

Piano Tile

Dokumentace maturitní práce

Kateřina Chmelová

školní rok: 2023/2024

třída:

8.V

Zadání maturitní práce:

Cílem práce je vytvořit mobilní aplikaci podobnou známé hře Piano Tiles. Součástí řešení bude generace herních levelů z MIDI souborů. Program bude vyvíjen v herním engineu Unity primárně pro platformu iOS.

Teoretická část - Formáty pro ukládání zvuku

Prohlašuji, že jsem na práci pracoval samostatně pouze za pomoci použitých zdrojů a že v práci i v dokumentaci jasně vymezuji, které části kódů jsou mým originálním dílem, které jsou upravenou verzí a které jsou převzaty v plném rozsahu.

podpis žáka

OBSAH

OBSAH	4
1. Teoretická část	6
1.1. Kodeky	6
1.1.1. Ztrátové kodeky	6
1.1.2. Bezztrátové kodeky	6
1.2. Kontejnery	7
1.2.1. Zvukové kontejnerové formáty	7
1.2.2. Multimediální kontejnerové formáty	7
2. Cíle práce	8
3. Způsoby řešení postupu	8
3.1. Hlavní menu	8
3.1.1. Animace	8
3.1.2. Tlačítka	8
3.2. Přehled úrovní	8
3.2.1. Animace hvězd	9
3.2.2. Přechod do hry	9
3.3. Generace úrovní	9
3.3.1. Načtení MIDI souboru	9
3.3.2. Uzpůsobení velikosti not	10
3.3.3. Generace not	10

3.3.4. Zvuk	10
3.3.5. Noty	11
3.4. Konec hry	11
3.5. Debugging	11
3.5.1. Debug.....	11
3.5.2. Simulátor.....	12
3.5.3. Unity Remote.....	12
3.5.4. ADB logcat	12
4.1. Programovací jazyk.....	13
4.2. Vývojové prostředí	13
4.2.1. Unity	13
4.2.2. Visual Studio.....	13
4.3. Knihovna WetDryMIDI.....	13
6. Instalace.....	14
7. Ovládání	15
7.1. Hlavní menu	15
7.2. Přehled úrovní	15
7.3. Level.....	15
7.4. Obrazovka se skóre	15
8. Seznam použitých informačních zdrojů.....	16

1. TEORETICKÁ ČÁST

1.1. Kodeky

Kodek je software (případně i zařízení), které komprimuje a dekomprimuje zvuková data. Určuje, jak budou zvuková data zakódována, aby bylo možné je efektivně přenášet či ukládat. ^[1]

1.1.1. Ztrátové kodeky

Ztrátové formáty fungují tak, že je část informací vypuštěna. Jedná se o tu část dat, která je nejméně důležitá pro poslech a kterou běžný posluchač v nahrávce neslyší. Velikost výsledného souboru je rovna přibližně 10–15 % původního souboru

MP3 (MPEG Audio Layer III): MP3 umožňuje zmenšit velikost hudebních souborů přibližně na desetinu, u mluveného slova však dává výrazně horší výsledky (může být potlačena počáteční a koncová slabika, zkracovány pauzy mezi slovy). Obsahuje méně vzorků než AAC – má tedy nižší kvalitu. Využívá se převážně na streamování hudby.

AAC (Advanced Audio Coding): AAC je používán produkty Apple. Je také podporován platformami jako je YouTube. Odstraňuje nepotřebné složky signálu a duplicitní informace v kódovaném zvukovém signálu. Dokáže čtvercové a pulzní vlny komprimovat mnohem lépe než MP3. Vzorky tohoto kodeku zabírají poměrně málo místa, je tedy efektivnější. Je využíván kontejnery MP4, ADTS a 3GP.

AMR (Adaptive Multi-Rate): AMR Oproti MP3 lépe zpracovává mluvené slovo, je totiž určen pro řeč. Je navržený pro kompresi hlasových hovorů. Jeho cílem je minimalizace šířky pásma přenosu a jednodušší a rychlejší přenos dat. Je využíván kontejnery jako AMR nebo 3GPP.

1.1.2. Bezztrátové kodeky

FLAC (Free Lossless Audio Codec): FLAC používá lineární predikci pro konverzi zvukových vzorků. Velikost výsledného souboru se většinou pohybuje okolo 40 - 50%. Je tedy přesnější a udržuje více detailů než bezztrátové kodeky. Je využíván kontejnery jako MP4, Ogg a stejnojmenný FLAC.

ALAC (Apple Lossless Audio Codec): ALAC je podobně jako AAC kodek vyvinutý společností Apple. Vyskytuje se tedy hlavně na zařízeních iOS a macOS. Pro ukládání hudby kódované s ALAC

je často použit kontejner MP4 - soubory tedy mají příponu .m4a. Stejně jako FLAC využívá lineární predikci. Výsledný soubor má většinou velikost mezi 45 – 60% původního souboru. ^[2]

1.2. Kontejnery

Kontejnerový formát (nebo také multimediální kontejner) je souborový formát, který slouží k uchovávání více typů multimediálního obsahu (jako například zvukové soubory, titulky, videa a obrázky) najednou. Kontejner většinou na každý soubor využívá jiný kodek. ^[3]

1.2.1. Zvukové kontejnerové formáty

WAV (Waveform Audio File Format): WAV je bezztrátový formát kontejneru, který byl vytvořen firmami IBM a Microsoft za účelem ukládání zvuku na PC. Jelikož je jeho zpracování snadné a nenáročné, využívá se nejčastěji jako pracovní formát při zpracování zvuku - v profesionálních nahrávacích studiích a pro archivaci zvukových dat.

AIFF (Audio Interchange File Format): AIFF je bezztrátový formát kontejneru vyvinutý společností Apple. Je běžně používán v profesionálních nahrávacích studiích na platformě macOS. Využívá ale dost místa na disku – přibližně 10 MB na minutu stereo audia.

M4A (MPEG-4 Audio): M4A je často používaný pro streamování hudby a audio obsahu na internetu. Vyšší kompresní poměr umožňuje menší velikost souborů než běžné standardy MP3. Zároveň je dosažena lepší kvalita reprodukce.

1.2.2. Multimediální kontejnerové formáty

MP4 (MPEG-4 Part 14): MP4 je velmi univerzální kontejnerový formát - do MP4 kontejneru může být totiž vložený téměř jakýkoliv typ dat. Nejčastěji se ale používá MPEG-4 pro obraz a MP3 nebo AAC pro zvuk.

AVI (Audio Video Interleave): AVI je kontejnerový formát vyvinutý společností Microsoft. Je považován za zastaralý formát (1992). Když je používán s populárními MPEG-4 kodeky, má značné nedostatky, které zvětšují velikost souboru více, než je nezbytné. ^[4]

2. CÍLE PRÁCE

Cílem mé práce je vytvořit 2D hru v Unity, podobnou známé hře Piano Tiles. Hra bude obsahovat čtyři řady obdélníků podobných klávesám na pianu. Principem hry bude kliknutí na bílé klávesy dříve, než zmizí z obrazovky. Úrovně budou generovány za pomoci MIDI souborů a hra bude vyvinuta na mobilní platformu Android.

3. ZPŮSOBY ŘEŠENÍ A POUŽITÉ POSTUPY

3.1. Hlavní menu

Po spuštění aplikace se jako první zobrazí scéna hlavního menu. Ta je složena pouze ze dvou komponentů – animace a tlačítka na spuštění hry.

3.1.1. Animace

Všechny animace obsažené v této hře jsem vytvořila ze snímků zakreslených na tabletu a zaznamenaných pomocí nahrávání obrazovky. Těm jsem (většině) odstranila pozadí, převedla je na soubory GIF a následně vybrala pouze 4–25 snímků, které jsem převedla do sprite sheetu (PNG soubor obsahující mnoho menších obrázků uspořádaných ve mřížce).

V Unity jsem si vytvořila obrázkový komponent, do kterého se po spuštění načítají snímky animace skrz přiložený skript. V hlavním menu je skript uzpůsoben tak, aby nejdříve spustil animaci první, poté druhou a tu donekonečna opakoval.

3.1.2. Tlačítka

Tlačítka na změnu scény fungují (až na přechod do levelu) v celé hře stejně. Na Unity UI Button komponentu (předem vytvořená šablona Unity) je připevněn skript, do kterého se v editoru pouze přepíše jméno destinace, kterou kód využije ke změně scény.

3.2. Přehled úrovní

Seznam levelů je dostupný formou projíždějícího seznamu. Tento seznam je ohraničený maskovacím komponentem, aby přesahující levely nebyly viditelné mimo určený obdélník a nenarušovaly prostředí.

Pod maskou se skrývá komponent druhý (Scroll rect), který umožňuje samotné projíždění. V něm jsou poskládány levely. Každý level má vlastní buňku v kontejneru, která obsahuje jeho jméno a dosažený počet hvězd.

3.2.1. Animace hvězd

Každá možnost počtu hvězd (0, 1, 2 a 3) má vlastní animaci. Podle jména rodiče (v tomto případě buňky obsahující referenci na level) se z PlayerPrefs načte hodnota dosažených hvězd. Podle toho se spustí jedna z animací (také za použití sprite sheetu). PlayerPrefs v Unity je prostředek pro ukládání a načítání jednoduchých dat v podobě klíč-hodnota. Nejsou ale vhodné pro ukládání velkých množství dat nebo citlivých informací.

3.2.2. Přechod do hry

Pokud hráč klikne na jméno úrovně nebo počet hvězd, je přesunut na scénu levelu a do PlayerPrefs je uloženo jméno vybrané skladby.

Při každém přechodu mezi scénami se spustí animace ztmavení obrazovky, která je tvořena animátorem Unity. Jde o černý obrázek přes celou obrazovku, kterému se postupně snižuje a zvyšuje průhlednost. Přepnutí scény se koprogramem odsune o jednu vteřinu.

3.3. Generace úrovní

3.3.1. Načtení MIDI souboru

Jelikož knihovna WetDryMIDI v tuto chvíli není přizpůsobená k tvorbě na telefony, rozhodla jsem se z MIDI souborů skrz kód vytáhnout informace potřebné ke generaci levelů hry a uložila jsem je do textových souborů.

Využila jsem k tomu prázdnou scénu v Unity, kterou jsem nezahrnula v konečném buildu hry. Nachází se zde skript, do kterého se pomocí editoru zapíše jméno MIDI souboru (který musí být ve složce Midi v Assetech) a BPM hodnota skladby. Po spuštění se vygeneruje textový soubor o třech řádcích – v prvním jsou hodnoty not, v druhém čas spuštění každé noty a ve třetím délka nejkratší noty v souboru.

3.3.2. Uzpůsobení velikosti not

Jelikož každý hráč bude mít jiné rozlišení obrazovky a v tom případě i not, je potřeba změnit její velikost. Ve skriptu spuštěném při startu levelu se načte velikost obrazovky a tvar noty (bílý obdélník) se změní na čtvrtinu délky a výšky. Také se stanoví místo generace not – čtyři možné pozice v ose x a jedna v ose y.

3.3.3. Generace not

S generovanými textovými soubory už pracuje skript MidiFileInfo v samotném levelu. Informace si z textu převede do listů.

Noty generuje skript GameControl. Scéna je rozdělena do čtyř řádků po čtyřech notách (řádky se v průběhu hry posouvají dolů a generují se další). V každém řádku je jedna nota viditelná a zbytek neviditelný. Pozice viditelné noty v řádku se vždy liší od poslední. Noty se generují nad obrazovkou hráče.

GameControl porovnává čas od začátku levelu a čas příštího spuštění noty z listu. Pokud se shodují s menší odchylkou, vygeneruje viditelnou notu.

Původně se řady not negenerovaly přímo za sebou, ale s menší mezerou. Řešením bylo zapamovat si poslední vygenerovanou notu a za použití její souřadnice na ose y vygenerovat notu další. To sice zajistilo, že od sebe všechny noty byly stejně daleko, pořád mezi nimi ale byla mezera, kterou vyřešil výpočet odchylky z rychlosti not a hodnoty 0,02, která poté byla od y pozice generace noty odečtena.

3.3.4. Zvuk

Každé viditelné notě je při generaci přiřazena její hodnota a zvukový komponent poslední viditelné noty. K přehrání zvuku noty jsem chtěla využít knihovnu, žádnou jsem ale nenašla. Většina knihoven je totiž vytvořena pro softwary, které notu zvládnou zvýšit o půl tón dolů i nahoru. Proto si hudebníci ulehčují práci a nahrávají například každou třetí notu. Vytvořila jsem si tedy vlastní knihovnu skrz DAW PreSonus Studio One, exportováním noty po notě.

Hodnota každé noty se dá vyjádřit numericky – tudíž se každý .ogg soubor s notou jmenuje svým číslem. Každá nota si tedy najde svůj soubor a po dotknutí jej spustí. Zároveň při spuštění nota ztlumí notu poslední, aby se zvuk nepřekrýval.

V některých skladbách se ke zvuku klavíru kontrolovaného hráčem připojuje i hudební podklad. Ten se načte a spustí na začátku levelu, zase ze skriptu za použití PlayerPrefs.

3.3.5. Noty

Noty se posouvají pouze směrem dolů, rychlostí vypočtenou z nejkratší noty v MIDI souboru a výšky not. Pokud už je některá nota dotčena, hráč již nemůže kliknout na žádnou notu v jejím řádku ani v řádku pod ní. To zamezí jak dotekům hráče dlaní, tak přiřítání bodů za jednu notu dvakrát. Viditelná nota po stisknutí změní barvu na šedivou, neviditelná nota změní barvu na černou a zobrazí se po zapnutí příslušného SpriteRendereru.

Skóre se zvyšuje každým dotknutím noty. Na obrazovce je textový box, do kterého se při každé změně skóre přepíše stávající hodnota.

3.4. Konec hry

Hra může být ukončena třemi způsoby. Buď hráč klikne na špatnou notu – hra dotčenou notu zobrazí černě a spustí zvuk chyby; na notu nestihne kliknout, než opustí obrazovku – je zastavena skrz skript na notě, kontrolující její pozici na ose y a její dotčení; nebo správně trefí všechny noty – počet dotčených not se rovná počtu not v MIDI souboru.

Level se po jakékoliv z těchto situací zastaví – noty se přestanou pohybovat (skrz boo) a ztmavne obrazovka. Zároveň už se hráč žádné noty na obrazovce nemůže dotknout (bool touchInput). Hru jsem původně ukončovala skrz nastavení Time.timeScale na 0f, to ale zastavilo všechny časově závislé procesy.

Po skončení levelu je ze skóre vypočítána úspěšnost v procentech, která je následně převedena na počet hvězd. Počet dosažených hvězd se také porovná s nejvyšší dosaženou hodnotou, případně jí přepíše. Pustí se příslušná animace a audio. Možnosti audia jsou na každý počet hvězd dvě, program je vybírá náhodně.

3.5. Debugging

K ladění programu jsem převážně využívala čtyři techniky.

3.5.1. Debug

Funkce Debug.Log() je jednoduchý způsob, jak sledovat chování hry a zobrazovat informace do konzole v Unity. Pomáhá diagnostikovat a ladit kód hry.

3.5.2. Simulátor

Simulátor v Unity je nástroj, umožňující sledování běhu hry přímo v editoru na počítači. Tento nástroj je užitečný pro rychlé testování a ladění různých prvků hry.

3.5.3. Unity Remote

Unity Remote je mobilní aplikace pro iOS a Android. Přes USB kabel nebo Wi-Fi připojení je možné hru zobrazit na telefonu. Samotný kód však běží stále v editoru na počítači.

3.5.4. ADB logcat

Skrz ADB logcat můžeme sledovat logy z Android zařízení přímo z příkazové řádky. Pomocí tohoto nástroje lze získat informace o běhu aplikací, varováních a chybách. Já ho využila hlavně k monitorování výkonu.

4. PROGRAMOVACÍ PROSTŘEDKY

4.1. Programovací jazyk

Pro vývoj jsem použila programovací jazyk C#, neboť je optimalizovaný pro prostředí Unity.

4.2. Vývojové prostředí

4.2.1. Unity

Zvolila jsem herní engine Unity, protože poskytuje širokou škálu nástrojů pro tvorbu her, včetně práce s audio soubory a animacemi. Poskytuje také možnosti vývoje pro 2D hry a je uživatelsky přívětivé.

4.2.2. Visual Studio

Pro psaní kódu jsem využila Visual Studio. Toto IDE nabízí pokročilé funkce pro editaci kódu v jazyce C#, jako je automatické doplňování kódu a ladění.

4.3. Knihovna WetDryMIDI

WetDryMIDI je open-source knihovna v jazyce C#, která umožňuje práci s MIDI soubory. Je navržena pro snadnou integraci do projektů vytvářených v Unity nebo jiných prostředích, které podporují .NET Framework.

Tato knihovna poskytuje různé funkce pro manipulaci s MIDI daty, včetně získávání informací o notách v MIDI souboru a změnu tempa (funkce mnou využité).

Knihovna není kompatibilní ani s platformou Android, ani iOS. Nepovedlo se mi ale najít žádnou knihovnu v C# plně odpovídající mým požadavkům, implementovala jsem tedy vlastní řešení pro práci s MIDI informacemi přímo v telefonech.

5. ZHODNOCENÍ DOSAŽENÝCH VÝSLEDKŮ

Určitě je několik věcí, co by ještě šlo vylepšit. Například v posledním levelu (Pro Elišku) se noty pohybují tak rychle, že hra nestíhá porovnávat doteky hráče a pozici noty. To by se asi dalo vyřešit podobně jako problém s generací not – vypočítala by se odchylka v závislosti na čas a délku not na ose y.

Hra také měla být původně na platformy iOS. Stále by měla být s platformou iOS kompatibilní, byla jsem ale konfrontován s omezeným přístupem k operačnímu systému macOS, který je nezbytný pro vývoj aplikací iOS pomocí platformy Unity.

Pokoušela jsem se ke spuštění v XCode (vývojové prostředí Applu) využít VirtualBox, vytváření virtuálního stroje s macOS bez přístupu k Macu je ale z hlediska licencování a technických omezení náročné, ne-li legálně neproveditelné.

S designem hry jsem ale spokojena, od názvů levelů, přes animace po průběh levelů. Zaregistrovala jsem posun od původní verze aplikace, kde jsem místo animací používala videa. Také jsem spokojená s tím, jak jsem vyřešila načítání MIDI souborů.

6. INSTALACE

K instalaci na Android telefon stačí stáhnout soubor „piano-tile.apk“. Po spuštění se aplikace sama nainstaluje. Velikost souboru je zhruba 80 MB. Spotřeba paměti se pohybuje okolo 60-70 MB. Po instalaci zabírá v telefonu skoro 170 MB.

Aplikace běží na verzi Androidu 6.0 a novějších verzích, až po Android 12.

7. OVLÁDÁNÍ

7.1. Hlavní menu

Po otevření aplikace se spustí první scéna, a to hlavní menu. V první polovině obrazovky se pouze promítá animace, v druhé půlce se nachází tlačítko na spuštění hry.

7.2. Přehled úrovní

Po kliknutí se scéna změní na přehled levelů. Z toho se můžeme buď přesunout zpátky do hlavního menu (kliknutím na animaci menu), nebo do jednoho z levelů.

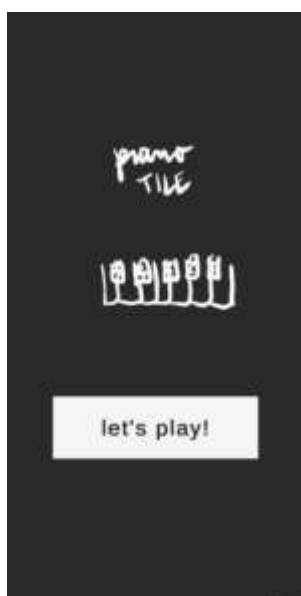
7.3. Level

Po výběru levelu se spustí hudba a začnou se objevovat noty. Cílem hráče je na všechny bílé vyznačené kliknout dříve, než zmizí mimo obrazovku. Dotčené noty změní barvu na šedivou, špatně označené noty změní barvu na černou. Ve vrchní polovině obrazovky se zobrazuje dosažené skóre.

Hra je ukončena v moment, kdy se zastaví chod not a v případě chyby hráče ozve zvuk chyby.

7.4. Obrazovka se skóre

Po konci hry se zobrazí scéna s dosaženým počtem hvězd. Hráč má na výběr mezi opětovným zvolením levelu a vrácením zpátky do přehledu úrovní.



Obrázek 1 - Hlavní menu



Obrázek 2 - Přehled úrovní



Obrázek 2- Level



Obrázek 4- Obrazovka se skóre

8. SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ

- (1) Wikipedie: Otevřená encyklopedie. *Kodek* [online]. [cit. 2024-02-04]. Dostupné z: <https://en.wikipedia.org/wiki/Codec>
- (2) MDN Web Docs. *Web Audio codec guide* [online]. [cit. 2024-02-04]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Audio_codecs
- (3) Wikipedie: Otevřená encyklopedie. *Kontejner* [online]. [cit. 2024-02-04]. Dostupné z: https://cs.wikipedia.org/wiki/Multimedi%C3%A1ln%C3%AD_kontejner
- (4) MDN Web Docs. *Web Audio codec guide* [online]. [cit. 2024-02-04]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Containers>