášho hlavného nepriateľa – žaby. Vytvorili sme štyri rôzne animácie, zodpovedajúce štyrom stavom, v ktorých sa žaba môže nachádzať.

Následne sme začali programovať žabu a všetky jej mechaniky. Žaba má štyri základné stavy: nečinnosť, začiatok skoku, skok a pristátie po skoku. V stave nečinnosti sme pridali animáciu dýchania, ktorá sa aktivuje náhodne, čím žaba pôsobí živšie. Implementovali sme tiež mechaniku spôsobovania poškodenia hráčovi a nakoniec sme nepriateľa pridali do hry.

Po pridaní nepriateľov sme sa zamerali na návrh grafického užívateľského rozhrania (GUI). Vytvorili sme HP bar, ktorý sme následne naprogramovali a implementovali do hry. Okrem toho sme pridali pasce so šípami a hroty, ktoré sme neskôr strategicky umiestnili do vybraných leveloch, aby sme zvýšili náročnosť a dynamiku hry.

S funkčným HP barom, hráčom, nepriateľmi a pascami sme mali už takmer všetko potrebné na vytvorenie viacerých levelov. Chýbal nám však spôsob, ako medzi nimi prechádzať. Na tento účel sme sa rozhodli použiť efekt „fade“. Prechádzanie medzi levelmi sme realizovali pomocou objektov *o_fade*, *o_transition* a funkcie *fade_to_room()*. *Fade_to_room()* vytvorí efekt postupného stmavenia obrazovky a následne presunie hráča do cieľovej miestnosti na zadané súradnice. *O_fade* je zodpovedný za vykreslenie efektu stmavenia a následného rozjasnenia obrazovky pri prechode medzi miestnosťami. *O_transition* je spúšťač prechodu medzi miestnosťami. Pri kolízii s hráčom aktivuje efekt prechodu cez *fade_to_room()*.

Dokončili sme prvých deväť levelov. V ôsmom leveli sme však narazili na problém – jedna z prekážok bola natoľko náročná, že ju nebolo možné plynulo prekonať len s dvoma skokmi. Mohli sme level upraviť tak, aby bol jednoduchší, no namiesto toho sme sa rozhodli rozšíriť hernú mechaniku o nový prvok – Magické čižmy. Navrhli sme ich tak, aby ich hráč mohol zdvihnúť tesne pred prekážkou, kde sú potrebné. Po ich získaní mu dočasne umožnia vykonať namiesto dvoch skokov až tri. Pri tvorbe zvyšných levelov sme sa nestretli s príliš veľkými problémami.

Chýbal nám už len desiaty level, v ktorom by hráč po jeho prejdení vyhral hru. Chceli sme ho odlíšiť od ostatných nielen náročnosťou, ale aj novými mechanikami, aby bolo jasné, že nejde o bežnú úroveň. Vytvorili sme preto niekoľko prvkov, ktoré ho robia výnimočným. Nepriatelia sa neobjavia hneď na začiatku, ale až po prechode cez určitý úsek. Taktiež sme pridali kľúč, ktorý musí hráč najskôr získať, aby mohol level dokončiť.

Posledným krokom pri tvorbe našej hry bolo vytvorenie obrazoviek pre úvod, smrť hráča a dokončenie hry. Hlavné menu sme obohatili o vlastný grafický návrh, pri ktorom nám mierne asistovala umelá inteligencia, no finálnu podobu sme si upravili podľa vlastných predstáv. Obrazovky víťazstva a hlavného menu sme umiestnili do samostatných miestností, zatiaľ čo obrazovka smrti sa zobrazuje priamo v miestnosti, kde hráč zomrel

Po dokončení všetkých herných prvkov sme hru otestovali a vyladili posledné detaily. Následne sme ju dali vyskúšať niekoľkým spolužiakom, aby sme získali spätnú väzbu a overili jej hrateľnosť.

.

Záver

Výsledkom tejto maturitnej práce je plne funkčná 2D platformová hra, ktorá naplnila všetky stanovené ciele a požiadavky. Podarilo sa nám vytvoriť hru, ktorá spája premyslené herné mechaniky, zábavné výzvy a vizuálne jednotný dizajn. Hráč má možnosť vychutnať si výzvy, ktoré kladú dôraz na rýchle reakcie, strategické myslenie a neustále zdokonaľovanie schopností.

Počas vývoja sme získali množstvo cenných skúseností v oblasti programovania, návrhu herných mechaník, tvorby grafiky a level dizajnu. Implementácia jednotlivých prvkov, ako sú plynulé animácie, prepracované mechaniky a vyvážená náročnosť, nám umožnila lepšie pochopiť proces tvorby hier a výzvy, ktoré s tým súvisia. Práca s GameMakerom nám poskytla cenné skúsenosti v oblasti vývoja hier a umožnila nám osvojiť si efektívne techniky programovania v jazyku GML.

Vytvorili sme hru, ktorá je prístupná pre široké spektrum hráčov – od tých, ktorí s platformovkami nemajú skúsenosti, až po skúsenejších hráčov, ktorí hľadajú náročnejšie výzvy. Testovanie hry nám ukázalo, že jej mechaniky sú intuitívne a zároveň dostatočne zábavné. Počas testovania sa neobjavili žiadne závažné chyby, a všetky menšie sme opravili.

Ukázalo sa, že dôležité je nielen pridávať nové prvky, ale aj zdokonaľovať existujúce mechaniky a udržiavať celkovú stabilitu hry. V budúcich projektoch by sme sa preto ešte viac zamerali na optimalizáciu kódu a vyváženie hrateľnosti.

Naším cieľom bolo vytvoriť hru, ktorá hráčom prinesie nielen výzvu, ale aj zábavu. Veríme, že sa nám to podarilo a že výsledný produkt dokáže zaujať hráčov svojou hrateľnosťou a atmosférou. Táto skúsenosť nám poskytla pevný základ pre ďalšie kreatívne projekty a prehĺbila náš záujem o vývoj hier.

ZOZNAM BIBLIOGRAFICKÝCH ODKAZOV

- [1] Introduction to GameMaker Dostupné na: [https://manual.gamemaker.io/monthly/en/#t=Introduction%2FIntroduction To GameM
aker Studio 2.htm](https://manual.gamemaker.io/monthly/en/#t=Introduction%2FIntroduction%20To%20GameMaker%20Studio%202.htm) 20.3.2025
- [2] The Room Editor Dostupné na: https://manual.gamemaker.io/monthly/en/The_Asset_Editors/Rooms.htm 20.3.2025
- [3] The Asset Browser Dostupné na: https://manual.gamemaker.io/monthly/en/Introduction/The_Asset_Browser.htm 21.3.2025
- [4] História počítačových hier Dostupné na: [https://historia-pocitacovych-
hier.estranky.sk/clanky/historia-pocitacovych-hier.html](https://historia-pocitacovych-hier.estranky.sk/clanky/historia-pocitacovych-hier.html) 20.3.2025
- [5] Čo je platformová hra? Dostupné na: [https://www.lifewire.com/what-is-a-platform-
game-812371](https://www.lifewire.com/what-is-a-platform-game-812371) 20.3.2025
- [6] Inšpirácia pre tile set Dostupné na: [https://tilation.itch.io/multi-size-mythical-
dungeon-tileset](https://tilation.itch.io/multi-size-mythical-dungeon-tileset) 23.3.2025
- [7] Inšpirácia pre postavu hráča Dostupné na: [https://itch.io/game-assets/tag-knight/tag-
pixel-art](https://itch.io/game-assets/tag-knight/tag-pixel-art) 24.3.2025