



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 HRA SOUND RUNNER V GODOT ENGINE

MOBILE GAME SOUND RUNNER IN GODOT ENGINE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

OLEKSII KORNIENKO

VEDOUCÍ PRÁCE

SUPERVISOR

TOMÁŠ MILET, Ing., Ph.D.

BRNO 2022

Zadání bakalářské práce



Student: **Korniienko Oleksii**
Program: Informační technologie
Název: **Mobilní hra Sound Runner v Godot engine**
Mobile Game Sound Runner in Godot Engine
Kategorie: Počítačová grafika

Zadání:

1. Nastudujte engine Godot a vývoj mobilních aplikací. Prozkoumejte trh her, které využívají herní mechaniky založené na zvuku.
2. Navrhněte mobilní hru, která využívá nevšedním způsobem zvuky pro herní mechaniky.
3. Implementujte navrženou hru.
4. Aplikaci otestujte na uživateliích 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

Pro udělení zápočtu za první semestr 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, doc. Dr. Ing.

Datum zadání: 1. listopadu 2021

Datum odevzdání: 11. května 2022

Datum schválení: 1. listopadu 2021

Abstrakt

Cílem práce je vytvoření nové hry, 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 vzhled k popularitě dané platformy.

Abstract

The purpose of the work is to create a new game in which sound will play 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 appearance to the popularity of the platform.

Klíčová slova

mobilní, hra, zvuk, godot, runner

Keywords

mobile, game, sound, godot, runner

Citace

KORNIENKO, Oleksii. *Mobilní hra Sound Runner v Godot engine*. Brno, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Tomáš Milet, Ing., Ph.D.

Mobilní hra Sound Runner 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

9. května 2022

Poděkování

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

Obsah

1	Úvod	2
2	Vyvoj mobilních her	3
2.1	Herní engine	4
2.2	Godot engine	9
3	Průzkum trhu	14
3.1	Runnery	14
3.2	Hry s použitím zvuku	18
4	Návrh	20
4.1	Komponenty hry	22
4.2	Kamera	28
5	Implementace	34
5.1	Hlavní menu	36
5.2	Vyběr hrdiny	38
5.3	Obchod	40
5.4	Hlavní část	41
6	Zhodnocení	48
6.1	Navíc	48
6.2	Feedback	49
7	Zavěr	52
	Literatura	53

Kapitola 1

Úvod

Tahle práce je původním záměrem, proč jsem se rozhodl studovat FIT. Vytvářet hry je moje vysněná práce a rád bych začal touto hrou. Jde o mobilní aplikaci typu „runner“, jež z názvu patrné, že účelem hlavního hrdiny neustále běžet a vyhýbat se překážkám. Her tohoto druhů existuje velmé mnoho, ale málo z nich využívá zvuk jako řídicí jednotku. Pro vytváření her se používají tzv. engines (dále se bude používat anglické slovo engine). Pro Sound Run byl použit Godot Engine – engine s otevřeným zdrojovým kódem, což je jedním z důvodů, proč jsem si ho zvolil, není třeba něco platit. Podporuje tři jazyky programování C++, C# a Godot Script. Godot Script (dále jen GS) je jazyk na základě python, v něčem je odlišný, ale když umíš python, tak umíš GS, takhle bych řekl.

Cílem této bakalářské práce je herní demo typu „runner“ v Godot Engine, kde se člověk bude řídit hlavně zvukem. Neustále běžící dopředu hrdina má tři pruhy, mezi kterými se může přepínat. Měnit pruh může v jakýkoliv okamžik. Během hry má možnost nasbírat vylepšení a mince, dokud se nenarazí na překážku.

Kapitola 2

Vyvoj mobilních her

Mobilní telefony se původně neplánovali jako herní zařízení. Mobilní hry, které se později dávali do mobilních telefonů, sloužily spíše jako doplněk softwaru, stejně jako např. kalkulačka. Všechno se změnilo s prvním mobilním operačním systémem (dále jen OS). Prvním mobilním OS byla Java ME¹. Společnost Nokia byla první, kdo si s tím začal počítat. Model Nokia 3410 už měla předinstalované následující hry: Bantumi, Bumper, Link5, Snake II, Space Impact. Dalším důležitým krokem bylo zavedení protokolu WAP (Wireless Application Protocol), který dal možnost vytvořit online-katalog her, kam vývojáři mohli nahrávat svoje hry.



Obrázek 2.1: Anglická Wikipedie ve WAP prohlížeči. Obrázek převzatý z internetu. ²

V tento okamžik už bylo jasné, že mobilní hry nejsou jen doplněk. Do vývoje her se zapojili tak velké vývojářské firmy jako Electronic Arts (The Sims 2, Doom, FIFA 06, Tetris, Tetris Mania), Ubisoft (Assassins's Creed) a Konami (PES). Tyto firmy jako prvních napadlo, že mobilní hry je atraktivní investiční prostředí. Jednoduše potřebovali přenést svoje populární hry na mobilní zařízení. A jim se to podařilo, částečně díky vývoji 3D grafiky a novým OS jako Android a IOS.

¹<https://trashbox.ru/topics/109325/istoriya-razvitiya-mobilnyh-igr>

²https://cs.wikipedia.org/wiki/Wireless_Application_Protocol

Na rozdíl od počítačových her, mobilní většinou nebyly zaměřeny na několik hodin non-stop hraní, spíše jako způsob příjemně strávit čas místo čekání ve frontách, na zastávce apod. Přestože většina her byla zdarma, bylo nutné nějak vydělávat, proto mnohem populárnější bylo přidávat nějaké placené úrovně, nové druhy zbraní nebo aut. Dalším problémem, na který narazili vývojáři mobilních her byly OS. Přesněji to, že jich bylo několik. Hry byly vytvořeny pro jeden operační systém (IOS nebo Android). Lidé si mohli kupovat mobil s ohledem na to, které hry podporuje OS. Tenhle problém byl ale celkem rychle vyřešen. Jednou z prvních her, která byla dostupná společně na Android a IOS byla Angry Birds. Jedna z nejznámějších mobilních her ve světě i v dnešní době.³ Dalším vzorcem, který nadchnul i mě, je Subway Surf. Je to nekonečný runner, kde je nutné sbírat mince a bonusy, a tak se soutěží s ostatními lidmi za místo v žebříčku.

Monetizace se liší od způsobu monetizace zavedeného u PC her. Lidé by neradi utráceli peníze za hru, kterou si možná nezhrají více než párkrát. proto jsou hry většinou zdarma. Zdarma je 93.8% všech aplikací App Store a 96.7% Google Play.⁴ Na druhou stranu by vývojáři také rádi získali peníze za svou práci, proto se objevily jiné druhy monetizace:

- První jsou nákupy ve hře, kdy uživatel nainstaluje hru zdarma a bude si moct zahrát několik úrovní, ale potom po něm hra bude chtít, aby zaplatil nějaké peníze (většinou malé), aby se mohl dostat na další level.
- Další možnost jsou taktéž nákupy ve hře, ale jiného druhu. Hráč má možnost koupit jiného hrdinu, kterým bude hrát nebo nějaké doplňky pro svoji zbraň/auto. Jinak řečeno, dá se hrát i bez toho, aby uživatel platil peníze, jenže bude mít nějaké omezení.
- Poslední možnost je reklama. Je to velmi populární varianta (z mé zkušenosti), když hra je úplně zdarma, ale jednou za několik hráčových neúspěchů nebo uhraných minut se objeví reklama. Dost často je reklama otravná a obtěžuje uživatele, ale občas může pomoci. Například za to, že se uživatel podívá na reklamu, tak dostane nějaké bonusy.

2.1 Herní engine

Termín „herní engine“ byl vytvořen v polovině 90. let v souvislosti se střilečkami z pohledu první osoby (FPS), jako jsou velmi populární hry hra Doom od id Software.

Doom byl navržen s poměrně jasným oddělením mezi základními softwarovými součástmi (jako je systém vykreslování 3D grafiky, systém detekce kolizí nebo audio systém) a grafickými aktivy, herními světy a herními pravidly, které dohromady tvoří herní zážitek. Hodnota tohoto oddělení se ukázala, když vývojáři začali licencovat hry a předělávat je na nové produkty, čímž vytvořili nové umění, rozvržení světa, zbraně, postavy, vozidla a pravidla hry s minimálními změnami v samotném enginu.

Dnes mohou vývojáři her licencovat herní engine a znovu použít většinu jeho klíčových softwarových komponent k vytváření her. I když tato praxe stále vyžaduje značné investice do vývoje softwaru na zakázku, je mnohem nákladově efektivnější než vývoj všech hlavních součástí motoru od začátku.[1, str.41–42]

Jinak řečeno, základem pro skoro jakoukoliv hru je herní engine. Mobilní hry nejsou výjimkou. Nejznámější jsou:

- Unreal Engine

³<https://dtf.ru/mobile/50025-industriya-v-karmane-kak-ustroen-mir-mobilnyh-igr>

⁴<https://42matters.com/stats>

- Unity
- Cryengine

Unreal Engine

Tento engine je určený především pro graficky náročné hry. Vývojáři tohoto enginu se zaměřili na to, aby jejich produkt zvládl nejnáročnější úkoly související s grafickou částí hry. Engine se využívá ve velmi graficky hezkých hrách, které dnes známe.

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



Obrázek 2.2: Batman: Arkham Knight je čtvrtá hra ze série Batman: Arkham. Vydání hry bylo původně naplánováno na 14. října 2014, ale později bylo odloženo na rok 2015. Podle tvůrců „Arkham Knight“ dovrší sérii her o Batmanovi, odpoví na všechny otázky vznesené fanoušky předchozích dílů příběhu Temného rytíře. Obrázek převzatý z internetu.⁶

⁶<https://knowtechie.com/att-customers-can-play-batman-arkham-knight-through-stadia-for-free-heres-how/>

https://uk.wikipedia.org/wiki/Batman:_Arkham_Knight

Může se zdát, že takový engine není určený pro mobilní hry a nejde ho použít. Není to pravda, engine se používá i pro vývoj mobilních her. Většinou však mobilní hry nepotřebují tak výkonné grafické zpracování, proto obvykle se používá. Ale má takovou možnost a jsou takové hry, například B.o.T.



Obrázek 2.3: V B.o.T hráč převezme kontrolu nad masivními mechy podobnými těm, které jsou v Titanfall. Není překvapením, že cílem je zničit mechy svých nepřátel, zatímco hráč bude bojovat o kontrolu nad mapou. Různé modely mají své silné a slabé stránky a samozřejmě jde je upgradovat pomocí herní měny. Obrázek převzatý z internetu.⁸

⁸https://www.phonearena.com/news/Best-Unreal-Engine-4-mobile-games-Android-iOS_id124882

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 jenom při výdělku nižším než 100 000 dolarů ročně. Jinak je nutné platit licenci ve výši 40 \$ měsíčně nebo 399 \$ ročně za každého člena týmu (Unity Plus). Když se přesáhne částka 200 000 \$, platí se licence ve výši 150 dolarů měsíčně. nebo 1 800 \$ ročně za každého člena týmu (Unity Pro). Používá se pro: Ori, Shipbreaker, Fall Guys.



Obrázek 2.4: Fall Guys je velmi zajímavá díky svému jednoduchému konceptu. Právě ta jednoduchost a možnost soutěžit s jinými lidmi dělá hru tak zábavnou. Hra je určena pro více hráčů a to i na místní síti. Pro hraní nepotřebujete žádné specifické dovednosti. Zaučení do hry bude trvat pár minut. Obrázek převzatý z internetu.¹⁰

¹⁰<https://www.vortex.cz/recenze-fall-guys-ultimate-knockout/>

Unity se pro vývoj mobilních her používá mnohem častěji než Unreal Engine. Lze říct, že je to neznámější engine pro tyto účely. Engine není tak dobrý na graficky náročné hry, ale, jak už bylo zmíněno, v případě mobilních her to ani není nutné. Jako příklad lze uvést jednu z neznámějších her na přežití LDOE.

Kefir (vývojářské studio) zvolil Unity, aby přenesl žánr na mobilní zařízení, rozvinul možnosti pro více hráčů a vytvořil jednoduché uživatelské rozhraní pro relativně komplexní hru. Studio také dokázalo zrychlit a optimalizovat jejich herní architekturu a kódovací úlohy pomocí Unity. Petr Kostylev, umělecký ředitel společnosti Kefir, řekl: „Jedním z důvodů, proč je pro nás Unity tak důležité je to, že usnadňuje sebevyjádření a zkoušení nových věcí“ [5].

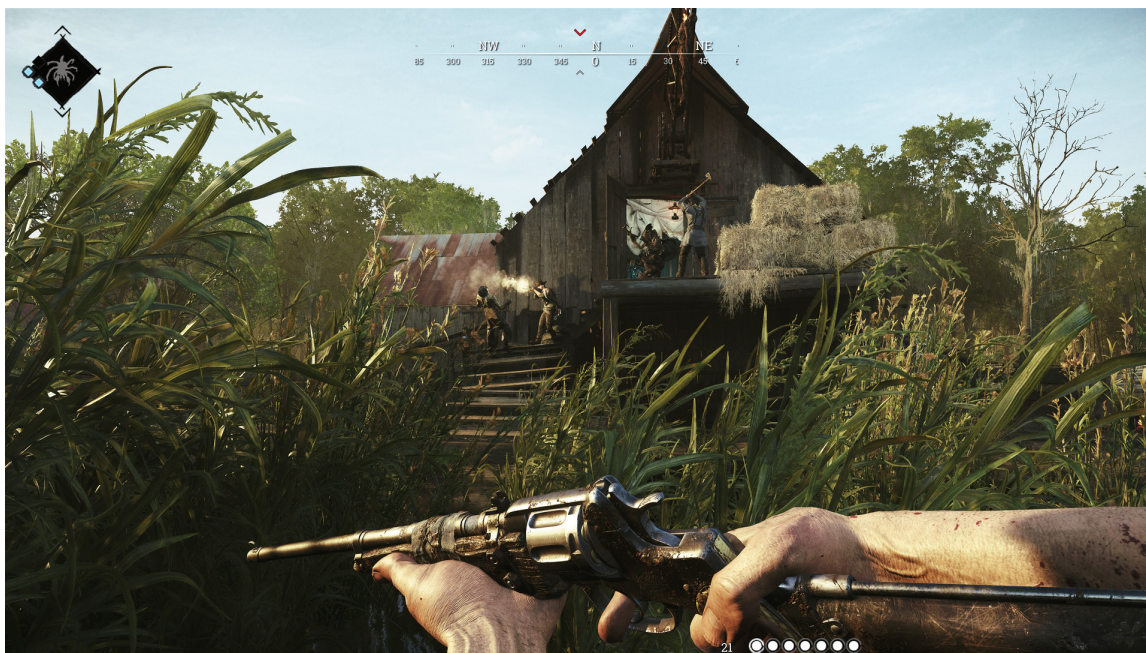


Obrázek 2.5: Last Day On Earth. Poslední den na Zemi – vytvořený studiem Kefir – je nejpobulárnější survival hra na mobilu. Kefir soft spustil LDOE za pouhých 62 dní a stal se okamžitě hitem, s dosud neuvěřitelnými 83 miliony stažení a více než 125 milionů dolarů v hrubém příjmu. Obrázek převzatý z internetu.¹²

¹²<https://logicsimplified.com/newgames/unity-game-engine-8-popular-mobile-games-that-rocked-the-game-charts/>

CryEngine

CryEngine poprvé představila velká vývojářská společnost Crytek v první hře Far Cry, je jistě jedním z nejvýkonnějších a dominujících herních enginů, které dnes existují. Co dělá CryEngine hodný tohoto seznamu, jsou jeho grafické schopnosti, které převyšují grafické schopnosti Unity a jsou ekvivalentní tomu, co má Unreal. Přestože jde o těžký a výkonný herní engine, uživateli obvykle chvíli trvá, než bude moci tento engine efektivně používat, a je o něco hůře uchopitelný pro začátečníky, kteří předtím nepoužívali žádné jiné herní enginy. Platí se 5% z vydělané částky v případě, že roční vydělek přesáhne 5 000 dolarů, v jiném případě je zdarma. Používá se pro: Crysis, Kingdom Come: Deliverance, Hunt: Showdown.



Obrázek 2.6: Hunt: Showdown. Tato hra kombinuje prvky hráč proti hráči a hráč proti prostředí. Vytváří vypjaté situace, které testují hráčovu postavu a vybavení. Za výhru hráč dostává odměnu. Jediná chyba může hráčovu postavu stát život a vybavení. Obrázek převzatý z internetu.¹⁴

2.2 Godot engine

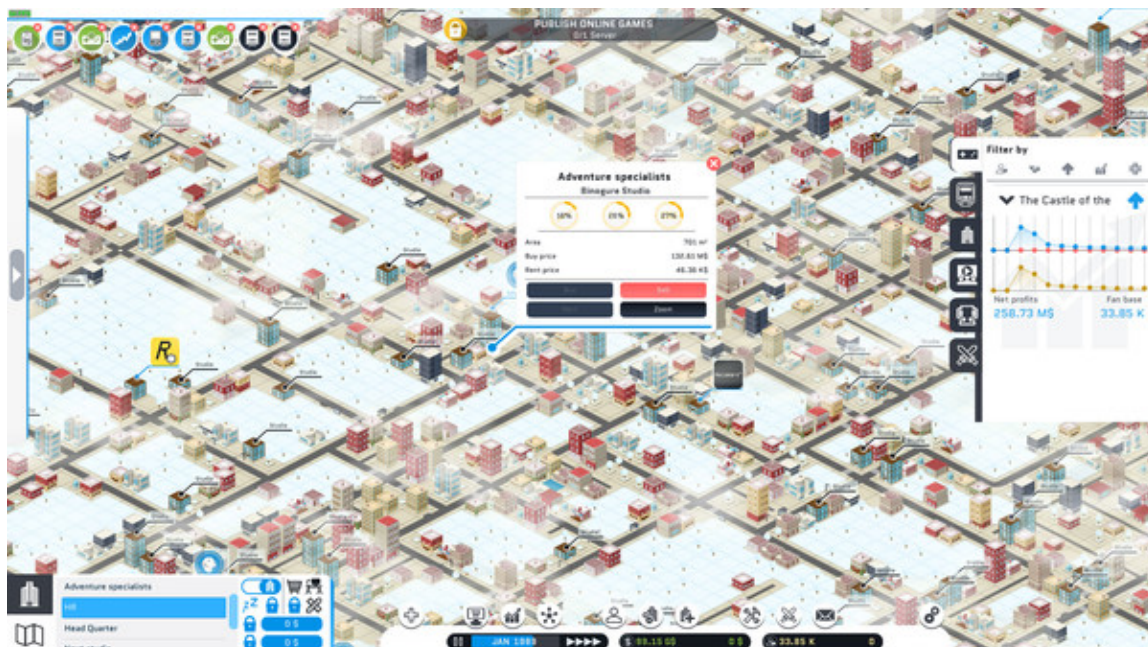
Godot engine je dle mého názoru určen především pro studenty a lidi pracující na startupu bez investic. Toto je 2D a 3D, multiplatformní open source herní engine s licencí od MIT.

„Ve skutečnosti byl Godot jen kódové označení pro něco jiného, pro obecnější účelový engine se správným uživatelským rozhraním namísto sady nejrůznějších nástrojů. Věděli jsme, že to bude trvat dlouho, a tak jsme pro vyjádření tohoto pocitu použili název založený na hře Samuela Becketa. Časem by se název změnil na něco s vazbou na naši rodnou zemi, Argentinu. V bizarním zvratu osudu jsme v určitém okamžiku a hluboko ve vývoji enginu hráli videohru s postavou, která toto spojení vytvořila (bonus ukazuje každému, kdo na to

¹⁴https://store.steampowered.com/app/594650/Hunt_Showdown/

příjde), takže engine je také pojmenován na počest k tomu. To je dobře, protože původní Godot nikdy neměl dorazit (no, to je jedna interpretace...)“^[4]

Kvůli tomu, že Godot je celkem nový, tak neexistují známé hry, které jej využívají. Je ale mnoho her, které se prodávají v steamu/Play Market/App Store. Jedna z takových je City Game Studio.



Obrázek 2.7: City Game Studio. Hra dovoluje založit společnost zabývající se vývojem videoher. Hráč pronajme si stísněný prostor, který umožní vytvořit jeho první hru. Vytvořené hráčem hry se rychle stávají úspěšnými a nutí uživatele přestěhovat na lepší místo, ale nechává na něm, aby přešel město a vybral si další kancelář. Při zvětšení počtů pracovníků bude hráč mít možnost najmout umělce, aby zlepšili vizuální část hry. Pak přicházejí vývojáři a testeři.¹⁶

¹⁶https://store.steampowered.com/app/726840/City_Game_Studio_a_tycoon_about_game_dev/

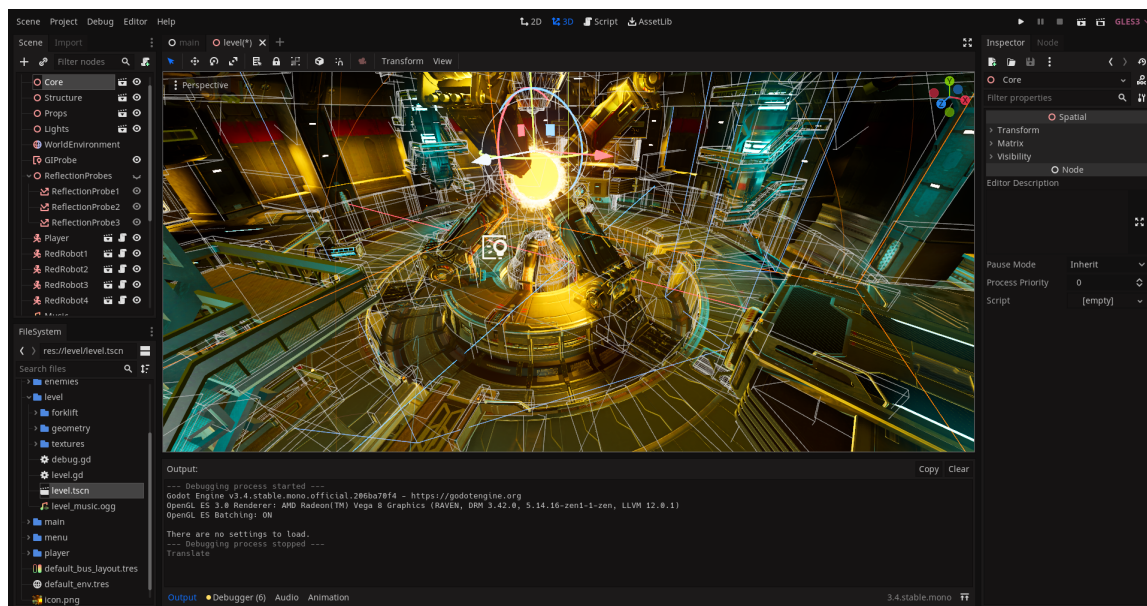
Pro mobilní zařízení existuje hra Deponia. Původně hra byla vytvořena pro PC, ale vývojáři z německé společnosti Daedalic ji přenesli na iPad, zatím pouze první díl.



Obrázek 2.8: Deponia. Deponia je klasická hra typu Point-and-Click, která byla vydána v roce 2012 na PC a byla vřele přijata kritiky i hráči. Nejprve byla Deponia koncipována jako trilogie a vše k tomu směřovalo: ve stejném roce 2012 vyšel druhý díl Chaos on Deponia a o rok později díl třetí Goodbuy Deponia. Skvělé zakončení série. Zdálo by se, že všechno ... Ale v roce 2016 nečekaně vyšla čtvrtá hra série - Deponia Doomsday.¹⁸

¹⁸<https://ipadstory.ru/obzor-deponia-na-ipad-otlichnyj-kvest.html>

Godot teď aktuálně je ve verzi 3.4.4. Rozhraní je hodně jednoduché na pochopení pro začátečníky. Většinou se dá najít všechno intuitivně. Základní jednotkou engineu je Node (dále jen noda). Nody bývají různých druhů a každá má vlastní parametry, například: Camera a KinematicBody. KinematicBody je určena pro hrdinu, něco, co se bude hybat a bude mít vlastní kolizi, váhu apod., Camera nemá žádnou kolizi a je neviditelná. Ale obě dvě jsou nody a mají společné parametry, třeba pozice. Z nod se skládají scény. Struktura scény je stromová, ze scén se pak skládají další scény a takhle se tvoří projekt.



Obrázek 2.9: Grafické rozhraní engineu. Vlevo nahoře jsou nody a scény. Vlevo dole je souborový systém. Dole je „output“. Vpravo máme aktuálně vybranou nodu. Rozhraní je možné poskládat tak, jak chce vývojář. Obrázek převzatý z internetu.²⁰

Skriptovací jazyk Godot Script (dále jen GS) je moc podobný nejpobulárnějšímu programovacímu jazyku Python. Taky je dynamicky typovaný a syntaxe je hodně podobná pythonu. Na rozdíl od pythonu, například má striktní deklarování proměnných. Naučit se GS, když umíte Python, není výzvou.

Nevyhody:

- Mladý projekt, není všechno naimplementováno, dalo by se rozšířit funkcionalitu.
- Jsou bugy, občas se opravují delší dobu.

Vyhody:

- Jazyk podobný pythonu, nejpobulárnějšímu ve světě (subjektivně ho mám rád).
- Stromová struktura scény, hned je vidět co je nad čím.
- Open source, všechno je zdarma. Pro ty, co to zajímá, tak se možná můžou k tomu i zapojit.
- Malý obsah jak engineu tak i hry. Třeba celá složka Godotu zabírá cca 950MB, ze kterých cca 210MB jsou dema.

²⁰<https://cs.wikipedia.org/wiki/Godot>

- Intuitivně pochopitelné a přívětivé GUI. Uživatel nepotřebuje hodně času a několik hodin videa na YouTube, aby pochopil kam kliknout. Je taky možnost přesunout elementy GUI tak, jak se chce.
- Možnost přidat vlastní knihovny. Tahle možnost částečně vyrovnává nevýhodu toho, že není všechno naimplementováno. Hodně lidí vytvářejí vlastní knihovny, které jsou hodně užitečné. Takže teoreticky, když není něco naimplementováno přímo v Godotu, je možné zkusit najít knihovnu, která tu možnost přidá.
- Godot hned po stažení obsahuje malé projekty, které ukazuje nejdůležitější mechaniky nebo metody. Jeden z příkladů je práce s „InAppPurchase“ pro Android. Další užitečné demo je „GUI in 3D“. Jsou i větší dema, je tu třeba demo verze celé hry.

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 hlavní hrdina se musí nacházet neustále v pohybu, například běžet, ostatní je specifické pro každou hru. Existuje hodně her tohoto typu, které buď někdy byli populární nebo jsou populární i dnes. Dále jsou jedny z nejznámějších her typu „runner“:

3.1 Runnery

Subway Surfers

Cílem této hry je co nejdéle utíkat před policistou a jeho psem. Hra začíná, když policista najde hlavní postavu, jak sprejuje po vlaku. Jakmile se začne hra, začíná hlavní hrdinu honit policista a ta před nim utíká. Mezitím se hráč musí vyhýbat překážkám nebo protijedoucím vlakům. Když stále běží, může v některých úsecích narazit na různá vylepšení jako například Coin magnet, Super Sneakers, Jetpack aj.



Obrázek 3.1: Subway Surfers byla vyvinutá dánskými společnostmi Kilo a SYBO Games. Od roku 2013 hra obsahuje aktualizací systém „World Tour“, ve kterém mohou hráči běhat po slavných městech po celém světě. Aktualizace jsou vydávány měsíčně a nabízejí nová témata, postavy a úkoly. Obrázek převzatý z internetu.²

²https://cs.wikipedia.org/wiki/Subway_Surfers

Minion Rush

Illumination, Universal a Gameloft se spojily, aby přinesly Minion Rush, rodinnou nekonečnou běžeckou hru, kde se s Miniony hráči vydají na řadu tajných misí pro Grua a gang.

Hráč má možnost závodit v okolí, uhýbat pastím v laboratoři nebo skákat do úkrytů superpadouchů, vždy se objeví nová dobrodružství a výzvy. Je možné narazit na El Macho, zahrát si hru s Margo a dívkami nebo se naučit být největším tajným agentem Miniona všech dob.



Obrázek 3.2: Minion Rush. Obrázek převzatý z internetu.³

³<https://play.google.com/store/apps/details?id=com.gameloft.android.ANMP.GloftDMHM&hl=cz&gl=CZ&pageId=none>

Temple Run 2

Hra Temple Run 2 v podstatě opakuje podstatu hry prvního dílu série. Hra začíná pokaždé v novém, samostatně vytvořeném místě s architekturou zvoleného tématu. Hráčova postava běží ruinami, prchá před monstrem a sbírá mince a bonusy. Cílem je získat co nejvíce bodů tím, že doběhnete co nejdále. V tomto případě platí, že čím větší vzdálenost postava uběhla, tím rychleji se pohybuje a překážky jsou obtížnější. Překážky v odbočení v čase, překážky, srázy a pasti, pod kterými musí postava proklouznout nebo přeskakovat, se stávají překážkou.



Obrázek 3.3: Temple Run 2 byl vydán 16. ledna 2013 v App Store na platformě iOS a 24. ledna v GooglePlay pro Android. Verze pro Windows Phone byla vydána 20. prosince 2013 a 14. října 2016 - pro Tizen. V roce 2014 se hra stala jednou z nejstahovanějších videoher v historii a byla stažena více než 1 miliardkrát. Obrázek převzatý z internetu.⁵

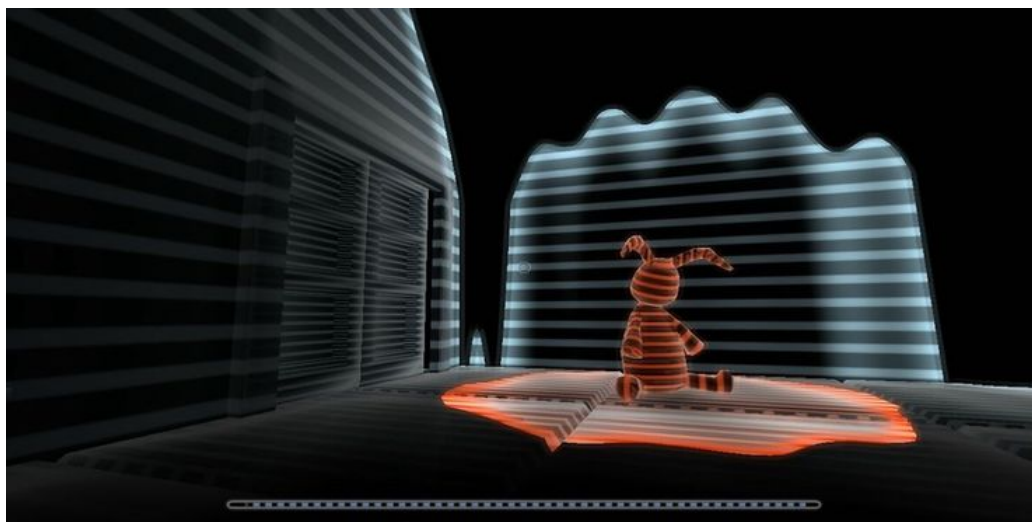
⁵https://uk.wikipedia.org/wiki/Temple_Run_2

Někteří lidé si myslí (já osobně patřím mezi ně), že runner je jeden z nejlepších typu her pro mobilní zařízení. I když v dnešní době smartphony jsou schopni podporovat i mnohem náročnější hry, pro většinu jsou to stále mobilní telefony, který není určený k nějakému “profesionálnímu” hraní. Proto zahrát nějakou jednoduchou hru během přestávky na univerzitě, v autobusu nebo ve frontě je celkem příjemná zábava. Jeden zápas může trvat tak dlouho, jak chce nebo dokáže hráč. Nestane se, že uživatel bude muset trávit hrou více času, než by chtěl, kvůli tomu, že nechce ztratit něco. Hra nedonutí uživatele k něčemu, pouze nahrazuje.

3.2 Hry s použitím zvuku

Devil’s Tuning Fork

Hra nevyužívá zvuk jako takový pro řízení, ale jeho vizuální podobu. Nápad je hodně zajímavý. V podstatě hráč jakoby vidí ten zvuk. Za mě bylo by ještě více zajímavé zahrát něco podobného, aby ten zvuk nebyl „vizuální“, ale opravdu zvukem.

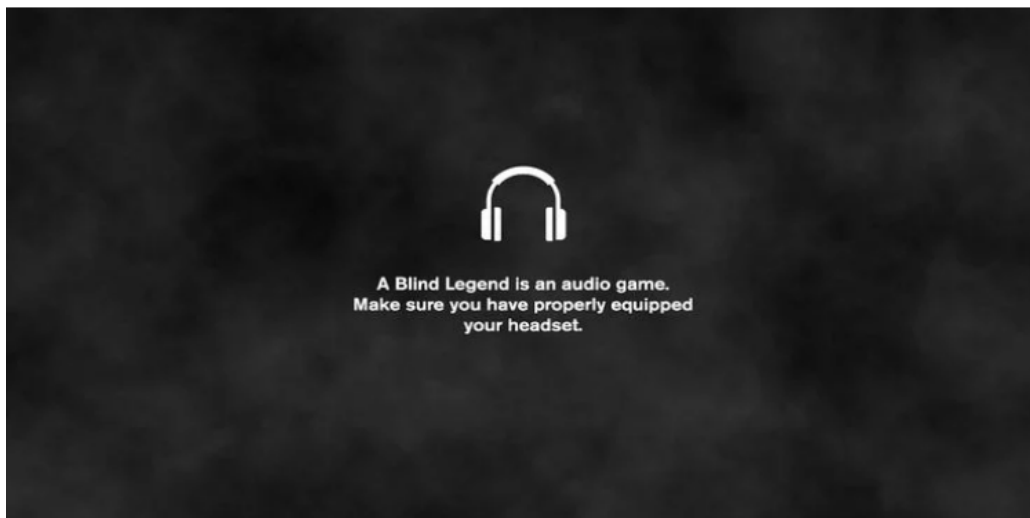


Obrázek 3.4: Devil’s Tuning Fork, vydaná v roce 2009 a založená na fyziologickém procesu echolokace, je puzzle hororová hra z pohledu první osoby, ve které hráč musí používat zvukové vlny, aby „viděl“ svět kolem sebe. Poté, co děti všude záhadně upadly do kómat, se hráčská postava, která je také dítětem, probudí v alternativní realitě, kde jsou ostatní děti uvězněny. Obrázek převzatý z internetu.⁷

⁷<https://screenrant.com/best-video-games-sound-significant-gameplay/>

A Blind Legend

Tato hra nemá žádnou vizuální část, je celá zaměřená pouze na zvuk. Je to typ her, které se jmenují „Audio Games“. O němž zná hodně málo lidí a není moc populární. Lidí nejsou zvykli na to, že hru pouze slyšíš a nic nevidíš, proto takové hry nejsou moc známé. Za mě, dalo by se kombinovat takový typ her s obyčejným a vyvíjet oba zároveň. Přičemž takovou kombinací by šlo rozšířit posluchačstvo pro hru.



Obrázek 3.5: A Blind Legend, vytvořená jako přístupná zkušenost pro zrakově postižené jedince a pro šíření povědomí o zrakových postiženích, je hra z roku 2015 o slepém rytíři jménem Edward Blake a byla vůbec první akční dobrodružnou hrou, která zcela spoléhala na zvuk. Edwardova žena, odehrávající se ve středověkém fantasy světě, byla právě zajata, a tak cestuje přes Království vysokého hradu, aby ji zachránil. Obrázek převzatý z internetu.⁹

Hry typu runner mám rád osobně a jsou hodně populární, ale dělat další hru, která se nebude lišit od předchozích je nuda. Zvukové hry nejsou populární, ale nápad použití jiného smyslu pro hru je zajímavý. Sound Run v podstatě je kombinaci těchto dvou typu her. Hráč vidí hrdinu, vidí cestu a nějaké vylepšení, ale uhnout se překážkám může pouze díky zvuku.

⁹<https://screenrant.com/best-video-games-sound-significant-gameplay/>

Kapitola 4

Návrh

Chtělo se do žánru přinést něco nového. Zatím Runnery jsou více méně sobě podobní. Liší se většinou pouze okolím. Bylo ale nutné vymyslet něco, co zatím nemá žádná populární hra, a to je použití zvuku. Zatím se zvuk nepoužíval pro něco důležitého, většinou se to dalo vypnout a nic podstatného se nezmění. Je to logické, protože většinou se takové hry hrají na společných prostorech, kde by zvuk hry mohl rušit ostatní.

V dnešní době tendence se trochu mění, vývoj bezdrátových zařízení se zrychluje, a to nejdůležitější pro “Sound Run” (dále SR) je, že většinou lidi mají sluchátka. Takže problém rušení ostatních zvukem už není tak aktuální díky nim. Tohle otevírá další možnost pro vývoj mobilních her, která se využívá v SR.

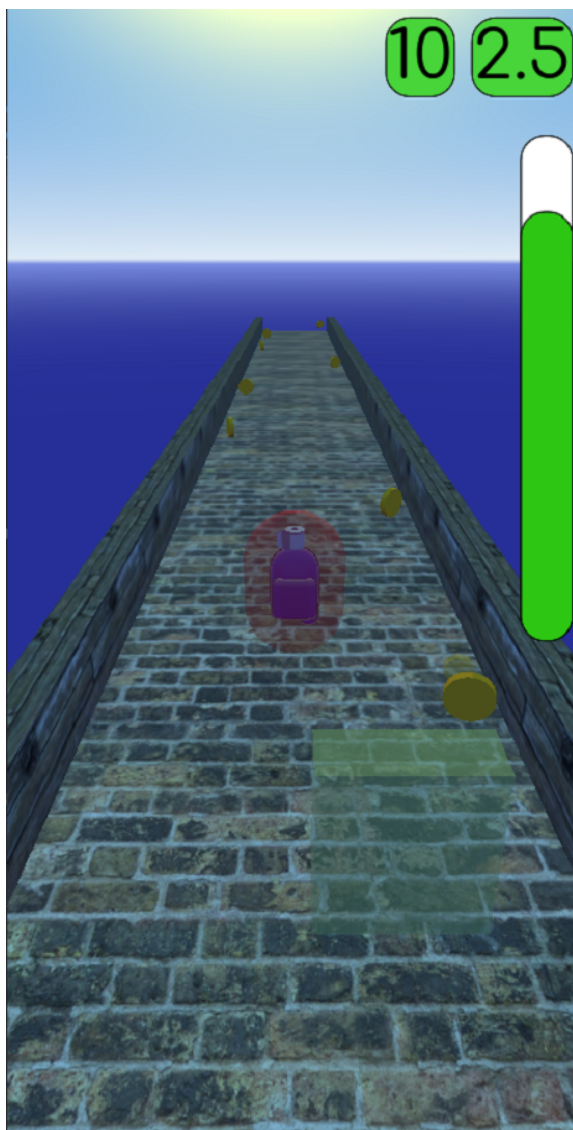
SR, jako i většina runnerů, má hrdinu, který nekonečně běží. Má tři pruhy, mezi kterými je možné se přemístit v jakýkoliv okamžik. Uhnout se překážkám jde způsoby: přejít na jiný pruh, skočit a přisednout. Čím delší dobu hráč dokáže přežít, tím rychleji bude běžet. Jsou tu překážky různého druhu, před kterými je nutné vyhnout a elementy typu „platforma“, které mění povrch. Rozdíl oproti ostatním je ten, že překážky jsou neviditelné.

Sound Run využívá zvuk jako způsob ukázat hráči, kde se nachází překážka. Záleží, jaký zvuk a jakou rychlostí je slyšet, na základě tohoto by hráč měl vědět na jaké straně, jak daleko od něj a jakého druhu je překážka, před kterou se má vyhnout. Z tohoto důvodu překážky jsou neviditelné. Cílem je nasbírat co nejvíce mincí.

Hra se začíná tím, že ve hlavním menu uživatel hru nastartuje. Z disku, v případě některých objektů, jako třeba překážky, tak z paměti, budou nahrané všechny elementy sceny. Pak se nahraje zvolená hrdina a nastaví se vylepšení, které má zabudovanou, když to má.

V tento okamžik se začíná samotný běh. Hned po začátku se spouští všechny timery, které jsou zodpovědní za generaci elementů hry: mince, platformy, překážky, vylepšení. Po skončení každého timeru (každý má vlastní čas), provede se generace příslušného objektu a timer se spustí znovu. Objekty se generují na nějaké vzdálenosti od hrdiny. Uživatel se zdá, že tihle objekty vždy byly tam, kde je potkal, protože má omezenou delku, na kterou vidí.

Zaučení do hry nebude trvat moc dlouho, zvuky jsou dost odlišné od sebe a po několika zápasech hráč už bude schopný jistě předvídat překážky dle zvuku. Náročnější je pochopit jak daleko překážka je. Tím, že mechanika je nová, tak jak ukazuje praxe, v prvních zápasech se lidi ztrácejí a jakmile uslyší zvuk, tak hned se snaží reagovat. Což není správně, protože překážka je tak daleko, že třeba po skoku hráč spadne přímo na ni. Po krátkém výcviku člověk zvykne a ti nejzkušenější dokážou přeskocit i to, co obvykle přeskocit není možné. Například další překážka (viz obrázek).



Obrázek 4.1: Jeden z druhů překážek. Překážka omezující celou linku.

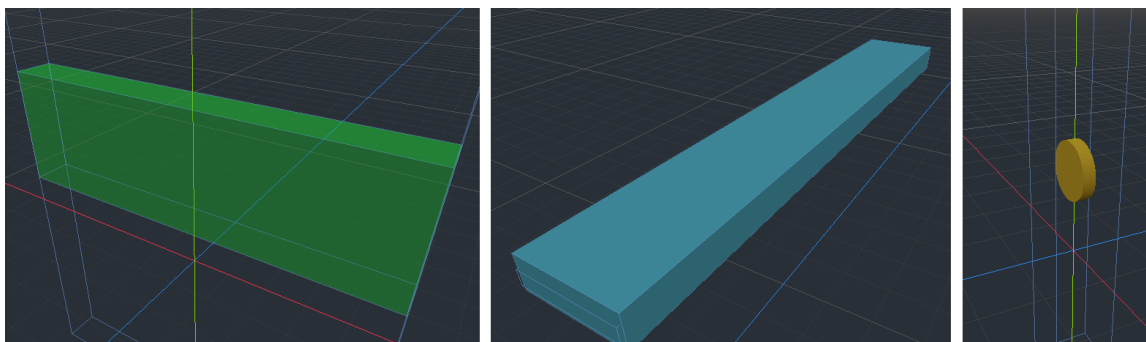
Tenhle druh překážky omezuje pouze jednu linku, ale celou, je možné ji jenom obejít. Není to ale celkem pravda, teoreticky tuhle překážku je možné přeskočit. Prakticky je to hrozně náročné, ale ti nejzkušenější to zvládnout (aspoň někdy).

Účelem práce není konečný produkt, ale demonstrace mechanik a základních věc. Z toho důvodu hra nemá skoro žádné vizuální efekty. Hrdina a veškeré modely/textury budou vyměněny v případě vydání hry.

4.1 Komponenty hry

Princip je hodně jednoduchý a intuitivně pochopitelný všem. Hrdina běží jedním směrem, pomocí swipů lze pruh změnit. Jsou pak další elementy mapy:

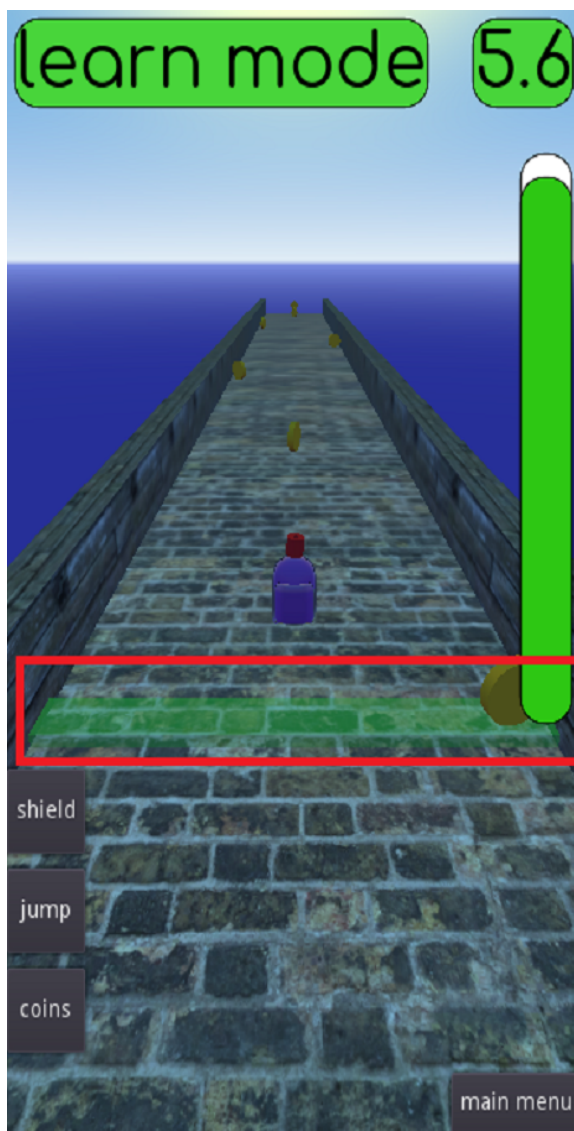
- Překážky - když narazí, tak buď zemře nebo ztratí jeden štít, pokud ho má
- Platforma - přední část je nebezpečná, hrdina zemře, když narazí zepředu. Je ale možné stát na ni shora
- Mince - herní peníze. Sbírají se během zápasu a je možné utrácet za vylepšení nebo nové hrdiny s bonusy.



Obrázek 4.2: Vlevo na obrázku je zobrazeno překážku, v centru je platforma a vpravo je mince.

Pochopit rozmístění a typ překážky je možné pomocí zvuku. Každá překážka má vlastní zvuk, frekvenci a rychlost. Čím blíže je hrdina k překážce, tím více se zrychluje zvuk. Po nějaké době cvičení už není problém rozlišit typ bariéry. Zatím existuje 4 druhy překážek:

bottom_full je to překážka, která je rozmístěna na zemi a to přes celou cestu. Takovou překážku je možné pouze přeskočit. Je nejjednodušší typ bariéry.



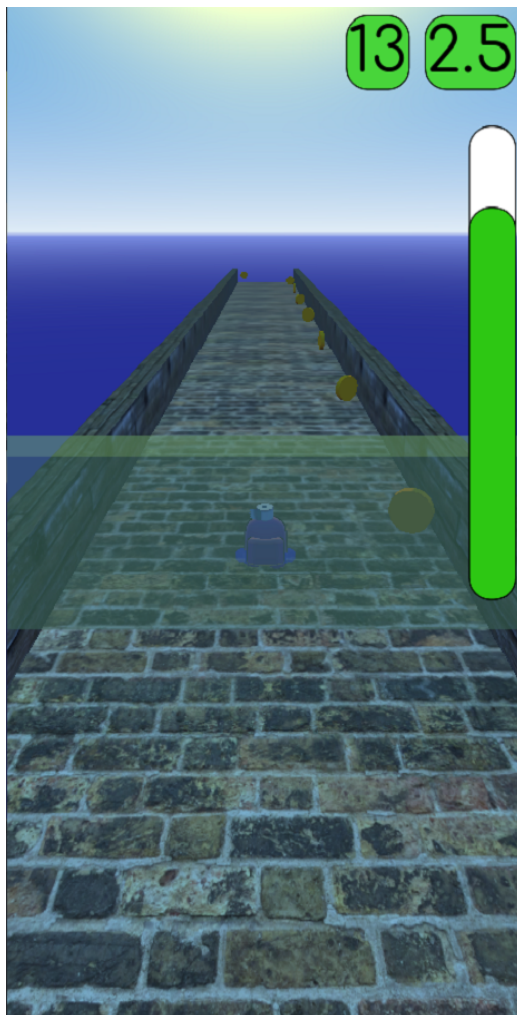
Obrázek 4.3: Překážka přes celou zem. Neznámější druh překážek v skoro jakémkoliv run-neru.

bottom_2_of_3 je další bariér, který je hodně podobný předchozímu. Taký je na zemi a vypadá skoro stejně, jen že není přes celou mapu, ale jen na 2/3 mapy. To znamená, že je možné ho nejen přeskočit ale i obejít. Tohle je jediná překážka, která nemá vlastní zvuk. Má stejné audio jako předchozí. Je to jakýsi chyták.



Obrázek 4.4: Překážka přes 2/3 zemí. Další variace předchozí bariéry.

top_full. Tato bariéra je absolutně jiná. Je to největší překážka ve hře. Je, stejně jako první, přes celou cestu, ale není na zemi. Tohle je zatím jediná překážka, před níž je možné vyhnout pouze přisednutím. Výjimka je, když hráč má vylepšení „Jumpers“, ale o tom později.

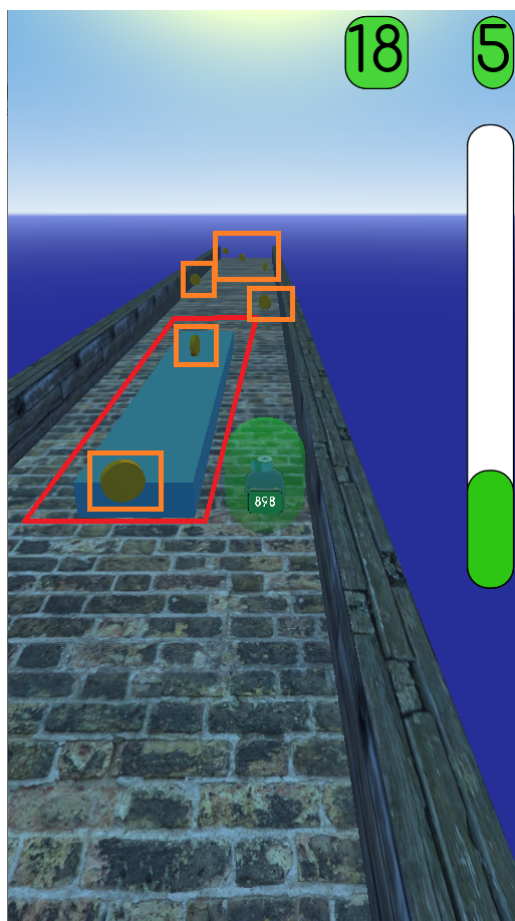


Obrázek 4.5: Překážka ve vzduchu.

one_side_full - je to překážka, která omezuje jednu linku ze vzduchu a ze zemi. Více viz v [návrhu](#).

Jsou také platformy, narazí na něž zepředu se rovná smrti. Pomocí platformy je možné přeskóčit překážky, které nešlo dříve. Například bariéru *top_full* bez pomoci přeskóčit nejde. Pomocí platformy hráč je ale výše než obvykle, proto je schopen ji přeskóčit bez větších problémů.

Platforma ale může i zachránit, když hráč na ni skočí, aby se nedotknul přední části platformy. Platforma je dost vysoká, první dva druhy překážek (*bottom_full* a *bottom_2_of_3*) nejsou pak schopni uškodit hráči. Takže je to spíš pomoc, jak něco špatného. Platforma žádným způsobem neovlivňuje sbírání mince nebo vylepšení.

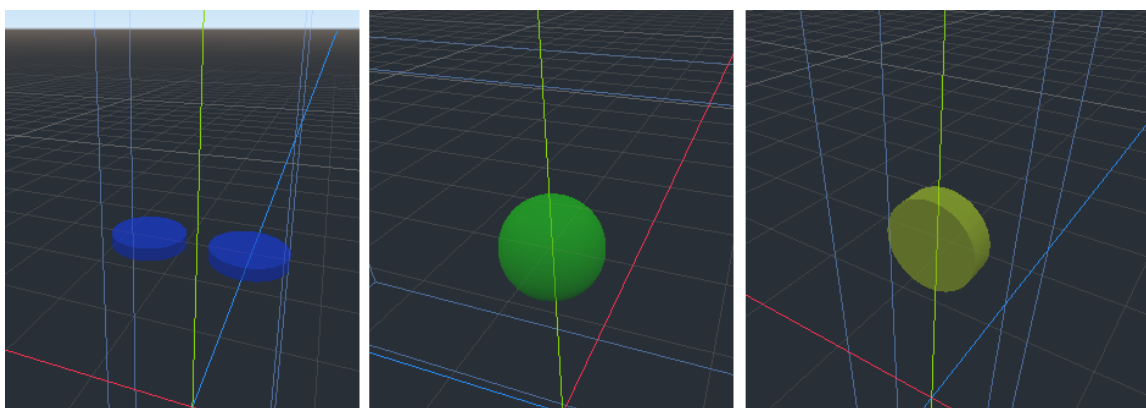


Obrázek 4.6: Platforma a mince.

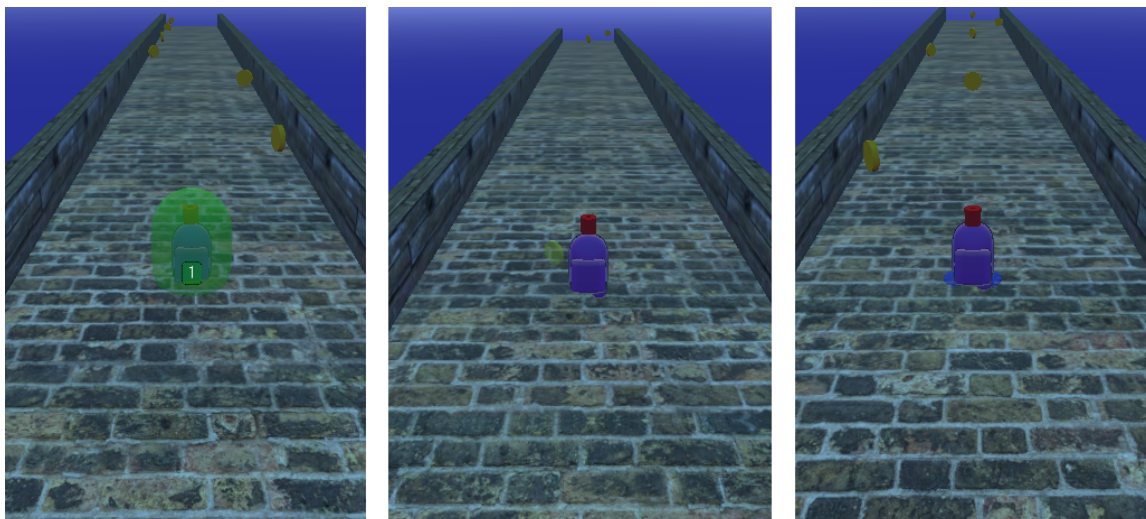
Je i jednoduchý režim, především určený pro zaučení, protože od začátku uživatel neví, jaké překážky odpovídá konkrétní zvuk, jak se ten zvuk chová při sblížení apod. Liší se od klasického režimu tím, že překážky se stávají viditelné v okamžik, když se zapne zvuk. V tomto režimu hráč nemůže nasbírat mince, jde pouze o zaučení. Takovým způsobem po několika pokusech už je možné začít hrát klasický režim. Jednoduchý režim je dostupný v jakýkoliv okamžik a je možné ho hrát i místo klasického.

Jen tak běžet by bylo hodně nudné, na cestě se občas vyskytují bonusy a vylepšení, například:

- Skokové vylepšení (jumpers) - hrdina dokáže skočit mnohem vyš než obvykle, jsou aktivní nějakou dobu, nejde použít více jak jedno vylepšení tohoto druhu. Jumpers dávají více možnosti pro hru. Kromě toho, že hodně ulehčují přeskočení jakékoliv překážky, tak hráč s nimi může přeskočit i to, co dřív možná nebylo, například překážku *top_full*. Překážky jsou opravdu fyzicky takové, jak vypadají, proto pomocí Jumpers hrdina skáče výše, než *top_full* se nachází. Na první pohled je to super, ale popravdě řečeno, častěji kvůli tomu hráč ztratí život.
- x2 mince (double coins) - každá sebraná mince se počítá za dvě, je aktivní nějakou dobu, nejde použít více jak jedno vylepšení tohoto druhu. Použití tohoto vylepšení se nevyplatí na male rychlosti. Je lepší si pořádně rozběhnout.
- Štít (shield) - kolem hrdiny se objeví štít, který zachrání oproti setkání s jednou překážkou. Je aktivní až dokud hráč ho neztratí. Štíty se hromadí a je možné vidět jejich aktuální počet. Štít je v podstatě něco jako život ve většině her. Původně hráč má jenom jeden, pak s každým dalším štítem jeden navíc. Pozor, štít nemůže zachránit proti setkání s přední částí platformy. Narazit se na ní bude znamenat smrt v jakémkoliv případě. Po smrti všechny štíty zmiznou a další zápas se nastartuje „od nuly“.



Obrázek 4.7: Všechny bonusy, které je možné potkat za běhu. V takové podobě se zobrazují na cestě. Vlevo jsou skokové vylepšení (jumpers), v centru je štít, vpravo je x2 mince (double coins).



Obrázek 4.8: Vylepšení při použití hráčem. Model všech vylepšení na hrdinně. Vlevo je štít, v centru je x2 mince (double coins) a vpravo je skokové vylepšení (jumpers).

4.2 Kamera

Když se bavíme o kameře ve hře, tak to není věc, která natáčí video. Je to objekt, který reprezentuje lidské oko. V kameře je možné změnit pozorovací úhel a ostatní věci. Její rozmístění a chování zajišťuje tak zvaný „typ kamery“. Často se neříká „typ kamery“, ale místo toho prostě „kamera“, například: „kamera z pohledu první osoby“.

Existuje hodně druhů kamer dle jejich rozmístění a chování. Základní ale jsou dva: z pohledu první osoby a z pohledu třetí osoby. Z pohledu první osoby znamená, že hráč vidí hru tak, jakoby byl ta hrdina, takže kamera je rozmístěna přímo v hrdinně (obvykle je ve hlavě). V takových případech hráč nevidí vlastní hrdinu ale to, co je před ní. Kamera z pohledu třetí osoby je naprosto opačný druh. Má různé poddruhy, záleží kde přesně je rozmístěna.

Nejpopulárnější jsou další [3]:

- „Ze vzduchu“ - kamera, která je umístěná nad herní plochou. Používá se ve strategických hrách jako Civilization apod. Obvykle takovou kameru je možné posouvat a ona není připojena k hrdinně nebo ještě něčemu. Pomocí velkého uhlu hráč vidí zároveň celou mapu a může řídit, třeba dvě armády ve dvou různých částech herního světa.



Obrázek 4.9: Civilization VI. Globální tahová strategická videohra Sida Meyera vyvinutá společností Firaxis Games. 2K Games byla vydána 21. října 2016 a v únoru následujícího roku byla videohra portována na Linux. Verze pro Nintendo Switch byla vydána 16. listopadu 2018. Od 22. listopadu 2019 je k dispozici také pro Xbox One a PlayStation 4 a od 13. srpna 2020 pro Android. Tradičně pro sérii hráč řídí vývoj národa, stará se o jeho vývoj, vede války a diplomatická jednání. Obrázek převzatý z internetu.²

²https://uk.wikipedia.org/wiki/Civilization_VI

- „Ze vzduchu (připojená)“: Existuje a používá se taky trochu jiná kamera a to v RPG hrách jako třeba Diablo. Kamera je taky ze vzduchu, ale je připojena k hrdinně. Většinou v takových hrách řídí jednu hodinu nebo několik ale má možnost se mezi nimi přepínat. Hráč může vidět okolí, ale nemůže kamerou hýbat.



Obrázek 4.10: Diablo. akční RPG videohra, zakladatel série Diablo od Blizzard Entertainment. Hra byla vydána 30. listopadu 1996. Existuje doplněk Diablo: Hellfire, vyvinutý společností Sierra Entertainment. Přidává nový příběh, lokace a třídy postav. Hra se odehrává ve městě Tristram, království Handaras, kde se probudil uvězněný démon Diablo. Hráč hraje roli hrdiny, který opakovaně sestupuje do žaláře, aby konečně našel a porazil ďábla. Obrázek převzatý z internetu.⁴

⁴<https://uk.wikipedia.org/wiki/Diablo>

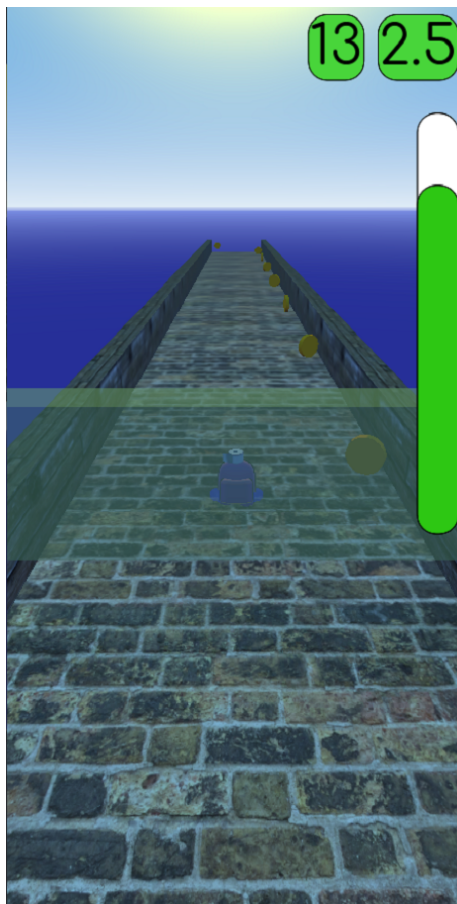
- Kamera rozmístěna vzadu. Je třetí z nejpobulárnějších typu rozmístění kamery z pohledu třetí osoby. V tomto případě kamera se bude „divat“ na hrđinu zezadu. Existuje více poddruhů takové kamery: někteří je možné točit, jiné posouvat, třetí jsou fixní aj. Je možné potkat i kombinaci těchto typů, třeba hlavní kamera je fixní zezadu, ale je možné přepnout na jinou kameru, která je zepředu. Je to možné vidět třeba ve hře Need For Speed: Payback. Při stisknutí tlačítka (záleží jaké je nastaveno) rozmístění kamery se mění a hráč se může podívat na auto zepředu.



Obrázek 4.11: Need for Speed: Payback - série videoher Need for Speed v žánru arkádového automobilového simulátoru vyvinutá společností Ghost Games. Hru vydala společnost Electronic Arts pro konzole PlayStation 4 a Xbox One a také pro osobní počítače se systémem Windows 10. listopadu 2017. Hra Need for Speed: Payback se zaměřuje na pouliční závody, zločin a policejní honičky. Obrázek převzatý z internetu.⁶

⁶https://uk.wikipedia.org/wiki/Need_for_Speed:_Payback

Pro Sound Run bylo zvoleno kameru z pohledu třetí osoby, třetí typ z těch, co jsou vysvětlení vyše a je to kamera umístěna vzadu:



Obrázek 4.12: Kamera z pohledu třetí osoby.

Takový druh kamery je nejčastější pro hry typu „Runner“. Hráč vidí svojí hrdinu a všechno, co je kolem ní. Rozmístění kamery nad hrdinou zvětšuje délku, na kterou hráč vidí a může naplánovat svůj běh (třeba vepředu bude vidět platformu nebo vylepšení). Pro hru je ale dobré, že hráč tu kameru nemůže otočit. To znamená, že všechno, co je za kamerou je možné promazat z paměti. Nebude to mít žádný špatný vliv na herní proces, ale hrozně zlepší výkon. Když se bavíme o mobilním zařízení, tak výkon je hodně důležitý. Žádný hráč nebude hrát hru, která vybije jeho mobil za par minut.

4.2.1 Zvuk

Na zvuk šířený vzduchem je možné pohlížet jako na změnu tlaku vzduchu v ovzduší. Má vlnovou délku, frekvenci a intenzitu. Zvuk se šíří od zdroje do místa přijetí v prostředí. Když udeří energie do molekul v prostředí, způsobí, že molekuly vibrují sem a tam a vytvářejí vlnu, která přenáší zvukovou energii. Lidské ucho je velmi citlivé a potřebuje jen nízkou míru energie, aby slyšelo. Sluch lidského ucha se pohybuje v oblasti mezi 0 dB (práh slyšení) a 120 dB (práh bolesti) o frekvencích mezi 20-20000Hz. Kmitočty, které spadají pod oblast sluchu, jsou známé jako infrazvuk a kmitočty nad 20000Hz se nazývají ultrazvuk.[11]

Pro Sound Run bylo nutné najít 3 různé zvuky, které by možno bylo odlišit od sebe. Zvuky mají být opakovatelné a moc se neměnit s časem. Je to hodně neprofesionální z pohledu vyvoju her, ale pro herní demo stačí.

Když je zvuk připraven zaujmout své místo ve hře, zvuky se přenesou na vývojáře. Poté jsou s pomocí zvukaře se uvadí do hry. Zkoušejí zvuk, koukají, jak zní v enginu. V případě potřeby se mění jeho frekvenci, optimalizuje se a tak dále.[7]

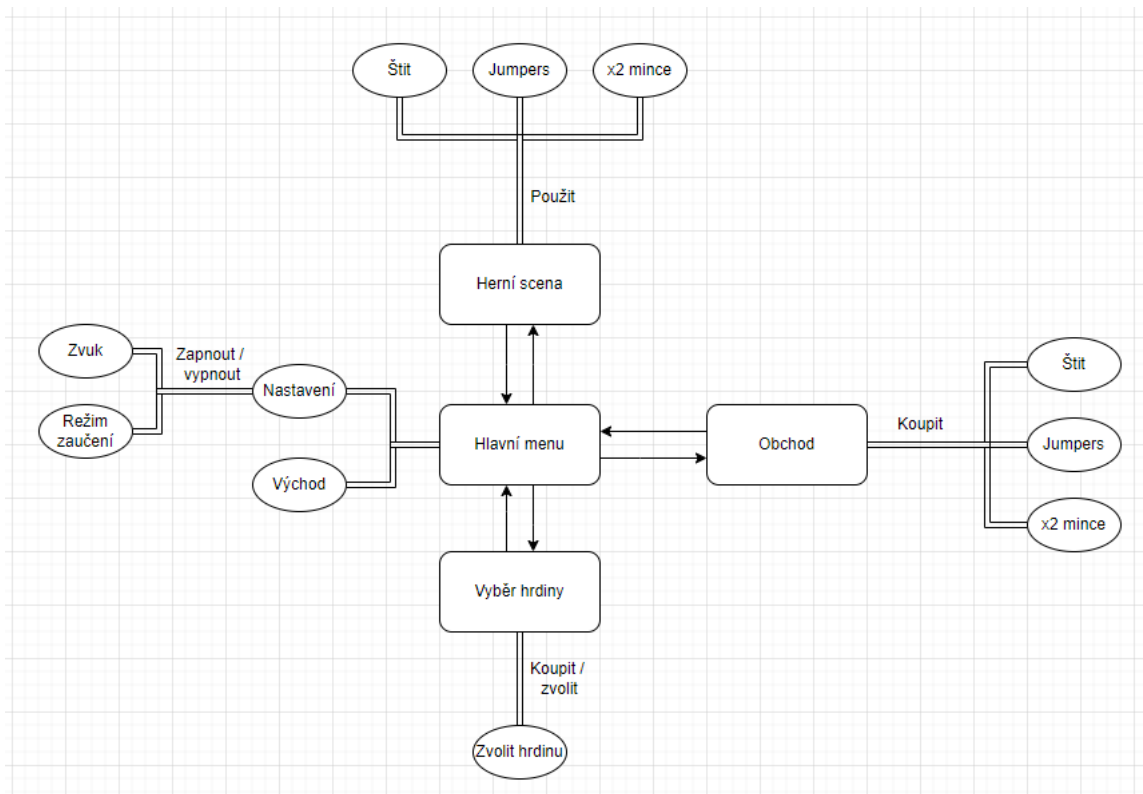
Zvuk je důležitou částí hry, dá se říct, že je její hlavní a unikátní část. Pro vybrané audio, na základě článku výše a dokumentace Godotu bylo zvoleno frekvencí 17500 Hz. Ve všech třídách, které jsou zodpovědní za zvuk, bylo nastaveno tuto frekvencí.

Godot udává možnost kromě nastavení frekvence nastavit ještě tempo audio. Tempo je důležité, díky němu zvuk se mění v čase. Právě pomocí tohoto parametru je možné pochopit jak blízko překážka je.

Kapitola 5

Implementace

Celý projekt se skládá ze scén. Scéna se rozumí buď celá obrazovka nebo jen nějaká její část. Scéna se může skládat z jiných scén. Na diagramu jsou všechny základní scény a jejich elementy, se kterými je možné interagovat. Šipka ukazuje ze které scény na kterou je možné přejít. Přepínání mezi scénami je možné pomocí metody *change_scene()*. Takhle metoda kompletně změní scénu, to znamená, že předchozí scéna bude promazaná z paměti. Je to důležité, protože mobilní zařízení, zatím, nemají tak velký počet RAM paměti jako počítač. Dvojitě linie jsou typ interakce, čtverečky zobrazují samotné obrazovky, a elipsy znamenají, s čím se provádí interakce.



Obrázek 5.1: Diagram základních scén a jejich interakce. Zobrazuje komunikaci mezi jednotlivými scénami a způsob komunikací. Z diagramu je vidět, že hlavní menu je první scéna, kterou hráč uvidí. Dostat se může do obchodu, na hlavní herní scénu a na výběr hrdiny. Rovnou ve hlavním menu může buď zavřít aplikaci nebo otevřít nastavení. Přejít mezi scénami je zpětný, tzn. uživatel se může vrátit na hlavní menu v jakýkoliv okamžik.

Godot má možnost nastavit tzn. „autoload“. To znamená, že všechno, co tam bude přidáno, bude se zpracovávat hned po spuštění aplikace a bude dostupné ve všech skriptech. V projektu „Sound Run“ se používá pro ukládání globálních proměnných a různého druhů nastavení. V případě této hry na to je určený soubor *global_vars.gd*. Je možné přidat i nějaké scény, nebo uzly (nody). Může se používat pro spuštění nějaké hudby na pozadí. V takovém případě hudba bude

5.1 Hlavní menu

Hlavní menu je jediná pouze 2D scéna. Při prvním spuštění se zobrazuje ale jiná scéna, která se jmenuje „main“. Scéna „main“ zahrnuje hlavní menu, script pro nahrání všeho uloženého na disku a zpracování shaderů/materiálů. Z toho všeho uživatel vidí pouze hlavní menu, všechno ostatní běží na pozadí.

První, co se má udělat, je nahrání uloženého z disku. To znamená, že všechny nastavení (je-li zapnutý zvuk, zaučovací režim) a progres (počet mincí, zakoupené zboží a hrdiny) se nahraje do paměti z disku. Pro tento účel existuje skript v „main“ scene, který zkontroluje, zda existuje soubor pro načítání, když není, tak to znamená první spuštění programu, a soubor se vytvoří. Jinak se všechno načte a uloží do globálních proměn nebo se rovnou nastaví.

Práce se soubory se skládá ze 3 kroků:

- Otevření souboru. Pomocí metody `open(„cesta do souboru“, typ komunikace)`, kde cesta do souboru je typu `String` a „typ komunikace“ je buď čtení nebo zápis. „`user://`“ je kořenový adresář pro jakýkoliv soubor, proto všechny cesty do souborů se začínají tímto. Funguje pro všechny OS a Godot je schopen vytvořit file na správném místě pro každý operační systém. Před čtením ze souboru je nutné zkontrolovat, jestli opravdu existuje, toto Godot taky umí a má na to nástroje.
- Dalším krokem je načtení samotného souboru nebo zápis do souboru. Čtení a zápis se může lišit, záleží co právě chceme uložit/přečíst: proměnnou, řádek, číslo apod.
- Poslední krok je uzavření souboru.

Druhé je zpracování shaderů a materiálů. Každý element hry, jako mraky nebo překážky, má svůj materiál. Materiál má barvu, průhlednost, shader a jiné nastavení. Když takový element je nutné zobrazit, tak se provádí výpočty na grafické kartě (nebo CPU, když GPU není) a potom je možné ho zobrazit. Ty výpočty jsou dost náročné, což může způsobit zaseknutí celé hry na velmi krátkou dobu. Je to ale hodně vidět ve hře takového typu, proto jedním z možných řešení je provést tyhle výpočty během spuštění hry. Když se to udělá jednou, potom se ukládá do paměti a objekt, při dalším zobrazení na kameře, nebude způsobit žádné zaseknutí.



Obrázek 5.2: Main scéna. Na obrázku je vidět, že scéna „main“ je poskládaná z dvou jiných scén, „main_menu“ a „preload_all_elements“, taky má přidáno script. První scéna je 2D scéna typu „GUI“ (má zelenou barvu), druhá je 3D scéna (růžová barva).

Samotné menu se skládá z 5 tlačítek a 1 políčka.

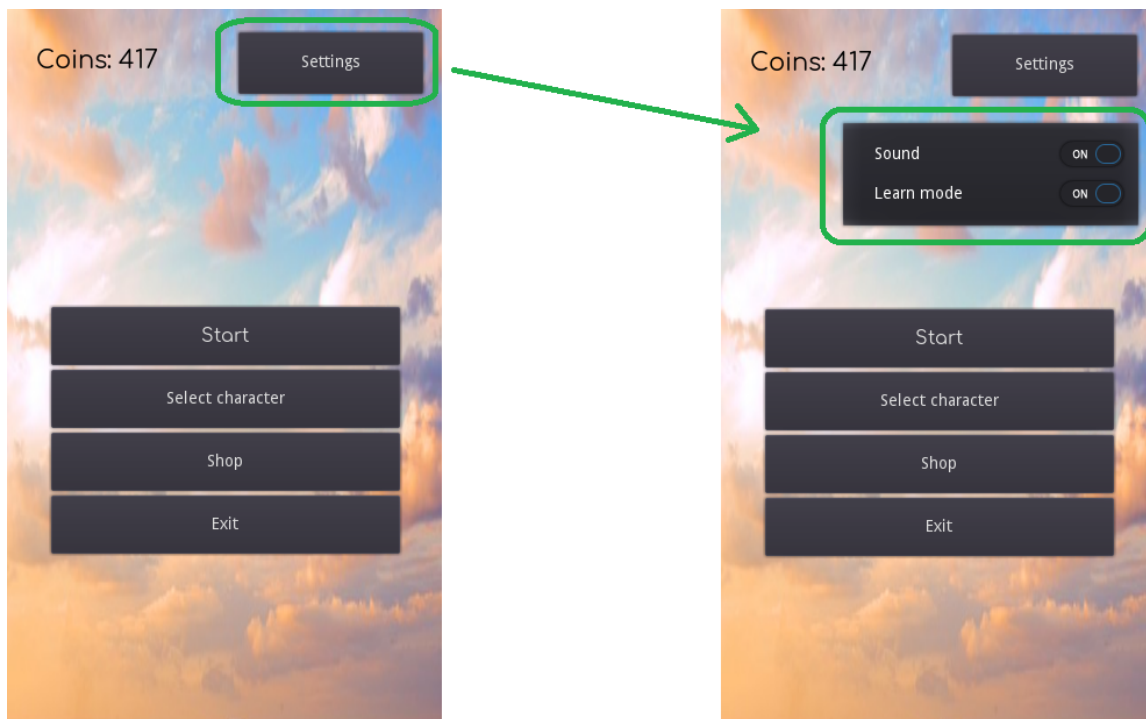
Pole ukazuje počet mincí, které hráč má. Pro zobrazení textu se používá noda *Label*, která je hodně jednoduchá a užitečná. Noda umožňuje regulaci rozmístění textu vertikálně a horizontálně, dává možnost nastavit automatický přenos textu, maximální počet viditelných znaku, řádků apod.

Taky jsou „CheckBox“, „CheckButton“, „MenuButton“ a další. Když se bavíme o tlačítkách, tak je důležité probrat signály. Signály je něco, co se posílá při nějakém ději, například kliknutí na tlačítko posílá signál „pressed“. Do skriptu je možné přidat zpracování signálu. Tímhle způsobem je implementováno zpracování skoro všech tlačítek. Existují pak další signály, jako třeba „button_down“, „button_up“. Signály mají všechny nody, nejen tlačítka. Třeba každý element menu má signál „focus_entered“, který je možné zjistit, že něco bylo zvoleno nebo někam bylo kliknutí. Existuje hodně druhů různých signálů [2, Signals].

Tlačítka scény menu jsou:

- Start - začít běh. Proveďte se změna scény na scénu běhu. Je to hlavní scéna hry.
- Select character - přechod na scénu výběru hrdiny.
- Shop - přechod na scénu obchodu
- Exit - východ. Uzavře se celá aplikace.
- Settings - otevření pop-upu s nastavením. Viz obrázek 5.3

Tlačítka jsou uvnitř nody, která se jmenuje *VBoxContainer*. Výhodou té nody je to, že ona dokáže automaticky rozmístit rovnoměrně všechny nody, pro které je rodičovskou. Proto je ideální pro tlačítka. Existuje taky hodně podobná noda, která se liší od *VBoxContainer* pouze směrem. *VBoxContainer* zajišťuje vertikální rozmístění, *HBoxContainer* řídí horizontální rozmístění. Pomocí těchto nod je možné poskládat tlačítka a jiné dceřiné elementy hezky a rychle.



Obrázek 5.3: Hlavní menu. 5.1

5.2 Vyběr hrdiny

V každé hře má být něco hazardního, něco, na co by měl hráč šetřit hodně mincí. V případě Sound Runu to jsou nové hrdiny. Každý hráč si může koupit jiné hrdiny, než ta základní. Základní nemá žádný bonus, na rozdíl od jiných. Zakoupené hrdiny se liší vzhledem a bonusem. Každá má něco navíc. Zatím existuje jen 3 druhy hrdinů: základní, rychlejší a ta, co může výše skákat. Já už bylo zmíněno, základní žádný bonus nemá. Rychlejší má vyšší počáteční rychlost, a to nejdůležitější - má větší zrychlení. Při zakoupení té poslední hráč má možnost skočit o něco výše. Může se zdát, že je to mnohem slabší bonus, jak to zrychlení. Ono to ale není tak, zkušený hráč s touto hrdinou dokáže přeskocit i překážku, která na to určená není. Plánuje se přidání i dalších hrdinů, s bonusy, jako třeba zvětšení délky paprsku, který reaguje na překážky nebo něco podobného.

Scéna výběru hrdiny umísťuje:

- Počet mincí na stejném místě jako v hlavním menu
- Tlačítko „Hlavní menu“ pro návrat do hlavního menu
- Modely hrdiny

Scéna je rozdělena na dvě části: 3D a 2D. Všechny tlačítka a labely jsou součástí 2D scény, pro kterou rodičovskou je 3D.

Model se zobrazuje jenom jedna, pomocí swipů je možné přejít na další a zpátky. Modely se točí kolem sebe pomocí uzlu *AnimationPlayer*, který dovoluje vytvářet animace různého druhů. Přepínání mezi hrdiny je uděláno dalším způsobem. Je to 3D scéna, která má kameru nasměrovanou na první model. Pomocí swipe skript spouští správnou animaci v *AnimationPlayer*, která rotuje kameru směrem další hrdiny na řadě.

Když hráč vybere hrdinu, kterou nemá zakoupeno, tak místo tlačítka „zvolit“ bude mít tlačítko „koupit“. Kliknutím na tlačítko „koupit“ se provede operace zakoupení. Zkontroluje se, zda hrdina není v seznamu zakoupených hrdinů, pak se provede přidání id této hrdiny do seznamu. Potom se odečte její cena a změní se textové pole s počtem mince. Rovnou uloží do paměti počet mincí a seznam zakoupených hrdinů. Poslední věc je změnit tlačítko „koupit“ na „zvolit“. Pak bude moct hrdinu zvolit.

Další důležitá věc je *WorldEnvironment*. Je to uzel, který zahrnuje všechno, co je vidět na pozadí: slunce, obloha, zem apod. *WorldEnvironment* se používá ve všech 3D scénách a vždy je to ten stejný.

Po tom, co hrdina je vybraná, změní se id a uchovává se na disku jako aktuální vybraná hrdina. Existuje pak slovník, kde pro každou id existuje cesta do správného modelu a při spuštění hry se nahraje vybraný model.

Před začátkem zápasu se nahrává model aktuální hrdiny. Každá hrdina má vlastní animaci pohybu, proto každá scéna hrdiny má uvnitř *AnimationPlayer* s animací pohybu, které se spustí hned po přidání. Tohle ale není všechno, je nutné zpracovat ještě bonusy. Každý model má něco vlastního, proto je nutné ten bonus nějak zapojit. Všechny bonusy jsou sepsané v globální proměnné jako slovník, kde „key“ je id hrdiny a „value“ je název bonusu, který se přidá hned před začátkem běhu.



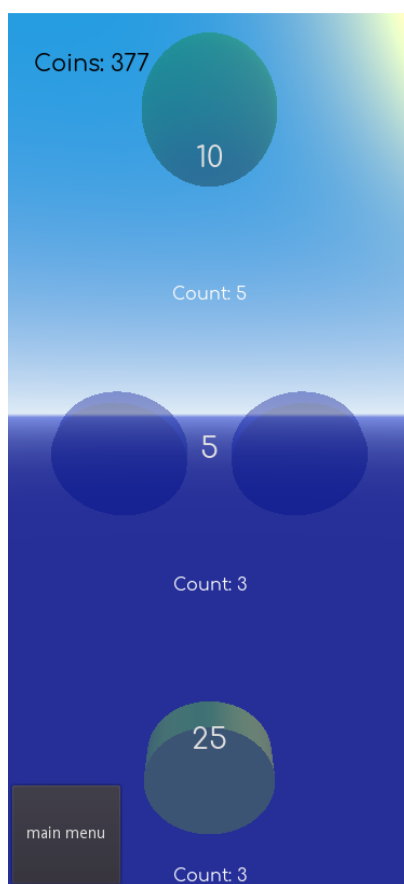
Obrázek 5.4: Scéna *Vyběr hrdiny*. Modely jsou použity pouze pro demo verzi a budou změněny během vývoje. 5.2

5.3 Obchod

Utrácet mincí pouze za hrdiny je nedostatečné. Bylo rozhodnuto přidat „Obchod“. Hráč má možnost ulehčit svůj další zápas a to tím, že zakoupí vylepšení do „své kapsy“. Existuje slovník, ve kterém se nachází všechny druhy vylepšení a jejich počet. Když hráč používá nějaké vylepšení během zápasu, tak se z tohoto slovníku odečítá jednička, až dokud nevyužije všechny.

Scéna „Obchod“ se používá pro zakoupení vylepšení pomocí mincí, které hráč nasbíral. Je možné tady koupit všechny vylepšení, které jsou ve hře. Scéna, jako i „Výběr hrdiny“ se skládá z 2D a 3D scén. 2D scéna zahrnuje textová polí pro aktuální stav peněženky hráče, počet vylepšení každého druhu a tlačítka, které jsou neviditelné a jsou umístěni ve stejném miste jako 3D modely. Kliknutím na nějaké vylepšení se provede hodně podobné chování jako zakoupení nové hrdiny. Provede kontrolu, jestli má dost peněz, odečte cenu a přenastaví textové pole s počtem mincí, zvětší počet vylepšení v slovníku o 1, uloží změny na disk. Typicky je tu i tlačítko „Hlavní menu“.

3D část umiřtuje všechny 3D modely vylepšení, které jsou trošku otočení a *WorldEnvironment* uzal. Po kliknutí na vylepšení se provede odečítání mince a přečtení 1 do aktuálního počtu vylepšení, pak **se uloží** nový počet do souboru na disku.



Obrázek 5.5: Scéna *Obchod*. Modely vylepšení jsou reprezentovány pomocí základních 3D objektů. Jsou pouze pro demo verzi. Budou změněni během vývoje.

5.4 Hlavní část

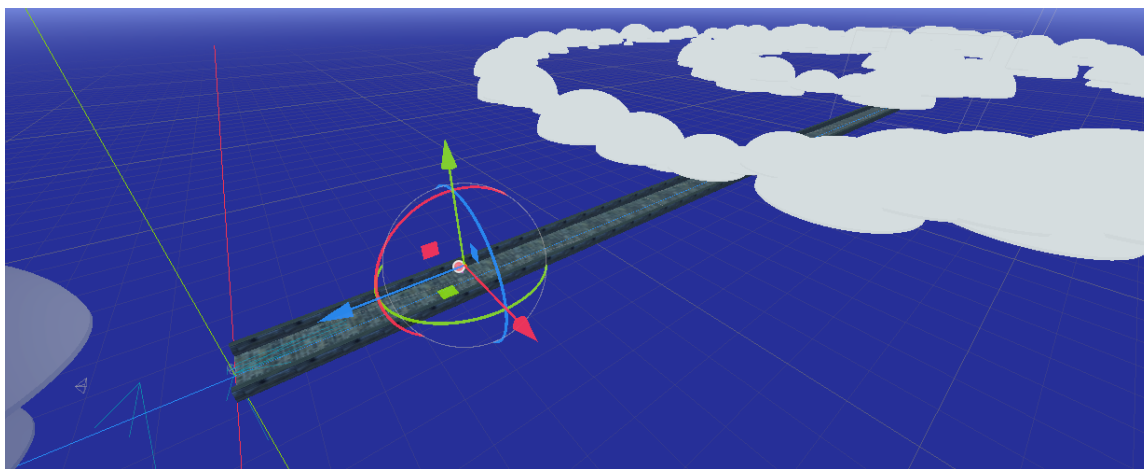
Samotná hra se začíná zde. Hlavní scéna je rodičovskou pro celou herní část, umísťuje v sobě všechny ostatní, na rozdíl od hlavní menu, které je pouze počáteční scéna pro hráče, ale pro přechod do všech ostatních 2D scén se používá metod přepínání scény. Hlavní scéna se skládá z dalších:

5.4.1 World

„World“ obsahuje mapu, objekty a logiku/skripty kterými se řídí. Jsou to další elementy: „maps“, „Items“ a „Cleaner“.

Samotná scéna „world“ má taky skript, který řídí přidávání nové mapy, promazávání předchozí mapy a zvětšení vzdálenosti mezi překážky. Nová mapa se přidává dalším způsobem. Zprv se nahraje z paměti a vytvoří se instance scény, rovnou se uloží do proměnné. Potom se nová mapa posune směrem souřadnice Z tak, aby byla rozmístěna hned po předchozí. Skript provede přidání mapy do rodičovské nody a spustí timer, který po uplynutí nastaveného času se dostane do předchozí mapy a pomocí `queue_free()` přidá předchozí mapu do fronty na promazání z paměti. Tímhle přidávání a mazání mapy končí. V tento okamžik v rodičovské nodě „maps“ znovu je pouze jedna mapa.

Zvětšení vzdálenosti mezi překážky probíhá dost jednoduchým způsobem. Jelikož generaci překážek se zabývá timer, který generuje je jednou za nějaký čas, tak navýšením tohoto časového limitu můžeme dosáhnout zvětšení vzdálenosti mezi překážky. Timer má parametr `wait_time` který se navyšuje. Pak další spuštění timeru bude trvat delší dobu.



Obrázek 5.6: Scéna „World“. Všechny modely, které je možné vidět jsou určeny pouze pro demo a to i shadery s jejich generačním kódem.²

„Maps“ je rodičovská noda pro mapu. Mapa je dopředu připravená cesta, která má jakési *collision objects*, aby hrdina nespadla skrz zem, dekorativní elementy, jako třeba mraky a neviditelný element, který signalizuje, že je nutno přidat další mapu. Po několika sekundách se starší mapa zmizí a zůstane se jen ta poslední.

„Items“ obsahuje všechny ostatní elementy, a to jsou:

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

- „coins“- rodičovská noda pro všechny mince, které budou vytvořeny. Scéna s mincemi je dopředu připravená a skládá se ze samotných mincí a *animation player*, který je točí kolem sebe. Animace se spouští hned po přidání mince do scény. Po přidání pomocí *RandomNumberGenerator* se náhodně vybere pruh, na kterém bude mince. Mince se generují jednou za sekundu pomocí timeru. Timer jednou za sekundu vysílá signál, který skript zachycuje, generuje minci a spouští timer znovu. Takhle dokola se generují mince.
- „barriers“- rodičovská noda pro všechny překážky. Logika je stejná jako u nody „coins“ s jednou výjimkou. Překážek existuje několik druhů, všechny jsou dopředu připravené a pomocí *RandomNumberGenerator* se vybere jenom jedna. V případě, že překážka není na celou šířku, tak se znovu pomocí *RandomNumberGenerator* vybere pruh.
- „pickables“- rodičovská noda pro vylepšení. Znovu stejný princip s jinou výjimkou. Než se začne generování a přidání vylepšení pomocí *RandomNumberGenerator* bude rozhodnuto, jestli se to vylepšení bude generovat vůbec. Ta implementace je v podstatě jednoduchá: spustí se generátor náhodných čísel od 0 do 9, kde 0 je *true* a všechno ostatní je *false*. Co znamená, že vylepšení se generuje s šancí 10 %.
- „map_elements“ - rodičovská noda pro elementy jako „platform“. Jsou to objekty, které mají kolizi. Na takové elementy je možné skočit, nebo do nich narazit.

„Cleaner“ je poslední a hodně důležitý objekt. Obsahuje tzv. *raycasts*. „Cleaner“ je určený pro uklízení. Maže z paměti všechno, čeho se dotknou paprsky. Paprsky zachytávají všechno kromě mapy. Je to realizováno pomocí „*Collision Mask*“. Každý element, který je možné takto ovlivnit, musí mít správně nastavené „*Collision Mask*“. Jakmile se nějaký objekt narazí na paprsek, tak se provede kontrola, jestli není ve frontě na promazání, když ne, tak bude přidán do fronty. Godot se pak o to promazání postará sám.

Je nutné zmínit ještě o samotném pohybu. Zaprvé se může zdát, že hrdina běží a všechno ostatní se nehýbe. Na začátku přesně tak všechno fungovalo. Působí to ale jeden problém, pro pohyb se musí zmenšovat souřadnice Z. Jak víme *integer* není nekonečný, a to je právě problém. Bylo rozhodnuto, že hrdina se posouvat nebude, místo toho se bude hýbat všechno ostatní, a to hned po přidání do svých rodičovských nod. Tímhle způsobem zajistíme nekonečnost hry, která bude probíhat kolem souřadnic (0,0,0).

5.4.2 Hlavní rozhraní

Základní nodá pro hlavní herní rozhraní je „Control“. „Control“ je v podstatě základní noda pro skoro jakýkoliv 2D menu element. Hlavní rozhraní zobrazuje počet mincí v levém horním rohu, aktuální rychlost hrdiny v pravém horním rohu a 3 tlačítka pro použití zakoupených vylepšení.

Kdyby hrdina běžela pořád se stejnou rychlostí, tak by to byla nuda. Proto bylo přidáno *ProgressBar* zrychlení. Zvýšení rychlosti běhu je standardní věc pro hry takového typu. Jelikož nejnáročnější je reagovat na překážky, proto čím rychleji hrdina běží, tím těžší a zajímavější to je. Na to je určený ten *ProgressBar* s jeho zabudovanou mechanikou.

ProgressBar má další nastavení, které nás zajímají:

- *Min Value* - začíná se progrese od nuly.
- *Max Value* - číslo, při kterém bude 100%.

- *Step* - krok o který budeme zvětšovat.

Navýšení probíhá pomocí timeru. Uvnitř *ProgressBar* má timer, který se spouští jednou za

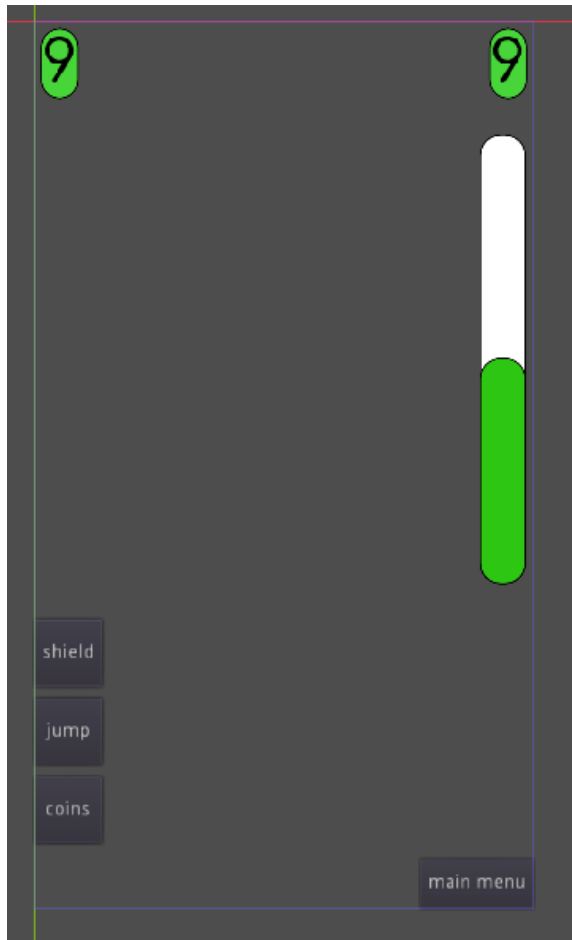
$$\text{float}(\text{GlobalVars.player.speed_up_timer_basic})/\text{progress_steps}$$

, kde *GlobalVars.player.speed_up_timer_basic* je proměna, která se bude zvětšovat v čase; *progress_steps* je *Max Value - Min Value*, což v našem případě je prostě *Max Value*.

Po uplynutí timeru se zavolá se zvětší aktuální hodnota o krok a provede se kontrola, jestli aktuální hodnota není ta maximální. Když jo, tak se provede zrychlení hrdiny pomocí metody *speed_up()*. Dále navýší se čas dalšího zrychlení, to znamená, že každé další zrychlení nastane čím dál tím víc později. Je to něco, čím by se dalo pochlubit před kamarády, že hráč dokázal dosáhnout nějaké velké rychlosti. Později by se dalo udělat na to online tabulku, například „Top 100 nejrychlejších“ nebo něco podobného.

Poslední věc, kterou timer dělá je nastavení nové rychlosti do textového polí rychlostí v pravém horním rohu.

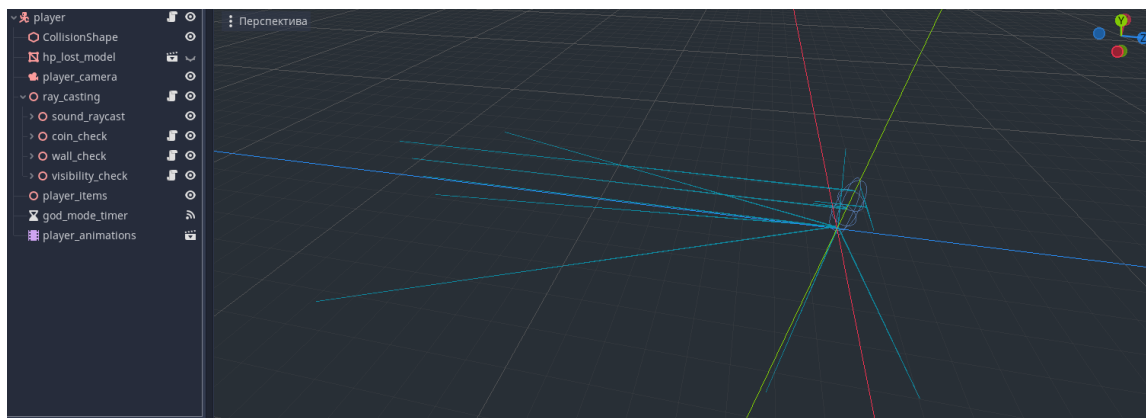
V této práci nejde o vizuálně hezkou aplikaci, proto většina tlačítek jsou standardní, ale chtělo by se trošku zmínit i o stylu. Změna stylu byla použita, například, pro „Progress bar“. Styl je možné změnit pomocí několika tříd, jedná z nich je „StyleBoxFlat“^[2], *StyleBoxFlat*. „StyleBoxFlat“ dává možnost změnit barvu, uhly, hranice. S jeho použitím byly nastaveny bílé pozadí a zelená progresse s černou barvou hranici. Taky bylo přidáno zaoblení pro pozadí a progresi. Podobný styl byl použitý i pro textová políčka v pravém a levém horních rozích.



Obrázek 5.7: Scéna „*main_interface*“. 5.4.2

5.4.3 Player

Implementace hrdiny je nejnáročnější a nejrozsáhlejší část práce. Rodičovská noda je typu *KinematicBody*. Sama o sobě žádnou kolizi nemá, proto uvnitř je noda *CollisionShape*, která to zajišťuje. Běh hrdiny je automatický, hráč není nutné něco pro to dělat. Ostatní věci, jako změna pruhu, skok, dřepnutí jsou řízení hráčem.



Obrázek 5.8: Scéna „Player“. Vlevo je možné vidět všechny nody a sub-scény, které jsou. Větší část obrazovky obsahuje samotná 3D scéna, na které je vidět hodně různých paprsků. V bodě (0,0,0) je vidět kapsle. Je to *CollisionShape* hráče. Právě tahle kapsle je zodpovědná za to, že hráč nepropadne skrz zem a bude mít kolizi. 5.4.3

Celou logiku řízení se dá rozdělit na dvě části:

`_input()`[2, Godot notifications] se používá pro zpracování vstupu. V případě mobilního zařízení, vstupem jsou „swipy“. Realizování jsou dalším způsobem: když se hráč dotkne obrazovky, tak se to projeví jako „event“ typu *InputEventScreenTouch* a zároveň ten „event“ je typu „dotknul se“. Zapamatujeme si souřadnice dotyku, až se hráč prst sejme z obrazovky tak se znovu načte „event“ typu *InputEventScreenTouch* ale tentokrát „event“ už není typu „dotknul se“. V tento okamžik znovu si zapamatujeme souřadnice a program zjistí, jakým směrem byl prst posunut a na základě toho rozhodne o jaký typ pohybu jde: skočit, přisednout, doleva či doprava.

`_physics_process()`[2, Godot notifications] řídí všechno ostatní. Nejdůležitější část je pohyb. Pro *KinematicBody* v Godot Engine existuje metoda *move_and_slide()* která zajišťuje pohyb. Pomocí proměny *direction* typu *Vector3(x,y,z)* je možné zadat směr a rychlost pohybu. Jsou pak další mechaniky:

- *gravity()*, tentokrát nejde o zabudovanou metodu. Metoda pro simulaci gravitace je nutná aby šlo pořádně nadefinovat skok. Jinak řečeno, hrdina se furt směruje dolů, a to tak, že s každým krokem `_physics_process()` se zrychluje svoje padání. Je ale jedná podmínka, to zrychlení by se mělo pozastavit, když hrdina je na zemi (není nutné zrychlovat padání, když hrdina nepadá). Na to je zabudovaná do třídy *KinematicBody* metoda *is_on_floor()*.
- *jump_left()* a *jump_right()* - nejjednodušší logika ze všeho uvedeného. Pruhy jsou reprezentovány pomocí čísel -1,0,1. Je nutné, aby hrdina se mohla posunout z levého pruhu hned na pravý. *jump_left()* a *jump_right()* v podstatě jenom zvyšují nebo redukuje to číslo, dále pohyb řídí metoda *check_line()*.
- *check_line()* je metoda, která mění směr a rychlost pro souřadnice *X*. To znamená, že řídí posuv hrdiny doleva a doprava. Funguje to tak, že se nastaví *direction.x* na nějaké číslo, číslo může být kladné nebo záporné, záleží jakým směrem má být hrdina posunuta. Čím větší je číslo, tím rychlejší bude posun. Existuje pak 3 pruhy: $x = -3$ (levý), $x = 0$ (centrální), $x = 3$ (pravý). Stále se provádí kontrola, když souřadnice *X*

hrdiny je v rozsahu $\langle -0.1; 0.1 \rangle$ tak je to centrální pruh, když $\langle -\text{inf}; -3.1 \rangle$ - levý, $\langle 3.1; +\text{inf} \rangle$ - pravý. Berou se rozsahy místo čísel kvůli tomu, že hrdina se může posunout za jeden krok metody `_physics_process()` o trošku dál jako třeba 0, kvůli čemu se nezastaví na centru.

- `check_collision()` - je hlavně určena pro kontrolu všeho, na co se může narazit hrdina. Zatím jsou to jenom dva objekty: „platforma“ a „překážka“. První znamená, že nezáleží na počtu štítu, když se narazí na tenhle objekt, tak je to konec hry. V případě platformy je to pouze její přední část. Na platformu je možné skočit nahoru, ale když se narazí zepředu, tak je to konec. Když se hráč narazí na překážku, tak teprve proběhne kontrola, jestli nemá štít. Když štít není, tak konec hry, jinak se zmenší počet štítů a spustí se metoda `hp_lost_process()`.
- `hp_lost_process()` - hrdinu nadhodí pomocí stejné mechaniky jako je skok (bude probráno dále). Nastaví se správný bit `collision_mask` na false, aby hráč se nemohl narazit na žádné jiné překážky, zapne se animace „hp_lost_anim“ a spustí se timer „god_mode_timer“. To znamená, že během další 1.1 sekundy hráč bude nezranitelný vůči překážkám. Animace „hp_lost_anim“ jednou za 0.1 sekundy mění viditelnost scény `hp_lost_model`, která je v podstatě napůl průhledný kapsle kolem hrdiny červené barvy.
- `jump()` - jak už bylo zmíněno, skok je možné implementovat hodně jednoduchým způsobem díky gravitaci. Při spuštění této metody se nastaví `direction.y` na nějaké číslo. Hrdina rychle „skočí“, s časem kvůli gravitaci `direction.y` se bude zmenšovat a hrdina se vrátí zpátky na zem.
- `crouch()` - je zajištěno pomocí `AnimationPlayer`. Hrdina už je na zemi, nejde ji posunout ještě níž, proto bylo rozhodnuto, že se bude měnit samotný model. `AnimationPlayer` zmenší výšku modelu a na obrazovce to vypadá jako dřepnutí. Zajímavé v této metodě je to, že když hrdina je ve vzduchu a bude použito dřepnutí, tak padání bude rychlejší než obvykle. Je to tak uděláno, aby hráč, když bude ve vzduchu a zjistí, že před ním je překážka, mohl se před ní uhnout.
- `speed_up()` - metoda je zodpovědná za zrychlení hrdiny. Rychlost se zadává pomocí parametru `movement_speed`, který se zvyšuje pokaždé, když se zavola tato metoda. Když se zvýší rychlost, tak se musí změnit i delka paprsků, jinak v nějaký okamžik hráč prostě nedokáže zareagovat na překážku. Zvětšení paprsků je naimplementováno dalším způsobem. Cyklem se proberou všechny paprsky, které jsou zodpovědné za zvuk a viditelnost (v případě režimu zaučení) a zvětší se jejich delka o stejné číslo jako je zrychlení hrdiny. Není to ale všechno, jiný problém je v tom, že překážky budou se příliš často objevovat. Je to taky propracované, po zrychlení hrdiny se zvětšuje i čas pro generaci překážky.

Další důležitá noda je `player_camera`. Je to kamera, která se nachází fixně za hráčem a je nasměrovaná na hrdinu. Kamera je uvnitř scény hráče, proto není nutné přidávat jí žádný řídicí skript. Kamera se bude hýbat spolu s hrdinou.

Noda `ray_casting` je rodičovská pro všechny použité paprsky. Paprsky jsou rozděleny na další 4 kategorie:

- paprsky zvuku - to jsou tři paprsky, které jsou umístěny nahoře, v centru a dole. Tyhle paprsky jsou určeny pro přehrávání zvuku, dle kterého hráč bude vědět, že před ním

je překážka. Každý paprsek je zodpovědný za různé překážky a má jiný zvuk. Zvuk se přehrává pomocí *AudioStreamPlayer*. *AudioStreamPlayer* má parametr *Pitch Scale* [2, *PitchScale*], který je zodpovědný za zrychlení zvuku. Rychlost se vypočítá uvedeným způsobem:

$$0.9 + \text{abs}(\text{cast_to.z} - \text{colide_object.global_transform.origin.z})/25$$

kde: konstanty jsou vybrány tak, aby zvuk bylo možné pořádně rozlišit; *cast_to.z* je délka paprsku; *colide_object.global_transform.origin.z* je *Z* souřadnice překážky, ale v našem případě je to i vzdálenost překážky od hráče, protože hráč je vždy v (0,0,0).

- paprsky objektů - to jsou taky tři paprsky, jen že rozmístění jsou jinak, než předchozí. Jeden je nasměrován dopředu, druhý doleva a poslední doprava. Účelem je zachytávání vylepšení a mince. Když se paprsek dotkne nějakého objektu, tak teprve zjistí, co to je. Dělá se to podle názvu nody, kterou se paprsek dotkne. Zatím je 4 druhy:
 - vylepšení skoku, když se narazí na tento objekt, pak hráč dostane skokové vylepšení, a to znamená, že do listu všech jeho vylepšen bude přidáno položku „*player_jumpers*“ a do nody pro 3D objekty se nahraje 3D scéna *player_jumpers*. Scéna *player_jumpers* má uvnitř sebe časovač, který se spustí hned po přidání a po ukončení smaže „*player_jumpers*“ z listu všech vylepšení a pak i samotnou 3D scénu.
 - zrychlený sběr mincí. Od předchozího se liší tím, že hned po přidání 3D scény, zvětší počet sbíraných mincí o dvakrát. Po ukončení timeru se pak o dvakrát zmenší.
 - mince se od ostatních liší tím, že nemá 3D model a nepřidává se do listu vylepšení, v podstatě to ani není vylepšení. Jakmile se paprsek dotkne *coin_static_body*, tak se do počtu mince přidá 1 a změní se počet mince v textovém polí grafického rozhraní.
 - štít je nejnáročnější ze všech. Je to způsobeno tím, že štítu najednou může být více jak jeden. Než se přidá další štít, tak se provede kontrola jeli „*player_shield*“ v seznamu všech vylepšení hrdiny. Když není, tak se tam přidá, navýší počet životů o 1, přidá 3D model štítu a nastaví textové pole počtu životů na aktuální počet štítů. Když „*player_shield*“ v listu všech vylepšení hrdiny už je, tak se provede navýšení počtu životů a přepsání textového polí počtů životů.
- kontrola ukončující zdí je rodičovská pro jenom jeden paprsek. Je to paprsek vepředu hrdiny, který je určený pro přidání další mapy. Každá mapa má na nějaké vzdálenosti od konce neviditelnou stěnu, když tenhle paprsek potká tu stěnu, tak ji smaže a spustí metodu přidání další mapy, která přidá další kus cesty hned po existujícím.
- kontrola viditelností má dvě skupiny paprsku *front* a *back*. Paprsky jsou určeny pro to, aby změnili viditelnost překážek. Dle názvu skupiny, je možné pochopit, že v případě zapnutého režimu zaučení, používají se paprsky, které jsou nasměrovány dopředu, jinak ty, co jsou nasměrovány dozadu.

Noda *players_items* je rodičovská noda, do které se ukládají všechny 3D scény vylepšení, které bude mít hráč.

Kapitola 6

Zhodnocení

6.1 Navíc

Jak už bylo zmíněno, účelem bakalářské práce je demonstrace mechanik hry, ale chtěl bych to rozvíjet i dále. Bude nutné přidat ještě hodně všeho, hlavně:

- Modely. Nejdůležitější část práce, kterou je nutné udělat jsou modely. Zatím hra vypadá jako „šedý čtverec vedle šedého cylindru“. Je nutné namodelovat vnějšek hrdiny, spíš několika hrdin (kvůli různým bonusem), 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 plusy a mínusy.
- Zhoršení. Další věc je udělat hru o něco těžší pro zkušenější a to pomocí zhoršení. Budou to věci, které, třeba, dokážou změnit swipy, když hráč bude chtít změnit pruh, tak místo toho, aby se posunul doprava, přejde na levý pruh. Podobné věci by mohli zbytečně zkomplikovat hru začátečníkům, proto se budou objevovat po delší době běhu.
- Více obsahu. Zatím jsou pouze 3 hrdiny a jedna platforma, ale šlo by přidat mnohem více. Další nápad pro hrdinové bonusy jsou: zvětšení délky paprsku, který zapíná zvuk; štít ze začátku (jinak hrdina, co má ze začátku 2 životy).
- Tabulka „Top100“. Jako ve většině her, lidé rádi soutěží. Šlo by udělat tabulku, ve které budou hráči, co nasbírali nejvíce mincí nebo běželi nejvíce času/nejdelší cestu.

Na jinou stranu, kromě základních mechanik, bylo uděláno dost všeho navíc:

- Bylo přidáno navíc 2 hrdiny a mechaniků bonusů.
- Možnost zakoupení vylepšení za mince, které hráč sbírá během zápasu a možnost tyhle bonusy využít během hry.
- Bylo vytvořené menu a přidáno uložení na disk.

6.2 Feedback

Pro testování a zpětnou vazbu, hru bylo předáno lidem, kteří nejsou zapojeni do samotného vývoje. Nápady testerů:

- Když překážka není přímo před hráčem, ale na sousedním pruhu, tak to není možné pochopit. Návrh na vylepšení je buď udělat zvuk 3D, kde každá překážka bude mít vlastní audio přehrávač a zvuk bude jít přesně od ní, nebo když překážka je na sousedním pruhu, tak bude mít trochu jiný zvuk, než obvykle. Problém prvního nápadu je, že hra je pro mobily, když hráč nebude mít sluchátka ale pouze reproduktor v mobilu, tak nedokáže rozlišit z jaké strany dochází zvuk. Problém druhé je, že v takovém případě počet různých zvuku bude cca 9 místo původních třech.
- Byl nápad přidat nějaké bonusové úrovně, které nebudou dynamicky generované (budou fixní, mapa bude vytvořena dopředu), ale během celého zápasu hráč bude slyšet hudbu a v nějaké určité okamžiky té písničky, třeba nějaký hlasitý bas bude znamenat, že je nutné skočit apod.
- Náročné pro začátečníky, když hráč zatím není aspoň trochu zkušený, tak většinou nedokáže běžet ani 15 sekund. Je to celkem špatné, z ohledem na aktuální tendenci her. Většinou, když hráče nezaujalo hned, tak hru zahodí. Možnost na vylepšení pak jsou: zvětšit délku pro paprsek, aby hráč věděl, že před ním je překážka mnohem více dopředu; zvětšit dobu pro skok a přisednutí; zrychlit animaci dřepnutí a zrychlit skok.
- Změnit zvuky, udělat více odlišné. Nejsou příjemné na poslech. Udělat zvuk nejenom nějaké audio, ale něco jako „toptoptop“, „botbotbot“, kde hráč zároveň bude slyšet směr, kam se má uhnout.
- Pomalá hra. Počáteční rychlost v pořádku, ale zrychlení je moc malé. Hra je mnohem zajímavější, když je více překážek, citace: „Když je rychlost dost velká, tak srdce se bije hodně často a pořád jsem ve stavu, jakoby čekám na ten další zvuk, na tu další překážku a je to pěkný pocit“.
- S ničím podobným se zatím nasetkal, mechanika je opravdu nová, ale je náročnější v realizaci. Cílová skupina hráčů je zvyklá hrát takové 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.
- Je to neobvyklé, kvůli tomu, že je ta mechanika nová, je to i nevýhodou. Lidí jsou zvykli vidět překážky a dopředu plánovat svůj běh, v tomto případě je to prostě nemožné. Začátečníky hodně tápou, když slyší zvuk. Je třeba zahrát nějakou dobu, než uživatel zvykne na tu samotnou mechaniku.
- Jedno z vylepšení, které je možné přidat, může být zrychlení. Může to být buď velké zrychlení, ale na krátkou dobu, nebo menší zrychlení, ale napořád. Šlo by přidat i nějakou kombinaci, třeba střední zrychlení na delší dobu, ale nebude navždy.
- Když hráč ztratí všechny životy, tak na obrazovce „Game over“ nevidilo by přidat informaci o aktuálním běhu, jako třeba počet nasbíraných mince, čas běhu, maximální rychlost apod. Taky na té stejné obrazovce nevidilo by tlačítko „Restart“, aby hráč neměl zpátky do hlavního menu, když neplánuje nic kupovat, ale chce hned zahrát další běh.

- Smrt hrdiny není nijak oznamovaná, hned po ztrátě posledního života změní scénu na „Game over“. Nevadilo by přidat animaci toho jak hrdina padá na zem, nebo něco podobného. Taky ještě nějakou melodii.
- Na scéně, kde hráč vybírá hrdinu, přidat popis všech hrdina cenu. Není jasné, která hrdina je která a čím je unikátní.
- Je možné udělat režim „pro slepce“. V tomto režimu každý element mapy bude mít vlastní zvuk, zvuk bude jistě udávat typ elementu a jeho rozmístění, třeba „platforma right, platforma right ..“. Je možné přidat zvuk každému elementu, a to tak, že ten zvuk bude přímo od samotného objektu.

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 kamarády. Bylo překvapením, že lidí z takové nové mechaniky mohou být tak silně zmatení. Na jinou stranu, je hodně zajímavá pro ti, co mají rádi výzvy. Pro ostatní bude nutné snížit vstupní práh. Taky byly i nápady, které mě nenapadly, jako třeba ten režim pro slepce.

„Zjistí testování hry problémy brzy, když je ještě čas je opravit? Ano. Buduje testování hry důvěru týmu, že vytváří správnou hru pro správné publikum? Ano. Je testování hry nezbytný pro vytvoření dobré hry? Ano. Naplňuje mě testování hry hrůzou tak intenzivní, že nedokážu ani myslet přímo? Ano ano ano!“[6, str.435]

Režim pro slepce

Režim pro slepce by se dalo rozšířit a to tak, že nejen samotný zápas bude pro slepce, ale celá hra. V dnešní době slepí lidi taky používají smartphony. Na to je určena funkce TalkBack nebo VoiceOver. Hlavní účel této funkce je číst všechno, co je napsáno na obrazovce nebo provádět operace pomocí hlasu.

TalkBack. Mluvíme o speciální aplikaci, kterou vyvinuli specialisté Google. Toto je program navrženy tak, aby ozvučil všechny akce uživatele, které provádí na svém gadgetu Android. Může to být chytrý telefon nebo tablet.

TalkBack je navíc standardní program. Původně byl začleněn vývojáři softwaru do každého zařízení s operačním systémem Android. Proto nebudete muset chodit na Play Market a hledat ho mezi miliony dalších aplikací. Pokud jste si zakoupili nový telefon nebo tablet, ujistěte se, že je tam již aplikace TalkBack předinstalovaná.

Tato aplikace také podporuje funkci automatické aktualizace. Aktualizace přicházejí, jakmile se objeví.

Je to skvělý nástroj pro lidi s problémy se zrakem. Využití pro něj ale našli lidé se zdravými očima. Často je využívají například řidiči. To vám umožní provádět určité manipulace s telefonem nebo tabletem, ale zároveň se příliš nerozptylovat od silnice.

Ale přesto je hlavním účelem pomoci lidem se slabým zrakem. Díky zvukové stopě budou přesně vědět, jaký text píše, do jaké nabídky vstupují a na co klikají. Toto je kompletní hlasový projev veškeré práce na gadgetu.[8]

VoiceOver. VoiceOver umožňuje zrakově postiženým uživatelům ovládat počítač pomocí široké škály klávesových příkazů a gest. Tato kapitola obsahuje přehled funkce VoiceOver a také obecné informace o kurzoru VoiceOver a kurzoru klávesnice, klávesových zkratkách a používání funkčních kláves na některých klávesnicích.

Když se používá VoiceOver k pohybu po obrazovce nebo textu v dokumentu, oblast, kde se nachází kurzor VoiceOver, je ohraničena tmavým obdélníkovým rámečkem. Tento snímek se nazývá kurzor VoiceOver.

Ve výchozím nastavení je kurzor VoiceOver stejný jako kurzor na klávesnici, takže když kurzor na klávesnici se pohybuje pomocí standardních navigačních kláves (klávesy Tab a šipky), kurzor VoiceOveru se pohybuje s ním.

Když se otevře nové okno, kurzor VoiceOveru a kurzor klávesnice jsou na stejném objektu. Když například uživatel otevře okno Nová zpráva v Mailu, kurzor VoiceOveru a kurzor klávesnice jsou v poli Komu, je možné to první pole změnit.[10]

VoiceOver – čtečka obsahu obrazovky založená na gestech – umožní iPhone používat, i když uživatel nevidí jeho displej. VoiceOver popisuje nahlas obsah obrazovky – od stavu baterie přes jméno volajícího až po název aplikace, na které uživatel drží prst. Rychlost předčítání a výšku hlasu je možné změnit tak, aby vyhovovaly potřebám.

Když se uživatel dotkne displeje nebo po něm posouvá prst, VoiceOver vyslovuje názvy položek, kterých se dotýká, včetně ikon a textu. K ovládání položek, jako jsou tlačítka a odkazy, a k přecházení na jiné položky je možné používat gesta VoiceOveru.

Pokud se otevře nová obrazovka, VoiceOver přehraje zvuk a potom vybere a přečte název první položky na obrazovce (obvykle v levém horním rohu). Informuje uživatele, když displej přejde do orientace na šířku nebo na výšku nebo když se ztlumí či uzamkne obrazovka. Při probuzení iPhone popíše aktivní obsah uzamčené obrazovky.[9]

Proto jako další možnost pro pokračování, vidím možnost udělat hru plně použitelný pro slepce. Třeba při prvním spuštění se hra zeptá, jestli chce hráč používat pouze zvuk nebo i vizuální část. Když hráč vybere jenom zvuk, tak bude pro něj speciálně vytvořené zaučení, ve kterém bude vysvětleno všechno, co se týká grafického rozhraní apod. Možná bude možnost nějak detekovat, jestli uživatel má zapnuto VoiceOver (a podobné funkce) na svém zařízení.

Je očividné, že většinou vývojáři začátečníci v tomto oboru nemůžou si dovolit takové rozšíření. Zabere to hodně času a není to výhodné finančně, ale samotný nápad je dobrý a, za mě, dá se to rozšíření udělat.

Kapitola 7

Zavěr

Původním cílem mé práce byla demonstrace nové mechaniky v herním žánru „Runner“. Záměrem bylo vytvořit hru typu „Runner“, kde hlavním smyslem pro řízení bude sluch.

Prvním úkolem bylo probrat samotný engine. Dříve jsem už pracoval s Godot Engine, proto to nebylo tak náročné. Problém byl ten, že já jsem pracoval s 2D jeho částí a jenom trošku se dotkl té 3D. Mnoho hodin jsem strávil tím, že se snažil pochopit proč a jak fungují nějaké zabudované metody. Díky oficiální dokumentaci Godotu, kterou jsem citoval během této práce, to probíráání bylo vůbec možné.

Bylo jasné, že budu muset kombinovat 2D a 3D. Vždy je nějaké rozhraní, které je 2D. Bylo nutné probrat, jak se to spojuje a je to nejjednodušší část celé práce, protože Godot dokáže rozlišit to všechno a stačí prostě přidat 2D rozhraní do 3D scény a rovnou bude v popředí. Dale jsem se pustil do samotné implementace hry.

Začínal jsem ze základních mechanik jako pohyb a kolize. Prvním elementem byla podlaha, bylo nutné udělat plochu, přes kterou hráč nebude padat. Další bylo vytvoření hrdiny a pohyb. Když hrdina už dokázala běžet, bylo nutné vymyslet způsob, jak udělat její běh nekonečný. Trvalo to delší doby, ale hrdina dostala nekonečnou cestu pod nohama. Zatím hrdina běžela jenom dopředu, bylo nutné přidat více pruhů. Bylo to jedno z nejtěžšího i když se to tak nezdá. Hrdina neustále chtěla přeskočit jeden pruh nebo zasekává se na nějakém čísle a furt měnila směr pohybu, jinými slovy, bylo to „zajímavé“.

Jakmile základní věci už byly přepravení, začal jsem vytvářet překážky. Bylo nutné důkladně probrat jak fungují matice kolize. Na radě pak byly materiály a „shaders“, chtěl jsem udělat překážky napůl průhledné. Rovnou jsem pracoval i na zvucích, nejnáročnější bylo najít zvuk a správně ho uříznout, aby to znělo více méně dobře.

V tento okamžik hlavní úkol už byl splněný, pak jsem přestoupil k přidání vylepšení, obchodu a nových hrdin. Bylo to náročné a nové, ale hrozně zajímavé. Chtěl bych i dále pokračovat směrem vývoje aplikace. V dnešní době popularita mobilních telefonů a tabletů se zvětšuje, proto dává smysl se soustředit na nich.

Takže úkol byl splněn, ba i přesáhl původní zadání. Byla vytvořena plně funkční hra se všemi základní mechaniky. I když hra je zaměřena na mobilní zařízení (IOS, Android), demo je možné spustit taky na Windows a Linux.

Jak už bylo uvedeno v předchozí kapitole, mám ještě nápady, co přidat a jak hru vylepšit. Během tohoto studijního roku jsem pochopil, že vytvoření aplikace je hodně časově náročné, proto plánuji zapojit k tomuto projektu ještě své kamarády a spolu udělat z té hry plnohodnotný produkt.

Literatura

- [1] GREGORY, J. *Game Engine Architecture*. 3. vyd. A K Peters/CRC Press, August 17, 2018. ISBN 978-1138035454.
- [2] JUAN LINIETSKY, A. M. a GODOT COMMUNITY the. *Godot Documentation* [online]. docs.godotengine, 2014-2022 [cit. 2022-04-06]. Dostupné z: <https://docs.godotengine.org/en/stable/>.
- [3] KUBRAK, N. *How virtual cameras works* [online]. dtf.ru, 2020 [cit. 2022-04-17]. Dostupné z: <https://dtf.ru/gamedev/128939-kak-ustroeny-virtualnye-kamery>.
- [4] LINIETSKY, J. *Godot history in images!* [online]. godotengine.org, leden 2014 [cit. 2022-04-05]. Dostupné z: <https://godotengine.org/article/godot-history-images>.
- [5] *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/>.
- [6] SCHELL, J. *The Art of Game Design*. 2. vyd. A K Peters/CRC Press, November 8, 2014. ISBN 978-1466598645.
- [7] *Teorie kvalitního zvuku ve videohrách* [online]. UE4 Daily, 2019 [cit. 2022-04-24]. Dostupné z: <https://ue4daily.com/blog/sound-in-games>.
- [8] *Na co je funkce TalkBack a jak se vypíná* [online]. nastroyvse.ru, 2022 [cit. 2022-04-29]. Dostupné z: <https://nastroyvse.ru/opersys/android/cto-takoe-talkback-kak-eyo-otklyuchit.html>.
- [9] *Zapnutí a procvičování VoiceOveru na iPhonu* [online]. support.apple.com, 2022 [cit. 2022-05-04]. Dostupné z: <https://support.apple.com/cs-cz/guide/iphone/iph3e2e415f/ios>.
- [10] *Představujeme VoiceOver* [online]. apple.com, 2022 [cit. 2022-05-04]. Dostupné z: https://www.apple.com/ru/voiceover/info/guide/_1121.html.
- [11] *Obecné informace o zvuku* [online]. paroc.cz, 2022 [cit. 2022-05-4]. Dostupné z: <https://www.paroc.cz/knowhow/zvuk/obecne-informace-o-zvuku>.