

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačové grafiky a interakce

Hudební hra ve virtuální realitě

Isa Skořepů

Vedoucí: Ing. David Sedláček, Ph.D.

Obor: Počítačové hry a grafika

Studijní program: Otevřená informatika

Leden 2020

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Skořepů** Jméno: **Isa** Osobní číslo: **466134**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačové grafiky a interakce**
Studijní program: **Otevřená informatika**
Studijní obor: **Počítačové hry a grafika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Hudební hra ve virtuální realitě

Název bakalářské práce anglicky:

Music game in virtual reality

Pokyny pro vypracování:

Proveďte rešerši hudebních her jak pro virtuální realitu, tak i obecně pro jiná elektronická zařízení (PC, tablet). Analyzujte používané herní mechaniky a navrhnete hudební hru pro virtuální realitu, kde hráč bude reagovat pohybem horní poloviny těla a paží na vizuální podněty synchronizované s přehrávanou/generovanou hudbou. Navrhnete a implementujete systém pro synchronizaci herních událostí s hudbou a vytvoříte grafické rozhraní pro modifikaci parametrů synchronizace (editor). Implementujte herní události napojitelné na hudbu a vytvoříte šest ukázkových úrovní hry se stupňující se složitostí. Postupujte dle metodiky UCD (User Center Design) pro návrh a testování uživatelského rozhraní s ohledem na dostupnost cílové skupiny.

Seznam doporučené literatury:

- [1] Moderní počítačová grafika: Bedřich Beneš, Jiří Sochor, Petr Felkel, Jiří Žára, 2005, ISBN: 978-80-251-0454-5
- [2] The VR Book: Human-Centered Design for Virtual Reality: Jason Jerald, 2016, ISBN: 978-1-97000-112-9
- [3] T. Lowdermilk, User-Centered Design, O'Reilly Media, 2013

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. David Sedláček, Ph.D., katedra počítačové grafiky a interakce FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **13.02.2019**

Termín odevzdání bakalářské práce: **24.05.2019**

Platnost zadání bakalářské práce: **20.09.2020**

Ing. David Sedláček, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování

Děkuji všem, kteří mě podporovali při tvorbě této práce a bez nichž by tato práce nikdy nevznikla. Zejména svému vedoucímu Ing. Davidu Sedláčkovi, Ph.D..

Prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně, a že jsem uvedla veškerou použitou literaturu.

V Praze, 7. ledna 2020

Abstrakt

V této práci popisují vývoj hry *PaintPong* pro virtuální realitu, jejíž hlavní mechanika je založená na odražení barevných kuliček do rytmu a obarvení co největší části okolí. Kuličky se chovají různými způsoby podle své barvy.

Tato práce obsahuje návrh této hry, implementaci, testování a následné úpravy na základě uživatelského testování. Dále obsahuje popis editoru úrovní a jeho integraci se hrou. V práci jsou popsány podobné hry ze kterých jsem čerpala inspiraci.

Klíčová slova: Virtuální realita, Počítačová hra, Unity, Hudba

Vedoucí: Ing. David Sedláček, Ph.D.

Abstract

This thesis describes development of game *PaintPong* for Virtual Reality. Its main game mechanic is deflecting colored balls in precise time to music and using them to color as much of the surroundings as possible. Balls have different behaviours depending on their color.

This thesis contains design, implementation, testing and changes based on testing with users. Further it contains description of level editor and its integration with the game. Thesis also contains descriptions of similar games whose I used for inspiration.

Keywords: Virtual reality, Computer game, Unity, Music

Title translation: Music game in virtual reality

Obsah

1 Úvod	1	Seznam nálezů	35
1.1 Cíle práce	1	Řešení nálezů	36
2 Rešerše existujících hudebních her	3	7 Závěr	39
2.1 Audioshield	3	A Literatura	41
2.2 Beat saber	4		
2.3 Beats fever	5		
2.4 Piano tiles 2	6		
3 Design hry	7		
3.1 Cílové publikum	7		
3.2 Herní systém	7		
3.2.1 Typy kuliček	8		
3.2.2 Skóre	8		
3.3 Rozložení úrovně	8		
3.4 Módy interakce	10		
3.5 Návrh hlavních obrazovek	10		
3.6 Zvuk	11		
3.6.1 Zdroje zvuku	12		
3.6.2 Zdroj hudby	12		
3.6.3 Poznámka k syntéze hudby ..	13		
4 Implementace	15		
4.1 Výběr technologií	15		
4.2 Hra	17		
4.2.1 Hlavní nabídka	17		
4.2.2 Hlavní herní mechanika	18		
4.2.3 Konec hry	19		
4.3 Načítání úrovní	20		
4.4 Scripty pro Unity editor	21		
5 Editor úrovní	23		
5.1 Rešerše existujících editorů	23		
5.1.1 Beatmapper	23		
5.1.2 Mediocre Mapper	24		
5.1.3 Zabudovaný editor	25		
5.2 Návrh řešení editoru	25		
5.3 Postup vytváření úrovně	26		
5.4 Hlavní rozhraní	26		
5.5 Editační nástroje	27		
5.6 3D náhled	28		
5.7 Implementace časové osy	29		
5.8 Implementace 3D náhledu	29		
5.9 Možná vylepšení editoru	30		
6 Testování	31		
6.1 Kognitivní průchod	31		
6.2 Testování s uživateli	32		
Výsledky testování s uživateli	34		

Kapitola 1

Úvod

Virtuální realita je v poslední době čím dál tím lepší a dostupnější. Nabízí výrazně lepší možnost ponořit se do dění. Co je další věcí, do které se lidé jsou schopni ponořit a vžít se do ní? Hudba. Tato práce je tedy spojením hudby a virtuální reality v podobě hudební hry.

V průběhu této práce vytvořím hru ve virtuální realitě[1] jejíž hlavní mechanika bude založena na synchronizaci akcí hráče k hudebnímu podkladu. K dosažení tohoto cíle provedu průzkum existujících her v této kategorii a také některých her mimo ní pro inspiraci.

Dále je potřeba vybrat technologie pomocí kterých tuto hru budu vytvářet a následně ji v těchto technologiích také implementovat. Protože zábavnost tohoto typu hry závisí také na množství a kvalitě úrovní je dalším, avšak sekundárním cílem vytvořit způsob tvorby úrovní.

Po vytvoření základu hry ji vyzkouším s uživateli a na základě jejich zpětné vazby provedu iteraci na designu a implementaci hry.

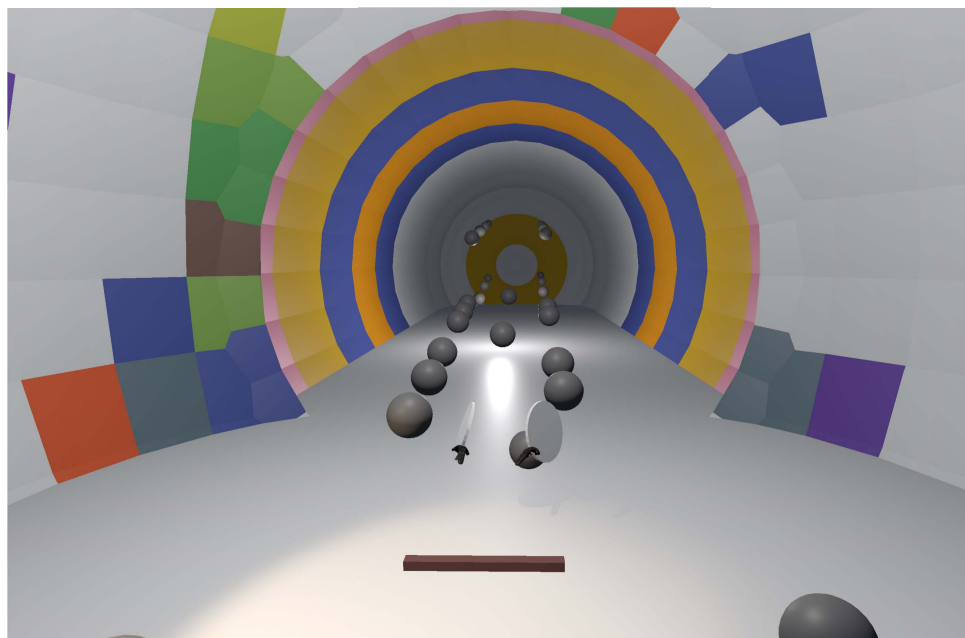
- V kapitole 2 projdu některé existující hry a jakým způsobem dané téma zpracovaly.
- V kapitole 3 navrhnu mou hru PaintPong
- V kapitole 4 tuto hru vytvořím
- V kapitole 5 pro ni vytvořím editor
- V kapitole 6 ji otestuji a na základě testů ji vylepším
- V kapitole 7 shrnu své výsledky a navrhnu možnosti dalšího pokračování

1.1 Cíle práce

Hlavním cílem této práce je vytvořit kompletní hru pro virtuální realitu se vším co k tomu náleží. Seznam cílů mé práce tedy je:

- Nalézt hry které jsou základní myšlenkou podobné mé hře
- Navrhnout vlastní hru, která bude založená na navázání herního prožitku na hudbu

- Vzniklý návrh hry implementovat
- A následně tuto hru otestovat
- A pokud čas dovolí upravit na základě výsledku testů



Obrázek 1.1: Výsledná hra

Kapitola 2

Rešerše existujících hudebních her

Před tím, než mohu hru začít vymýšlet a vytvářet je dobré se podívat na už existující tituly v podobné kategorii. Kategorie moje hra bude splňovat dvě. První jsou hry pro virtuální realitu (VR). Druhou kategorií jsou hudební hry. Následuje neúplný seznam her převážně pro virtuální realitu seřazený podle abecedy.

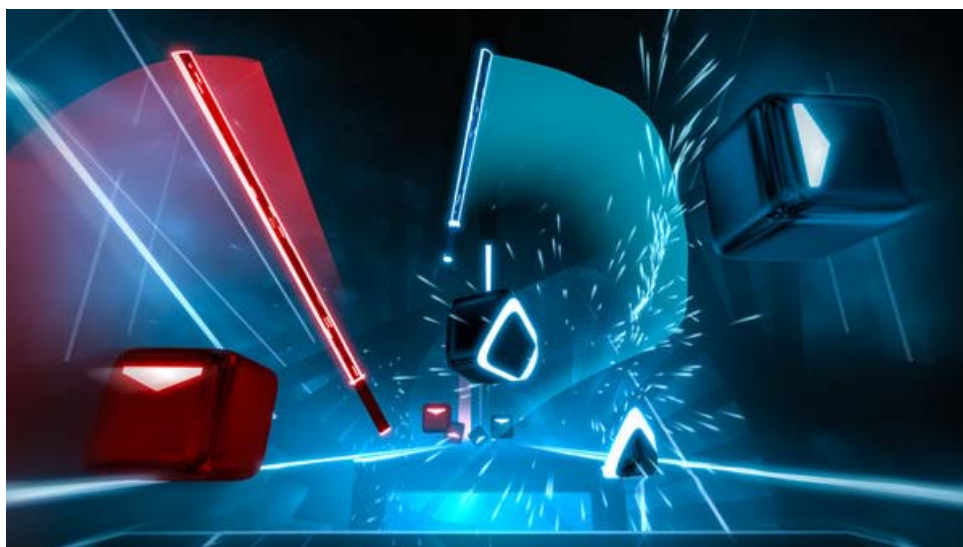
- Airtone
- Audioshield
- Beat Saber
- Beats Fever
- BOXVR
- Hatsune Miku VR
- Holodance
- Piano tiles (android)
- Rock Band VR (rift)
- Soundboxing

V této práci však budu zkoumat jen některé a to konkrétně ty, které používají vybranou herní mechaniku (synchronizace akcí hráče k hudbě) a jsou mi dostupné.

2.1 Audioshield

Hlavní mechanikou v této hře je, že hráč dostane štíty pomocí kterých musí trefit modré a červené kuličky - červené pravou rukou a modré levou rukou (barva štítů tento fakt odráží). Dále se v úrovních vyskytují fialové kuličky, které hráč musí odrazit oběma štíty naráz.

Tato hra, která vyšla spolu s HTC Vive používá automaticky generované časy kdy má hráč odrazit danou kuličku takže velkou výhodou této hry je, že



Obrázek 2.2: Herní obrazovka hry Beat saber[3]

sažených v řadě.

Dalším prvkem je, že hráč si může upravit některé vlastnosti hry a tím si ji usnadnit nebo zesložitit. Když si hráč hru usnadní tak na konci dostane nižší skóre a naopak.

2.3 Beats fever



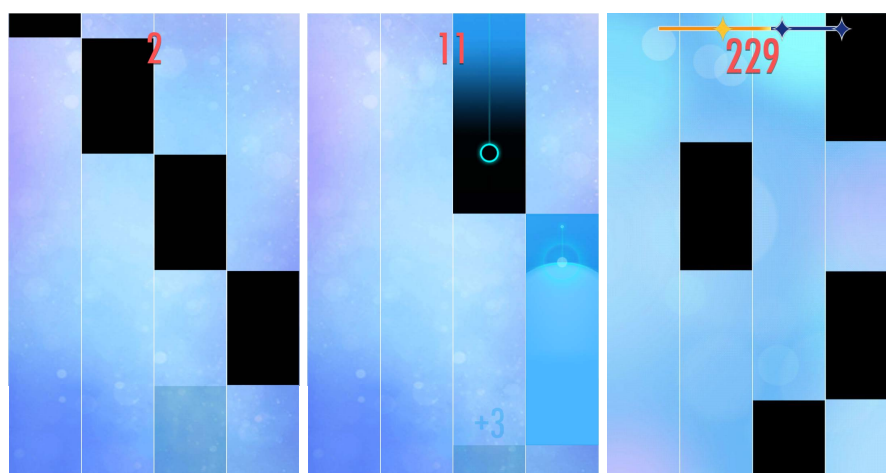
Obrázek 2.3: Herní obrazovka hry Beats fever[4]

Další hrou, kterou jsem zkoumala z hlediska herních mechanik je beats fever. Hra je opět založena na synchronizaci hráčových pohybů vůči písni, ale v tomto případě toho dosahuje pomocí pálek, které se musí střetnout s

letící notou ve správný čas. Ve výsledku to znamená, že hráč umístí pátku na mřížku, kterou před sebou vidí a hra je v důsledku vlastně 2D a nejde ani tak o to se do hudby vcítit, ale následovat terčíky.

Osobně mi přijde chybný design ovladačů v této hře, protože hráči mají tendenci s nimi do cílů mlátit (jako na buben), což je sice zábava, ale snižují si tím šanci na dosažení vyššího skóre, které se odvíjí od celkového počtu zasažených kuliček a od množství kuliček zasažených v řadě. Toto by nebyl problém, kdyby například rychlost pohybu ovladače byla započtena do výsledného skóre.

2.4 Piano tiles 2



Obrázek 2.4: Herní obrazovka hry Piano tiles 2[5]

Hra pro mobilní telefony založená na herní mechanice, kde se na obrazovce objevují černá políčka, na která má hráč kliknout a zároveň nesmí kliknout mimo. Při kliknutí mimo se úroveň ukončí. Tato hra jde hrát do nekonečna - po dohrání písně se píseň přehrává znovu a znovu s tím, že je pokaždé mírně rychlejší.

Zajímavý rozdíl v této hře je to, že hra nesleduje přednahranou zvukovou stopu, ale hráč ji svými akcemi generuje. Tato zvuková stopa je ve hře definována jako sekvence not či akordů a jejich délek, takže hráč nemůže upravit melodii, ale může změnit rytmus.

Skóre je založené na počtu stisknutých dlaždic.

Kapitola 3

Design hry

Vzhledem k nálezům z rešerše (viz. kapitola 2) můžu začít navrhovat svou hru. Pro lepší představu nejdřív zadefinuji cílové publikum a na základě toho vytvořím hlavní herní mechaniku. Následně okolo hlavní herní mechaniky postavím nezbytné prvky, které hráči umožní hru hrát.

3.1 Cílové publikum

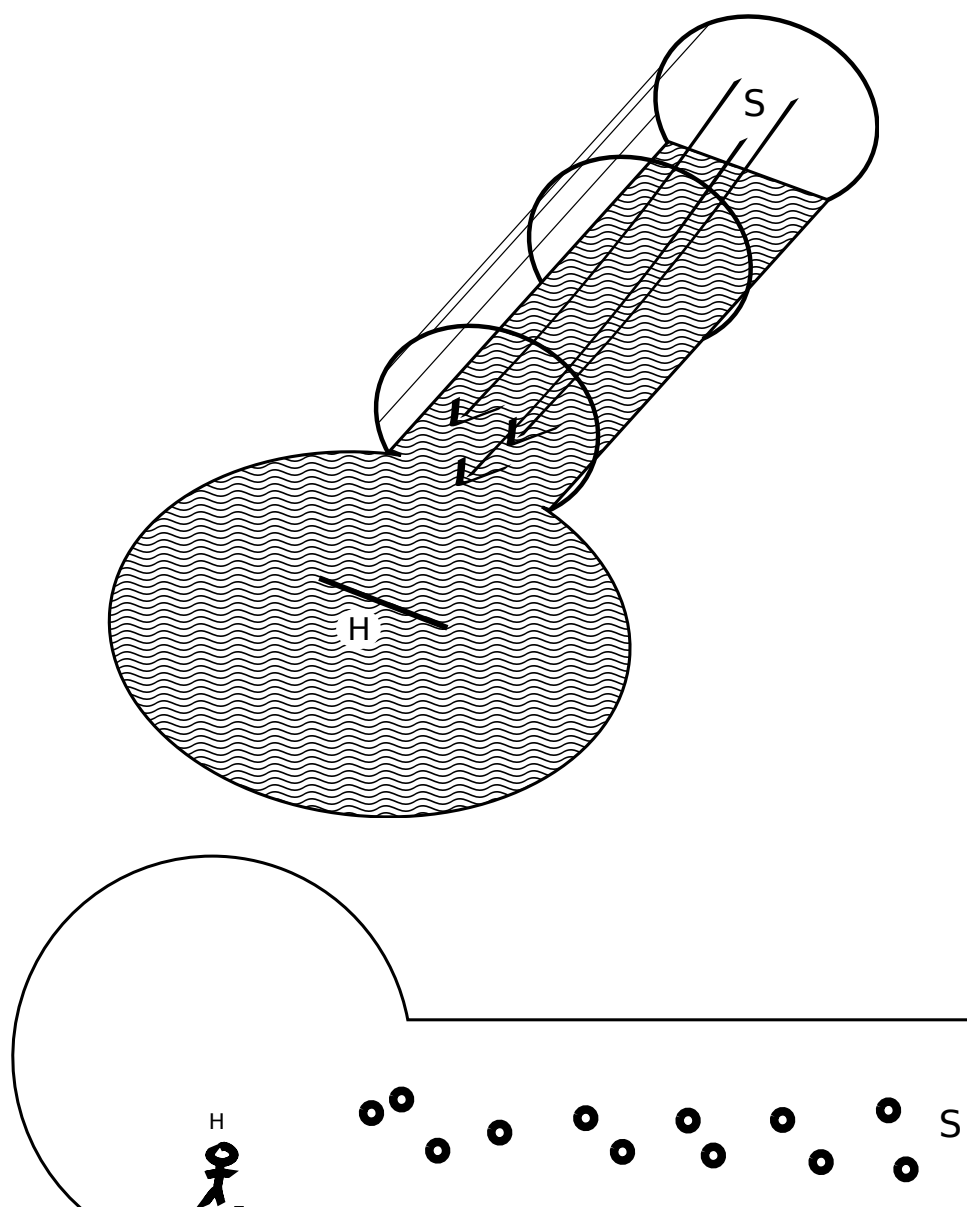
Mohlo by se zdát, že cílové publikum pro tuto hru jsou zkušení vlastníci VR headsetů, ale není tomu nutně tak. Jakožto hudební hra tato hra bude mít relativně krátkou minimální dobu hry a z toho vyplývá, že je vhodná jako takzvaná *party hra*, kde se u hraní hráči střídají. To mění cílové publikum na lidi, kteří s virtuální realitou mají menší zkušenosti a tím pádem nezvládnou složitější schéma ovládání - ze zkušenosti vím, že například u HTC Vive je občas složité vysvětlit pozici grip tlačítek a ještě složitější vysvětlit možné funkcionality touchpadu. Také to znamená, že by model ovladačů ve hře neměl být příliš abstraktní, aby bylo snadné se u hry střídat. Dále to pro design hry znamená, že by neměla příliš záviset na náhodě, aby se hráči mohli smysluplně poměřovat.

3.2 Herní systém

Po spuštění úrovně hráč dostane do rukou dvě páčky pomocí nichž bude mít za úkol odrážet nálet barevných kuliček. Kuličky může odrážet libovolnou páčkou. Kuličky budou přilétat do hudby a jejich trajektorie bude po přímce tak, aby je hráč měl v zorném poli pokud stojí rovně a kouká se dopředu.

Generování přilétajících kuliček bude na hudbu synchronizované ručně. To je z důvodu komplikovanosti dobré automatizované analýzy hudby - synchronizace na hudbu je klíčová k dosažení dobrého zážitku.

Rychlost kuliček bude poměrně velká. Důvod k tomuto je, že potřebuji hráče donutit aby byl s hudbou synchronní a vyšší rychlost kuliček sníží časové okno ve kterém hráč může kuličku zasáhnout. Toto mi přijde jako lepší způsob jak dosáhnout tohoto cíle než mřížka, kterou používá Beats Fever nebo Holodance.



Obrázek 3.1: Rozložení úrovně

Hráč stojí na rovné ploše uvnitř kulové plochy. Umístění hráče je na obrázku označeno H. Ke kulové ploše je připojen tubus ze kterého přilétají kuličky. Místo odkud přilétají kuličky je označeno S. Před hráčem je čára, která označuje místo, kde má hráč kuličky trefovat.

Obarvovat se dá vše okolo hráče přičemž obarvování funguje po jednotlivých plochách - když kulička zasáhne danou část plochy, plocha se obarví celá. Jediné části hráčova okolí, které se nedají obarvit jsou podlaha a startovní čára.

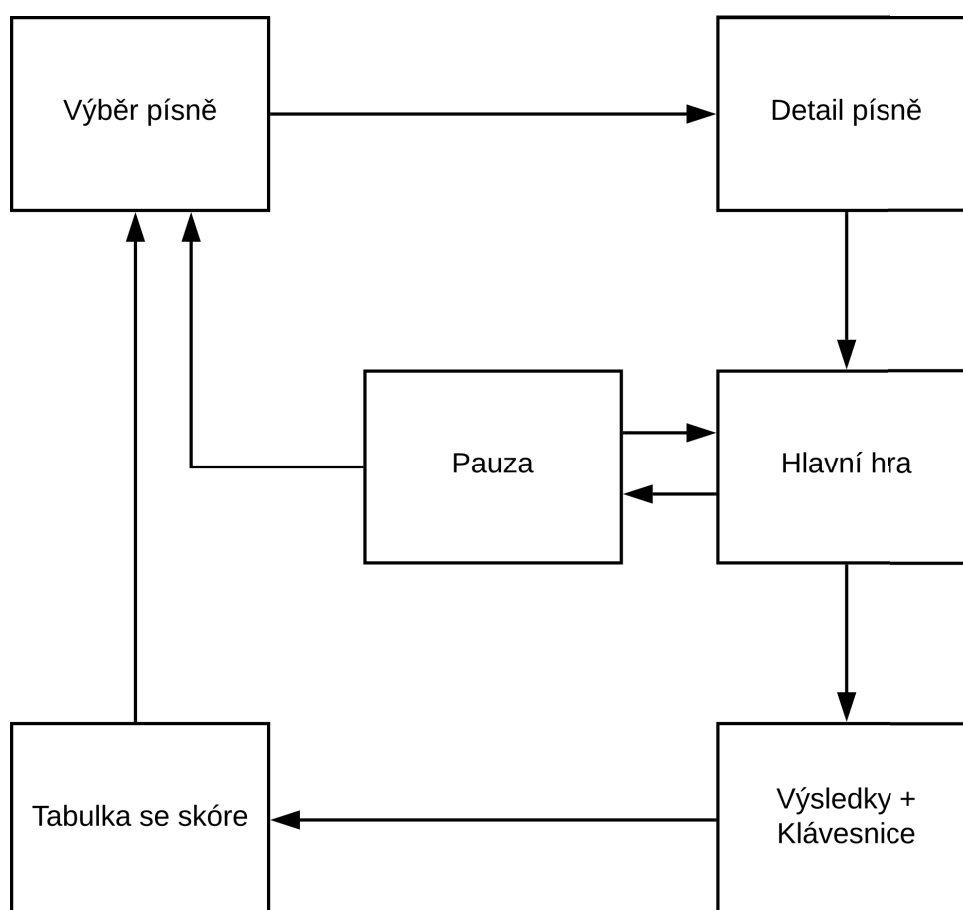
3.4 Módy interakce

Hlavní herní smyčka používá k výběru interakční vzor přímé manipulace pomocí ruky (direct hand manipulation pattern) [6] sekce 28.2. Toto je dobré pro hlavní herní mechaniku, která má být co nejintuitivnější, má realisticky odpovídat pohybům uživatele a má být snadná na vysvětlení (viz cílové publikum - sekce 3.1).

Tento režim interakce ale není vhodný pro interakci s menu, která by měla být co nejrychlejší a kde jsou uživatelé zvyklí používat myš - tedy na nepřímou interakci. Proto jsem pro tyto využití implementovala výběr pomocí míření (pointing pattern) [6] sekce 28.1.

3.5 Návrh hlavních obrazovek

Hra se kromě hlavní herní bude skládat ještě z několika interaktivních obrazovek.



Obrázek 3.2: Průchod obrazovkami

Hlavní obrazovky často obsahují různá tlačítka, která se vybírají pomocí

míření. Aby hráč poznal, že je vybral a věděl, že s nimi může interagovat tak se při výběru dané tlačítka zvětší. Výsledný vzhled jednotlivých obrazovek uvedu v kapitole 4

■ Výběr písně

Obrázek 4.2. První obrazovka, se kterou se hráč setká. Zde se zobrazí seznam úrovní spolu s jejich délkou. Nad seznamem úrovní je popisek "vyberte píseň".

■ Detail písně

Obrázek 4.3. Zde se zobrazí další detaily o písni. Při otevření této obrazovky se začne přehrávat část písně, která odpovídá dané úrovni. Je zde tlačítko pro spuštění hry s popisem start. Zároveň je to také největší prvek na této obrazovce.

Napravo od detailu písně se zobrazí tabulka s nejlepším skóre. Při zobrazení této obrazovky zůstává obrazovka výběru písně viditelná - pouze se odsune vlevo, aby bylo možné snadno změnit volbu úrovně.

■ Hlavní hra

Obrázek 4.4. Zde se bude odehrávat hlavní herní mechanika. Je možné ji v průběhu přerušit.

■ Pauza

Když hráč na ovladači zmáčkne tlačítko menu tak se hra pozastaví a přes ní se mu zobrazí tato obrazovka. Obsahuje akce: pokračovat, restartovat a zpět do hlavního menu.

■ Klávesnice

Obrázek 4.5. Pokud se hráč umístil do tabulky vítězů tak se mu zobrazí virtuální klávesnice ve které bude moci zadat své jméno. Také zde bude na výběr historie posledních zadaných jmen, aby hráč své jméno nemusel pořád vypisovat znova.

■ Tabulka se skóre

Poté co hráč zadal své jméno tak se mu zobrazí jeho umístění v tabulce.

■ 3.6 Zvuk

Pro kompletní zážitek ze hry je často vhodné vizuální zážitek doplnit zvukem. Pro hudební hru to platí dvojnásob.

■ 3.6.1 Zdroje zvuku

To, že hra je primárně založená na hudbě a tedy na zvuku může vyvolat pocit, že pro implementaci bude potřeba složitý audio systém. Není tomu tak. V této sekci projdu jednotlivé zvuky, které se budou ve hře vyskytovat a jejich vlastnosti.

■ Hudební stopa

Hlavní zvuková složka této hry. Přehrává se ze zvukového souboru a není diegetická (její zdroj není nijak ukotven ve virtuálním světě). Toto je nejjednodušší složka na zpracování - jednoduše se přímo pustí do sluchátek nebo reproduktorů uživatele. Je zde otázka co dělat se stereo zvukem (někteří uživatelé nemusí mít funkční jednu stranu - například při předvádění si uživatel chce nasadit jenom jedno sluchátko a tím přijde o jeden kanál), ale většina zvukových stop nemá silný rozdíl mezi kanály, takže nejjednodušší řešení a to přehrát hudbu tak jak je by mělo být dostačující.

■ Zvuk nárazu kuliček

Pro zvuk nárazu kuliček na pátku jsem se rozhodla pro hudebně neutrální zvuk metronomu, který je výrazný a tudíž se v hudbě neztratí a také podobný zvuku nárazu pingpongového míčku, což odpovídá vzhledu hry.

■ Zvuk menu tlačítek

Interakce s tlačítky nabídkách by také měla vydávat zvuk. Zde by zvuk měl vycházet z pozice tlačítek, se kterými je interagováno.

■ Hudba v hlavním menu

V momentě kdy hráč vybere píseň, což je první interakce v hlavním menu tak se vybraná píseň začne přehrávat a tedy není potřeba příliš řešit hudbu na pozadí v menu.

■ 3.6.2 Zdroj hudby

Možnosti pro zdroj odkud se bude pouštět hudba jsou: lokální soubory, úrovně distribuované s aplikací, youtube video, spotify, nebo vlastní servery pro distribuci.

Úrovně distribuované s aplikací - toto je pro hráče nejpohodlnější možnost, která však nabízí nejmenší flexibilitu. Tato možnost obsahuje pouze předdefinovanou hudbu, která se hráči nemusí líbit. Také je zde problém s licencováním hudby, přičemž většinu hudby nemohu spolu s hrou legálně distribuovat.

Lokální soubory jsou jednoduchá alternativa, která je pohodlná pro lidi, kteří mají sbírku stažených hudby. Problém zde nastává v tom, že je složité přiřadit správnou úroveň dané písničce (stejně jako je občas složité najít správný soubor s titulky pro daný film). Jednoduchý hash souboru v tomto

případě nemusí nutně pomoci, protože píseň může být převedena do jiného formátu, než pro který byla úroveň vytvořena. Toto by nebyl problém, pokud by hra obsahovala systém automatického generování úrovní.

Youtube videa jsou v tomto aspektu lepší - dané video je snadno identifikovatelné pomocí URL. Zde by však nejspíš nastal stejný problém, jak již bylo zmíněno výše, jako u audioshieldu. Tato možnost chce víc výzkumu, protože by mohlo jít přehrát youtube reklamy a tímto tomuto problému předejít.

Spotify zde uvádím pouze pro kompletnost co se populárních zdrojů hudby týče. Problém zde je s tím, že spotify ve svých podmínkách užití explicitně zakazuje použití ve hrách.

Vlastní servery pro distribuci mají stejné problémy jako assety distribuované spolu s aplikací. Ale byly by vhodné pro distribuci úrovní spolu s nějakým identifikátorem (youtube url / hash souboru / otisk písně [7]).

Jelikož toto není zásadní pro mnou vytvořené úrovně tak jsem se rozhodla pro soubory integrované do hry respektive načítané ze složky se hrou, protože je to nejjednodušší možnost na implementaci.

Dále si ale uživatelé budou moci pomocí editoru vytvářet vlastní úrovně, které se budou sestávat z jimi poskytnutého hudebního souboru a textového souboru. Pro vytváření úrovní implementuji editor.

■ 3.6.3 Poznámka k syntéze hudby

Všude výše předpokládám, že budu používat herní systém na synchronizaci herních událostí k existující přednahrané hudbě. Zde stojí za zmínku i druhá alternativa, ve které by hra znala tóny, které se mají v daný čas přehrát (například z MIDI souboru[8]) a na základě toho by vytvořila herní události při jejichž provedení by se teprve tón přehrál. Toto má ale nevýhodu, že to vyžaduje téměř či úplně dokonalé provedení, protože jakmile začne hráč chybovat tak se ztratí hlavní motiv melodie (jako při hře na hudební nástroj). Toto je způsob, který je použit v piano tiles.

Další problém je omezené množství skladeb a hudebních stylů, které se tímto způsobem dají vytvořit. Toto však by mohla být částečně i výhoda, protože tento způsob se dá použít k syntéze hudby, jež je volným dílem jako je klasická hudba a lidové písně. To by umožnilo tuto hudbu distribuovat spolu se hrou bez problémů popisovaných v kapitole zdroj hudby.

Finální problém je v samotné syntéze zvuků, což je komplikovaná záležitost, která by nejspíš zabrala netriviální množství času, který by tím pádem nemohl být stráven vylepšováním zbytku hry. Zde by bylo možné se inspirovat jinými dostupnými projekty, ale kód, který by šel integrovat přímo do unity (bez zakoupení assetu z asset storu) jsem nenašla, takže by šlo pouze o inspiraci a kód bych musela napsat sama.

Ve výsledku jsem se tedy rozhodla touto cestou nejít.

Tato možnost by šla použít k hybridnímu přístupu pro některé úrovně - v externím syntetizátoru vytvořím zvukovou stopu a pomocí kódu ze zdrojového souboru vygeneruji události. Tím se z toho však v podstatě stává přístup předchozí akorát s tím rozdílem, že jsou herní události vygenerované automaticky.

Kapitola 4

Implementace

Když mám dobrou představu o tom, jakou hru chci vytvářet je na čase ji vytvořit. Tato kapitola popisuje zvolené technické řešení a důvody které k němu vedli.

4.1 Výběr technologií

První krok při vytváření hry je vybrat, jaké technologie použiji. Výběr by měl být takový, aby mi co nejnázce umožnil dosáhnout cílů, které jsem si stanovila v designu hru.

Herní engine

Hlavním stavebním kamenem každé hry je herní engine, který poskytuje hlavní stavební bloky ze kterých mohu svou hru sestavit. Zásadní požadavek je, aby vybraný engine podporoval virtuální realitu, což však dnes všechny hlavní engine již podporují.

Jelikož chci engine, který bude pro vývoj dostupný zdarma tak mám dvě hlavní možnosti - Unreal Engine[9] a Unity[10]. Pro tuto práci jsem si vybrala Unity. Hlavní důvod je, že v něm umím. Dalším je, že C# je pro mě příjemnější jazyk než C++.

Systém interakce

Při vytváření VR hry v Unity existuje hned několik možností ze kterých si můžu vybrat pro vytváření interaktivních prvků. Tento výběr ovlivní konkrétní implementaci, její náročnost, portabilitu a v některých případech co lze a nelze dosáhnout.

Mým hlavním požadavkem na systém interakce je, že musí fungovat na HTC Vive a preferovaně by mělo být co nejjednodušší ho upravit tak, aby podporoval i jiné náhlavní displaye. Hlavní volbou pro mě je zda-li použít či nepoužít SteamVR plugin[11] a jestli používat jeho primitiva pro interakci jako je Hand a Interactable nebo ne.

Základní možnost je použít čistě Unity - toto je možnost nejjednodušší, ale chybí zde několik zásadních funkcí. První z nich je absence modelů VR



Obrázek 4.1: Různé ovladače pro virtuální realitu. Zleva doprava: Oculus touch controller, Vive controller, Windows mixed reality controller a Valve knuckles controller

ovladačů - zde je možnost ovladače zcela nahradit vlastními modely, což je způsob, který úspěšně aplikuje Beat Saber, ale do složitějších modelů (jako jsou štíty z audioshieldu a také moje páčky) je lepší zakomponovat reálný model ovladače, aby bylo pro hráče snazší je uchopit při vstupu do virtuální reality. Další funkce která zde chybí, ale není tolik kritická, protože lze doplnit v kódu je absence mapování na různé ovladače (oculus touch, vive wand, windows mixed reality controllers, knuckles), které se výrazně liší v množství a rozložení různých tlačítek (obrázek 4.1). Toto ale není tak velký problém pro tuto hru protože nejspíš pro většinu funkcí budou potřeba pouze dvě tlačítka (jedno pro hlavní interakce a jedno pro pozastavení hry), která jsou na všech výše zmíněných ovladačích.

Alternativní možností je SteamVR plugin, který všechny výše zmíněné funkce podporuje. Jeho hlavní nevýhodou je nutnost mít nainstalováno SteamVR pro chod aplikace.

Výsledné rozhodnutí v tomto bodě je tedy použít SteamVR, ale omezit jeho použití na jednu třídu XRHand, která by měla být co nejjednodušší a implementovat veškerou interakci se SteamVR pluginem. Dále bude SteamVR plugin použit pouze v jednom prefabu a to konkrétně Camera Rig, do kterého budou přidány mnou vytvořené a na SteamVR nezávislé skripty. Cílem je aby to celé fungovalo i po případném odstranění SteamVR. A pro podporu více platform by mělo stačit generalizovat vytváření XRHand a při startu načíst jiný Camera Rig prefab.

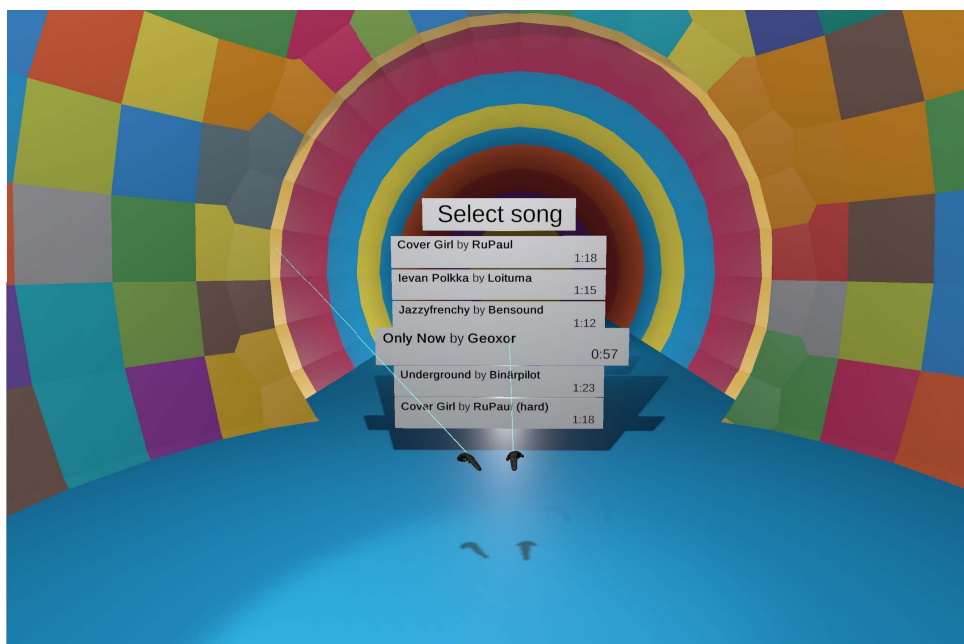
Toto rozhodnutí se v průběhu ukázalo jako výhoda i při aktualizování SteamVR pluginu protože jsem díky tomu měla v kódu jenom jedno místo, které jsem musela upravit.

4.2 Hra

Po zvolení základních technologií je čas pustit se do tvorby hry samotné. Ta se sestává ze dvou hlavních celků: hlavní nabídka a hra samotná. Každý jsem vytvořila jako separátní scénu. Pro předání informací mezi těmito scénami používám objekt, který vytvořím před načtením nové scény a nastavím, aby se při přechodu nezničil.

4.2.1 Hlavní nabídka

Při otevření hry se jako první scéna zobrazí hlavní herní nabídka, kde uživatel dostane na výběr ze seznamu úrovní.



Obrázek 4.2: Obrazovka výběru písně

Interakce v hlavní nabídce probíhá pomocí míření. Pro orientaci uživatele je ke každému ovladači připojeno ukazovátko. Ukazovátko je válec, jehož spodní podstava je připojena k ovladači a horní podstava je umístěna v bodu interaktivního objektu na který je mířeno.

Pro interaktivní objekty je implementován script, který ukazovátko detekuje. Ukazovátko na tomto scriptu volá metody `HoverStart` a `HoverEnd`, které indikují, že dané ukazovátko ukazuje na tento prvek. Při stisknutí tlačítka `trigger` se vyvolávají akce `OnTriggerDown` a `OnTriggerUp`, které umožňují jiným scriptům na tyto akce reagovat.



Obrázek 4.3: Obrazovka detailu písně

Všechna interaktivní tlačítka se při najetí kurzorem zvětší. Toto uživateli indikuje, že tlačítko je aktivní. Při stisknutí se spustí požadovaná akce, takže další specifická zpětná vazba není potřeba.

Jako drobný bonus jsem umožnila uživateli obarvování úrovně pomocí ukazovátka. Toto zároveň slouží jako drobný ukazatel toho, o čem hra je.

4.2.2 Hlavní herní mechanika

Nejdůležitější částí celé hry je hlavní herní smyčka. To je část hry, kde se spustí hudba, hráč dostane do rukou pátku a má odrážet nálet kuliček.

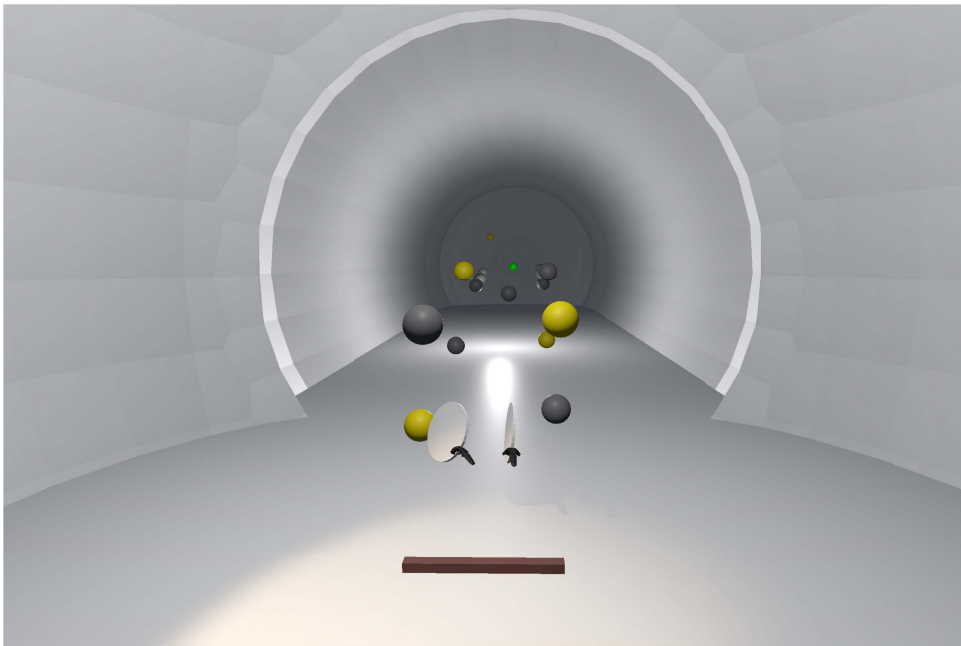
Přilétající kulička musí splňovat dvě důležité vlastnosti. Zaprvé musí k hráči přiletět přesně v momentě kdy ji má odrazit. Zadruhé musí všechny přilétat stejně rychle, aby hráč věděl kdy je má očekávat.

Obě tyto vlastnosti jsem zajistila tím, že jsem pozici kuliček napojila přímo na čas přehrávané hudby. Jinými slovy z času hudby spočítám pozici kuličky. Pozici počítám jako lineární interpolaci počátečního bodu a koncového bodu na základě časů ve kterých se má kulička v těchto bodech vyskytovat.

Při kontaktu kuličky s pátkou se kulička odrazí. Kontakt kuličky s pátkou mi poskytne unity včetně bodu kontaktu a jeho normálového vektoru. S pomocí této informace spočítám rychlost a směr pohybu kuličky pokud by se pátky nepohybovala. Tato rychlost se rovná rychlosti přilétající kuličky a směr je prostým odrazem.

Druhou a významější složkou, která přispívá k výslednému pohybu odražené kuličky je směr a rychlost pohybu pátky. Tato rychlost se k první složce přičte.

Obarvování se také děje při kontaktu, ale v tomto případě je se jedná o kontakt se zdí. Unity opět poskytne informace o kolidujících objektech. V

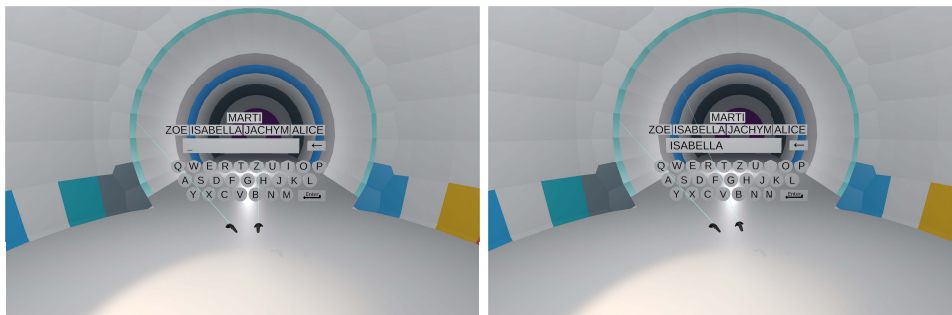


Obrázek 4.4: Obrazovka hlavní herní mechaniky

tomto momentě se vybere barva s předdefinovaného seznamu barev a ta se nastaví jako primární barva dotčené části zdi.

Další částí je, že různé kuličky na kontakt se zdí reagují různě. Proto se při kontaktu vyvolá metoda, která na různých typech kuličky má různé implementace a tato metoda provede požadované akce.

■ 4.2.3 Konec hry



Obrázek 4.5: Obrazovka klávesnice

Hra končí pět sekund po poslední kuličce. Tento interval byl zvolen protože umožní posledním odraženým kuličkám doletět ke stěně. Po tomto intervalu se hráči zobrazí virtuální klávesnice, která mu umožní zadat své jméno.

Klávesnice je implementována pomocí dvojice komponent Keyboard a KeyCap. Komponenta Keyboard očekává jako potomky objekty s napojenou komponentou KeyCap. Komponenta KeyCap se stará o přeposílání kliknutí

do komponenty Keyboard a umožňuje nastavit o jakou klávesu se jedná: písmeno, backspace, enter. Abych nemusela každé písmeno vyplňovat dvakrát tak KeyCap také zjistí svojí hodnotu z TextMeshPro potomka. Dále KeyCap fyzicky posouvá klávesu v reakci na kliknutí.

Komponenta Keyboard se stará o stav vyplněného textu a přeposílá konečný stav vyplnění komponentě, která zobrazuje nejvyšší skóre. Dále zajišťuje, aby hráč nevyplnil příliš dlouhé jméno. Poslední funkce této komponenty je čtení vstupu z fyzické klávesnice připojené k počítači. Toto usnadňuje hraní ve větší skupině, kdy může člověk sedící u počítače jméno zadat výrazně rychleji, nebo ho zadat potom co si hráč sundá náhlavní display příliš brzo.

Kromě zobrazení klávesnice se po konci úrovně také postupně utlumí hlasitost hudby, což je implementováno jako lineární snižování hlasitosti po dobu pěti sekund. Toto se samozřejmě stane pouze pokud ještě zbývá část hudební stopy.

4.3 Načítání úrovní

Jedním z cílů je mít uživatelsky dostupný editor úrovní což vyžaduje umožnit načítání ze souboru. Soubor s daty je ve formátu JSON[12] a hra očekává soubor ve formátu ogg se stejným jménem. Datový soubor obsahuje:

Jméno pole	Typ hodnoty	Popis
songName	Textový řetězec	Obsahuje jméno písně, zobrazuje se v detailu úrovně
songAuthor	Textový řetězec	Obsahuje jméno autora písně, zobrazuje se v detailu úrovně
levelAuthor	Textový řetězec	Jméno autora úrovně, zobrazuje se v detailu úrovně
levelName	Textový řetězec	Jméno úrovně, zobrazuje se v detailu úrovně a v seznamu úrovní
beats	Seznam úderů (přesná definice úderu níže)	Obsahuje seznam všech přilétajících kuliček a jejich vlastností.

Každá herní událost (odpovídající jedné přilétající kuličce) dále obsahuje

Jméno pole	Typ hodnoty	Popis
id	Kladné celé číslo	ID úderu musí být v dané úrovni unikátní. Slouží pro živé úpravy z editoru.
time	Desetinné číslo	Počet sekund od začátku písně kdy má hráč trefit přilétající kuličku.
color	Textový řetězec	Jedna hodnota ze seznamu: gray, yellow, green, black. Pokud obsahuje něco jiného je kulička ignorována. Ovlivňuje barvu kuličky.
position	Objekt obsahující klíče x a y, jejichž hodnoty jsou desetinná čísla	Vyjadřuje pozici přilétající kuličky. x=0 znamená ve stejné pozici jako je hráč. y=0 znamená podlahu. Hodnoty jsou v metrech.

Čas generování události se spočítá jako čas očekávané reakce mínus doba letu, která je fixní pro celou úroveň.

Tento systém v zadaný čas vytvoří unity gameobject z prefabu a předá mu zadané parametry plus instanci reprezentující stav herního světa. Dále na základě typu kuličky vybere třídu, vytvoří její instanci a následně jí nastaví jako řešitele konkrétních událostí. Formát úrovně se tedy skládá ze zvukového souboru a JSON souboru. Cílem je mít možnost přidávat úrovně bez zásahu do kódu hry a tím umožnit editor úrovní.

4.4 Scripty pro Unity editor

Implementovala jsem dva scripty, které usnadňují vývoj hry a nijak se nepodílejí na herním zážitku.

Prvním scriptem je Note PropertyAttribute. Tento script se dá připojit k libovolné veřejné vlastnosti dané komponenty. V tom momentě se tato vlastnost skryje z editoru a místo ní se zobrazí informativní text. Tento script používám pro popis komponent, který není čistě programátorský, ale popisuje konkrétní požadavky na hierarchii scény, kterou daný script požaduje.

Druhý script mám připojen k jednomu jinak prázdnému objektu ve scéně a umožňuje mi přímo z unity editoru přepnout mezi jednotlivými úrovněmi bez toho, abych se musela dotýkat ovladače. Toto výrazně urychlilo tvorbu některých funkcí.

Kapitola 5

Editor úrovní

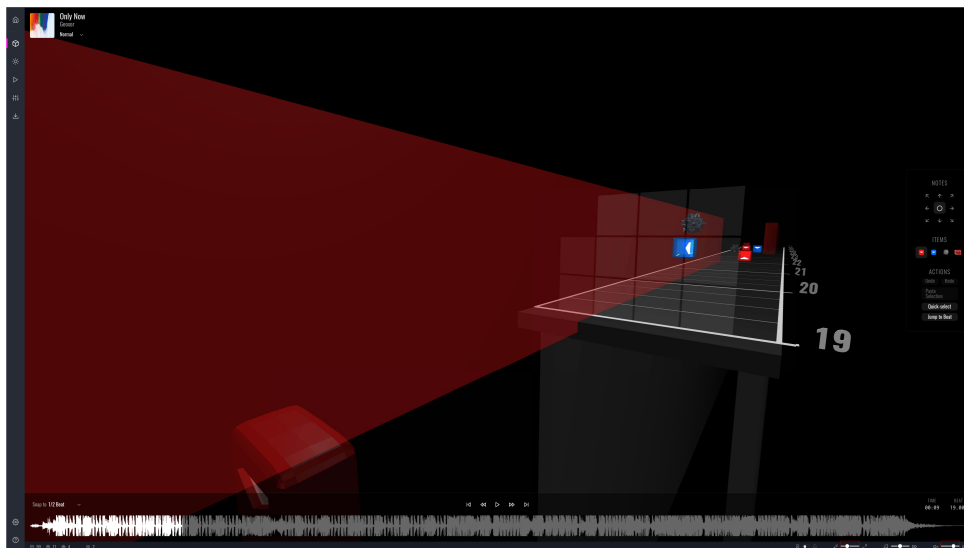
Editor úrovní je klíčový nástroj při vývoji hry a způsob pro uživatele, jak si do hry doplnit nový obsah. Hlavním cílem editoru je umožnit a usnadnit vytváření kvalitního obsahu pro hru.

5.1 Rešerše existujících editorů

Nejprve se pro porovnání podívejme na několik existujících editorů úrovní a to konkrétně editorů pro hru BeatSaber.

5.1.1 Beatmapper

Beatmapper[13] je dle mého názoru nejintuitivnější editor úrovní pro Beat Saber.



Obrázek 5.1: Hlavní obrazovka editoru beatmapper

Hlavním cílem tohoto editoru je dle jeho dokumentace především jednoduchost a uživatelská přívětivost. Toho dosahuje tím, že hlavním prvkem ed-

itačního rozhraní je náhled vytvářené úrovně, kterou uživatel může přímo editovat.

Prvním krokem při vytváření úrovně je v tomto editoru nahrání zvukového souboru a vyplnění základních informací jako jméno písně, autora a obtížnosti. Dále uživatel musí vyplnit počet úderů za minutu (BPM) a jejich posun vůči začátku. Toto je složitá informace na zjištění, což beatmapper v tomto případě naštěstí řeší výbornou dokumentací.

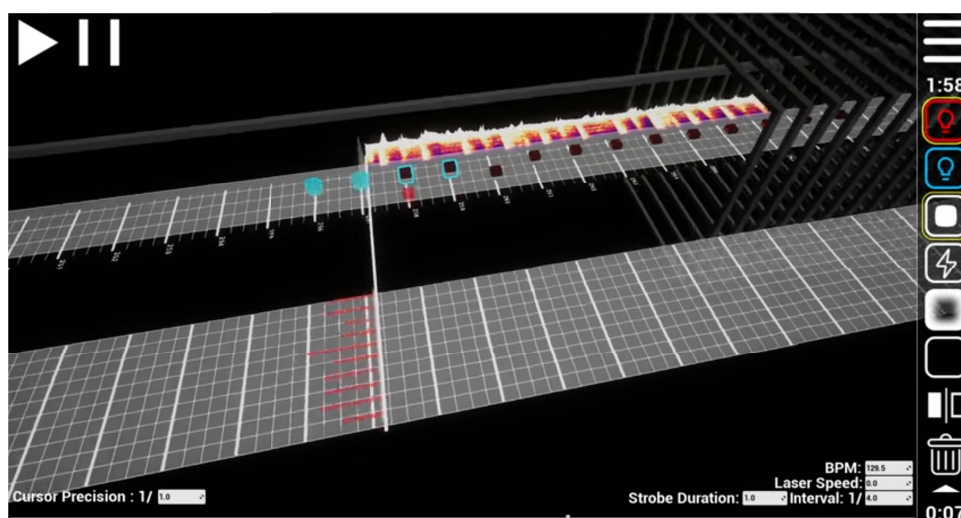
Nastavování BPM na začátku vytváření úrovně je největší výhodou a zároveň nevýhodou tohoto editoru. Výhoda spočívá v tom, že pokud máme píseň která následuje konzistentní BPM tak je editování mnohem snazší, protože máme přesně stanovené doby. Nevýhody jsou dvě: první a důležitější nevýhoda je, že jsou písně jimž se v průběhu BPM mění - tyto písně beatmapper jednoduše není schopen namapovat. Druhá nevýhoda je, že pokud chceme některé údery o kousek posunout vůči pravidelným taktům tak to nelze. Toto lze částečně obejít použitím možnosti mezidob avšak to shledávám náročným udělat dobře.

Hlavního rozhraní (Obrázek 5.1) tohoto editoru je intuitivní a uhlazené. Hlavním prvkem je velký náhled editované úrovně ve které může uživatel přímo umisťovat kostky. Směr kostek se umisťuje přetažením. Mazání kostek se provádí pravým tlačítkem. K pohybu v písni slouží časová osa, která má podobu grafu zobrazujícího hudební stopu.

Celkově se s tímto editorem pracuje dobře.

5.1.2 Mediocre Mapper

Mediocre Mapper[14] je editor úrovní v podobě aplikace pro Windows. To má jednu zásadní výhodu a tou je, že úrovně může uložit přímo do složky se hrou a uživatel tedy nemusí přesouvat žádné soubory pro vyzkoušení své mapy.



Obrázek 5.2: Hlavní obrazovka editoru Mediocre Mapper

Je to editor, který podporuje mnoho pokročilejších funkcí, avšak dělá to na úkor uživatelské přívětivosti. Osobně jsem měla problém vůbec novou úroveň vytvořit. Hlavní rozhraní je však celkem přívětivé - sestává se také především z náhledu písně v 3D pohledu, ale k tomu přidává náhled na frekvenční rozklad (obrázek 5.2, horní část) dané písně, což je užitečné pro kontrolu zarovnání úderů.

Tento editor také spoléhá na počáteční nastavení BPM, ale umožňuje ho změnit v průběhu, když uživatel zjistí, že ho nastavil mírně špatně. Dále umožňuje měnit BPM v průběhu jedné úrovně, což je užitečné pro mapování písní, které mění BPM v průběhu.

■ 5.1.3 Zabudovaný editor

Poslední editor úrovní pro Beat Saber, který jsem vyzkoušela je zabudovaný editor přímo ve hře. Tento editor volí výrazně jiný postup od předchozích dvou editorů tím, že je celý 2D.

Jeho hlavní výhodou je přímá zabudovanost do hry a tedy, že pro vyzkoušení úrovně stačí pouze kliknout na tlačítko vyzkoušet.

Hlavní rozhraní také spoléhá na fixní BPM a obsahuje jeden řádek pro každý úder a jeden sloupec pro každou možnost nastavení. Je zde sada sloupců pro horní, střední a dolní umístění úderů. Dále je zde sada sloupců pro miny a další.

Co v tomto editoru shledávám zvláštním je, že jeho rozlišení je svázané s rozlišením hry, což na mém monitoru znamená, že celé rozhraní je rozmazané a těžko se na něj kouká.

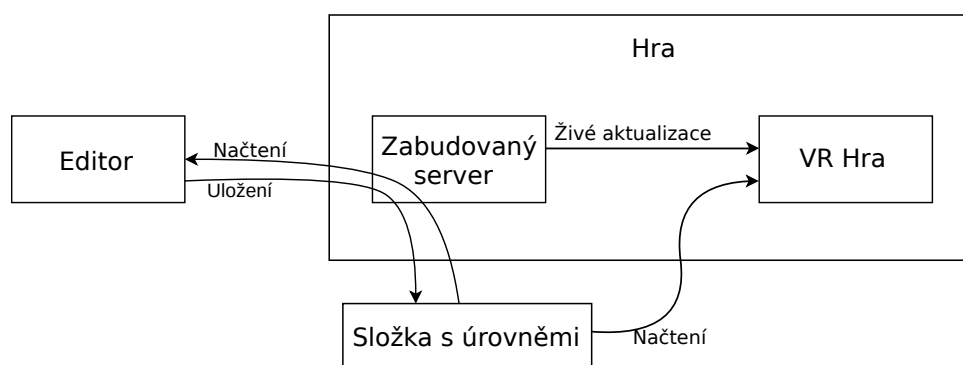
■ 5.2 Návrh řešení editoru

Editor úrovní jsem se rozhodla vytvořit jako webovou stránku a to z důvodu rychlé iterace při jeho vytváření (se správnou konfigurací uložím změny a téměř ihned je vidím), snadné navázání serveru pro sdílení úrovní (není součástí této práce), absenci nutnosti cokoli instalovat a v neposlední řadě také proto, že s webovými technologiemi umím pracovat.

Když uživatel chce vytvořit úroveň tak mu jednoduše stačí otevřít v moderním webovém prohlížeči adresu `http://127.0.0.1:8018` se hrou spuštěnou v pozadí.

Ukládání úrovní probíhá přes jednoduchý do hry integrovaný HTTP server - hra ověří, že je k ní přístupováno z důvěryhodné stránky (aby nebylo možné na disk ukládat soubory z libovolné stránky) a následně editoru zpřístupní seznam existujících úrovní přes REST API[15]. Dále umožní editoru vytvářet a editovat úrovně viz Obrázek 5.3.

Úrovně vytvořené pomocí editoru se ukládají do složky se hrou a jsou tudíž dostupné i po jejím restartu. Hra také umožňuje živě upravovat již běžící úroveň - tedy není potřeba pro načtení upravené úrovně ani přecházet do menu nebo zpět.



Obrázek 5.3: Vnitřní fungování ukládání a načítání úrovní

Hlavní nevýhodou webového editoru by za normálních okolností bylo to, že uživatel musí být připojen k internetu, aby si ho zobrazil. To neplatí především protože soubory editoru jsou přiloženy ke hře a ke spuštění editoru je potřeba mít hru spuštěnou v pozadí.

5.3 Postup vytváření úrovně

Nejdříve musí uživatel vybrat buď zvukovou stopu pro vytvoření nové úrovně nebo vybrat už existující úroveň. Výběr zvukové stopy se provádí pomocí systémového dialogu pro výběr souborů. Existující úrovně se načtou pokud hra běží na pozadí a zobrazí se jejich seznam.

Následně se editor uživatele zeptá na základní informace o úrovni (informace definované v sekci 4.3 mimo seznam úderů).

Po vyplnění těchto informací se uživateli zobrazí hlavní rozhraní editoru.

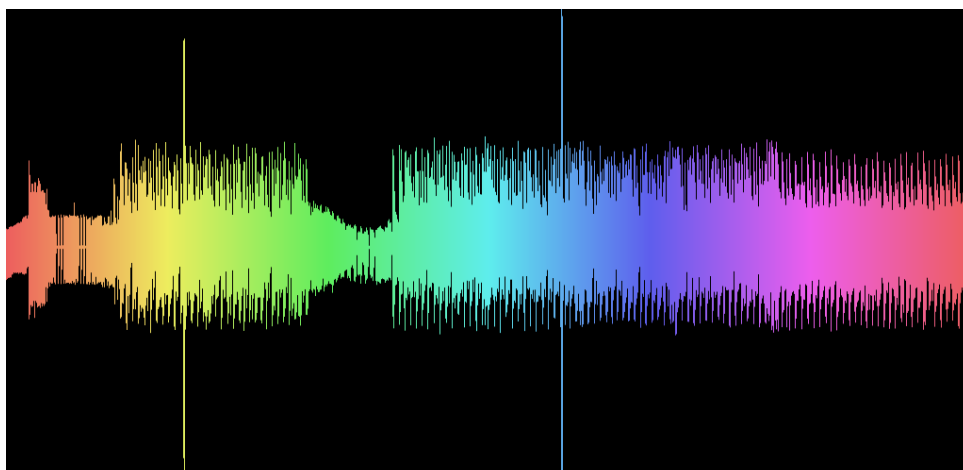
5.4 Hlavní rozhraní

Toto rozhraní obsahuje především časovou osu ve formě grafu zobrazujícího průběh signálu (Obrázek 5.4).

Touto časovou osou se dá pohybovat především pomocí kolečka myši, které při držení klávesy ctrl ovládá přiblížení/oddálení a při držení klávesy shift ovládá posun v čase (vodorovná osa).

Jak je vidět z obrázků tak časová osa je vybarvená v barvách duhy. To splňuje dva účely. Prvním účelem je vizuální souvislost se hrou, pro kterou je to editor - hra je o obarvování a tedy editor je taky barevný. Druhý účel je usnadnění orientace při posouvání se v čase - jedna konkrétní část vlny má stabilní barvu, takže i při obvyklé pravidelnosti hudebních souborů je snadné se orientovat. Barevnost tedy působí jako kotva.

Dále se na tento graf umísťují štítky označující herní události. Je zde také indikátor aktuální pozice v písni. Hlavní výhodou tohoto přístupu je, že umožňuje přesné mapování herních událostí na hudbu i při výrazně měnícím se BPM.



Obrázek 5.4: Příklad grafu průběhu signálu

Podstatnou částí tohoto editoru je samozřejmě možnost píseň přehrát. Ke kontrole správné synchronizace na hudbu slouží jednak vizuální reprezentace popsaná výše, ale také přidání zvuku metronomu do přehrávané hudby.

Druhou částí rozhraní je 3D náhled (o tom více v sekci 5.6)

5.5 Editační nástroje

Editor umožňuje dělat několik operací: vytvořit úder, smazat úder, zpět, znovu, přebarvit úder a posunout úder. Další pár operací je kopírování a vložení pozice a barvy úderu najednou - toto slouží k vytvoření dvou totožných úderů.

Vytvoření úderu se provádí na hlavní časové ose. Uživatel má dvě možnosti, jak tento nástroj použít: klávesová zkratka a tlačítko. Tlačítko funguje podobně jako nástroje v Microsoft Malování - po zmáčknutí se editor přepne na tento nástroj a používá se dokud uživatel nevybere jiný. Když uživatel aktivuje klávesovou zkratku tak se akce provede na místě, kde má uživatel momentálně kurzor myši. Toto umožňuje rychle používat několik různých nástrojů najednou.

Smazání úderu funguje podobně jako jeho vytváření akorát s tím rozdílem, že uživatel nemusí najet myší na přesné místo, ale provede se pro úder, kterému je kurzor nejbližší.

Pro každou barvu na kterou jde úder přebarvit existuje separátní tlačítko a klávesová zkratka. Tato sada nástrojů funguje stejně jako mazání úderu.

Nástroje zpět a znovu jsou z hlediska uživatelské interakce snadné - prostě se při aktivaci jak klávesovou zkratkou tak kliknutím na odpovídající tlačítko provedou.

Jelikož se v editoru nepíše text tak jsou klávesové zkratky jednotlivá písmena bez modifikačních kláves. Tato písmena jsou zobrazena na odpovídajících tlačítkách, aby si je uživatel nemusel pamatovat.

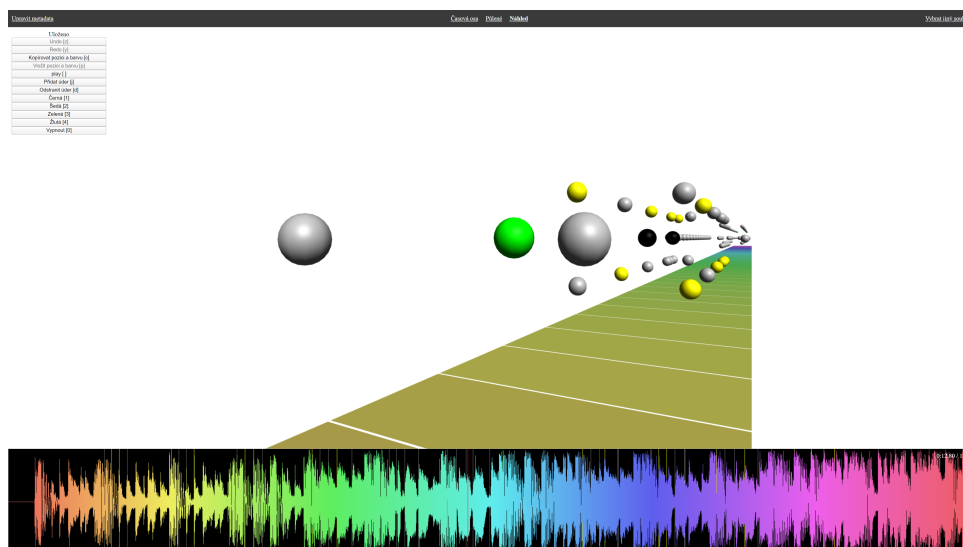
Pro editaci horizontální a vertikální pozice (tj. ne v čase) není časová osa vhodná. Proto jsem implementovala 3D náhled.

5.6 3D náhled

Kromě časové osy rozhraní také obsahuje 3D náhled rozložení scény. V tomto náhledu se zobrazují kuličky v prostoru (obrázek 5.7). Časové posunutí je svázáno s časovou osou, takže ta slouží jako navigátor a zároveň toto prostým přehráním písně umožňuje hrubý náhled, jak bude úroveň vypadat. K tomuto pomáhá fakt, že se nezobrazují úder v minulosti (ty jejich čas je nižší než je současná pozice v písni).

V tomto zobrazení fungují nástroje přebarvení a smazání úderů tím způsobem, že se aplikují na kuličky, kterou má uživatel pod kurzorem myši.

Hlavní účel tohoto zobrazení je umístění úderů v prostoru. To se provádí jednoduše chycením úderu a jeho přetažením. Toto přesouvání probíhá v rovině odpovídající časovému okamžiku zvoleném v časové ose. Editor neumožní úder umístit mimo limitovaný prostor - pod zem, mimo hráčův vertikální dosah ani příliš daleko od středu horizontálně.

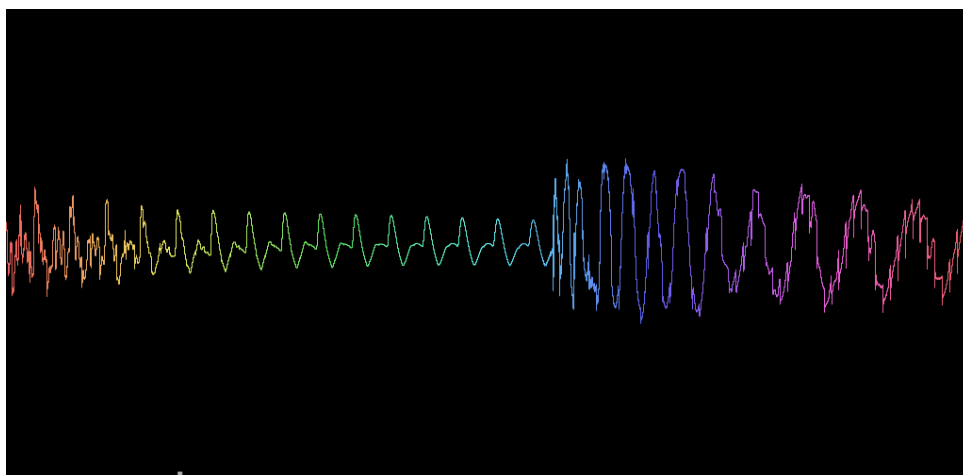


Obrázek 5.5: Příklad 3D náhledu v editoru

Jelikož by bylo nepohodlné a náročné všechny úder přesouvat ručně tak jsem vytvořila nástroj pro kopírování a vkládání pozice úderu. Uživatel jednoduše vybere nástroj kopírování a klikne na úder, vybere nástroj vkládání a klikne na jiný úder pro duplikaci pozice a barvy.

5.7 Implementace časové osy

Pro zobrazení časové osy se ukázalo nejjednodušší pro každý horizontální pixel vykreslit čáru šířky 1 pixel se spodní hranou v minimální a horní hranou



Obrázek 5.6: Příklad přiblíženého grafu průběhu signálu

v maximální hodnotě v časovém úseku, který tento pixel reprezentuje.

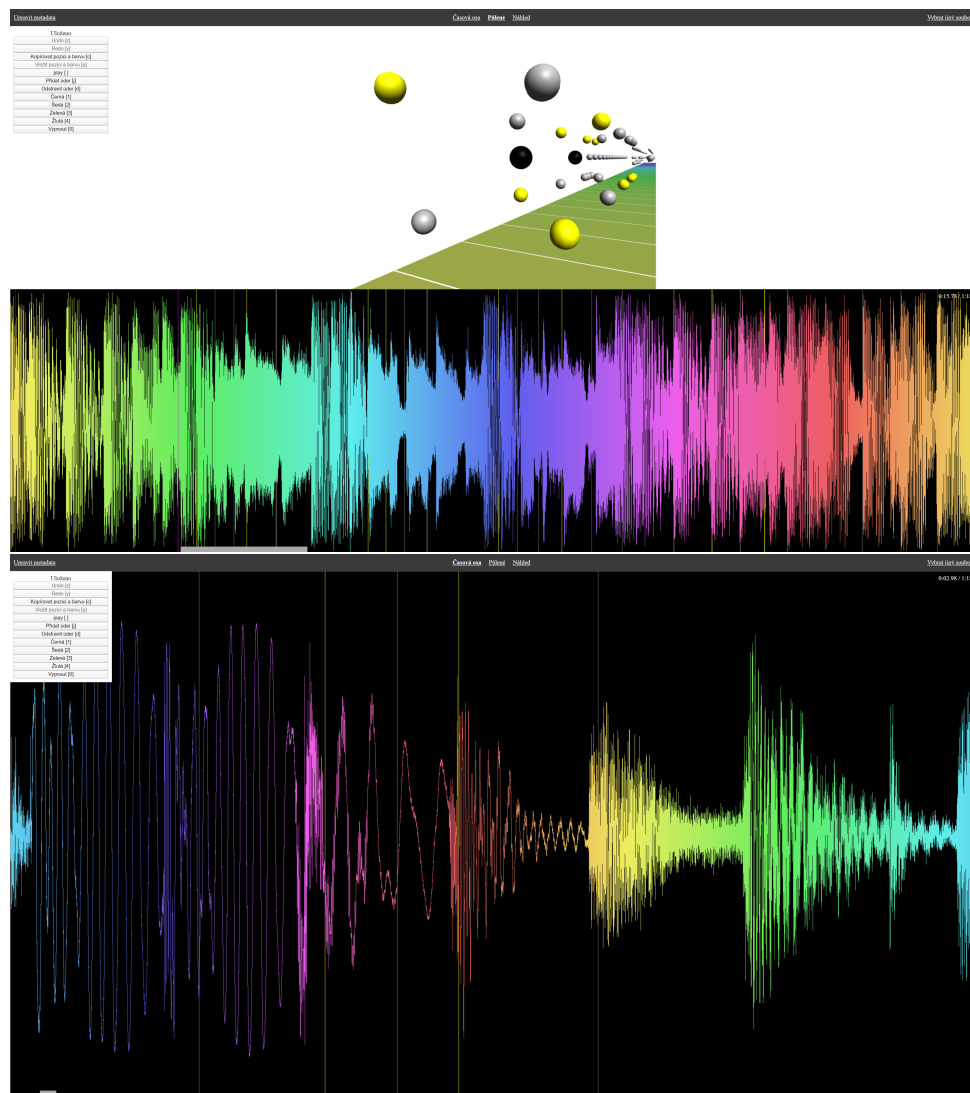
Časová osa je hlavní interakční prvek při vytváření úrovně a je tedy klíčové, aby byl plynulý - v ideálním případě by se měla zvládnout spolehlivě vykreslovat 60x za sekundu (60 FPS). Spočítat maxima a minima pro interval při obvyklé vzorkovací frekvenci 44100Hz v reálném čase je téměř nemožné, takže jsem se rozhodla při otevření souboru spočítat hodnoty pro všechna možná přiblížení. To sice zabere v závislosti na délce přesně několik sekund, to však při vytváření úrovně není problém, protože vzhledem k postupu při vytváření úrovně uživatel ze začátku časovou osu ani nevidí a ta se tak může připravovat v pozadí.

5.8 Implementace 3D náhledu

K implementaci 3D náhledu jsem použila knihovnu React Three Fiber[16]. Ta mi umožňuje jednoduše specifikovat rozložení a vzhled scény a jak odpovídá stavu editoru, aniž bych musela implementovat logiku, která bude zobrazení aktualizovat při změnách.

K provedení interakce editačních nástrojů používám nízkoúrovňové raycast nástroje, protože abstrakce poskytnutá touto knihovnou se ukázala být příliš pomalá (pro nalezení objektu na který uživatel kliknul provede raycast pro každý interaktivní objekt, když stačí provést pouze jeden raycast pro všechny najednou).

Přesouvání kuliček je implementováno tak, že v momentě, kdy uživatel stiskne tlačítko myši nad určitým objektem tak se vytvoří neviditelná plocha, která odpovídá všem pozicím středu kuličky v daném čase. Následně dokud uživatel tlačítko nepustí tak se kulička přesouvá vždy na místo v této ploše, nad kterým je umístěna myš. Tento přístup funguje i pokud uživatel myši pohne rychle nebo pokud vyjede z omezeného prostoru.



Obrázek 5.7: Další dvě možná rozložení editoru

5.9 Možná vylepšení editoru

Možných vylepšení editoru úrovní je mnoho. Nejvýraznějším z nich by bylo přidat možnost kopírovat celou sadu úderů. Toto by urychlilo vytváření úrovní. Další užitečnou funkcí by byla možnost kopírovat x a y pozice zvlášť.

Kapitola 6

Testování

Testování je důležitou součástí tvorby jakéhokoli programu. Je to způsob jak zjistit, jestli program splňuje zadané cíle a požadavky a je to také proces, který je schopný ukázat na možná vylepšení.

6.1 Kognitivní průchod

Jednou z nejrychlejších testovacích metod je kognitivní průchod[17]. Je to testovací metoda, která nevyžaduje uživatele a přesto je schopna poskytnout určitý vhled do aplikace. Odpovídá na otázku "Jak úspěšně tento design dovede nového uživatele k jeho cíli?"

Tento test jsem provedla v průběhu vytváření designu hry.

Kognitivní průchod je původně směřován na tradiční uživatelská rozhraní. Proto ho zde aplikuji na hlavní menu. Cílem uživatele je "spustit úroveň". Jako počáteční stav považuji uživatele v hlavním menu s nasazeným náhlavním displayem a držícím ovladače. Dále uživateli bylo vysvětleno kde je umístěno hlavní tlačítko trigger.

Tento test probíhá tak, že provedu kroky potřebné k dosažení požadovaného cíle a u každého kroku se zeptám na následující tři otázky:

- Q1. Bude správná akce uživateli zřejmá?
- Q2. Bude si uživatel schopen spojit popis akce se svým cílem?
- Q3. Dostane uživatel dostatečnou zpětnou vazbu?

Krok 1. Výběr písně

Q1. Pouze pokud se uživatel dívá dopředu (nález 1)

Q2. Pokud ví, že má vybrat písně (nález 2)

Q3. Ano.

Krok 2. Spuštění úrovně

Q1. Ano.

Q2. Ano.

Q3. Ano.

■ Nálezy

Nálezy z tohoto testu jsou dva.

První nález je, že pokud se uživatel nedívá dopředu tak neví, že se má otočit. Priorita tohoto nálezu je nízká, neboť je hra koncipovaná jako party hra a jde očekávat, že je v místnosti zkušenější hráč, který bude schopen poradit, že se uživatel má otočit. Avšak řešení je jednoduché - stačí za hráče přidat šipky, které ho nasměrují k menu.

Druhý nález je, že uživatel nemusí vědět, že má vybrat píseň. Priorita tohoto nálezu je střední, protože to může uživatele zmást. Řešením je přidat nad seznam úrovní popisek "vyberte píseň".

■ 6.2 Testování s uživateli

Pro vyhodnocení funkčnosti mého návrhu jsem provedla testování s uživateli. Při testování jsem se držela zásad stanovených Jakobem Nielsenem[18]. Testování probíhalo v místnosti nainstalovaným náhlavním displayem HTC Vive, kde moderátor seděl u počítače a viděl to, co viděl účastník.

Pro vyřazení problémů s ovládáním virtuální reality, které nemají nic společného s testovanou hrou jsem vypnula SteamVR tlačítka na ovladačích. Tímto se zamezilo zmatení uživatele, který by omylem mohl toto tlačítko stisknout.

Pro výběr účastníků jsem vytvořila krátký dotazník, který ověřil, že spadají do cílové skupiny a zároveň nebudu muset příliš vysvětlovat virtuální realitu samotnou. Požadavky na účastníka byly následující:

- Účastník má zkušenosti s virtuální realitou
- Alespoň jednou hrál jinou hudební hru pro virtuální realitu
- Nemá výraznou ztrátu zraku ani sluchu
- Základní znalost angličtiny

Dále jsem dbala na to, aby měly zastoupení kategorie jak hudebně vzdělaných uživatelů tak hudebně méně zdatných. Toto je podstatné, protože styl hraní podobných her se často odvíjí od způsobu vnímání rytmu.

Testování probíhalo v následujících krocích:

- Před příchodem účastníka jsem se ujistila, že je vše plně funkční a připravené. Hra byla spuštěna na obrazovce výběru písně (úvodní obrazovka) a ovladače byly dostatečně nabity.
- Při příchodu účastníka byl účastník uvítán a byl seznámen s postupem testování. Ujistila jsem ho, že není testován on, ale hra a že pokud s něčím bude mít problém tak to nejspíš není jeho chyba. Dále byl informován, že může testování kdykoli ukončit. Následně jsem účastníkovi položila dodatečné otázky, které slouží k ujištění se o vhodnosti účastníka a získání dalšího kontextu o získaných nálezech.

- Ve fázi testování účastník postupoval podle pokynů sepsaných níže. Obvykle jsou tyto pokyny napsány na papír a moderátor nezasahuje. To vzhledem k tomu, že má uživatel na hlavě náhlavní display nejde, takže jsem je účastníkovi sdělovala ústně.
- Následně jsem účastníkovi nejdřív položila předpřipravené otázky.
- Dále jsem účastníkovi sdělila informace o tom, jak přesně fungují různé barvy kuliček a o způsobu výpočtu skóre.
- Následně jsem umožnila účastníkovi hru si zahrát znovu, protože některé informace nebyly okamžitě zřejmé a chtěla jsem zjistit, jak jejich znalost ovlivní způsob hraní. Toto simuluje situaci, kdy si uživatel hru zahraje a poté si najde dodatečné informace v manuálu či na internetu a následně si hru zahraje znovu.
- Nakonec jsem se zeptala na několik konečných otázek a dala jsem účastníkovi prostor na dotazy a možnost vyjádřit své myšlenky, které nevyjádřil v průběhu.

Testování jsem provedla se třemi uživateli. Pokyny pro uživatele byly následující:

- Nasadit si náhlavní display a uchopit ovladače
- Následně jsem dala účastníkovi čas se zorientovat
- Dokončit jednu úroveň
 - Vybrat píseň
 - Stisknout start
 - Odehrát úroveň
 - Zadat své jméno
 - Potvrdit stisknutím klávesy enter
 - Vrátit se zpět do menu
- Dokončit další úroveň
- Dokončit všechny zbylé úrovně

Cíleně jsem se snažila nezasahovat a pokyny jsem sdělovala pouze pokud je účastník neudělal sám od sebe. Do fáze testování jsem vstupovala především mezi úrovněmi, kdy jsem se účastníka zeptala na jeho postřehy a poznatky.

■ Výsledky testování s uživateli

Participant 1 osobně nevlastní náhlavní display, ale má zná někoho kdo ano. Má tedy s virtuální realitou větší zkušenosti než někdo kdo si jí jednou vyzkoušel, ale menší než někdo kdo má náhlavní display doma. Podobná hra, kterou hrál je Beat Saber. Dále tento účastník nehraje na žádný hudební nástroj.

Participant neměl žádný problém vybrat píseň a spustit úroveň. Účastník velmi rychle pochopil, že má odpalovat přilétající kuličky. Přišlo mu však, že jsou páčky otočené o 90° od toho, než jak by je očekával (nález 1). Toto však nebral jako velký problém - prostě uchopil ovladače jinak.

Při vyplňování svého jména neměl větší problémy. Po dokončení úrovně uvedl, že si průběžně nevšiml obarvování úrovně a že toho bylo příliš mnoho a že se cítil zmateně (nález 2).

Po dohrání druhé úrovně uvedl, že pochopil obarvování stěn pomocí kuliček. Při vyplňování svého jména podruhé však měl problém, protože si myslel, že jeho jména je předvyplněné jméno a rovnou stiskl enter (nález 3).

Po dokončení všech úrovní při vysvětlování uvedl, že pochopil, že různé barvy kuliček mají různé vlastnosti, ale kromě žluté netušil jaké (nález 4).

Při druhém průchodu hrou ji hrál mírně jinak - především se snažil kuličky odpalovat větší rychlostí. Uvedl, že když hrál konkrétní úroveň podruhé tak se mu hrála snáz. Také se snažil odrážet zelené kuličky pomaleji, což mu moc nešlo, ale uvedl, že to je zábavný prvek.

Celkově se mu hra líbila. Ocenil především výběr hudby. Frustrovaly ho však kuličky umístěné příliš nízko. Participant konstatoval, že by bylo zajímavé nehrát na skóre ale ve skupině vzájemně hodnotit umělecký projev při hraní úrovní.

Participant 2 podobně jako první participant nevlastní náhlavní display a také zná někoho kdo ano. Úroveň zkušeností je tedy podobná. Také hrál Beat Saber a hrál Audioshield. Na rozdíl od prvního účastníka však tento účastník má hudební vzdělání a mnoho let hraje na různé hudební nástroje.

Tomuto participantovi trval výběr úrovně déle, což bylo způsobeno tím, že byl okouzlen přebarvováním pomocí ukazovátka v menu. Když se jeho pozornost přesunula na zadaný úkol, neměl problém úroveň vybrat.

Na rozdíl od prvního participanta mu páčky přišly správně, protože odrážel kuličky jiným směrem. Stejně jako první participant se cítil v průběhu první úrovně zmateně a přehlčeně (nález 2). Ke konci úrovně už však byl schopen spolehlivě odrážet většinu přilétajících kuliček.

Po dokončení úrovně uvedl, že bylo těžké odhadnout, kdy trefit kouli - po druhém průchodu už však tento problém neměl a uvedl, že stačilo si zapamatovat danou píseň.

Participant také neměl problém s prvním zadáním jména, ale při druhém zadání nepochopil tlačítka pro rychlé zadávání. Místo toho prostě jméno zadal znovu. Toto se ukázalo být chybou ve hře, která způsobila, že se tlačítka pro rychlé zadání jména nezvětšovala při najetí kurzoru (nález 5).

Po dohrání všech úrovní uvedl, že chápe význam zelené kuličky v tom,

že se rozdělí na několik ale nechápe mechanismus (nález 4). U žluté kuličky pochopil, že nemizí při kontaktu se stěnou.

Při druhém průchodu změnil způsob hraní hlavně tím způsobem, že se snažil obarvit co největší část úrovně okolo něj.

Celkově ho hra bavila a ocenil dobré namapování kuliček na rytmus písni. Dále ocenil, že hra umožňuje dělat "vše co by očekával"- konkrétně, že může s kuličkou například driblovat. Dále mu přišly zábavnější těžší úrovně ať už byly těžší vzdáleností kuliček nebo rychlostí úderů.

Participant 3 vlastní náhlavní display, takže má větší zkušenosti s virtuální realitou než participant 1 a 2. Hrál hudebních her pro virtuální realitu včetně Beat Saberu. Hraje na hudební nástroj.

Po spuštění hry neměl žádný problém spustit úroveň a v úrovni se zorientoval rychle. Celkem rychle pochopil, že má odrážet kuličky. Průběh druhé úrovně strávil zkoumáním co dělá která barva kuličky a obecně se moc nesnažil mít dobré skóre. Z tohoto byl schopen celkem přesně vyvodit fungování jednotlivých barev kuliček.

Po skončení první úrovně byl chvíli zmatený co se děje a doba pro zobrazení klávesnice mu přišla příliš dlouhá (nález 6). Při druhé písni mu stále přišla doba pro zobrazení klávesnice příliš dlouhá, ale už nebyl zmaten. Se zadáním jména ani s použitím rychlé volby pro druhé zadání neměl problém.

Vibrace ovladačů mu přišly příliš slabé (nález 7).

Celkově zhodnotil, že hra má potenciál, ale že by potřebovala víc dotáhnout aby stála za nákup.

■ Seznam nálezů

Následuje seznam všech nálezů seřazený podle závažnosti. U každého nálezu uvedu jeho závažnost, seznam dotčených participantů, popis a navrhované řešení. Závažnost rozdělují na následující kategorie:

- Závažnost 1 - nízká. Jedná se spíš o kosmetické problémy
- Závažnost 2 - střední. Jedná se o nálezy, které komplikují hraní a je vhodné je odstranit.
- Závažnost 3 - vysoká. Jedná se o důležité problémy. Je nutné je odstranit.

Nález 2

- Závažnost: 2 (střední)
- Dotčení participant: participant 1 a 2
- Participant se po první úrovni cítili zmateně a přehlčeně. Měli pocit, že toho bylo příliš mnoho.

Nález 3

- Závažnost: 2 (střední)

- Dotčení participantů: participant 1
- Nepochopení funkce rychlé volby a spletení si této funkce s předvyplněným jménem.

Nález 4

- Závažnost: 2 (střední)
- Dotčení participantů: participant 1 a 2
- Ze hry sám nepochopil funkci různých barev kuliček.

Nález 7

- Závažnost: 2 (střední)
- Dotčení participantů: participant 3
- Příliš slabé vibrace ovladačů

Nález 1

- Závažnost: 1 (nízká)
- Dotčení participantů: participant 1
- Pálky byly otočené o 90° od očekávaného natočení.

Nález 5

- Závažnost: 1 (nízká)
- Dotčení participantů: participant 2
- Nepochopení funkce rychlé volby a její následné ignorování

Nález 6

- Závažnost: 1 (nízká)
- Dotčení participantů: participant 3
- Doba před zobrazením klávesnice byla příliš dlouhá

■ Řešení nálezů

Řešení některých nálezů byla celkem snadná. Jiná byla celkem složitá. Následuje seznam implementovaných řešení výše uvedených nálezů. Uvádím je ve stejném pořadí jako jsou nálezy uvedeny v předchozí sekci.

Nález 2 Problém s přehlcením participantů v první úrovni jsem vyřešila vytvořením ještě jednodušší první úrovně.

Nález 3 Tento problém jsem se rozhodla neřešit plně kvůli složité validaci řešení. Rozhodla jsem se pro částečné řešení tím, že neumožním zmáčknout enter bez vyplnění jména.

Nález 4 Toto by se dalo plně vyřešit vytvořením tutoriálové úrovně, která by uživatele informovala o funkci jednotlivých kuliček. Já jsem se rozhodla opět pro částečné řešení tím, že jsem vytvořila jednoduchou první úroveň ve které má hráč čas sledovat chování jednotlivých kuliček.

Nález 7 Příliš slabé vibrace ovladačů jsem vyřešila jednoduše zvýšením intenzity vibrací.

Nález 1 Tento nález jsem se rozhodla neřešit protože to jiným participantům přišlo intuitivní. Potenciální řešení by mohlo spočívat v nastavitelnosti otočení pálek.

Nález 5 Jelikož toto bylo způsobeno chybou ve hře tak stačilo tuto chybu opravit - nastavit aby se tlačítka rychlé volby zvětšovala při najetí kurzoru.

Nález 6 Toto jsem se také rozhodla neřešit, neboť byl pouze jeden participant zmaten po dobu pěti sekund. Potenciální řešení by spočívalo v zobrazení klávesnice po pěti sekundách nebo po tom co zmizí všechny kuličky, to co přijde první.



Kapitola 7

Závěr

V úvodu práce jsem si stanovila několik cílů pro tuto práci. Hlavním cílem bylo vymyslet a následně implementovat zábavnou herní mechaniku založenou na hudbě. K tomuto cíli jsem nejprve prozkoumala. Ukázalo se, že podobných her existuje i pro virtuální realitu hned několik a že je to populární žánr.

Dále jsem na základě inspirace z rešerše a také tak, aby hra nebyla prostým klonem vytvořila hlavní herní mechaniku. Cílem hráče tedy je do rytmu odrážet barevné kuličky. Na základě designu jsem vytvořila hru v herním enginu Unity a paralelně s ní jsem vytvořila i editor úrovní.

Následné testování odhalilo několik drobných nedostatků v původním designu i implementaci, z nichž jsem nejzávažnější opravila. Testování také ukázalo, že základní koncept hry je zábavný, jelikož si testeři hru chtěli zahrát znovu nad rámec testování.

Hlavními možnostmi pro rozšíření této hry jsou tvorba více úrovní a umožnění různých vzhledů jednotlivých úrovní. Tvorba více úrovní je díky editoru úrovní celkem snadná. Umožnění různých vzhledů jednotlivých úrovní je složitější, ale také by mělo být dosažitelné.

Příloha A

Literatura

- [1] Steven M. LaValle. *Virtual Reality*. Cambridge University Press, 2016.
<http://vr.cs.uiuc.edu>.
- [2] Dylan Fitterer. *Audioshield*. [online] Steam [cit. 2019-12-21] Dostupné z: <https://store.steampowered.com/app/412740/Audioshield/>.
- [3] *Beat saber*. [online] Beat Games [cit. 2019-12-21] Dostupné z: <https://beatsaber.com/>.
- [4] Arrowiz. *Beats fever*. [online] Steam [cit. 2019-12-21] Dostupné z: https://store.steampowered.com/app/566780/Beats_Fever/.
- [5] Cheetah Games. *Piano tiles 2*. [online] Google play [cit. 2019-12-21] Dostupné z: <https://play.google.com/store/apps/details?id=com.cmplay.tiles2>.
- [6] Jason Jerald. *The VR Book: Human-Centered Design for Virtual Reality*. ACM Books, 2015.
- [7] Jovan Jovanovic. *How does Shazam work? Music Recognition Algorithms, Fingerprinting, and Processing*. [online] toptal.com [cit. 2019-12-21] Dostupné z: <https://www.toptal.com/algorithms/shazam-it-music-processing-fingerprinting-and-recognition>.
- [8] *MIDI*. [online] MIDI Association [cit. 2019-12-12] Dostupné z: <https://www.midi.org/specifications-old/item/the-midi-1-0-specification>.
- [9] Epic Games. *Unreal Engine*. [online] Epic Games [cit. 2019-12-21] Dostupné z: <https://www.unrealengine.com/en-US/>.
- [10] Unity Technologies. *Unity*. [online] Unity Technologies [cit. 2019-12-21] Dostupné z: <https://docs.unity3d.com/Manual/index.html>.
- [11] Valve Corporation. *SteamVR Unity Plugin*. [online] Github [cit. 2019-12-21] Dostupné z: https://github.com/ValveSoftware/steamvr_unity_plugin.

- [12] T. Bray. The javascript object notation (json) data interchange format. RFC 7159, RFC Editor, March 2014. <https://www.rfc-editor.org/rfc/rfc7159.txt>.
- [13] *Beatmapper*. [online] Joshua Comeau [cit. 2020-01-04] Dostupné z: <https://beatmapper.app/>.
- [14] *Mediocre Mapper*. [online] Mediocre Mapper contributors [cit. 2020-01-04] Dostupné z: <https://github.com/squeaksies/MediocreMapper/releases>.
- [15] Leonard Richardson and Sam Ruby. *Restful Web Services*. O'Reilly, first edition, 2007.
- [16] Paul Henschel. *React Three Fiber*. [online] github [cit. 2019-12-21] Dostupné z: <https://github.com/react-spring/react-three-fiber>.
- [17] W. M. Newman and M. G. Lamming. *Interactive System Design*. Addison-Wesley, 1995.
- [18] Jakob Nielsen. *Usability engineering*. Boston: AP Professional, 1993. ISBN 978-012-5184-069.