



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

MOBILNÍ DEMO SOUND RUN V GODOT ENGINE

MOBILE DEMO SOUND RUN IN GODOT ENGINE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

OLEKSII KORNIENKO

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. TOMÁŠ MILET, Ph.D.

BRNO 2023

Zadání bakalářské práce



146320

Ústav: Ústav počítačové grafiky a multimédií (UPGM)
Student: **Korniienko Oleksii**
Program: Informační technologie
Specializace: Informační technologie
Název: **Mobilní demo Sound Run v Godot engine**
Kategorie: Počítačová grafika
Akademický rok: 2022/23

Zadání:

1. Nastudujte engine Godot a vývoj mobilních aplikací. Prozkoumejte současné hry, které využívají herní mechaniky založené na zvuku.
2. Navrhněte mobilní demo hru, která nevšedním způsobem využívá zvuky pro herní mechaniky.
3. Implementujte navrženou demo hru.
4. Aplikaci otestujte na uživatelích a zhodnoťte.
5. Vytvořte demonstrační video.

Literatura:

- Gregory, Jason. *Game engine architecture*. crc Press, 2018. ISBN 1351974289, 9781351974288
- Bishop, Lars, et al. "Designing a PC game engine." *IEEE Computer Graphics and Applications* 18.1 (1998): 46-53.
- Adams, Ernest, and Joris Dormans. *Game mechanics: advanced game design*. New Riders, 2012. ISBN 0321820274, 9780321820273

Při obhajobě semestrální části projektu je požadováno:

Body 1 a 2 a kostra aplikace.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Milet Tomáš, Ing., Ph.D.**
Vedoucí ústavu: Černocký Jan, prof. Dr. Ing.
Datum zadání: 1.11.2022
Termín pro odevzdání: 31.7.2023
Datum schválení: 13.3.2023

Abstrakt

Cílem práce je vytvoření nové herní demo, ve které hraje zvuk důležitou roli. Výsledkem je hra typu runner, kde se hráč za pomoci zvuků vyhýbá neviditelným překážkám. Překážky jsou různého druhu a každá je reprezentována jiným zvukem. Hra je realizována jako mobilní vzhledem k popularitě dané platformy.

Abstract

The purpose of the work is to create a new game demo in which sound plays an important role. The result is a runner-type game where the player avoids invisible obstacles with the help of sounds. Obstacles are of different types and each is represented by a different sound. The game is implemented as a mobile application due to the popularity of the platform.

Klíčová slova

mobilní, hra, zvuk, godot, runner

Keywords

mobile, game, sound, godot, runner

Citace

KORNIENKO, Oleksii. *Mobilní demo Sound Run v Godot engine*. Brno, 2023. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Tomáš Milet, Ph.D.

Mobilní demo Sound Run v Godot engine

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Mileta. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Oleksii Korniienko

28.07.2023

Poděkování

Chtěl bych poděkovat vedoucímu Ing., Ph.D. Tomáši Miletovi za čas, rady, nápady, trpělivost a zpětnou vazbu. Také za vstřícnost při dohodě na konzultacích a ostatních věcech.

Obsah

1	Úvod	2
2	Vývoj mobilních her	3
2.1	Herní engine	4
2.2	Godot engine	5
2.3	Komponenty hry	7
3	Průzkum trhu	11
3.1	Runnery	11
3.2	Hry s použitím zvuku	14
4	Návrh	16
4.1	Hlavní menu	17
4.2	O hře a Tabulka běhů	18
4.3	Výběr hrdiny a Obchod	19
4.4	Herní scéna	20
5	Implementace	32
5.1	Hlavní menu	35
5.2	Tabulka běhů	37
5.3	O hře	37
5.4	Výběr hrdiny	38
5.5	Obchod	39
5.6	Dynamická část	40
6	Zhodnocení	50
6.1	Zpětná vazba	50
6.2	Analýza zpětné vazby	52
6.3	Možná budoucí rozšíření	53
7	Závěr	54
	Literatura	55

Kapitola 1

Úvod

Tato práce se týká vývoje počítačových a mobilních her. V práci se vysvětluje rozdíl mezi herními motory (dále se bude používat anglické slovo engine) a jejich účely. Dále se vysvětlují typy her, základní principy vývoje a struktura typické mobilní hry.

Výsledkem práce je mobilní aplikace typu „runner“. Již z názvu je patrné, že úkolem hlavního hrdiny je neustále běžet. Základní náplní her tohoto typu jsou: překážky, mince, předměty vylepšující a zhoršující schopnosti postavy – vylepšení a zhoršení. Her tohoto typu je vyvíjeno velmi mnoho, ale málo z nich využívá zvuk pro navigaci. V posledních letech je vývoj mobilních her zaměřen na vydělávání peněz. Hodně vývojářů vytváří napodobeniny známých a populárních her bez přidání něčeho nového. Cílem Sound Run (název vytvořené hry) je přinést novou herní mechaniku do konceptu „runnerů“.

Pro Sound Run je využit engine Godot – engine s otevřeným zdrojovým kódem. Engine je zcela zdarma. Podporuje tři programovací jazyky: C++, C# a Godot Script. Godot Script (dále jen GS) je jazyk založen na jazyku Python.

Cílem této bakalářské práce je herní demo typu „runner“ v engine Godot, ve kterém se člověk řídí hlavně zvukem. Hráč slyší neviditelné překážky, které vydávají zvuk, a proto se jim dokáže vyhnout. Hrdina, který běží neustále vpřed, může vybírat mezi třemi pruhy. Pruh lze vyměnit v libovolný okamžik. Během hry je možné sbírat vylepšení a mince. Také je možné narazit na zhoršení, platformy a překážky.

Díky neobvyklému aspektu, kdy je sluch stejně důležitý jako zrak pro řízení postavy, může práce posloužit jako podklad a inspirace pro podtypy „runnerů“.

Kapitola 2

Vývoj mobilních her

Mobilní telefony se původně neplánovaly jako herní zařízení. První mobilní telefony neměly žádné hry. Byl to přístroj pouze pro volání. Později se začaly objevovat telefony s předinstalovanými hrami, které sloužily jako doplněk – podobně jako například kalkulačka.

Všechno se změnilo s prvním mobilním operačním systémem (dále jen OS) a zavedení protokolu WAP (Wireless Application Protocol, obrázek 2.1), který dal možnost vytvořit online katalog her, kam vývojáři mohli nahrávat svoje hry.



Obrázek 2.1: Anglická Wikipedie ve WAP prohlížeči. Obrázek převzatý z internetu¹

Prvním mobilním OS byl Java ME. Společnost Nokia byla první, která s tím začala počítat. Model Nokia 3410 už měl předinstalované následující hry: Bantumi, Bumper, Link5, Snake II, Space Impact.

Od této doby popularita mobilních her začala růst a mobilní hry už nebyly jen doplňkem. Do vývoje her se zapojily tak velké vývojářské firmy jako Electronic Arts (The Sims 2, Doom, FIFA 06, Tetris, Tetris Mania), Ubisoft (Assassins's Creed) a Konami (Pro Evolution Soccer). Tyto firmy jako první napadlo, že mobilní hry jsou atraktivní investiční prostředí. Jednoduše potřebovaly přenést svoje populární hry na mobilní zařízení. Podařilo se to částečně také díky vývoji 3D grafiky a novým OS jako Android a iOS.

Dalším problémem, na který narazili vývojáři mobilních her, byl OS. Přesněji to, že jich bylo několik. Hry byly vytvořeny pro jeden operační systém (IOS nebo Android). Lidé si

¹https://cs.wikipedia.org/wiki/Wireless_Application_Protocol

mohli kupovat mobilní telefon s ohledem na to, které hry podporuje OS. Tento problém byl ale celkem rychle vyřešen. Vývojáři začali vytvářet hry pro oba operační systémy. Například hra Angry Birds v roce 2009 byla dostupná pro iOS a později i pro Android. Angry Birds je jedna z nejznámějších mobilních her ve světě i v dnešní době². Jednou z nejznámějších her typu „runner“ je Subway Surf. Je to nekonečný runner, kde je nutné sbírat mince a bonusy, a tak se soutěží s ostatními lidmi v žebříčku.

Na rozdíl od počítačových her nebyly mobilní hry většinou zaměřeny na několik hodin nepřerušovaného hraní, ale spíše byly způsobem, jak příjemně strávit čas během čekání. Monetizace se liší od způsobu monetizace zavedeného u PC her. Lidé neradi utrácejí peníze za hru, kterou si zahrají jen párkrát, proto jsou hry většinou zdarma. Zdarma je 93,8 % všech aplikací App Store a 96,7 % Google Play³. Na druhou stranu by vývojáři také rádi získali peníze za svoji práci, proto se objevily jiné druhy monetizace:

- První jsou nákupy ve hře: uživatel nainstaluje hru a bude si moct zahrát několik úrovní zdarma. Potom po něm hra bude chtít, aby zaplatil nějaké peníze (většinou malé), aby se mohl dostat na další úroveň.
- Další možností jsou taktéž nákupy ve hře, ale jiného druhu. Hráč má možnost koupit si jiného hrdinu, za kterého může hrát nebo nějaké doplňky pro svoji zbraň/auto. Nákupy tohoto druhu nejsou povinné. Uživatel může hrát hru bez placení, ale bude mít nějaká omezení.
- Poslední možnost je reklama. Je to velmi populární varianta. Hra je úplně zdarma, ale jednou za několik hráčových neúspěchů nebo uhraných minut se objeví reklama. Velmi často je reklama nepříjemná a obtěžuje uživatele, ale občas může pomoci. Uživatel například dostane bonusy za to, že se na reklamu podívá.

2.1 Herní engine

Termín „herní engine“ byl poprvé použit v polovině devadesátých let. Jednou z her, kde se použil, byla střílečka z pohledu první osoby (FPS) – hra Doom.

Hra Doom byla navržena s jasným oddělením mezi základními softwarovými součástmi (např. vykreslování 3D grafiky, detekce kolizí, audio systém) a herními pravidly a světy, které dohromady tvoří hru.

Užitečnost takového rozdělení se projevila velmi rychle. Vývojáři začali předělávat hry na jiné tím, že základní věci jako 3D vykreslování, pohyb, kolize přebírali a přidali nové textury a mechaniky. Takovým způsobem se objevila úplně nová hra, ale mnohem rychleji.

V dnešní době vývojáři licencují herní engine a používají základní komponenty k vytváření nových her. Tato praxe je mnohem efektivnější než vyvinout všechny hlavní součásti softwaru od začátku, Game Engine Architecture [2, s.11].

Z tohoto důvodu je základ pro téměř jakoukoliv hru herní engine. Mobilní hry nejsou výjimkou. Mezi nejznámější enginey se řadí:

- Unreal Engine;
- Unity;

²[https://angrybirds.fandom.com/wiki/Angry_Birds_\(series\)](https://angrybirds.fandom.com/wiki/Angry_Birds_(series))

³<https://42matters.com/stats>

- Cryengine.

Unreal Engine – tento engine je určený především pro graficky náročné hry. Vývojáři tohoto engine se zaměřili na to, aby jejich produkt zvládl nejnáročnější úkoly související s grafickou stránkou hry. Engine se využívá pro graficky velmi náročné hry.

Cena je 5 % z vydělané částky, když přesáhne milion dolarů, jinak je zdarma. Používá se třeba pro: Fortnite, Sea of Thieves, Batman: Arkham Knight.

Mobilní hry většinou nepotřebují tak výkonné grafické zpracování, proto se obvykle tento engine nepoužívá pro vývoj mobilních her. Avšak možnost vytvářet mobilní hry existuje – příkladem je hra B.o.T.

Unity – engine nabízí svým uživatelům širokou škálu nástrojů a funkcí, které jsou snadno dostupné, i když vývojář není technicky zdatný. Jednou z nejlepších funkcí tohoto nástroje je jeho integrace mezi platformami, která umožňuje vývojářům her přepínat libovolnou herní platformu mezi Android, Blackberry, iOS a Windows. Unity je zdarma při výdělku nižším než 100 000 dolarů ročně. Jinak je nutné platit licenci ve výši 40 dolarů měsíčně, nebo 399 dolarů ročně za každého člena týmu (Unity Plus). Když se přesáhne částka 200 000 dolarů, platí se licence ve výši 150 dolarů měsíčně, nebo 1 800 dolarů ročně za každého člena týmu (Unity Pro). Používá se pro: Ori and The Blind Forest, Shipbreaker, Fall Guys.

Unity se pro vývoj mobilních her používá mnohem častěji než Unreal Engine. Lze říct, že je to nejnámější engine pro tyto účely. Engine není tak dobrý na graficky náročné hry, ale v případě mobilních her to ani není nutné. Jako příklad lze uvést jednu z nejnámějších her na přežití – Last Day on Earth (LDOE).

Pro vývoj LDOE vývojářským studiem Kefir bylo zvoleno Unity. Tímhle studio přeneslo žánr „po apokalypse“ na mobilní zařízení. Pomocí engine byla hra rozšířena pro více hráčů a vytvořeno jednoduché uživatelské rozhraní. Pomocí Unity dokázalo studio zefektivnit herní architekturu a kvalitu kódu. Petr Kostylev [7], umělecký ředitel společnosti Kefir, řekl : „Jedním z důvodů, proč je pro nás Unity tak cenné je, že usnadňuje sebevyjádření a zkoušení nových věcí.“ Kefir soft spustil LDOE za šedesát dva dní. Hra má více jak sto milionů stažení ⁴.

CryEngine - engine poprvé představila známá vývojářská společnost Crytek v první části hry Far Cry. CryEngine je jedním z nejnáročnějších a nejpoužívanějších herních engineů. Jde o složitý a komplexní herní engine, proto uživatelům obvykle chvíli trvá, než jsou schopni jej efektivně použít. Je o něco hůře pochopitelný pro začátečníky, kteří nemají žádné zkušenosti s používáním herních engineů. Platí se 5% z vydělané částky v případě, že roční výdělek přesáhne 5 000 dolarů, v opačném případě je zdarma. Používá se pro: Crysis, Kingdom Come: Deliverance, Hunt: Showdown.

2.2 Godot engine

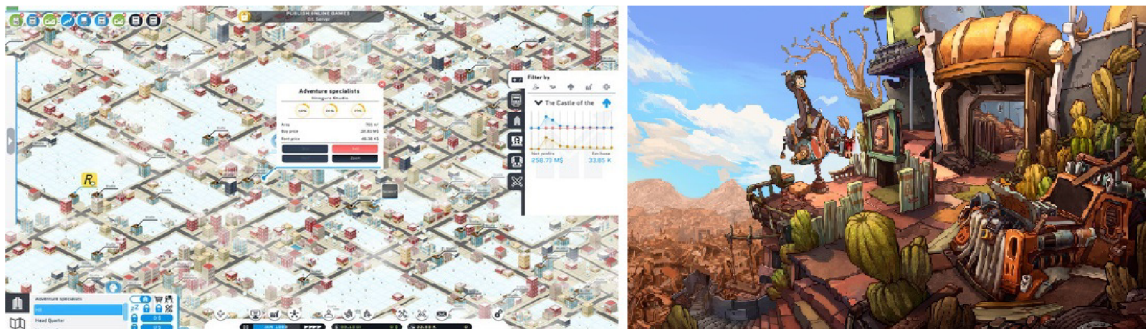
Godot engine je určen především pro studenty a lidi pracující na startupu bez finančního kapitálu. Je to 2D a 3D multiplatformní open source herní engine s licencí od MIT [1].

Citace [5]: „Ve skutečnosti byl Godot jen kódové označení pro něco jiného, engine pro obecnější účel s řádným uživatelským rozhraním namísto sady nejrůznějších nástrojů.“

⁴<https://logicsimplified.com/newgames/unity-game-engine-8-popular-mobile-games-that-rocked-the-game-charts/>

Kvůli tomu, že je Godot relativně nový, neexistují známé hry, které by jej využívaly. Avšak existuje mnoho her, které se prodávají v obchodech Steam / Play Market / App Store, které jsou na něm postaveny. Jedna z takových je City Game Studio, obrázek 2.2.

Pro mobilní zařízení existuje hra Deponia. Původně byla hra vytvořena pro PC, ale vývojáři z německé společnosti Daedalic ji přenesli na iPad, zatím pouze první díl.



Obrázek 2.2: City Game Studio a Deponia. Obrázky převzaty z internetu.⁵

Pro začátečníky je rozhraní velmi jednoduché na pochopení. Většinou lze najít všechno intuitivně. Základní jednotkou engine je uzel. Uzly bývají různých druhů (například Camera a KinematicBody) a každý má vlastní parametry. KinematicBody je určen pro hrdinu – objekt, který se bude hýbat a bude mít kolize, váhu apod. Camera nemá žádnou kolizi a je neviditelná. Ale oba dva jsou uzly mají společné parametry, třeba pozice. Z uzlů se skládají scény. Skripty se používají k ovládání uzlů a scén a programování jejich chování. Většinou je skript spojený s uzlem. Struktura scény je stromová, ze scén se pak skládají další scény a tak se tvoří projekt.

Skriptovací jazyk Godot Script (dále jen GS) je velmi podobný programovacímu jazyku Python. Také je dynamicky typovaný a syntaxe je hodně podobná. Na rozdíl od Pythonu vyžaduje deklaraci proměnných. Nevýhody engine jsou:

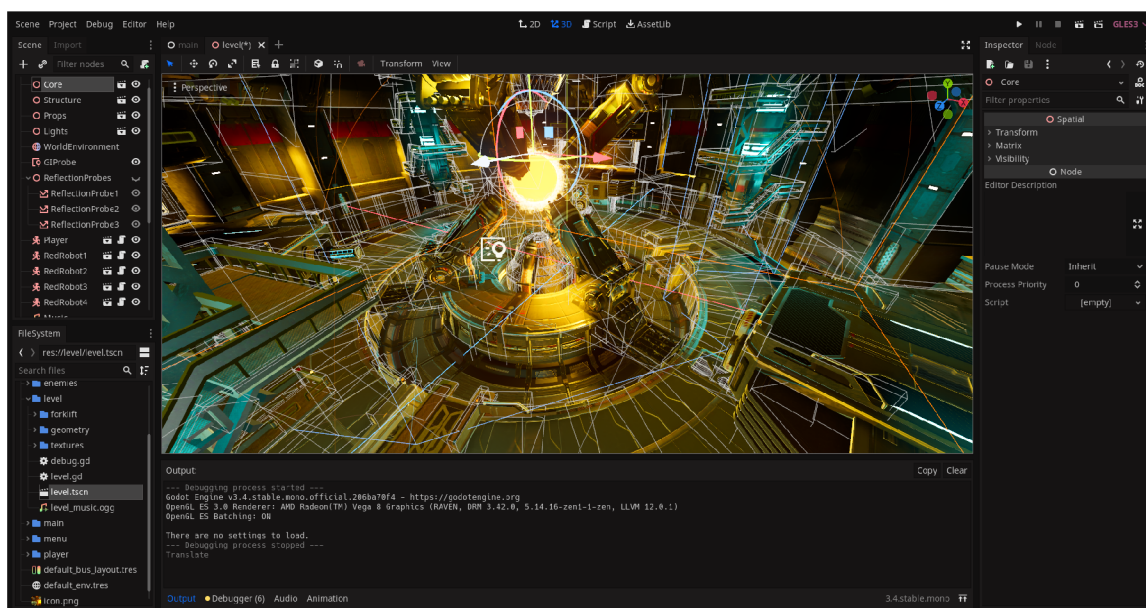
- Mladý projekt s omezenou funkcionalitou;
- Obsahuje chyby, které se řeší delší dobu.

Výhody:

- Jednoduchý jazyk podobný velmi rozšířenému Pythonu;
- Stromová struktura scény, hned je vidět, co je rodič a potomek;
- Open source, vše je zdarma. Je možné se zapojit do vývoje;
- Malá velikost engine i samotných her. Například celá složka Godotu zabírá cca 950 MB, ze kterých cca 210 MB jsou dema;
- Intuitivně pochopitelné a přívětivé GUI, obrázek 2.3. Je možné přesunout elementy GUI podle potřeby uživatele;

⁵https://store.steampowered.com/app/726840/City_Game_Studio_a_tycoon_about_game_dev/
<https://store.steampowered.com/app/214340/Deponia/>

- Možnost přidat vlastní knihovny. Tato možnost částečně vyrovnává nevýhodu omezené funkcionality. Mnoho lidí vytváří vlastní knihovny, které jsou velmi užitečné. Když není nějaká funkcionality v Godotu naimplementováno přímo, je možné najít knihovnu, která ji přidává;
- Godot hned po stažení obsahuje malé projekty, které ukazují nejdůležitější mechaniky nebo metody. Jeden z příkladů je práce s „InAppPurchase“ pro Android. Další užitečné demo je „GUI in 3D“. Jsou i větší dema, jako třeba demo verze celé hry.



Obrázek 2.3: Grafické rozhraní engine. Vlevo nahoře jsou uzly a scény. Vlevo dole je souborový systém. Dole je *output*. Vpravo máme aktuálně vybraný uzel. Rozhraní je možné poskládat tak, jak chce vývojář. Obrázek převzatý z internetu.⁷

2.3 Komponenty hry

Hra se skládá s velkého množství elementů. V této sekci se uvádí některé základní elementy, které jsou součástí skoro každé mobilní hry.

2.3.1 Kamera

Pojmem kamera se v oblasti počítačových her rozumí objekt, který reprezentuje lidské oko a produkuje obraz scény, kterou snímá. Kameře je možné změnit vnitřní parametry, jako jsou pozorovací úhel, poměr stran a zkosení. Orientace a umístění jsou vnější parametry, které jsou řízeny typem kamery, například: kamera z pohledu třetí osoby.

Existuje mnoho typů kamer dle jejich umístění a chování. Základní typy jsou tři: z pohledu první osoby, z pohledu třetí osoby a izometrická kamera. Informace o vlastnostech a druzích kamer lze nalézt v *A Beginner's Guide to Video Games: Perspectives* [8]. Z

⁷<https://cs.wikipedia.org/wiki/Godot>

pohledu první osoby je kamera umístěna přímo v hrdinovi (obvykle v hlavě) a hráč vidí scénu virtuálníma očima. V takovém případě nevidí vlastního hrdinu, ale to, co je před ním.

Kamera z pohledu třetí osoby je odlišný typ. Má různé poddruhy lišící se jejich umístěním a chováním. Jeden z nich je kamera umístěná vzadu. Je to jeden z nejpobulárnějších typů umístění kamery z pohledu třetí osoby. V tomto případě kamera sleduje hrdinu zezadu. Existuje více poddruhů takové kamery: některé je možné točit, jiné posouvat, další jsou fixní. Vyskytují se i kombinace těchto typů. V takovém případě je umožněno jejich přepínání. Je to možné vidět třeba ve hře Need For Speed: Payback, obrázek 2.4. Při stisknutí klávesy se rozmístění kamery mění a hráč se může podívat na auto ze všech stran. Jedny z nejpobulárnějších typů izometrických kamer jsou:

- *Ze vzduchu* – kamera, která je umístěna nad herní plochou. Používá se ve strategických hrách jako Civilization, obrázek 2.4. Obvykle je možné s takovou kamerou posouvat a není připojena k hrdinovi nebo něčemu jinému. Hráč díky velkému pozorovacímu úhlu vidí celou mapu a může ovládat objekty na dvou různých koncích;
- *Ze vzduchu (připojená)*. Existuje a používá se i mírně odlišná verze, a to v RPG hrách jako je například Diablo, obrázek 2.4. Kamera je také vysoko nad herní plochou, ale je připojena k hrdinovi. Hráč může vidět okolí, ale nemůže kamerou hýbat. Ve většině takových her je jen jeden hrdina a není proto potřeba ovládat objekty na opačných koncích mapy.



Obrázek 2.4: Na levém horním obrázku je zobrazena hra Civilization, kde je kamera umístěna vysoko nad herní plochou a je umožněno s ní pohybovat. Vpravo je hra Diablo, kde kamera sleduje postavu. Dole, u hry Need For Speed, je kamera umístěna za autem. Obrázky jsou převzaty z internetu⁹.

⁹https://en.wikipedia.org/wiki/Civilization_VI
[https://en.wikipedia.org/wiki/Diablo_\(video_game\)](https://en.wikipedia.org/wiki/Diablo_(video_game))
<https://klardendum.com/review/disapprove-need-for-speed-payback-and-few-words-on-heat/>

2.3.2 Zvuk

Zvuk je vlnění šířící prostředím. Má amplitudu, frekvenci a vlnovou délku. Molekuly vibrují a vytvářejí vlnu, která přenáší zvukovou energii. Lidské ucho je schopné slyšet v rozmezí 20 Hz až 20 000 Hz. Kmitočty pod 20 Hz se nazývají infrazvukem a nad horní limitem ultrazvukem, jak uvádí Obecné informace o zvuku [13].

Při práci se zvukem je důležité znát pojem sonifikace. Citace: „Sonifikace se definuje jako využití neřečového audia k přenosu informací. Konkrétněji, sonifikace představuje transformaci datových vztahů na vnímané vztahy v akustickém signálu za účelem usnadnění komunikace nebo interpretace.“ Více informací o sonifikaci je uvedeno v Sonification Report: Status of the Field and Research Agenda [4], informace byly čerpány ze zveřejněného dokumentu ¹⁰.

Pro proces přenosu informace zvukem lze použít sonifikaci. Existuje pět kanálů, které se běžně používají v softwaru pro sonifikaci: frekvence, hlasitost, klepání, délka a tónbr (zabarvení). Lidé obvykle slyší vyšší frekvence hlasitěji než nižší. Zvýšení frekvencí přidává zvuku také efekt zvýšení hlasitosti [12, s. 74].

Zvuk ve hře hraje velmi důležitou roli. V některých hrách pomáhá hráčům ponořit se do atmosféry hry. V jiných může být důležitou herní mechanikou, například hráč může slyšet kroky protivráče. Zvuk může také varovat proti nebezpečí. Velmi často hráč dokáže podle typizované melodie zjistit, že na něj čeká někdo za rohem nebo stromem. Více o použití zvuku ve hrách lze nalézt v Designing sound for video games [10].

Se zvukem je možné pracovat různě, například změnit rychlost (tempo) audia nebo výšku tónu. Jsou i jiné způsoby: modulace, efekty vibrato, tremolo a zvukové filtry.

Proces změny charakteru zvuku (označovaného jako nosný signál) pomocí modulujícího signálu se nazývá modulace. Nejčastěji se používá pro přenos signálu na delší vzdálenost. Většinou není původní signál vhodný pro dálkový přenos, proto je nutné jej posílit pomocí modulace.

Další efekt je vibrato. Efekt spočívá v pravidelných a rychlých změnách výšky tónu nebo zpěvu. Jedná se o změny kolem 5 Hz. Podobný efekt je tremolo. Při tomto efektu se mění hlasitost (amplituda) tónu. Často se používají oba efekty zároveň.

Existují i zvukové filtry mezi které se řadí dolní propust, horní propust, pásmová propust apod. Každý má vlastní chování. Například dolní propust (Low-pass filter) se používá když je potřeba propouštět pouze signály s nižší frekvencí než nastavené omezení. Horní propust (High-pass filter) je doplňkem k dolní propusti. Tento filtr má opačné chování a nepropouští signál o nižší než stanovené frekvenci.

2.3.3 Herní mechaniky

Souboru pravidel, dle kterých hráč interaguje s hrou, se říká herní mechaniky. Jedná z nejjednodušších herních mechanik je například skok. Množina herních mechanik bez vizualizace tvoří logickou část hry – herní proces.

Důležitou herní mechanikou je kolize. Kolize ve hře znamená srážku dvou (nebo více) objektů. Aby se hráč mohl srazit s překážkou, dotknout se zbraní nebo běhat po zemi, je nutné využít systém kolizí. Většinou lze vytvořit různé kolekce objektů. Objekty v rámci kolekce spolu mohou interagovat, ale objekty z různých kolekcí ne. Interakce mohou být různých typů. Například když postava narazí na zeď, zastaví se, ale kolize postavy s vodou hrdinu jen zpomalí.

¹⁰<https://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=1443&context=psychfacpub>, s. 4

Různé objekty ve hře mají různé vlastnosti. Postava se od židle liší způsobem interakce se scénou a možnostmi ovládní. Základem pro postavu je třída, která má zabudováno mechaniku pohybu, kterou židle nemá. Herní enginy počítají s tím, že herní vývojáři potřebují vytvořit mnoho různých typů objektů. Proto mají předem připravené šablony s různými vlastnostmi.

Kromě mechanik samotné hry, se kterými hráč interaguje přímo, jsou i takové, o kterých skoro nic neví. Jedna z takových je logika ukládání. Pro offline hry se obvyklé používá ukládání na disk. Uchovávat se může velké množství informací: nastavení, průběh úrovně, umístění postavy a protihráčů apod. Realizace systému ukládání dat se může lišit. Jeden z nejjednodušších systémů načítá data při spuštění hry a ukládá je hned při jejich změně. Nevýhodou je časté přepisování dat, což může mít negativní efekt na stabilitu hry a životnost disku.

Animace herních objektu je další etapou ve vývoji her. Animace většinou slouží jen jako vylepšení vizuální stránky a nemají žádný vliv na pravidla hry. Postavu je možné animovat ručně v 3D editoru pomocí interpolace klíčových snímků pozic končetin. Dalším způsobem tvorby animace je tzn. motion capture, více o tom uvádí Emma Rose [9]. Motion capture vytváří animace například pomocí senzorů a speciálního oděvu, který si oblékne herec. Každý pohyb člověka je ukládán a po manuální korekci je přímo využit pro animaci virtuální postavy. Druhý způsob je náročnější a potřebuje více nástrojů a zařízení, ale animace vypadá mnohem přirozenější.

2.3.4 Ovládání v mobilních hrách

Ovládání v mobilních hrách se liší od ovládání v počítačových hrách. Mobilní zařízení nemají myš a klávesnici, proto je ovládání většinou realizované pomocí gest (tzn. swipes). Gesto začíná dotykem prstu obrazovky, pokračuje tahem a končí sejmutím prstu. Tvar, délka a čas tahu určuje jeho význam. Existuje mnoho různých způsobů použití, například ve hře Angry Birds délka tahu určuje sílu výstřelu.

Další způsoby ovládání umožňují gyroskop. Gyroskop poskytuje informace o poloze mobilního zařízení, které mohou být následně využity pro ovládání hry. Pohyb mobilního zařízení je možné přímo přenést na pohyb virtuálních objektů. Jedna ze známých her, ve které se používá gyroskop, je Doodle Jump. Ve hře je nutné vyskákat co nejvýše. Pomocí náklonu mobilního zařízení vlevo a vpravo je možné přeskakovat z jedné platformy na jinou a nespadnout.

Kapitola 3

Průzkum trhu

Hry typu runner nejsou novinkou. První runner byl vytvořen v roce 1983 a byl to Quest for Tires. První 3D runner se objevil v roce 2011 a jmenoval se Temple Run. Společné pro každý runner je to, že se hlavní hrdina musí neustále pohybovat, například běžet. Ostatní mechaniky jsou specifické pro každou hru. Žánr runner je velmi populární a existuje mnoho her, které jsou na něm založeny. Mezi hry, které posloužily jako inspirace Sound run patří: Subway Surfers, Minion Rush a Temple Run 2.

3.1 Runnery

Subway Surfers Cílem Subway Surfers, obrázek 3.1, je co nejdéle utíkat před policistou a nenarazit do vlaku nebo jiné překážky. Hra začíná tím, že policista uvidí postavu kreslit grafity po vlaku a začne hrdinu pronásledovat. V proti směru běhu hráče jezdí vlaky, kterým se musí vyhýbat. Hráč může narazit na vylepšení, které mu nějakým způsobem pomůžou. Například vylepšení skoku (Super Sneakers) dovolí přeskočit vlak.



Obrázek 3.1: Subway Surfers byla vyvinuta dánskými společnostmi Kiloo a SYBO Games. V roce 2013 vývojáři přidali systém „World Tour“. Každý měsíc se mění město a hráči běhají po známých městech celého světa. Na pravém obrázku je možné vidět mince a bonus, které je možné sebrat. Obrázek převzatý z internetu.²

²<https://nokiapoweruser.com/subway-surfers-for-windows-10-mobile-arrives/>

Minion Rush Illumination, Universal a Gameloft společně vytvořili hru Minion Rush, obrázek 3.2. Jedná se o typický „runner“, kde se hráč musí vyhýbat různým překážkám a sbírat banány. Lokace jsou založené na animovaných filmech: Despicable Me a Minions.



Obrázek 3.2: Minion Rush. Banány figurují jako měna, za kterou lze koupit vylepšení postavy a předměty. Obrázek převzatý z internetu.⁴

⁴<https://www.educationalappstore.com/app/minion-rush>

Temple Run 2 Mechaniky hry Temple Run 2, obrázek 3.3, jsou stejné jako v prvním dílu. Postava běží ruinami, překonává překážky a sbírá mince a bonusy. Čím déle hráč běží, tím větší nabírá rychlost a překážky na dráze jsou obtížnější. Když hlavní hrdina zemře, je možné ji oživit zaplacením ceny nebo prohlížením reklamy. Jako ve většině „runnerů“, hrdina běží jedním ze tří pruhů a překážku může překonat skokem, přisednutím nebo vyhnutím se v jiném pruhu. Ve hře je možné narazit na vylepšení, například štít, který zachrání proti překážkám.



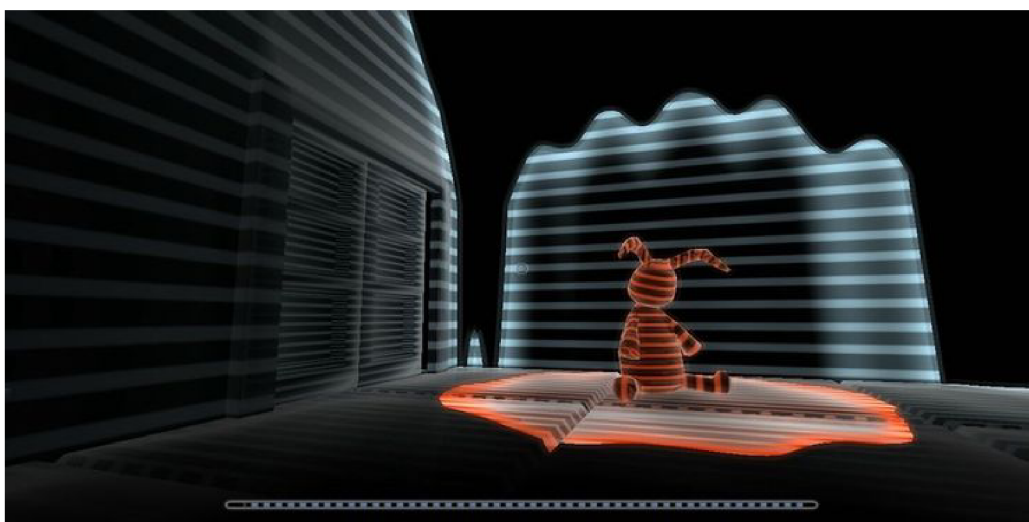
Obrázek 3.3: V roce 2014 byl Temple Run 2 a jeho předchůdce stažen více než miliard krát. Hra se počítá mezi nejstahovanější hry v historii. Ve hře je možné odblokovat postavy známých lidí z celého světa. Například Usain Bolt. Obrázek převzatý z internetu.⁶

⁶https://en.wikipedia.org/wiki/Temple_Run_2

Hry typu runner jsou jedny z nejpobulárnějších typů her pro mobilní zařízení. I když jsou v dnešní době mobilní zařízení schopny spustit i mnohem náročnější hry, pro většinu lidí jsou to stále zařízení, které nejsou určeny k profesionálnímu hraní. Většinou se hraje nějaká jednoduchá hra během přestávky na univerzitě, ve veřejné dopravě nebo ve frontě. Jeden zápas může trvat tak dlouho, jak hráč chce nebo jak dokáže. Nestane se, že uživatel bude muset trávit hrou více času, než by chtěl. Neztratí skóre a není potrestán za to, že hru ukončí dříve, než bylo očekáváno (jako například ve hře Dota 2). Hra může hráče pouze odměnit a do ničeho nenutí.

3.2 Hry s použitím zvuku

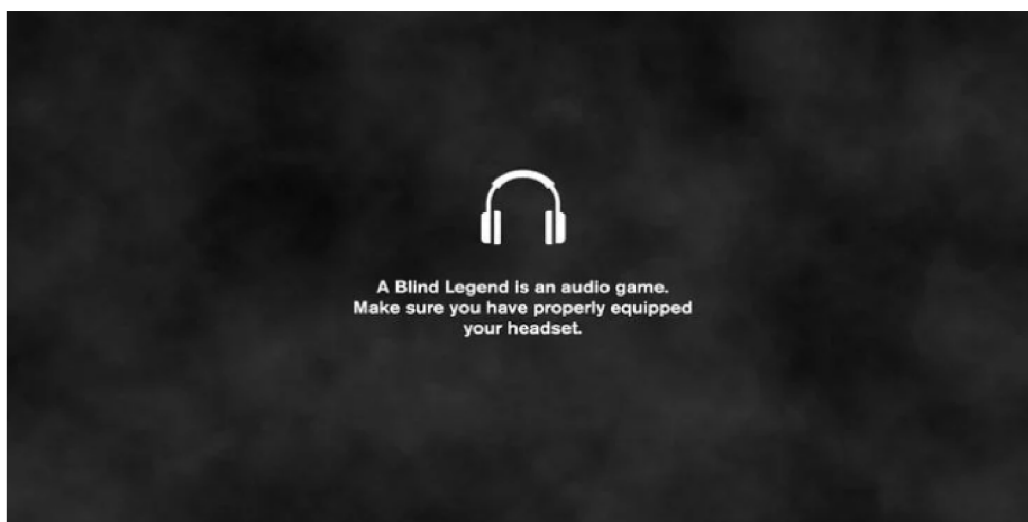
Devil's Tuning Fork je hra, která nevyužívá zvuk jako takový, ale pouze jeho vizuální podobu, obrázek 3.4. Hráč využívá zvuk pro echolokaci – vidí interakce zvuku s prostředím. Hra může být základem a dobrou inspirací pro toho, kdo by realizoval plnohodnotné řízení zvukem.



Obrázek 3.4: Devil's Tuning Fork, vydaná v roce 2009 a založená na fyziologickém procesu echolokace. Je to hororová hra z pohledu první osoby, ve které hráč vidí svět kolem sebe pomocí zvukové vlny.⁸

⁸<https://screenrant.com/best-video-games-sound-significant-gameplay/>

A Blind Legend nemá žádnou vizuální část, obrázek 3.5, hra je celá zaměřená pouze na zvuk. Je to typ her, které se jmenují „Audio Games“. Tento typ her není moc populární, ale může inspirovat lidi, kteří se zabývají vývojem her pro slepce. Lidé nejsou zvyklí na hry, které je možné jen slyšet. Proto nejsou takové hry velmi známé. Je možné kombinovat takový typ her s typem obyčejným a vyvíjet oba zároveň, přičemž takovou kombinací by šlo zvětšit uživatelskou základnu.



Obrázek 3.5: A Blind Legend, vytvořená v roce 2015 s cílem šíření povědomí o zrakovém postižení. Hra je zaměřena hlavně pro slepce a je to první akční dobrodružná hra, která je postavená pouze na zvuku. Příběh hry je o slepém rytíři, který cestuje královstvím, aby zachránil svou ženu.¹⁰

Hry typu runner se liší jenom vizuálně a v nějakých drobnostech, ale samotné mechaniky jsou stejné. Zvukové hry nejsou zajímavé pro lidi bez poruchy zraku, ale nápad používat jiný smysl pro hraní než zrak není běžný. Sound Run je kombinací těchto dvou typů her. Hráč vidí hrdinu, vidí cestu a nějaké vylepšení, ale vyhnout se překážkám může pouze pomocí zvuku.

¹⁰<https://screenrant.com/best-video-games-sound-significant-gameplay/>

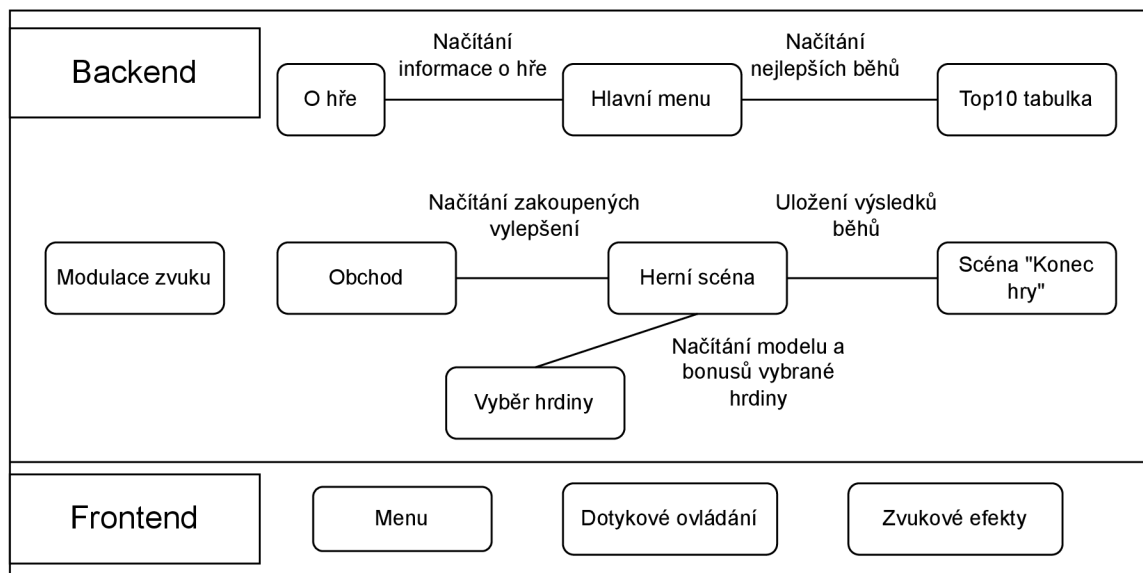
Kapitola 4

Návrh

Cílem práce bylo také do žánru přinést něco nového. Bylo nutné vymyslet něco, co zatím nemá žádná populární hra, a to je použití zvuku v kombinaci se zrakem. Zatím se zvuk nepoužíval pro něco důležitého, většinou se dal vypnout a neměl žádný důležitý vliv na herní proces. Vzhledem k tomu, že uživatelé mobilních zařízení většinou nesedí doma, ale hrají ve společných prostorech, kde by zvuk hry mohl rušit ostatní, vývojáři dělají vše, aby zvuk nebyl nepostradatelný. A v případě, že bude vypnutý, neovlivní hraní.

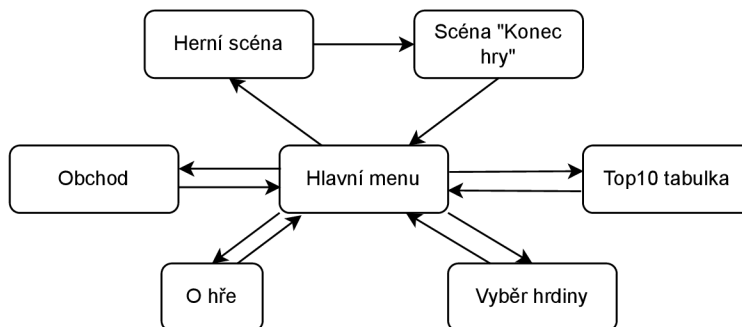
V dnešní době se tendence trochu mění, bezdrátová sluchátka jsou čím dál častější, což je důležité pro Sound Run (dále SR). Problém rušení ostatních zvukem už není tak aktuální. Toto otevírá další možnost pro vývoj mobilních her, kterou využívá SR. Sluchátka dovolují používat zvuk bez strachu ze způsobení nepohodlí ostatním lidem.

Celková struktura hry je naznačena na diagramu 4.1. Hra je rozdělena do pěti větších částí, které si mezi sebou předávají informace.



Obrázek 4.1: Diagram zobrazující hlavní podčásti hry. Hra obsahuje menu, hlavní herní scénu, ve které probíhá samotná hra. Dále obsahuje tabulku nejlepších výsledků, obchod, ve kterém je možné nakoupit různá vylepšení.

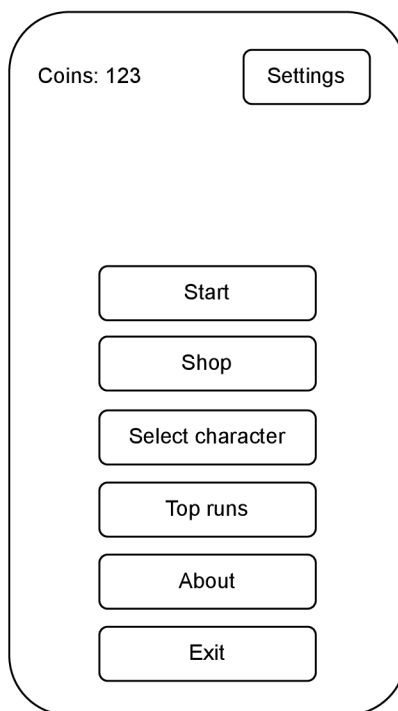
Následující schéma 4.2 ukazuje strukturu a propojení všech herních menu. V dalších sekcích této kapitoly jsou jednotlivé části popsány podrobněji.



Obrázek 4.2: Diagram zobrazující menu hry a přechody.

4.1 Hlavní menu

Hlavní menu slouží ke spojení všech ostatních částí hry. Z hlavního menu je možné přejít na ostatní části a zpět. Schematické zobrazení hlavního menu je na obrázku 4.3

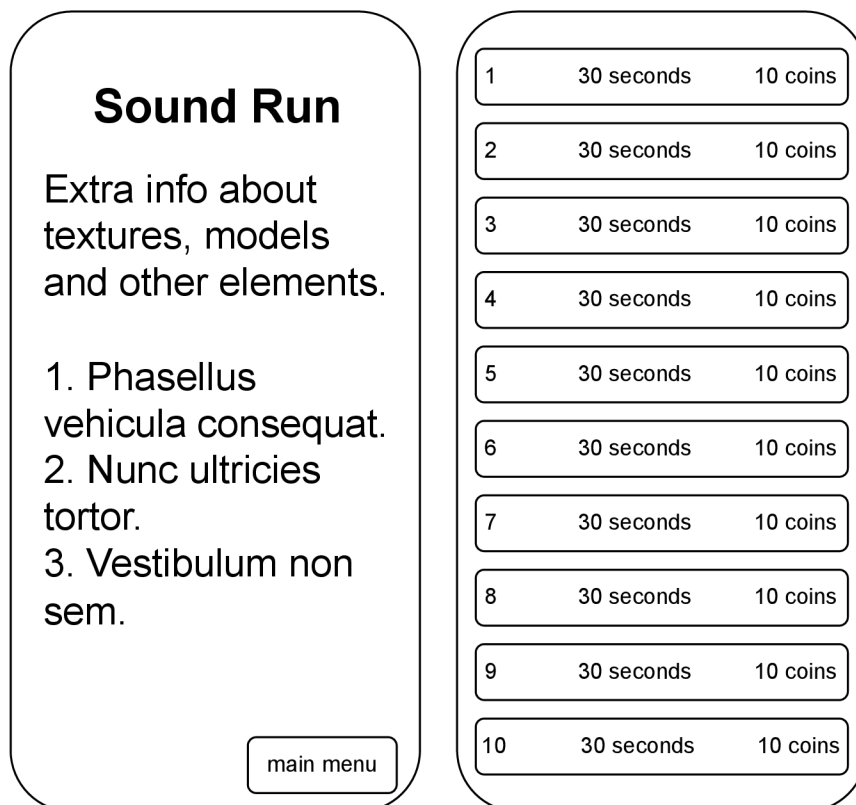


Obrázek 4.3: Schematické zobrazení hlavního menu. V levém horním rohu je umístěn počet mincí a uprostřed scény jsou tlačítka pro navigaci.

4.2 O hře a Tabulka běhů

V této sekci je popsána tabulka deseti nejlepších běhů se záznamy času a nasbíraných mincí a scéna *O hře*. Po dokončení běhu se výsledek uloží jako záznam v tabulce pouze v případě, že není horší než poslední záznam. Po uložení nového záznamu se poslední záznam z tabulky odstraní. Tabulka je seřazena podle času běhu a nasbíraných mincí v případě, že čas je stejný.

Funkcionalita scény *O hře* není spojená se samotným herním procesem. Hlavním účelem je zobrazení informací. V demo verzi hry zobrazuje jenom seznam převzatých elementů, jejich zdroj, autora apod.

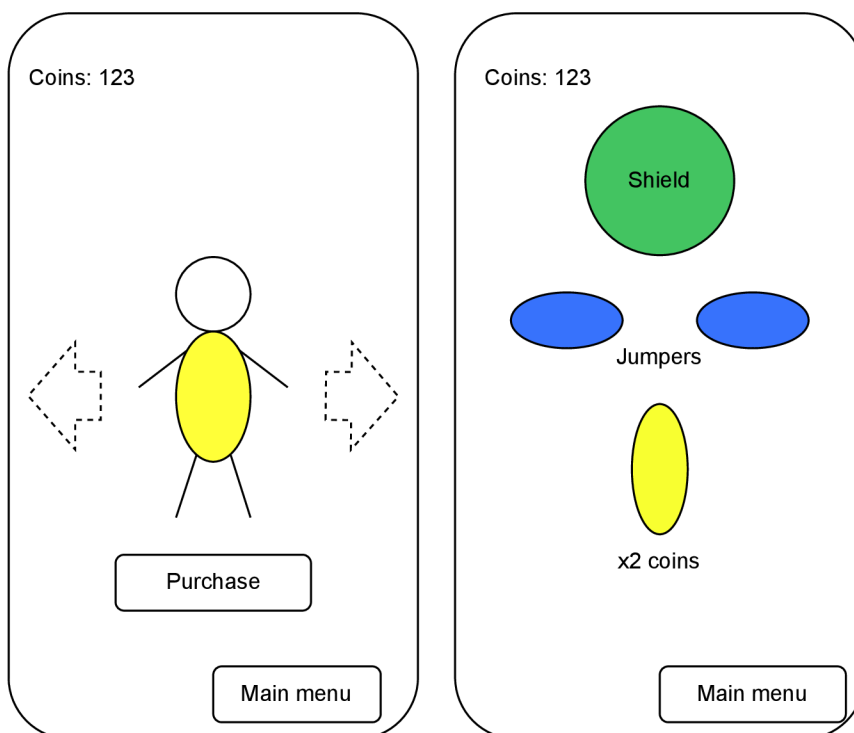


Obrázek 4.4: Schematické zobrazení tabulky běhu vpravo a scény *O hře* vlevo. Tabulka umísťuje záznamy jednotlivých běhů. *O hře* má předem připravené záznamy jednotlivých převzatých elementů a další informací o nich.

4.3 Výběr hrdiny a Obchod

Hry typu runner nemají různé úrovně s odlišnými mechanikami. Úrovně sice mohou existovat, ale je to jen změna ve vizuální stránce. Zajímavosti herního procesu se přidávají jinými způsoby. V Sound Runu si hráč může koupit jiné hrdiny. Základní hrdina nemá žádný bonus, na rozdíl od jiných. Liší se bonusem a zevnějškem.

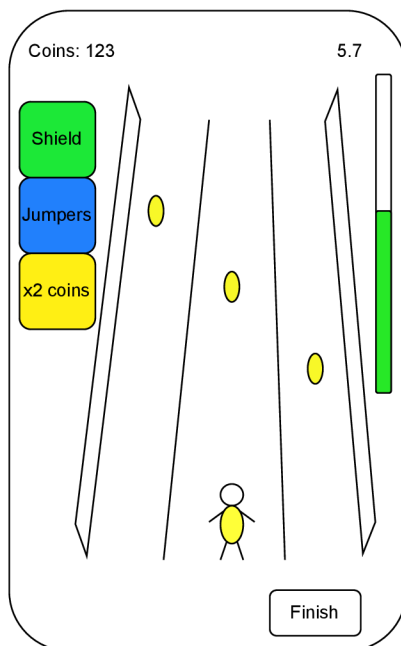
Obchod se používá pro zakoupení vylepšení pomocí mincí, které hráč nasbíral. Je možné tady koupit všechna vylepšení, která jsou ve hře.



Obrázek 4.5: Schematické zobrazení výběru hrdiny vlevo a obchodu vpravo. Pomocí gesta, znázorněného čárkovanou čarou, se přepíná mezi hrdiny. Obchod slouží pro zakoupení vylepšení pro ulehčení běhu. Existují tři druhy vylepšení: Štít, Vylepšení skoku a Dvojnásobek mincí. Kliknutím na vylepšení se provede zakoupení.

4.4 Herní scéna

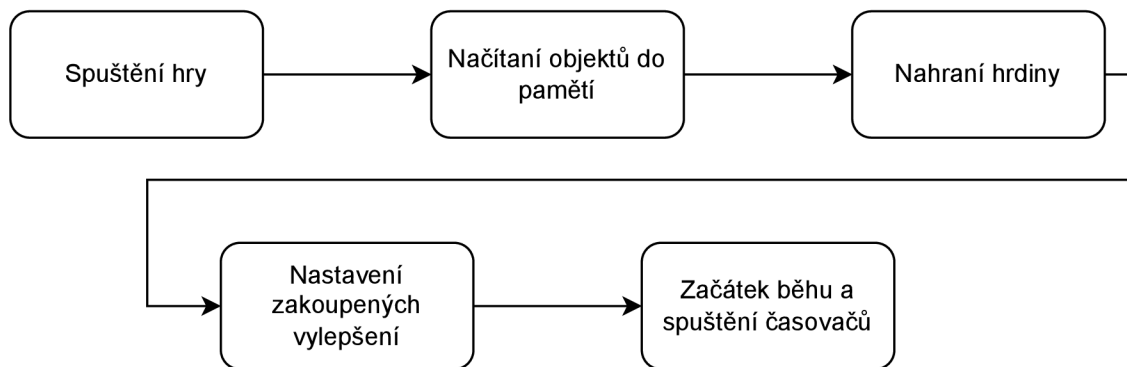
SR, jako i většina runnerů, obsahuje hrdinu, který nekonečně běží. Hra nabízí tři pruhy, mezi kterými je možné se přemístit v jakýkoliv okamžik. Na pruzích jsou umístěny překážky různých druhů. Překážkám se lze vyhnout třemi způsoby: přejít na jiný pruh, skočit a přikrčit se. Čím delší dobu dokáže hráč přežít, tím rychleji bude běžet. Dalším elementem hry je platforma. Rozdíl oproti ostatním runnerům je ten, že jsou překážky neviditelné.



Obrázek 4.6: Diagram zobrazuje hlavní scénu, kde se odehrává většina času hry. Hlavní scéna obsahuje tři pruhy, překážky, platformy, vylepšení a zhoršení a postavičku, která běží v před.

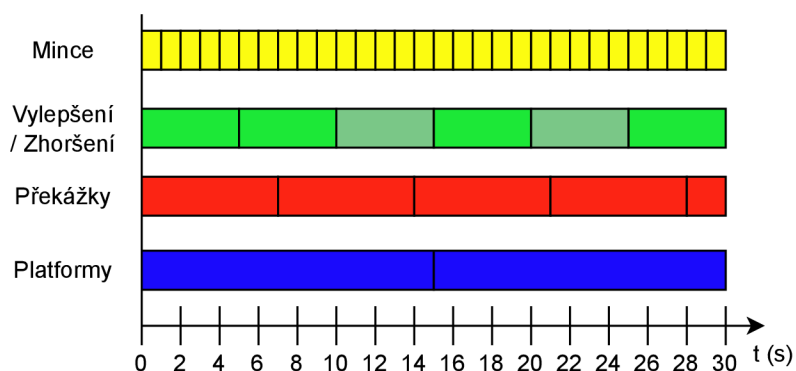
Sound Run využívá zvuk pro signalizaci překážky hráči. Na základě toho, jaký zvuk jakou rychlostí je přehráván, dokáže hráč určit, jak daleko od něj překážka je a jakého je typu. Z tohoto důvodu jsou překážky neviditelné. Cílem je nasbírat co nejvíce mincí a co nejdéle běžet.

Hra začíná tím, že uživatel hru spustí v hlavním menu. Z disku se načtou všechny potřebné objekty do paměti. Pak se nahraje zvolený hrdina a nastaví se vylepšení, která hráč koupil, obrázek 4.7.



Obrázek 4.7: Hlavní etapy spuštění hry.

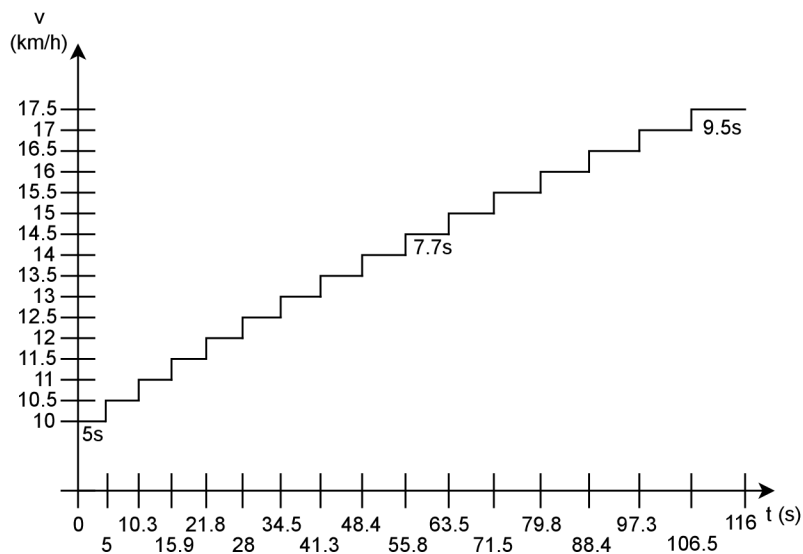
Po nahrání začíná samotný běh. Hned po začátku se spouští všechny časovače, které jsou zodpovědné za generaci elementů hry: mince, platformy, překážky, vylepšení, obrázek 4.8. Po skončení každého časovače (každý má vlastní prodlevu), se provede generace příslušného objektu a časovač se spustí znovu. Objekty se generují ve vzdálenosti od hrdiny, kdy ještě nejsou vidět. Toto navozuje iluzi, že objekty nevznikají, ale byly vždy přítomny na dráze díky tomu, že kamera nemá nekonečný dohled.



Obrázek 4.8: Obrázek zobrazuje časovače hry. Každý časovač slouží pro generování jiného objektu. Po uplynutí časového intervalu se spustí proces generování objektu a časovač se restartuje. Na rozdíl od ostatních, vylepšení a zhoršení se generují s šancí 10%.

Jednou ze základních mechanik runneru je zrychlení. V Sound Runu má hrdina počáteční rychlost deset kilometrů za hodinu. Na obrazovce se zobrazuje progress bar zrychlení, který značí, kdy se navýší rychlost. Jakmile je progress bar plný, rychlost se u obyčejného hrdiny zvýší o půl kilometru v hodině. Progress bar funguje jako časovač, který jednou za uplynulý interval zvýší rychlost hrdiny. S časem se tento interval prodlužuje. Počáteční délka intervalu je pět sekund. Interval se prodlužuje o 300 milisekund při každém zrychlení. To znamená, že čím déle hráč běží, tím pomaleji zrychluje, viz graf 4.9.

Naučení hry netrvá dlouho dobu, zvuky jsou od sebe dostatečně odlišné a po několika bězích by měl být hráč schopen předvídat překážky dle jejich zvuku. Náročnější je pochopit, jak daleko překážka je. Kvůli tomu, že je to nová mechanika, v prvních bězích se lidé ztrácejí. Jakmile uslyší zvuk, snaží se okamžitě zareagovat, což není správně, protože překážka je

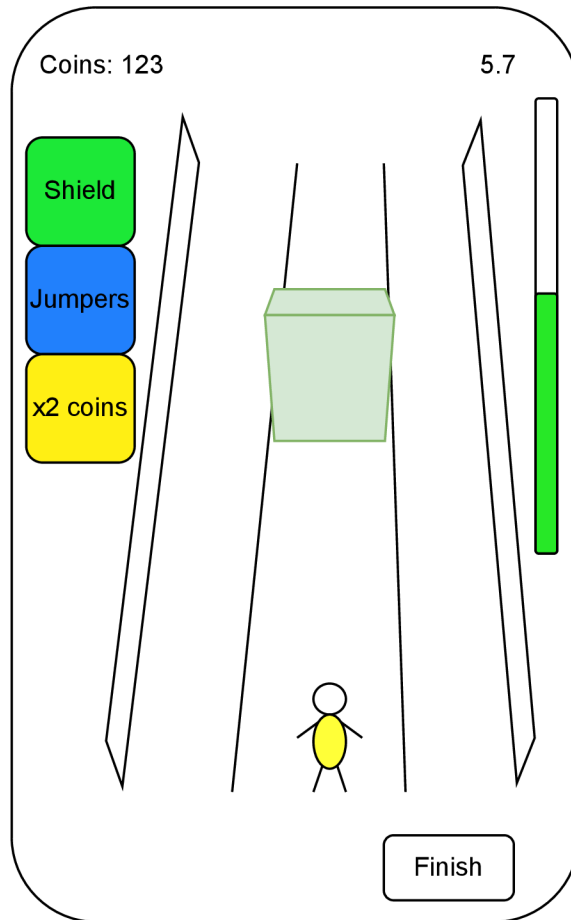


Obrázek 4.9: Graf rychlosti hrdiny. Každé další navýšení rychlosti je zpožděno o 0.3 sekundy. Rychlost se zvyšuje o 0.5 km/h

ještě daleko. Po krátkém výcviku se člověk přizpůsobí a ti nejzkušenější hráči dokážou přeskóčit i to, co není téměř možné přeskóčit, obrázek 4.10.

Tento druh překážky omezuje pouze jeden pruh, většinou je možné jej jenom obejít. Teoreticky je možné tuto překážku i přeskóčit, ale prakticky je to velmi náročné. Jedná se o typ, který je výzvou i pro nejzkušenější, aby se ji snažili překonat náročným způsobem.

Účelem práce není plnohodnotná hra, ale demonstrace mechanik a jejich využití. Z toho důvodu nemá hra skoro žádné vizuální efekty. Hrdina a veškeré modely/textury budou vyměněny v případě vydání hry. V současném stavu obsahuje jen pár vizuálních objektů, obrázek 4.11.

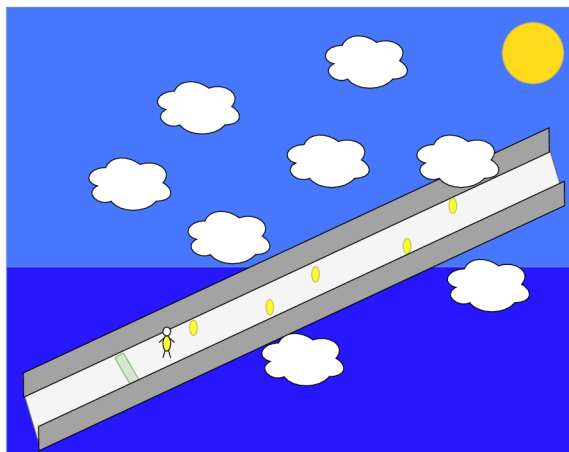


Obrázek 4.10: Jeden z druhů překážek. Vysoká překážka omezující jeden pruh.

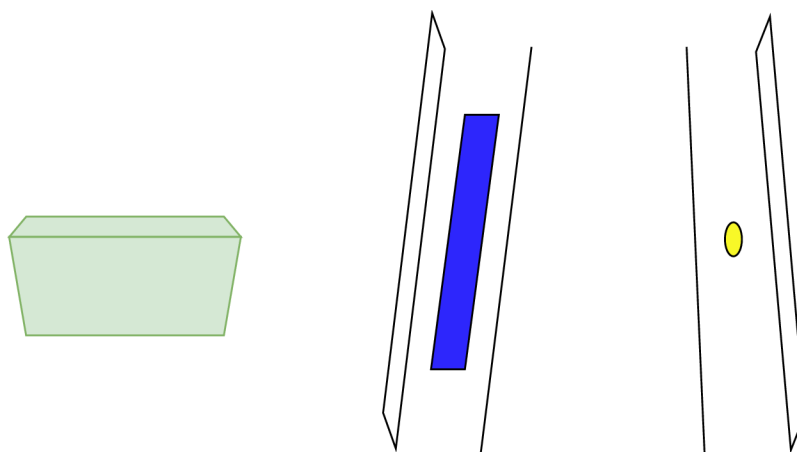
4.4.1 Komponenty hlavní scény

Hra obsahuje několik typů komponent. Například: základní objekty dráhy, vylepšení, zhoršení, kamera, zvuky apod. Základní objekty dráhy jsou na obrázku 4.12 a zahrnují objekty:

- Překážka – když do ní hrdina narazí, zemře nebo ztratí jeden štít, pokud ho má;
- Platforma – přední část je nebezpečná, hrdina zemře, když do platformy narazí ze předu. Je ale možné na ní stát;
- Mince – herní peníze. Sbírají se během závodu a je možné je utrácet za vylepšení nebo nové hrdiny s bonusy. Cena mincí roste s rostoucí rychlostí hráče. Každá mince má hodnotu jeden bod a každý přírůstek rychlosti o deset kilometrů za hodinu přidá další.



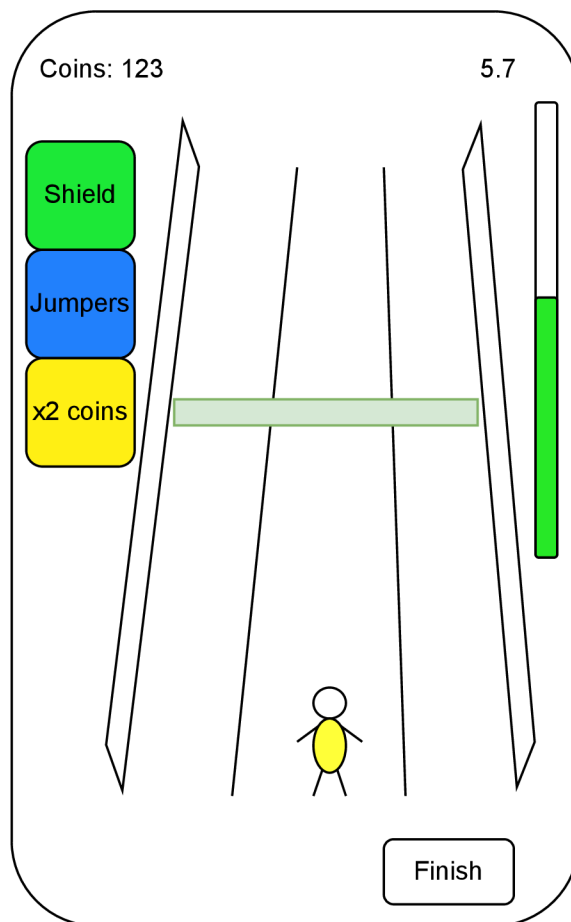
Obrázek 4.11: Schematické zobrazení hry. Vizuální část zahrnuje zemi, stěny, mraky a pozadí.



Obrázek 4.12: Vlevo na obrázku je zobrazena překážka, v centru je platforma a vpravo je mince.

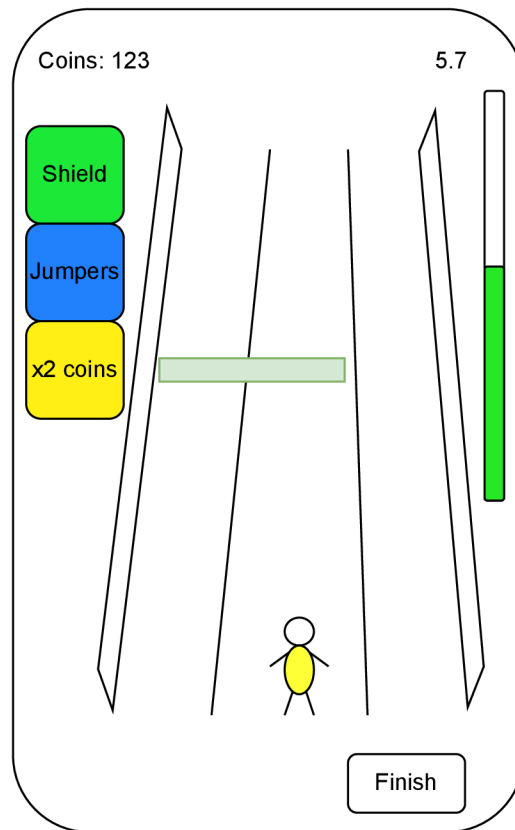
Překážky jsou rozmístěny na dráze. Pochopit jejich rozmístění a typ je možné pomocí zvuku. Každá překážka má vlastní zvuk. Čím blíže je hrdina k překážce, tím více se zvuk zrychluje. Rozlišit typ bariéry je náročné jenom na začátku. Existují čtyři různé překážky:

Pozemní překážka je překážka, která je umístěna na zemi přes celou cestu, obrázek 4.13. Takovou překážku je možné pouze přeskocit. Je nejjednodušším typem bariéry.



Obrázek 4.13: Překážka přes celou zem. Nejznámější druh překážek v téměř jakémkoliv runneru.

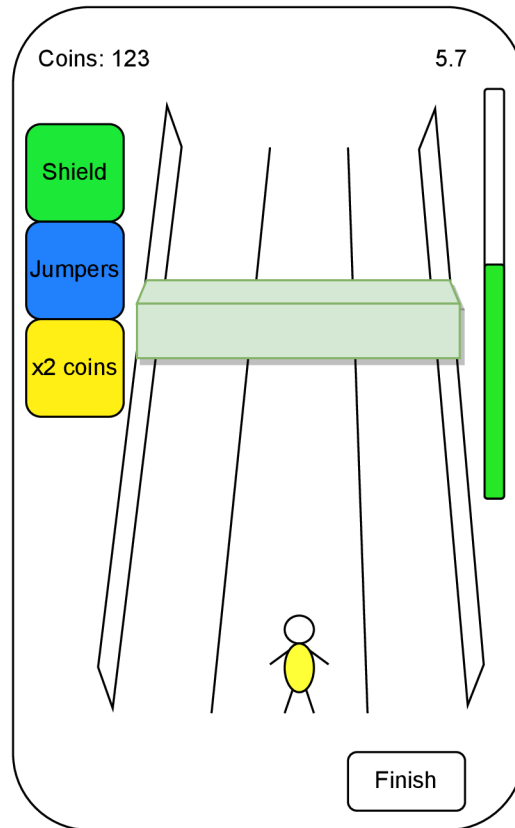
Zkrácená pozemní překážka je další bariéra, která je velmi podobná předchozí, obrázek 4.14. Také je na zemi a vypadá téměř stejně, pouze není přes celou mapu, ale jen na dvou třetinách. To znamená, že je možné ji nejen přeskočit, ale i obejít. Tohle je jediná překážka, která nemá vlastní zvuk. Z důvodu, že způsob překonání je stejný jako u předchozí, tak má i stejné audio. Odlišit od pozemní překážky je možné tak, že se zvuk nepřehrává, když je hráč před prázdnou třetinou.



Obrázek 4.14: Překážka přes dvě třetiny země. Další variace předchozí bariéry.

Nadzemní překážka bariéra je odlišná, obrázek 4.15. Je to největší překážka ve hře. Je stejně široká jako první překážka – přes celou cestu, ale je umístěna nad cestou. Toto je jediná překážka, které se bez vylepšení nebo platformy nelze vyhnout jinak než příkrčením. Když má hráč vylepšení skoku nebo stojí na platformě, dokáže tuto překážku přeskočit.

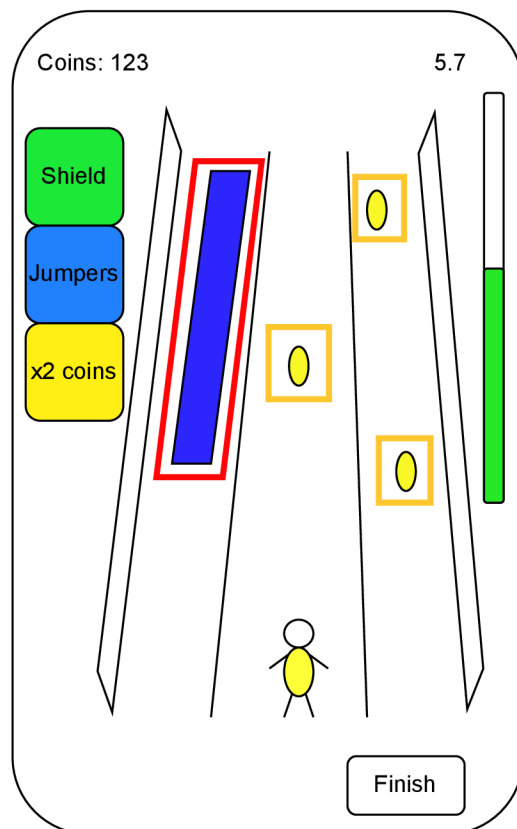
Jedno pruhová překážka je překážka, která omezuje jednu linku a hráč se jí musí vyhnout. Více v návrhu 4.4.



Obrázek 4.15: Překážka ve vzduchu.

Platforma je další komponenta umístěná na dráze, obrázek 4.16. Lze na ni narazit zepředu, což se rovná smrti. Pomocí platformy je možné přeskočit překážky, které obvykle nelze přeskočit. Například bariéru *Nadzemní překážka*, protože na platformě se hráč nachází výše, než obvykle a normální výška skoku je dostatečná.

Platforma může hráče i zachránit. Když po ní hráč běží, nemusí se vyhýbat pozemním překážkám. Platforma je dostatečně vysoká, aby se prvním druhům překážek vyhnul (*Pozemní překážka* a *Zkrácená pozemní překážka*). Takže je to spíše pomoc než způsob, jak ztížit hru. Platforma žádným způsobem neovlivňuje sbírání mincí nebo vylepšení. Ty je možné sbírat i na platformě.



Obrázek 4.16: Platforma a mince. Červenou barvou je vyznačená platforma, oranžovou mince.

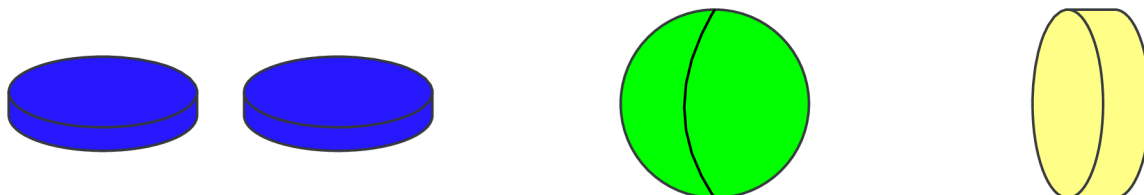
Na cestě se objevují vylepšení a zhoršení, které může hráč sbírat, obrázek vylepšení 4.17, obrázek zhoršení 4.19. Při jejich sebrání se na hráče aplikují různé efekty obrázek vylepšení 4.18, obrázek zhoršení 4.20. Většinu efektů stejného typu není možné aplikovat několikrát, je aktivní maximálně jednou. Výjimku tvoří štít. Efekt většinou trvá určitou dobu (5 sekund).

Skoková vylepšení dává možnost hrdinovi skočit mnohem výš než obvykle. Skoková vylepšení umožňují jiné způsoby zdolávání překážek a ovlivňují průběh hry. Kromě toho, že značně ulehčují přeskočení jakékoliv překážky, hráč s nimi může přeskočit i to, co dříve nebylo možné. Příkladem je *Nadzemní překážka*. Na první pohled je to velmi užitečné vylepšení, ale často kvůli tomu hráč ztratí život, protože riskuje. Když je vylepšení aktivní, model vylepšení je animován pohybem nahoru a dolů.

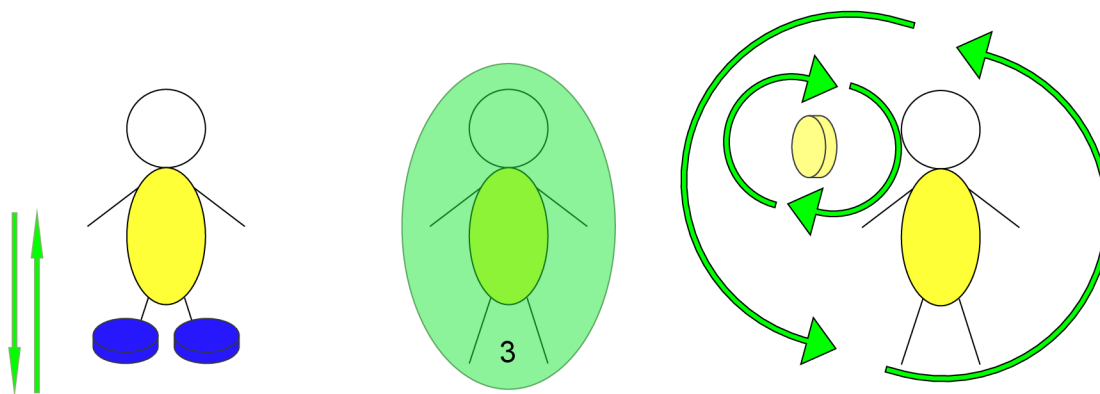
Mince $\times 2$ je vylepšení, které násobí množství sebraných mincí. V případě zakoupení tohoto vylepšení v obchodě, je lepší počkat, až hrdina nabere rychlost. Například, když je rychlost hrdiny deset kilometrů za hodinu, každá mince má hodnotu dva body, ale s aktivovaným vylepšením čtyři body. Model tohoto vylepšení má kombinovanou animaci. Otáčí se kolem vlastní osy a zároveň kolem hrdiny opačnými směry.

Štít je podobný konceptu životů ve většině her. Kolem hrdiny se objeví štít, který chrání před střetnutím s překážkou. Vylepšení je aktivní, dokud ho hráč neztratí. Štíty se akumulují a je možné vidět jejich aktuální počet. Hráč začíná jen s jedním a postupně může posbírat další. Štít nechrání před střetnutím s přední částí platformy. Srážka s přední částí

platformy znamená konec hry v jakémkoliv případě. Po konci hry všechny štíty zmizí a další běh se spustí bez nich.



Obrázek 4.17: Obrázek zobrazuje všechny bonusy, které je možné posbírat v průběhu hry. V takové podobě se zobrazují na cestě. Vlevo je *Skokové vylepšení*, uprostřed je *Štít*, vpravo je *Mince × 2*.

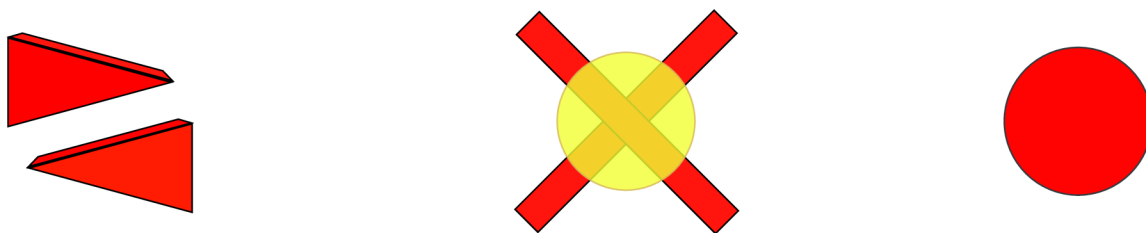


Obrázek 4.18: Obrázek zobrazuje vylepšení při jejich aplikaci na hráče. Aktivované vylepšení je reprezentováno zvláštním modelem a animací (výjimka je štít, nemá žádnou animaci). Animované jsou tak, že model vylepšení se posouvá směrem šipek. Vlevo je *Skokové vylepšení*, uprostřed je *Štít* a vpravo je *Mince × 2*.

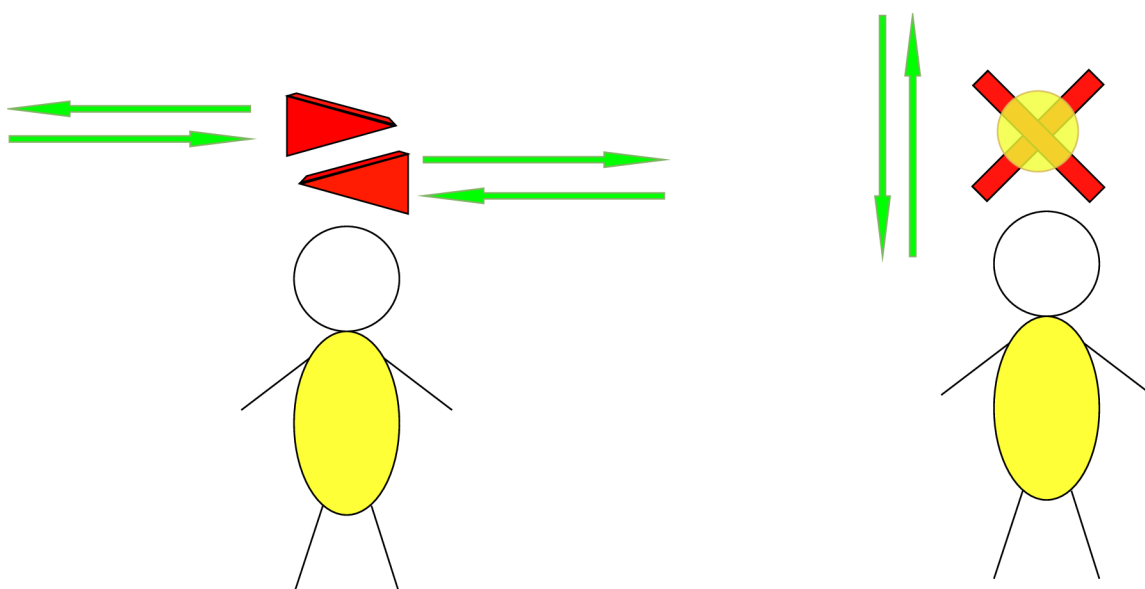
Bez mincí je zhoršení, které během trvání nedovoluje sběr mincí. Po ukončení času zhoršení zmizí. Samo o sobě není nebezpečné a neohrožuje životy hrdiny. Animace je stejná jako u *Skokového vylepšení*, ale model je umístěn nad hrdinou.

Přehození řízení je velmi nepříjemné zhoršení pro začátečníky. Mění efekty gest na opačné. To znamená, že gesto vlevo znamená pohyb vpravo apod. Týká se to pohybu vlevo a vpravo, skoku a příkrčení. Zkušenější hráči jsou na toto zhoršení zvyklí a při menší rychlosti pro ně není problém. Když je zhoršení aktivní, nad hráčem se pohybují šipky. Ty se k sobě přibližují a oddalují.

Ztráta štítu je nejsilnější a zároveň nejslabší zhoršení. V případě, že se hráč toho zhoršení dotkne, ztratí všechny štíty, které má. Když hráč žádný štít nemá, tak se nic nestane. Z větší vzdálenosti se toto zhoršení maskuje a vypadá jako štít. Když se hráč dostane na určitou vzdálenost, model zhoršení změní barvu. Toto zhoršení se aktivuje hned a nezobrazuje se na hrdinovi.



Obrázek 4.19: Obrázek zobrazuje všechny zhoršení, které je možné posbírat v průběhu hry. V takové podobě se zobrazují na cestě. Vlevo je *Přehození řízení*, uprostřed je *Bez mincí* a vpravo je *Ztráta štítů*.



Obrázek 4.20: Obrázek zobrazuje zhoršení při jejich aplikaci na hráče. Aktivované zhoršení je reprezentováno zvláštním modelem a animací. Animované jsou tak, že model zhoršení se posouvá směrem šipek. Vlevo je *Přehození řízení* a vpravo je *Bez mincí*.

4.4.2 Kamera

Pro Sound Run byla zvolena kamera z pohledu třetí osoby – třetí typ z těch, které jsou vysvětleny v sekci 2.3.1 – kamera umístěná vzadu, obrázek 4.15.

Takový druh kamery je nejčastější pro hry typu „runner“. Hráč vidí svého hrdinu a všechno, co je kolem něho. Umístění kamery nad hrdinou zvětšuje vzdálenost do které hráč vidí. To umožňuje hráči naplánovat svou trasu. Pro hru je dobré, že s kamerou nemůže hráč manipulovat. To znamená, že všechno, co je za kamerou je možné smazat z paměti. Nebude to mít žádný špatný vliv na herní proces, ale zvýší to výkon. V případě mobilních zařízení je šetření výkonem velmi důležité. Čím horší je optimalizace, tím rychleji se vypotřebuje baterie.

4.4.3 Zvuk

Demo verze Sound Run obsahuje tři různé zvuky. Zvuky jsou opakovatelné a nemění se s časem. Použití zvuku jako herní mechaniky pro navigaci je inovativní, proto musí být co nejjednodušší na pochopení. Zvuky nesmí být nepříjemné a musí být od sebe dostatečně odlišné. Hráč má během hry málo času na jejich rozlišení.

Godot umožňuje měnit rychlost přehrávání audia, což zároveň mění výšku tónu. Změna tempa je důležitá, protože je díky ní možné pochopit, jak blízko překážka je. Blízké překážky vydávají vyšší tón.

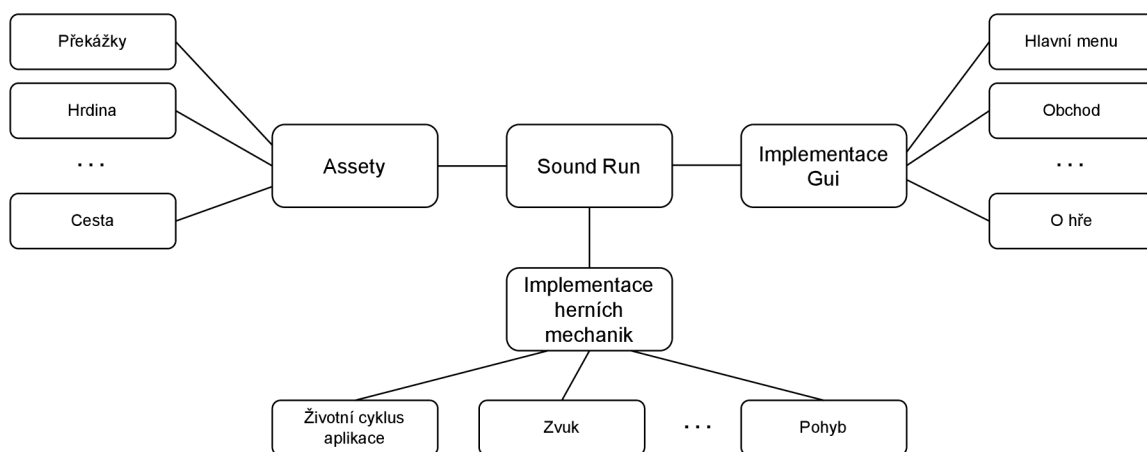
4.4.4 Režimy hry

Aby si hráč měl možnost zvyknout na zvuky a jejich efekty při přiblížení k překážce, je ve hře zahrnut i výukový režim. Od klasického režimu se liší tím, že se překážky stávají viditelné v okamžiku, kdy se začne přehrávat zvuk. V tomto režimu hráč nemůže sbírat mince a soutěžit o místo v tabulce nejlepších běhu, jde pouze o výcvik. Díky výukovému režimu je po několika pokusech mnohem snazší začít hrát klasický režim. Výukový režim je dostupný v jakýkoliv okamžik a je možné ho hrát i místo klasického.

Kapitola 5

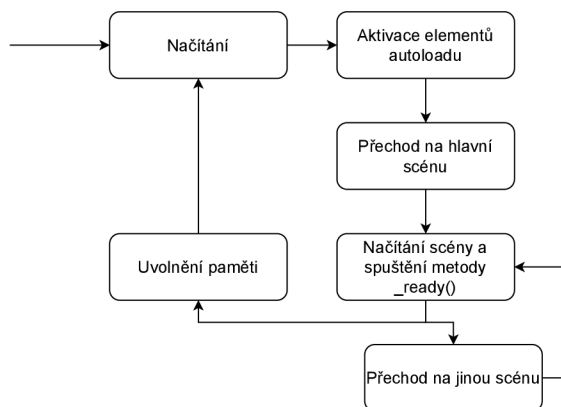
Implementace

Realizace je rozdělená na tři základní části: implementace GUI, implementace herních mechanik a vizuální část (assets). Rozdělení je zobrazeno na obrázku 5.1



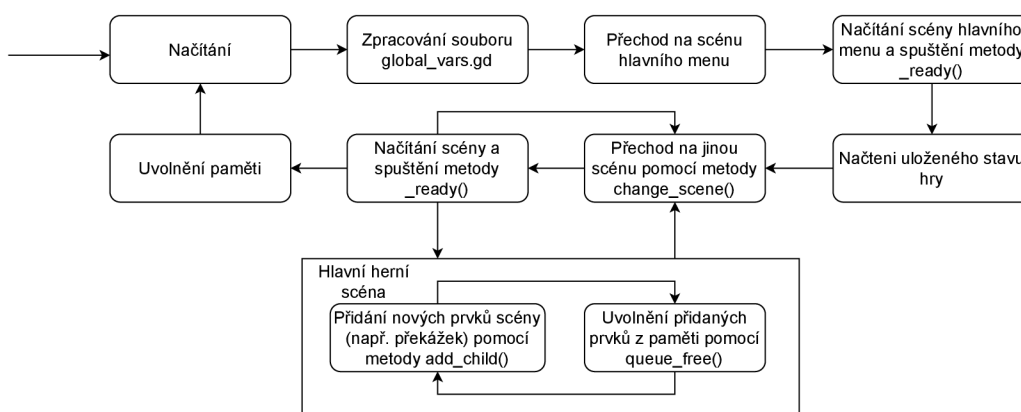
Obrázek 5.1: Diagram základních implementačních elementů aplikace Sound Run.

Celý projekt se skládá ze scén. Scénou se rozumí buď všechny objekty nebo jen nějaká jejich podčást. Chování scén je naprogramováno ve skriptech. Hlavní metodou téměř každého skriptu je `_ready()`. Metoda se spustí automaticky, jakmile se nahraje skript. Scéna se může skládat z jiných scén. Přepínání mezi scénami je možné pomocí metody `change_scene()`. Tato metoda kompletně změní scénu, to znamená, že předchozí scéna bude promazána z paměti. Je to důležité, protože mobilní zařízení zatím nemají tak velkou paměť jako počítač. Životní cyklus obecné aplikace v Godotu je vizualizován na obrázku 5.2.



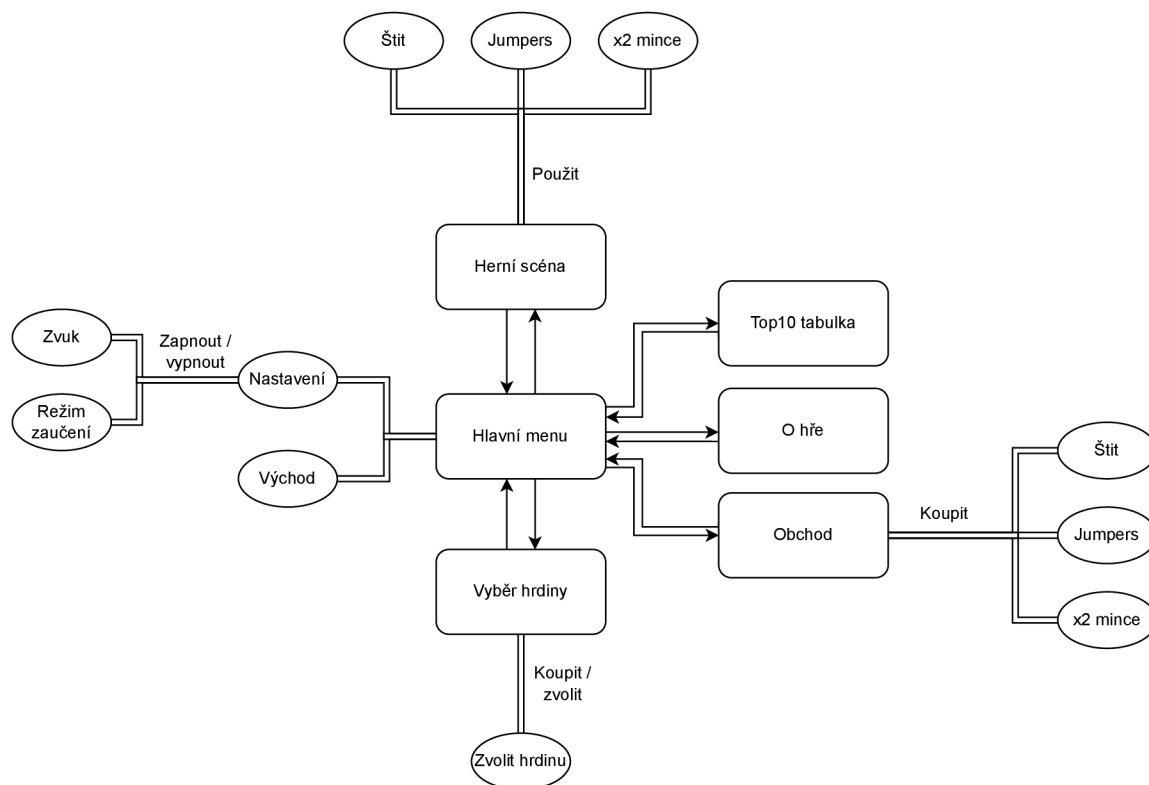
Obrázek 5.2: Životní cyklus obecné aplikace v Godotu. Na diagramu jsou zobrazeny základní etapy. Godot má možnost nastavit seznam *autoload*. Vše uvedené v tomto seznamu je zpracováváno hned po spuštění aplikace a je k dispozici bez ohledu na aktuální scénu.

Životní cyklus Sound Runu je podobný základnímu, ale je rozsáhlejší. Diagram životního cyklu je zobrazen na obrázku 5.3.



Obrázek 5.3: Životní cyklus Sound Runu. V projektu Sound Run se používá *autoload* pro ukládání globálních proměnných, konstant apod. V případě Sound Runu je na to určený soubor *global_vars.gd*. Do *autoload* je možné přidat i scény nebo uzly. Může se používat pro spuštění hudby na pozadí. V takovém případě se hudba bude přehrávat bez ohledu na aktuální scénu. Stav hry se ukládá v souborech a načítá se při každém spuštění hry. Zvláštní částí hry je herní scéna. V ní probíhá hlavní děj, kde se odstraňují a přidávají různé objekty za běhu.

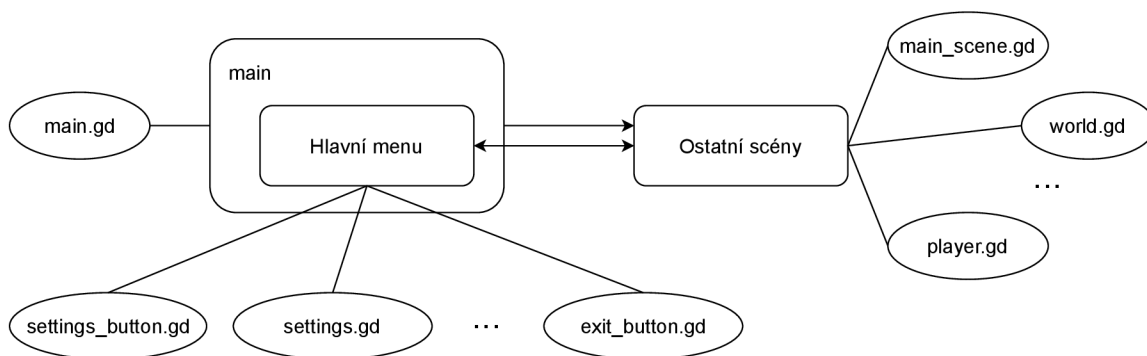
Na diagramu scén 5.4 jsou všechny základní scény GUI a jejich elementy, se kterými je možné interagovat. První scénou, kterou hráč uvidí je hlavní menu. Z něj může přejít do obchodu, na hlavní herní scénu, na výběr hrdiny a na další scény. Přechod mezi scénami je zpětný, tzn., uživatel se může vrátit do hlavního menu v jakýkoliv okamžik. Také může v hlavním menu buď aplikaci zavřít nebo otevřít nastavení.



Obrázek 5.4: Diagram zobrazuje základní elementy GUI a jejich interakce. Zobrazuje komunikaci a její typ mezi jednotlivými scénami. Přechody mezi scénami jsou naznačeny šipkami. Dvojitě značí typ interakce, čtverečky zobrazují samotné scény a elipsy znamenají, s čím se interakce provádí.

5.1 Hlavní menu

Při prvním spuštění se zobrazuje scéna, která se jmenuje *main*. Scéna *main* zahrnuje hlavní menu a skript pro nahrání uloženého stavu hry z disku. Uživatel vidí pouze hlavní menu, skript běží na pozadí. Na obrázku 5.5 je zobrazený rozdíl scény *main* a hlavního menu.

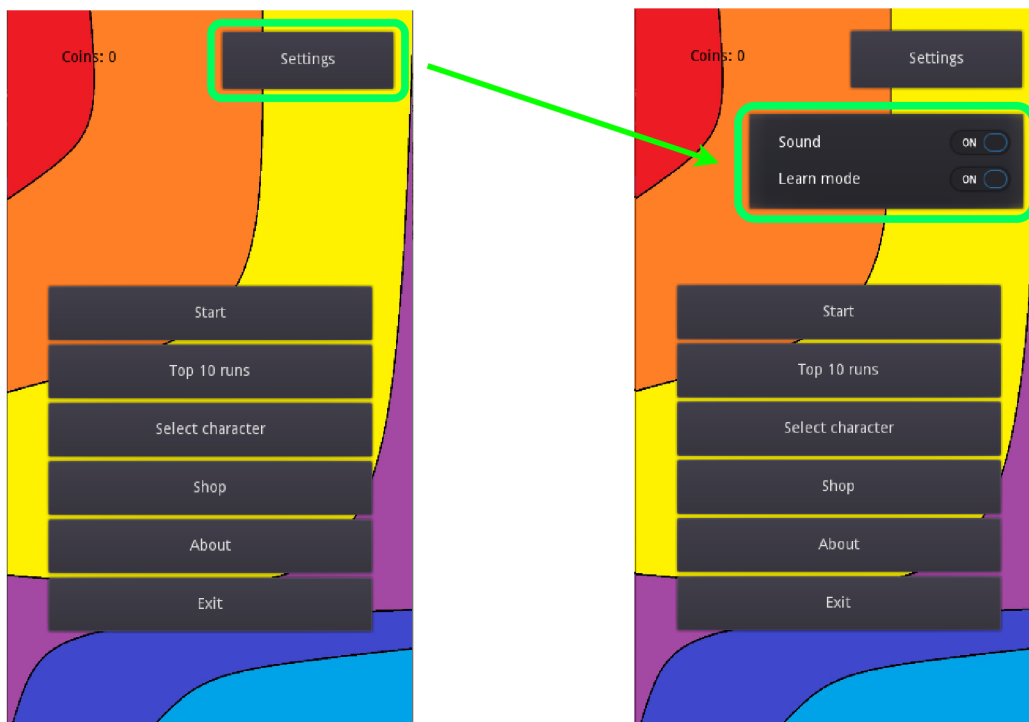


Obrázek 5.5: Scéna *main* je jediná, na kterou se není možné vrátit. Po návratu do hlavního menu z jiné scény, scéna *main* se nenačítá.

Nejprve se provede nahrání uložených dat z disku. To znamená, že se do paměti z disku nahraje nastavení (je-li zapnutý zvuk, zaučovací režim) a aktuální stav (počet mincí, zakoupené zboží a hrdinové). Skript scény *main* zkontroluje, zda existuje soubor pro načítání. Když soubor nebyl nalezen, tak to znamená první spuštění programu a soubor se vytvoří. Jinak se všechno načte a uloží do globálních proměnných nebo se rovnou nastaví.

Práce se soubory se skládá ze základních kroků. Kořenový adresář pro soubory je *user://*, který je stejný pro všechny podporované OS. Godot je schopen vytvořit soubor na správném místě pro každý operační systém. Čtení a zápis odpovídají typu dat, který se ukládá nebo načítá: proměnná, řádek, číslo apod. a Godot obsahuje speciální funkce pro každý typ.

Hlavní menu se skládá ze sedmi tlačítek a jednoho pole, které ukazuje aktuální počet mincí. Pro zobrazení textu se používá uzel *Label*, který je velmi jednoduchý a užitečný. Uzel umožňuje regulaci rozmístění textu vertikálně a horizontálně. Také udává možnost nastavit automatické zalomení na další řádek, maximální počet viditelných znaků, řádků apod.



Obrázek 5.6: Hlavní menu. Zelená šipka ukazuje otevření pop-upu s nastavením.

Tlačítka mají několik uzlů, dle typu a účelu použití. Základní uzel je *Button*, který představuje obyčejné tlačítko. Jsou i další druhy, například *CheckBox*, *CheckButton*, *MenuButton* apod. Při práci s tlačítky jsou signály velmi důležité. Signál se posílá při nějaké události, například kliknutí na tlačítko posílá signál *pressed*. Signály se odchyťávají ve skriptech a použít se předem připravené metody. Existuje mnoho druhů signálů pro různé uzly. Všechny uzly mají signály, nejenom tlačítka. Například elementy vizuálního rozhraní mají signál, který se posílá když je element aktivní [6, Signals]. Hlavní menu má pouze základní typ tlačítek. Jejich názvy a funkce jsou:

- Start – spustí běh. Provede se změna scény na scénu běhu;
- Top 10 runs – otevře tabulku s deseti nejlepšími běhy;
- Select character – otevře scénu výběru hrdiny;
- Shop – přejde na scénu obchodu;
- About – zobrazí informace o autorech, licencích apod.;
- Exit – uzavře celou aplikaci;
- Settings – otevře pop-up s nastavením. Viz obrázek 5.6.

Tlačítka jsou uvnitř uzlu, který se jmenuje *VBoxContainer*. Výhodou tohoto uzlu je automatické rovnoměrné rozmístění všech uzlů, pro které je rodičovským. Proto je ideální pro tlačítka v hlavním menu. Existuje také velmi podobný uzel, který se liší od předchozího pouze směrem. *VBoxContainer* zajišťuje vertikální rozmístění, *HBoxContainer* řídí horizontální rozmístění. Pomocí těchto uzlů je možné poskládat tlačítka a jiné elementy zarovnaně a rychle.

5.2 Tabulka běhů

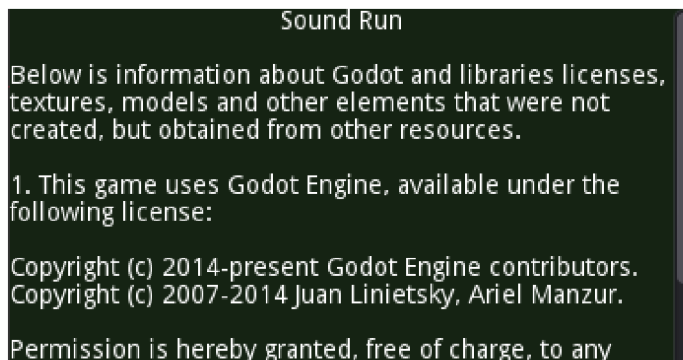
Tabulka deseti nejlepších běhů hráče, obrázek 5.7, se skládá z pořadí, časů jednotlivých běhů a počtu nasbíraných mincí. Tabulka je seřazena nejprve dle času, potom dle množství mincí. Nemá žádnou interakci a kliknutím na jakékoliv místo se provede návrat do hlavního menu.

1	7 minutes 33 seconds	359 coins
2	2 minutes 51 seconds	79 coins
3	2 minutes 44 seconds	21 coins

Obrázek 5.7: Tabulka nejlepších běhů.

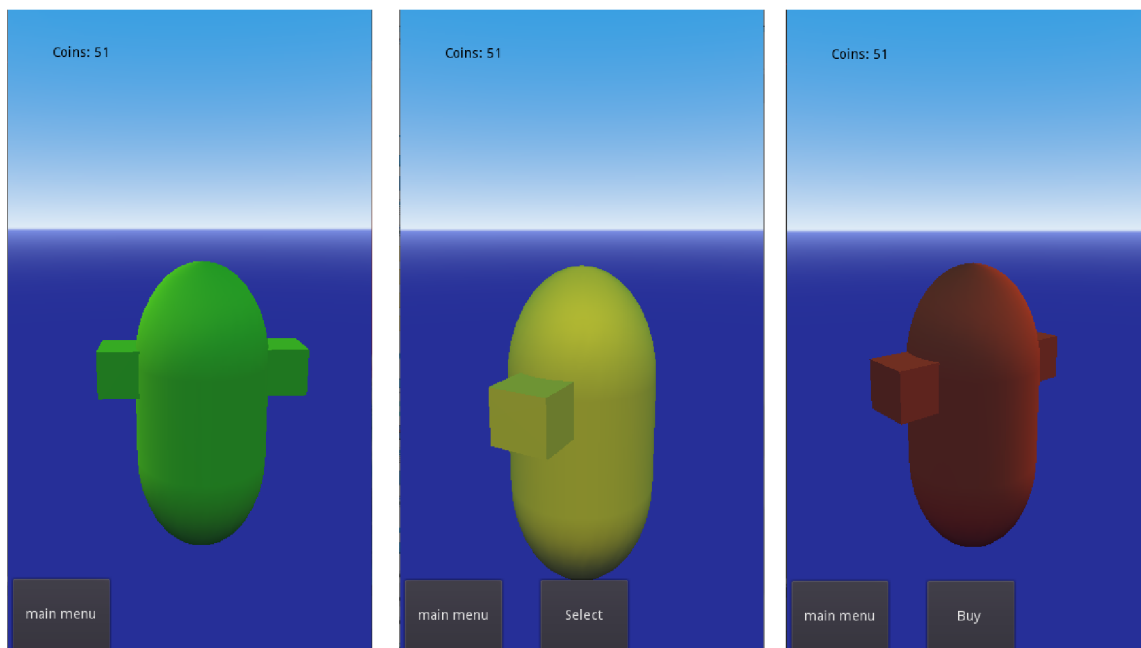
5.3 O hře

Scéna obsahuje informace o obrázcích, texturách, modelech apod, obrázek 5.8. Všechny informace jsou umístěny v textovém poli s podporou formátování pomocí *bbcode*. Popisuje zdroje převzatých obrázků a zvuků, jejich autory a licenční podmínky k použití. Skrolování a tlačítko návratu do hlavního menu jsou jediné interaktivní prvky této scény. Text není možné měnit.



Obrázek 5.8: Informace o hře.

5.4 Výběr hrdiny



Obrázek 5.9: Scéna *Výběr hrdiny*. Modely se liší pouze barvou. Skládají se z *CapsuleMesh* a *CubeMesh*. Vlevo je rychlý, uprostřed základní a vpravo ten, co má vylepšený skok.

Scéna *Výběr hrdiny* umožňuje výběr z několika postav. Zatím existují jen tři typy hrdinů: základní, rychlejší a ten, který může skákat výš. Základní hrdina žádný bonus nemá. Rychlejší má vyšší počáteční rychlost a větší zrychlení. Při zakoupení posledního hrdiny má hráč možnost skočit o něco výše.

Scéna je rozdělena na dvě části: 3D a 2D. Všechna tlačítka a labely jsou součástí 2D scény, pro kterou je 3D scéna rodičovskou. Scéna výběru hrdiny umísťuje:

- Počet mincí na stejném místě jako v hlavním menu;
- Tlačítko pro návrat do hlavního menu;
- Tlačítka pro zakoupení a výběr hrdiny. Pokud hráč hrdinu nevlastní, zobrazuje se tlačítko pro zakoupení. Po zakoupení se vymění za tlačítko výběru. U vybraného hrdiny se tlačítko nezobrazuje;
- Modely hrdinů.

V daný okamžik se zobrazuje jen jeden model hrdiny a pomocí gesta je možné přejít na další a zpátky. Modely se točí kolem svislé osy pomocí uzlu *AnimationPlayer*, který dovoluje vytvářet animace různého druhu. Druhý uzel *AnimationPlayer* slouží pro plynulý přechod mezi hrdiny a je spuštěn pomocí gesta. Na základě gesta a aktuálně zvoleného hrdiny se vybírá předem připravená animace.

Pokud hráč vybere hrdinu, kterého nemá zakoupeného, zobrazí se místo tlačítka *zvolit* tlačítko *koupit*. Kliknutím na tlačítko *koupit* se zkontroluje, zda hrdina není v seznamu zakoupených hrdinů, pak se provede přidání identifikátoru tohoto hrdiny do seznamu. Potom

se odečte jeho cena a změní se textové pole s počtem mincí. Rovnou se uloží do paměti počet mincí a seznam zakoupených hrdinů.

Pozadí je realizované pomocí uzlu *WorldEnvironment*. Stejný uzel *WorldEnvironment* se používá ve všech 3D scénách. Jeho součástí je parametr *Environment*, který umožňuje nastavit mnoho vlastností pozadí, například: slunce, oblohu, zem apod.

Jakmile je hrdina vybrán, změní se identifikátor aktuálně vybraného hrdiny a uloží se na disk. Každý identifikátor je uložen ve slovníku, který obsahuje cesty k modelům.

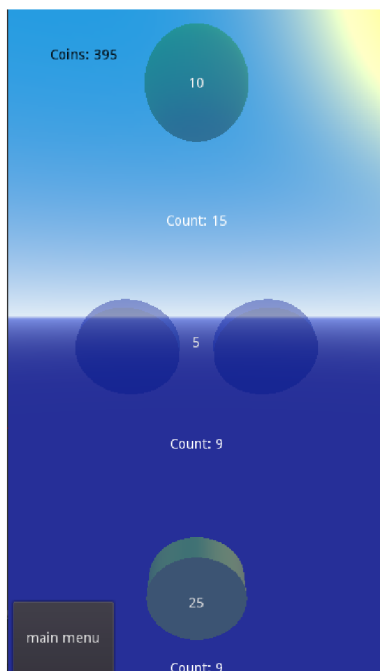
Před začátkem běhu se nahraje model aktuálního hrdiny a přidají se jeho bonusy. Scéna, která tvoří hrdinu, obsahuje uzel *AnimationPlayer* s animací pohybu, která se spustí hned po nahrání.

5.5 Obchod

Jinou možností využití mincí je obchod, obrázek 5.10. Existuje slovník, ve kterém se nachází všechny druhy vylepšení a jejich počet. Pokud hráč nějaké vylepšení využije, počet se dekrementuje.

Scéna obchodu se také skládá z 2D a 3D pod scén. 2D scéna obsahuje textové pole stavu peněženky a počtu vylepšení a tlačítka pro nákup. Tlačítka jsou neviditelná a překrývají oblast zobrazení 3D modelu. Kliknutím na nějaké vylepšení se provede velmi podobná akce jako při zakoupení nového hrdiny. Provede se kontrola, jestli má dostatek peněz, odečte se částka a přenastaví se textové pole s počtem mincí. Potom inkrementuje počet vylepšení ve slovníku a změny se uloží na disk. Jako většina ostatních scén, *Obchod* má tlačítko návratu do hlavního menu.

3D část umísťuje všechny 3D modely vylepšení, které jsou mírně otočené a uzel *WorldEnvironment*.



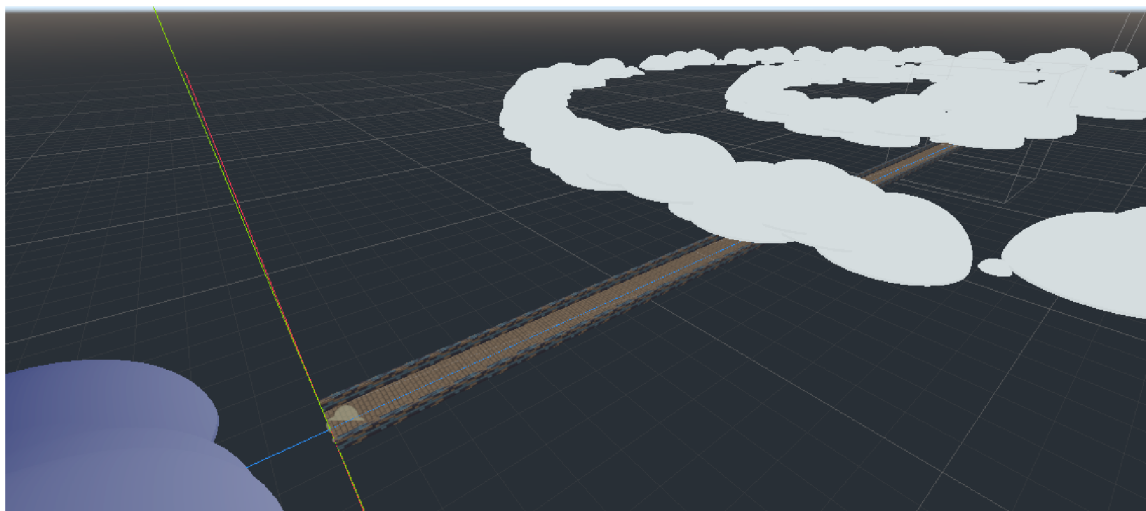
Obrázek 5.10: Scéna *Obchod*. Modely vylepšení jsou reprezentovány pomocí základních 3D objektů.

5.6 Dynamická část

Hlavní scéna je rodičovskou scénou pro celou herní část. Scéna v sobě umísťuje všechny ostatní na rozdíl od hlavního menu, které je pouze počáteční scénou pro hráče. Pro přechod do všech ostatních 2D scén se používá metoda přepínání scény. Hlavní scéna se skládá z dalších: *world*, *hlavní rozhraní*, *player*, *WorldEnvironment*, *time_counter*.

5.6.1 World

Scéna *world* představuje svět, ve kterém se odehrává samotná hra, obrázek 5.11. *world* obsahuje mapu, objekty a skripty kterými se řídí. Objekty a skripty jsou rozděleny do tří skupin: *maps*, *items* a *cleaner*.



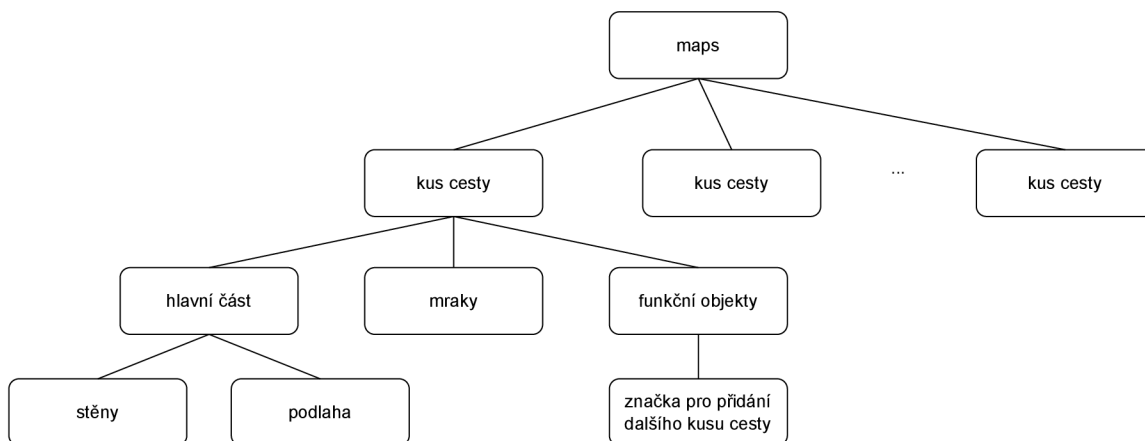
Obrázek 5.11: Scéna *world* obsahuje běžeckou trasu a modely mraků, které jsou generovány procedurálně pomocí shaderů. Mraky jsou převzaty z internetu a mírně upraveny.²

Samotná scéna *world* obsahuje skript, který řídí přidávání a promazávání kusů běžecké trasy.

Postup přidání nového kusu trasy je následující: za prvé se nahraje z paměti a vytvoří se instance. Instance se uloží do proměnné a potom se nový kus trasy posune směrem vpřed tak, aby byl umístěn hned za stávajícím kusem. Skript provede přidání nového kusu do rodičovského uzlu a spustí časovač. Po uplynutí nastaveného času se pomocí metody *queue_free()* přidá předchozí kus do fronty na promazání z paměti.

²<https://github.com/TinyLegions/stylised-clouds-tutorial>

Licence: <https://github.com/TinyLegions/stylised-clouds-tutorial/blob/main/LICENSE>

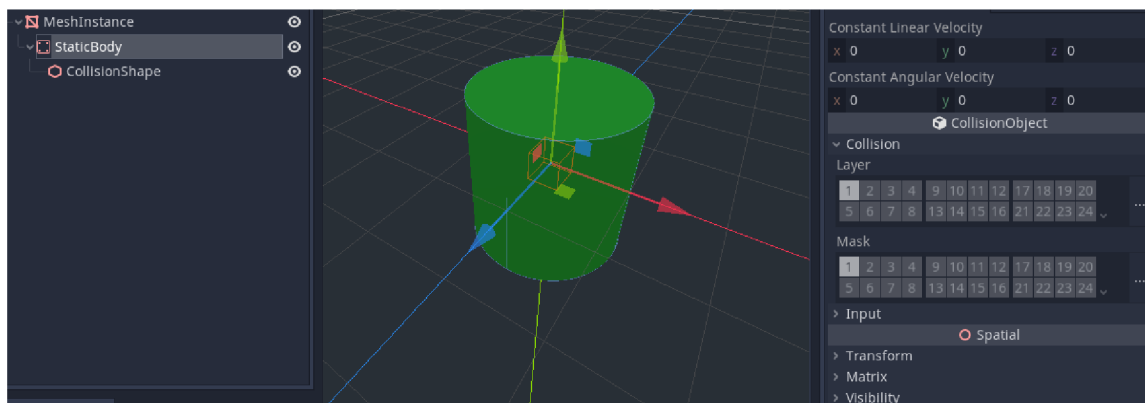


Obrázek 5.12: Obrázek zobrazuje uzel *maps* a kus cesty. Ukazuje z čeho se skládá kus cesty.

Rodičovský uzel pro cestu se nazývá *maps*. Tento uzel obsahuje kusy cesty, obrázek 5.12. Kusy cesty obsahují *collision objects*, aby hrdina nepropadl skrz zem, dekorativní elementy, jako převzaté procedurálně generované mraky a neviditelný element, který při kolizi spustí přidání dalšího kusu cesty.

Většina modelů Sound Rumu je vytvořena pomocí základních geometrických objektů. Realizované jsou pomocí vizuálních uzlů *MeshInstance*. Parametr *Mesh* udává typ geometrické objektu. Parametr *Material* umožňuje nastavit barvu, průhlednost, texturu a mnoho jiných vlastností.

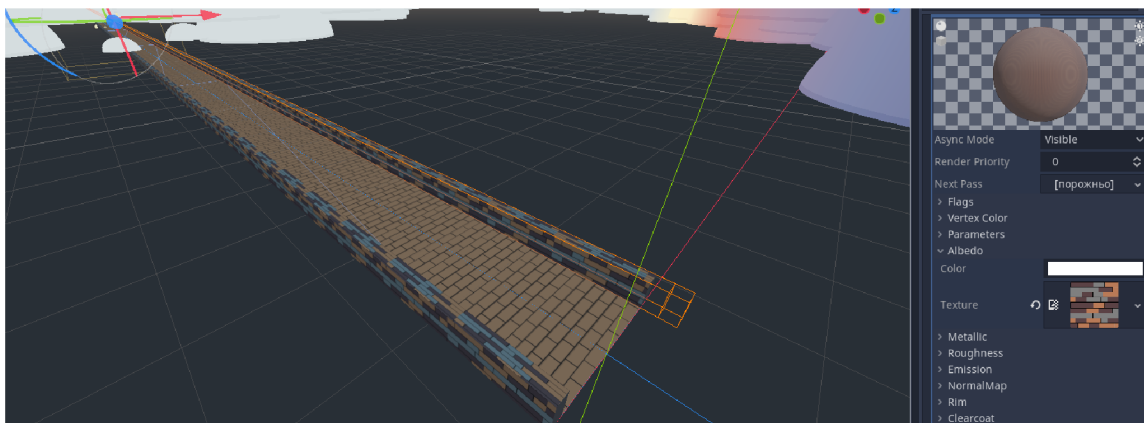
MeshInstance obsahuje dva pod uzly, které se používají pro nastavení kolizi. Jsou to *StaticBody* a *ColisionShape*, obrázek 5.13.



Obrázek 5.13: Jednoduchý 3D objekt. Vlevo je zobrazená struktura objektu. Vpravo je kolizní vrstva a maska.

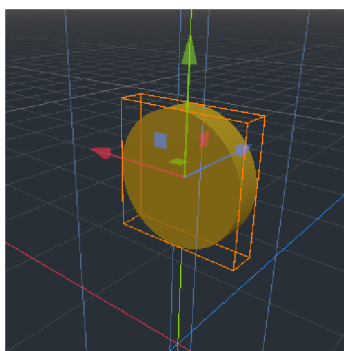
Ve *StaticBody* se nastavuje kolizní vrstva a maska. Kolizní vrstva je vrstva, ve které se objekt nachází. Kolizní maska udává vrstvy, se kterými může objekt kolidovat. Například, aby hrdina nepropadl skrz podlahu, je nutné, aby maska hrdiny obsahovala číslo vrstvy, ve které je umístěna podlaha. Pro určení přesné kolizní geometrie se používá uzel *ColisionShape*.

Cesta se skládá pouze z podlahy a stěn. Pro její vizualizaci se používá uzel *MeshInstance* dvou typů: *PlaneMesh* pro podlahu a *CubeMesh* pro stěny. Každý element má vlastní texturu ve formátu JPEG.



Obrázek 5.14: Scéna *map_part*. Vpravo je vidět materiál s parametry.

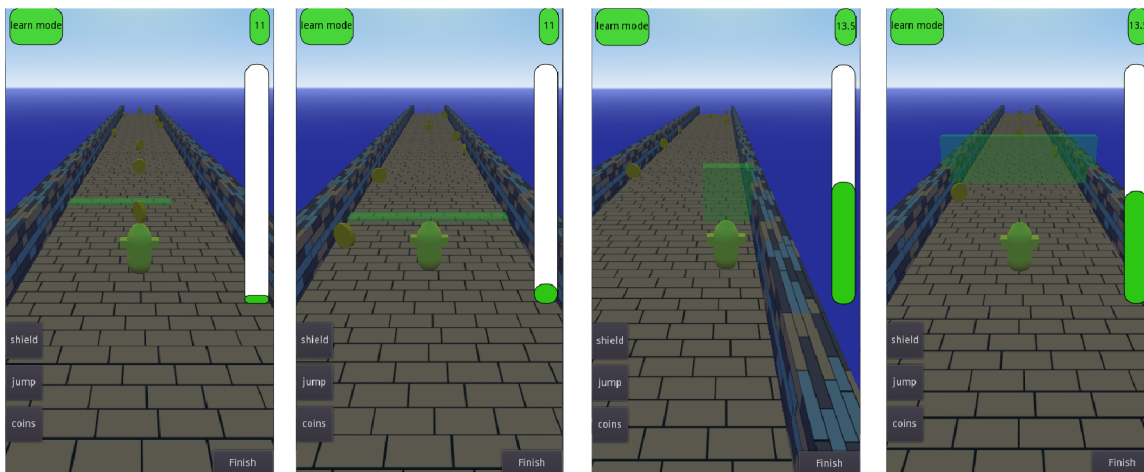
Uzel *items* obsahuje všechny ostatní elementy. Jsou to: *coins*, *barriers*, *pickables* a *map_elements*. Uzel *coins* je rodičovský uzel pro všechny mince, které se v průběhu hry vytváří, obrázek 5.15.



Obrázek 5.15: Vizuální část mince. Základním geometrickým elementem pro mince je cylindr.

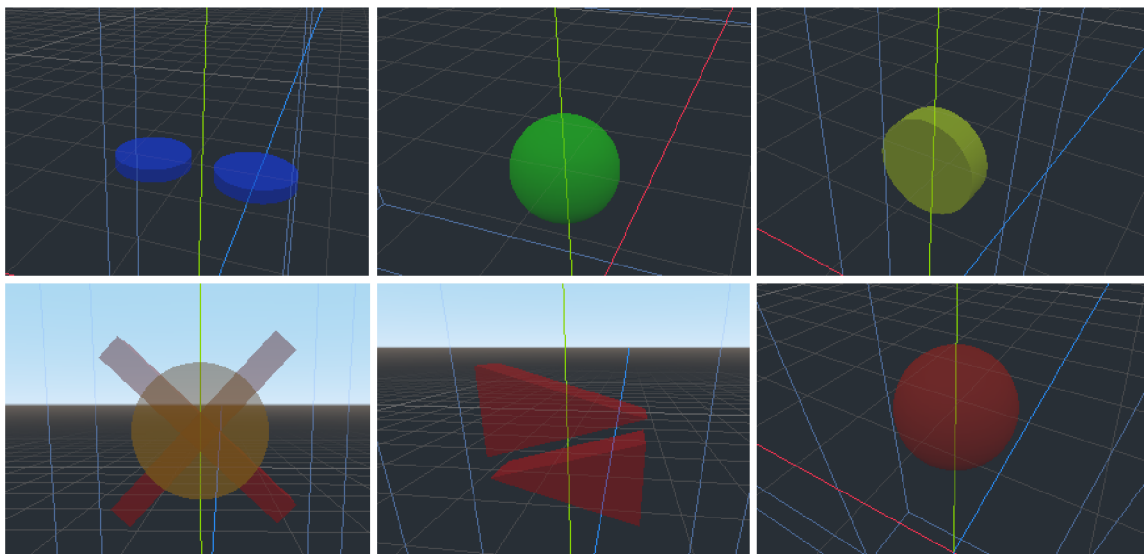
Scéna s mincí je dopředu připravená a skládá se ze samotné mince a uzlu *AnimationPlayer*, který s ní otáčí kolem svislé osy. Animace se spouští hned po přidání mince do scény. Pomocí třídy *RandomNumberGenerator* se náhodně vybere pruh, do kterého se mince umístí. Mince se generují pomocí časovače. Časovač jednou za sekundu vyšle signál, který skript zachytí a vygeneruje minci. Poté spustí časovač znovu.

Uzel *barriers* je rodičovský uzel pro všechny překážky, obrázek 5.16. Logika je stejná jako u uzlu *coins* s jednou výjimkou. Překážek existuje několik druhů, všechny jsou dopředu připravené a pomocí třídy *RandomNumberGenerator* se vybere jenom jedna. V případě, že překážka není na celou šířku, vybere se náhodný pruh.



Obrázek 5.16: Obrázek zobrazuje překážky. Překážky mají nastavenou kolizi tak, aby se ve správný čas spustil zvuk a aby do nich hrdina mohl narazit.

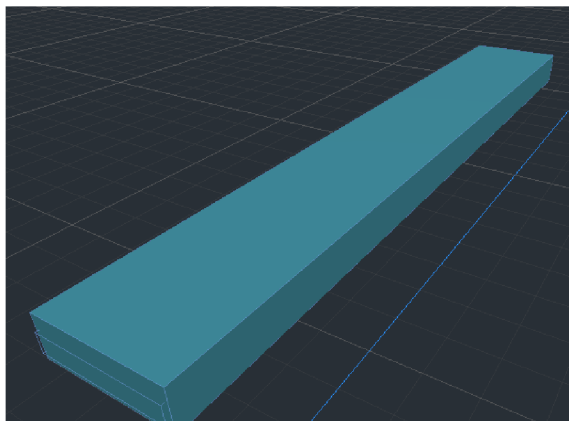
Uzel *pickables* je rodičovský pro vylepšení a zhoršení, obrázek 5.17. Princip je stejný jako u mincí, ale obsahuje výjimku. Objekty vylepšení a zhoršení jsou generovány s pravděpodobností 10 % na rozdíl od mincí, které jsou generovány vždy.



Obrázek 5.17: Obrázek zobrazuje všechny vylepšení a zhoršení. Skládají se z různých typů *Mesh*. Různé typy se můžou kombinovat pro vytvoření nového tvaru.

Uzel *map_elements* je rodičovský pro elementy, které mají kolizi s hráčem. Na takové elementy je možné skočit, nebo do nich narazit. V Sound Runu je pouze jeden objekt, který patří do této skupiny a je to *platforma*, obrázek 5.18.

Další velmi důležitý uzel scény *world* je uzel *cleaner*. Stará se o uklízení nepotřebných objektů. Obsahuje tři uzly typu *RayCast* (každý pro jeden pruh). Uzel *RayCast* umožňuje detekovat kolizi paprsku a geometrie. Při detekci kolize s libovolným objektem jej smaže z paměti. Paprsky zachytávají všechno kromě mapy. Je to realizováno pomocí *Collision Mask*. Každý element, který je možné takto ovlivnit, musí mít správně nastavenou vrstvu kolizi.



Obrázek 5.18: Platforma má dva *StaticBody* s různými kolizními vrstvami. První a větší *StaticBody* má stejné nastavení jako podlaha, aby hrdina mohl běžet po platformě. Druhý je malý a je na přední straně platformy. Má stejnou vrstvu jako překážky, proto působí ztrátu životů. Dle názvu *StaticBody* je možné rozlišit platformu od překážky a přizpůsobit chování.

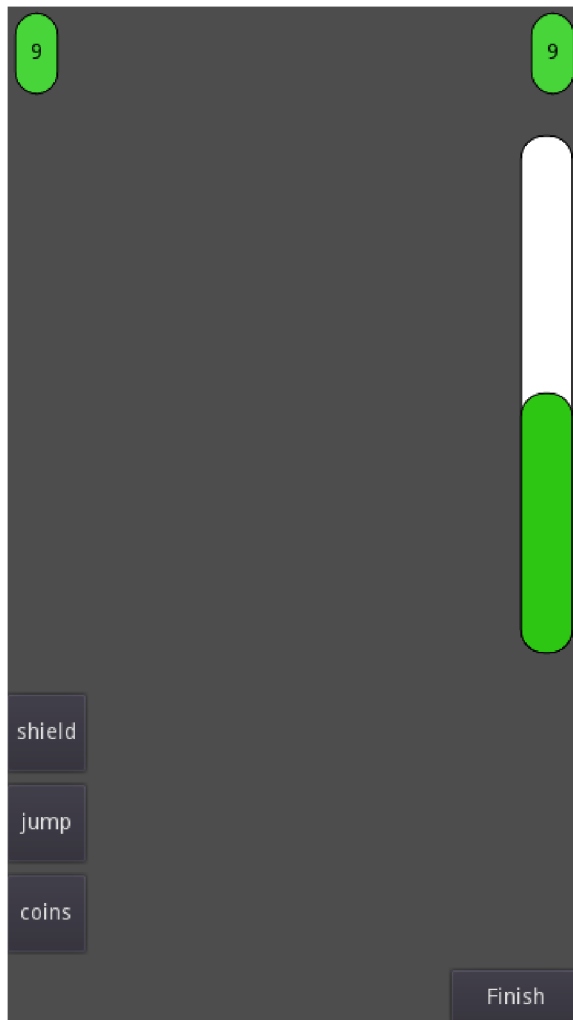
Jakmile nějaký objekt narazí na paprsek, je jen jednou přidán do fronty na promazání. Promazání z paměti je řízeno Godotem.

Všechny modely a elementy se načítají až když jsou potřeba, což způsobuje krátké zamrznutí hry. Jakmile se například objeví štít na hrdinovi, hra se na chvíli zastaví. Je to způsobeno načítáním z disku, kompilací shaderů apod. Pro vyřešení tohoto problému je potřeba všechny potřebné elementy dopředu načíst a zobrazit. K tomu slouží uzel *preload_all_elements*. Každý element hry, jako mraky nebo překážky, má svůj materiál. Materiál má barvu, průhlednost, shader a jiná nastavení. Scéna *preload_all_elements* má v sobě většinu elementů, které jsou umístěné pod cestou na začátku mapy a nejsou pro hráče viditelné. Jsou umístěné před kamerou a proto se zpracují hned před začátkem běhu. Tím je veškerá inicializace přesunuta na začátek běhu.

Při běhu hrdiny se neustále zvětšuje (zmenšuje) jedna z jeho souřadnic. To může způsobit problém při velmi dlouhé hře. Jedním z možných řešení je, že se hrdina posouvat nebude. Místo toho se bude hýbat všechno ostatní hned po přidání do svých rodičovských uzlů. Tímhle způsobem je zajištěna nekonečnost hry.

5.6.2 Hlavní rozhraní

Hlavní rozhraní je grafické rozhraní pro běh, obrázek 5.19. Zobrazuje počet mincí a další informace. Základní uzel pro hlavní herní rozhraní je *Control*.



Obrázek 5.19: Scéna hlavního rozhraní běhu. Hlavní rozhraní zobrazuje počet mincí v levém horním rohu, aktuální rychlost hrdiny v pravém horním rohu, tři tlačítka pro použití zakoupených vylepšení v levém dolním rohu, tlačítko ukončení běhu v pravém dolním rohu a *ProgressBar* vpravo.

Jedna ze základních mechanik her typu „runner“ je zrychlování s časem. Při vyšší rychlosti je reakce na překážky obtížnější. V Sound Runu je pro tento účel určen uzel *ProgressBar*. Tento uzel vizualizuje čas do dalšího zrychlení a zároveň volá metodu pro zrychlení *speed_up()*. Každé zrychlení nastává po delším časovém intervalu. Po zrychlení se zobrazí nová rychlost v textovém poli v pravém horním rohu herní obrazovky.

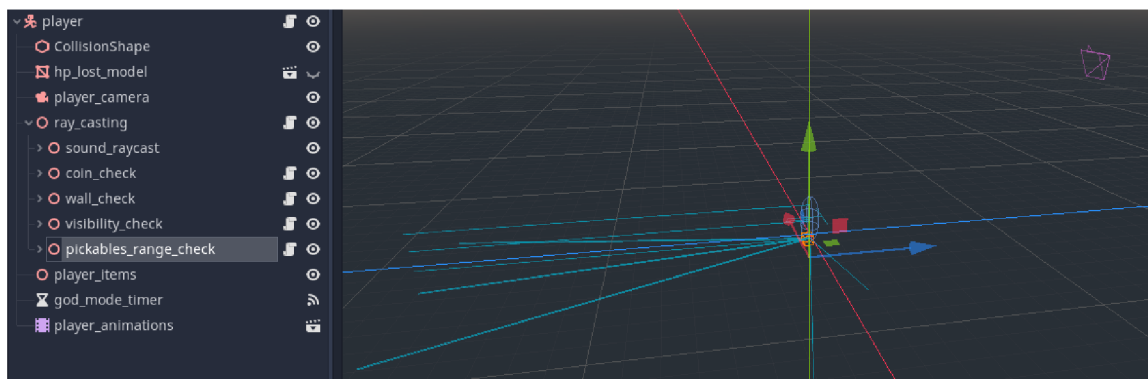
Funkce *speed_up()* je zodpovědná za zrychlení hrdiny. Rychlost se zvyšuje pokaždé, jak se tato funkce zavolá. Po navýšení rychlosti se musí změnit i vzdálenost, ve které je zvuk překážky přehráván.

Sound Run zobrazuje většinu elementů rozhraní pouze se standardní grafikou, ale engine Godot umožňuje elementy stylizovat. Změna stylu je použita například u elementu *Progress*

bar. Styl je možné změnit pomocí několika tříd, jedna z nich je třída *StyleBoxFlat* [6, *StyleBoxFlat*]. Třída *StyleBoxFlat* umožňuje změnit barvu, zaoblení a hranice. Pomocí této třídy je změněno pozadí u elementu *Progress bar* na bílou barvu se zelenou výplní. Dále má element černé hranice se zaoblenými rohy. Podobný styl je použit i pro textová políčka v pravém a levém horním rohu herního grafického rozhraní.

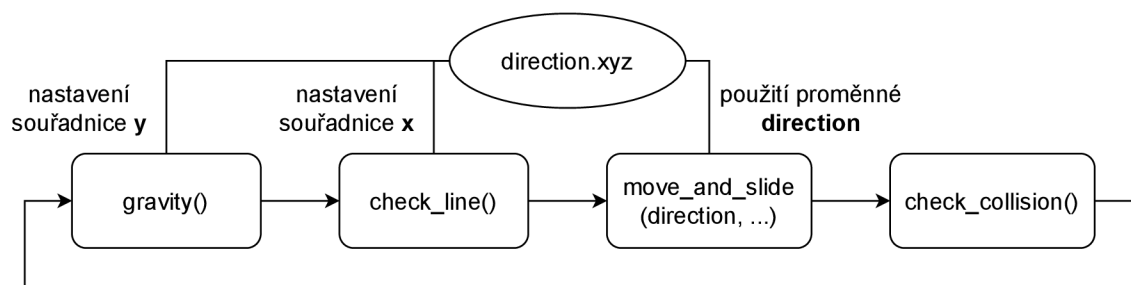
5.6.3 Player

Nejrozsáhlejší část práce je implementace hrdiny, který je opět reprezentován jako scéna, obrázek 5.20. Rodičovský uzel scény hrdiny je typu *KinematicBody*. Běh hrdiny je automatický, hráč řídí jen změnu pruhu, skok a dřep.



Obrázek 5.20: Scéna *player*. Vlevo je možné vidět všechny uzly a sub-scény. Větší část obrazovky obsahuje samotná 3D scéna, na které je vidět mnoho různých paprsků. V bodě (0,0,0) je vidět kapsle. Je to uzel *CollisionShape* hrdiny, který je zodpovědný za to, že hrdina nepropadne skrz zem a bude mít kolizi.

Celá logika řízení hrdiny je rozdělena do dvou částí: zpracování vstupů od hráče a samotné řízení pohybu, kolizí a gravitace. Tyto dvě části jsou řešeny pomocí metod: *__physics_process()* a *__input()*.



Obrázek 5.21: Diagram ukazuje funkce, které se volají ve *__physics_process()* ve smyčce. *gravity()* a *check_line()* nastavují proměnnou, která se použije v *move_and_slide()* pro pohyb hrdiny.

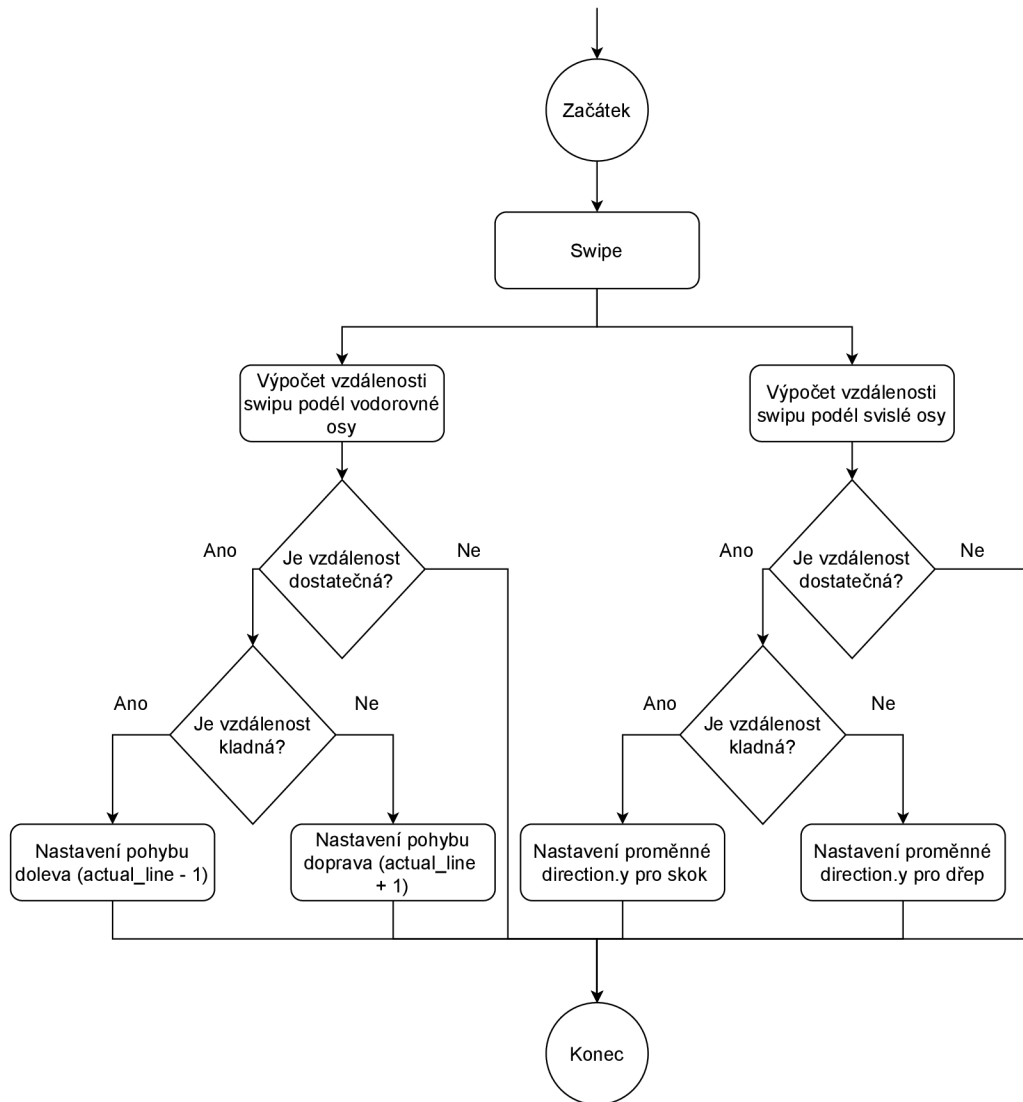
__physics_process() [6, Godot notifications] řídí pohyb, kolize, gravitaci apod., obrázek 5.21. Pro uzel *KinematicBody* v Godot Engine existuje metoda *move_and_slide()*, která zajišťuje pohyb. Proměnná *direction* obsahuje x,y a z souřadnice. Pomocí ní se do

metody *move_and_slide()* předává směr a rychlost pohybu. *__physics_process()* zahrnuje další tři funkce kromě *move_and_slide()*: *gravity()*, *check_line()* a *check_collision()*.

- Metoda *gravity()* zajišťuje výpočet gravitace. Hrdina neustále padá dolů. Každou iterací *__physics_process()* se zrychluje pád až dokud hrdina není na zemi.
- Metoda *check_line()* nastavuje směr a rychlost pro změnu pruhu na základě aktuálního umístění hrdiny a proměnné *actual_line*.
- Metoda *check_collision()* kontroluje kolize hrdiny s nebezpečnými objekty. Jsou to jen dva objekty: *platforma* a *překážka*. Při kolizi s *platformou* zepředu, nezáleží na počtu štítů a hra končí. Při kolizi s *překážkou* se nejprve sníží počet štítů a spustí se metoda *hp_lost_process()*. Teprve v případě, že už žádný štít nemá, hra končí.

Skript scény *player* obsahuje několik dalších funkcí. Například funkce *hp_lost_process()* se volá ve funkci *check_collision()* pouze za podmínky, že hrdina narazí na překážku. Tato funkce zapíná animaci, která mění viditelnost scény *hp_lost_model* což vypadá jako blikání. Také spouští časovač s názvem *god_mode_timer*. Během doby trvání časovače a animace, nemůže hrdina narazit do jiné překážky.

Existují čtyři typy gest: nahoru, dolů, doleva a doprava. Doleva a doprava znamenají změnu pruhu. Nahoru je skok a dolů je dřep. Skok a změna pruhu jsou realizované pomocí změny souřadnic proměnné *direction*, viz obrázky 5.22 a 5.21. Dřep zrychluje pád hrdiny, pokud je ve vzduchu. Důvodem této mechaniky je umožnit hráči vyhnout se vznášející překážce. Dřep je zajištěn pomocí uzlu *AnimationPlayer*, který změní výšku modelu a na obrazovce to vypadá jako dřep. *AnimationPlayer* je umístěný ve scéně hráče a nazývá se *player_animations*.



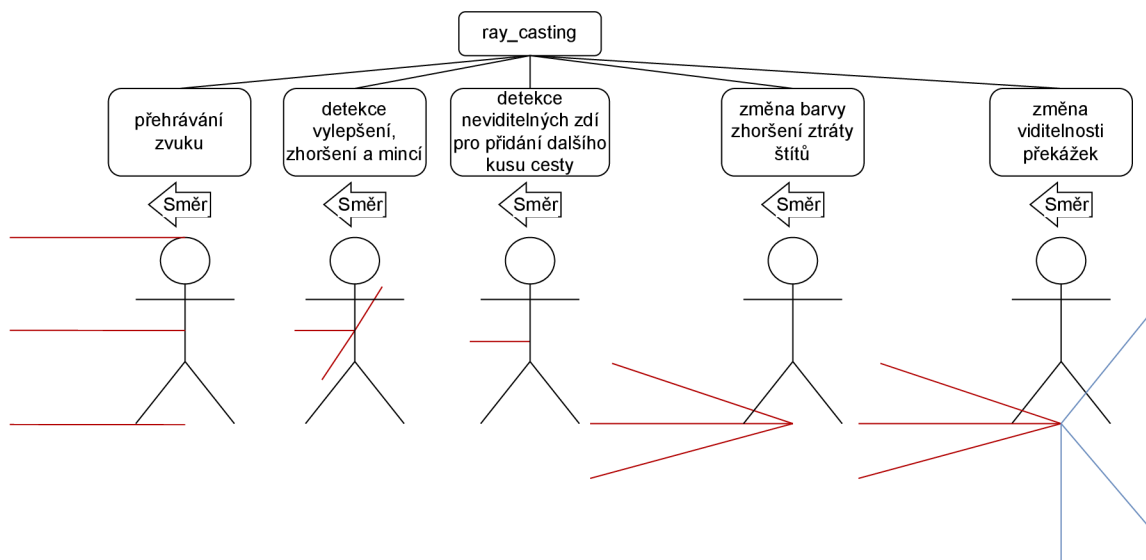
Obrázek 5.22: Diagram ukazuje kroky pro nastavení směru pohybu hrdiny.

Zpracování vstupů obsluhuje vstupní zařízení jako je klávesnice a myš, ale v případě mobilní aplikace se hlavně zabývá vstupem v podobě tzn. „swipe“ a je řešeno pomocí funkce `_input()` [6, Godot notifications]. Sound Run ukládá pozici začátku a konce gesta, na jejich základě rozhodne o jaké gesto se jedná a provede nastavení proměnných `actual_line` a `direction.y`, viz obrázek 5.22.

Další důležitý uzel je `player_camera`. Je to kamera, která se nachází fixně za hráčem a je nasměrovaná na hrdinu. Kamera je uvnitř scény hráče, proto nepotřebuje řídicí skript. Kamera se hýbe společně s hrdinou.

Uzel `ray_casting` je rodičovský pro všechny uzly paprsků. Paprsky jsou rozděleny do pěti kategorií: 1. pro přehrávání zvuku, 2. pro detekci vylepšení, zhoršení a mincí, 3. pro detekci neviditelných zdí přidávající nový kus cesty, 4. pro změnu viditelnosti překážek a 5. pro změnu barvy (odmaskování) zhoršení, které způsobuje ztrátu štítů, obrázek 5.23.

Poslední uzel hráče `players_items` je rodičovský uzel, do kterého se ukládají všechny 3D scény vylepšení a zhoršení, které hráč vlastní.



Obrázek 5.23: Diagram ukazuje všechny paprsky. Paprsky *přehrávání zvuku* přehrávají audio překážky se kterou kolidují. Paprsky *detekce vylepšení, zhoršení a mincí* jsou malé a určeny pro sbírání mincí, vylepšení a zhoršení. Paprsek *neviditelných zdí pro přidání dalšího kusu cesty* je pouze jeden. Koliduje s neviditelnou zdí, která spouští proces přidání dalšího kusu cesty. Paprsky *změny barvy zhoršení ztráty štítů* mění barvu zhoršení ztráty štítů ze zelené na červenou. Paprsky *změny viditelnosti překážek* jsou rozděleny do dvou skupin: přední a zadní. Paprsky jsou určeny pro to, aby změnily viditelnost překážek. V případě zapnutého režimu zaučení se používají paprsky, které jsou nasměrovány dopředu, jinak ty, co jsou nasměrovány dozadu.

5.6.4 Časovač a WorldEnvironment

Poslední uzly hlavní scény jsou *time_counter* a *WorldEnvironment*. *time_counter* se používá pouze pro měření doby běhu. Po ukončení běhu se počet sekund přepočítá na minuty a sekundy a uloží se do tabulky nejlepších běhů, když je nutné. Uzel *WorldEnvironment* zobrazuje grafiku pozadí a používá se stejný, jako pro ostatní scény.

Kapitola 6

Zhodnocení

Hra je v základním stavu, tzn., že je vše implementováno pouze na základní úrovni a hru je nutné dále vyvíjet. Je v plánu provést mnoho etap testování a přidat nebo změnit mnoho elementů hry. Uvažovány jsou změny i některých herních mechanik, než se hra dostane do stavu produktu, který je možné ukázat veřejnosti.

6.1 Zpětná vazba

Pro testování a zpětnou vazbu byla hra předána lidem, kteří nejsou zapojeni do samotného vývoje. Testeři se vyjádřili ke třem kategoriím: **mechanika zvuků**, **gameplay** a **design**.
Mechanika zvuků:

- Pokud je překážka v jiném pruhu než je hráč, nemůže o ní vědět, protože nevydává zvuk. Návrh řešení je buď udělat zvuk 3D, kde každá překážka obsahuje vlastní audio přehrávač a zvuk se bude šířit přesně od ní. Jiným řešením je přehrávat mírně odlišný zvuk. Problém prvního nápadu je ten, že je hra určena pro mobilní zařízení. Pokud hráč nebude mít sluchátka, ale pouze reproduktor v mobilu, nedokáže rozlišit z jaké strany zvuk přichází. Problém druhého nápadu spočívá v znásobení množství zvuků, které hráč musí rozpoznat. Místo původních tří by potřeboval rozlišit devět.
- Změnit zvuky neboť nejsou příjemné na poslech. Udělat je odlišnější. Místo monotónního zvuku přehrávat opakující se mluvené slovo. Například *top* nebo *bottom*.
- Mechanika je opravdu nová, ale je náročnější naleznout cílovou skupinu uživatelů. Cílová skupina hráčů je zvyklá hrát podobné hry ve škole, občas i během školní hodiny nebo přednášky. Z pohledu podnikání je jasné, že hra nebude populární v těchto případech a je to nevýhoda, ale pro samotné studenty to možná bude lepší.
- Herní mechanika založená na zvuku je neobvyklá, avšak to může být i nevýhodou. Lidé jsou zvyklí vidět překážky a dopředu plánovat svůj běh. V tomto případě je to nemožné. I přes zaučení mají začátečníci problém na základě zvuku určit, o jakou překážku se jedná. Je třeba hrát nějakou dobu, než si uživatel na samotnou mechaniku zvykne.
- Je možné udělat režim pro slepce. V tomto režimu bude mít každý element mapy vlastní zvuk, který jednoznačně udá typ elementu a jeho umístění pomocí klíčových

slov. Například slova: „platforma right“, „coin center“ nebo „shield left“ mohou slepému hráči oznámit, kde se objekty nacházejí. Je možné, aby každý objekt vydával vlastní zvuk ve 3D prostoru.

Gameplay:

- Přidat bonusové úrovně, které nebudou procedurálně generované. Během hry by hrála hudba a v určitých okamžicích (například při bubnech) by hráč musel provést nějakou akci.
- Hra je náročná pro začátečníky. Pokud není hráč alespoň trochu zkušený, většinou nedokáže běžet ani 15 sekund. Možnosti usnadnění jsou: zvětšit délku paprsků pro dřívější detekci překážek; zvětšit dobu pro skok a dřep; zrychlit animaci dřepu a zrychlit skok.
- Hra je pomalá. Počáteční rychlost je v pořádku, ale zrychlení je moc malé. Hra je mnohem zajímavější, když je v ní více překážek. Přímé vyjádření hráče: „Když je rychlost vysoká, srdce mi bije velmi rychle a neustále a úpěnlivě očekávám další zvuk překážky. Je to pěkný pocit.“
- Jedno z možných dalších vylepšení je zrychlení. To může být v několika druzích a v různých časových intervalech.

Design:

- Smrt hrdiny není nijak oznámena. Hned po ztrátě posledního štítu se změní scéna na Game over. Nevadilo by přidat animaci toho, jak hrdina padá na zem nebo něco podobného. Také by bylo vhodné přidat ukončující melodii.
- Na scéně výběru hrdiny přidat popis všech hrdinů a jejich ceny. Není jasné, který hrdina je který a čím je unikátní.

Podle zpětné vazby bylo jasné, že na hře je nutné ještě hodně pracovat. Nejpravděpodobněji bude nutné provádět více etap testování a to na větším počtu lidí, než jenom na kamarádech. Bylo překvapením, že lidé z takové nové mechaniky mohou být tak silně zmatení. Na druhou stranu je hodně zajímavá pro ty, kteří mají rádi výzvy. Pro ostatní bude nutné snížit počáteční obtížnost. Testeré také přišli se zajímavými nápady jako je režim pro slepce.

6.2 Analýza zpětné vazby

Režim pro slepce zmíněný v sekci 6.1 lze rozšířit. Pro slepce by mohla být přizpůsobena celá hra nejen samotný běh. V dnešní době i slepí lidé používají chytré telefony. Na to je určena funkce TalkBack nebo VoiceOver. Hlavní účel těchto funkcí je číst vše, co je napsáno na obrazovce nebo provádět operace pomocí hlasu.

6.2.1 TalkBack

TalkBack je speciální nástroj [3], který umožňuje používání mobilního zařízení bez vizuálního kontaktu. Program používá různé způsoby pro komunikaci s člověkem například zvuk a vibrace.

Program je předinstalovaný na zařízeních Android společně s ostatními aplikacemi od Google pro Android. Je velmi pravděpodobné, že si většinou lidé této aplikace nevšimají, když nemají vadu zraku. Doplněk je ale udržovaný a automaticky se aktualizuje.

Funkce je velmi jednoduchá. Uživatel hýbe prstem po obrazovce dokud nenarazí na nějaký element. TalkBack namluví informace o elementu, například přečte textový odstavec na webu.

6.2.2 VoiceOver

VoiceOver je čtečka obrazovky [11], která umožňuje použití iPhonu, aniž by se uživatel díval na obrazovku. Princip je stejný jako u TalkBack. VoiceOver popisuje nahlas všechny položky, kterých se uživatel dotkne. Čtečka umožňuje nastavit výšku hlasu a rychlost předčítání tak, aby vyhovovaly uživateli.

Při zapnutém VoiceOveru mají některá standardní gesta jiný účinek. Kvůli tomu existují další gesta pro ovládání iPhonu. Například výběr první nebo poslední položky obrazovky je možné provést klepnutím čtyřmi prsty u horního nebo dolního okraje.

6.2.3 Režim pro slepce

Další možnost pro pokračování je možnost udělat hru plně použitelnou pro slepce. Třeba při prvním spuštění se hra zeptá, jestli hráč chce používat pouze zvuk nebo i vizuální část. Když hráč vybere jenom zvuk, tak bude pro něj speciálně vytvořené zaučení, ve kterém bude vysvětleno jak se hra ovládá. Režim pro slepce by pravděpodobně mohlo být možné automaticky zapnout, pokud má uživatel spuštěný režim VoiceOver.

Je očividné, že noví vývojáři nemají prostředky pro realizaci režimu pro slepce. Zabere to hodně času a možná to nebude finančně výhodné. Samotný nápad je však zajímavý a šlo by pokračovat s vývojem v tomto směru.

6.3 Možná budoucí rozšíření

Účelem této bakalářské práce je demonstrace herních mechanik, ale autor práce by ji chtěl rozvíjet i dále. Bude nutné přidat další rozšíření:

- Modely. Nejdůležitější část práce, kterou je nutné udělat, jsou modely. Je nutné namodelovat vnějšek hrdiny, spíš několika hrdinů (kvůli různým bonusům), vymyslet a vytvořit styl aplikace apod.;
- Zatačky nebo jiné zajímavé modifikace cesty. Možná nějaká větvení, kde by si hráč mohl vybrat cestu. Každá z takových cest by měla nějaké vlastní výhody a nevýhody;
- Více obsahu. Zatím jsou pouze tři hrdinové a jedna platforma, ale šlo by přidat mnohem více. Další nápad pro hrdinovy bonusy jsou: zvětšení délky paprsků, které zapínají zvuk; štít ze začátku (hrdina začíná s dvěma životy);
- Tabulka *Top100*. Jako ve většině her, lidé rádi soutěží. Úkolem je rozšíření top 10 běhů do top 100 běhů online. To znamená, že v tabulce budou nejenom zápasy hráče, ale sto nejlepších zápasů všech hráčů.

Základní úkol bakalářské práce byl splněn a navíc byly některé rozšíření částečně realizovány. Jedná se o tyto mechaniky:

- Byli přidáni další dva hrdinové a mechanika bonusů.
- Byly přidány tři druhy zhoršení.
- Byla přidána možnost zakoupení vylepšení za mince, které hráč sbírá během hry a možnost tyto vylepšení využít.
- Bylo vytvořeno menu a přidáno uložení na disk.

Kapitola 7

Závěr

Původním cílem práce byla demonstrace nové mechaniky v herním žánru „runner“. Záměrem bylo vytvořit herní demo typu „runner“, kde má zvuk důležitou funkci a slouží pro navigaci.

Prvním krokem bylo studium herního enginu Godot. Kapitola 2 se zabývá základními poznatky o herních enginech, vysvětluje jejich silné a slabé strany a příklady použití. Popisuje zvolený Godot Engine a jeho výhody. V další kapitole 3 byl probrán zvolený typ hry, hry s podobnými mechanikami a byla vysvětlena hlavní mechanika Sound Runu – zvuk.

Vývoj byl rozdělen na tři etapy: realizace základních mechanik, implementace překážek a kolizí a poslední etapa se zabývala vylepšeními a zhoršeními. Kapitola 4 popisuje veškeré elementy hry.

Implementace začínala realizací **základních mechanik** jako pohyb a kolize. Prvním elementem byla dráha. Bylo nutné udělat plochu, skrz kterou hrdina nepropadne. Dále se realizoval hrdina a jeho pohyb. Cílem hry je nekonečný běh, což způsobilo implementační problémy.

Následujícím krokem byly **překážky**. Základem byly kolizní vrstvy a masky, pomocí kterých lze nastavit správnou reakci elementů hry. Po vyřešení mechanického fungování překážek bylo potřeba vyřešit jejich vizuální stránku a připravit materiály.

Jedna z nejdůležitějších částí hry je zvuk, ale jeho realizace není nejnáročnější. Obtížnější bylo nalezení vhodných zvuků, které dobře naznačují překážky. Samotné jejich přidání pak trvalo kratší dobu.

Poslední částí implementace byla realizace **vylepšení a zhoršení**, obchodu a dalších hrdinů. Chtěl bych i dále pokračovat směrem vývoje aplikace. V dnešní době se popularita a výkon mobilních zařízení zvětšuje, proto dává smysl soustředit se na vývoj mobilních aplikací.

Úkol byl splněn a dokonce přesáhl původní zadání. Bylo vytvořeno plně funkční demo se všemi základními mechanikami hry. I když je hra zaměřena na mobilní zařízení (IOS, Android), demo je možné spustit také na operačním systému Windows a Linux.

Jak už bylo uvedeno v předchozí kapitole, mám ještě nápady, co přidat a jak hru vylepšit. Během psaní bakalářské práce jsem si uvědomil, že vytvoření hry je velmi časově náročné, a proto plánuji zapojit ještě své kamarády a společně vytvořit plnohodnotný produkt. Shadery (s jejich kódem) a ostatní prvky, které nebyly vytvořeny, ale byly získány z jiných zdrojů jsou pouze pro demonstrační verzi a nebudou použité v konečném produktu.

Literatura

- [1] LINIETSKY, J., MANZUR, A. a GODOT ENGINE CONTRIBUTORS. *Godot License* [online]. godotengine [cit. 2023-07-24]. Dostupné z: <https://godotengine.org/license/>.
- [2] GREGORY, J. *Game Engine Architecture*. 3. vyd. A K Peters/CRC Press, August 17, 2018. ISBN 978-1138035454.
- [3] HILDENBRAND, J. *What is Google TalkBack?* [online]. www.androidcentral.com, 2014 [cit. 2023-02-19]. Dostupné z: <https://www.androidcentral.com/what-google-talk-back>.
- [4] KRAMER, G., WALKER, B., BARGAR, R. a AUDITORY DISPLAY, I. C. for. *Sonification Report: Status of the Field and Research Agenda*. International Community for Auditory Display, 1999. ISBN 9780967090405. Dostupné z: <https://books.google.cz/books?id=Kv7yNwAACAAJ>.
- [5] LINIETSKY, J. *Godot history in images!* [online]. godotengine.org, leden 2014 [cit. 2022-04-05]. Dostupné z: <https://godotengine.org/article/godot-history-images>.
- [6] LINIETSKY, J., MANZUR, A. a GODOT COMMUNITY. *Godot Documentation* [online]. docs.godotengine, 2014-2022 [cit. 2022-04-06]. Dostupné z: <https://docs.godotengine.org/en/stable/>.
- [7] *Unity game engine: 8 popular mobile games that rocked the game charts* [online]. logicsimplified.com, 2022 [cit. 2022-05-04]. Dostupné z: <https://logicsimplified.com/newgames/unity-game-engine-8-popular-mobile-games-that-rocked-the-game-charts/>.
- [8] PHILENDRA, V. *A Beginner's Guide to Video Games: Perspectives* [online]. www.thesmujournal.ca, 2021 [cit. 2023-02-05]. Dostupné z: <https://www.thesmujournal.ca/video-games/a-beginners-guide-to-video-games-perspectives>.
- [9] ROSE, E. *Motion Capture Animation: The Pros and Cons for Marketing Video* [online]. <https://idearocketanimation.com>, 2021 [cit. 2023-02-05]. Dostupné z: <https://idearocketanimation.com/22689-motion-capture-animation/>.
- [10] TITEUX, N. *Designing sound for video games* [online]. www.nicolastiteux.com, 2021 [cit. 2023-02-05]. Dostupné z: <https://www.nicolastiteux.com/en/blog/sound-design-for-video-games/>.

- [11] *Zapnutí a procvičování VoiceOveru na iPhonu* [online]. support.apple.com, 2023 [cit. 2023-04-23]. Dostupné z:
<https://support.apple.com/cs-cz/guide/iphone/iph3e2e415f/ios>.
- [12] WANG, R., JUNG, C. a KIM, Y. Seeing Through Sounds: Mapping Auditory Dimensions to Data and Charts for People with Visual Impairments. *Computer Graphics Forum* [online]. 2022, sv. 41, č. 3, s. 71–83, [cit. 2023-04-02]. Dostupné z:
<https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14523>.
- [13] *Obecné informace o zvuku* [online]. paroc.cz, 2022 [cit. 2022-05-4]. Dostupné z:
<https://www.paroc.cz/knowhow/zvuk/obecne-informace-o-zvuku>.