

Gymnázium, Praha 6, Arabská 14
Obor programování



ROČNÍKOVÁ PRÁCE

Nella Carraro, 4E

Melinoe

březen 2024

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská¹⁴ oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V dne

Nella Carraro.....

Název práce: Melinoe

Autoři: Nella Carraro

Anotace: Cílem projektu bude jednoduchá skákací 2D hra. Hra se bude skládat z několika úrovní, které se hráči odemknou po splnění úrovně předchozí. Cílem jednotlivé úrovně bude dostat se ze startu do cíle. Jednotlivé úrovně se budou postupně ztěžovat V tom mu budou bránit různé překážky, kterým se hráč musí vyhnout. Některé překážky vrátí hráče na start.

1 Obsah

2 Úvod.....	5
3 Použité technologie.....	6
3.1 Python.....	6
3.2 Pygame.....	6
4 Architektura.....	7
5 Třídy.....	8
5.1 Button.....	8
5.1.1 Statické metody Button.....	8
5.2 Scene.....	8
5.2.1 Úprava scén.....	9
5.3 Level.....	9
5.3.1 Skóre a jeho počítání.....	9
5.4 Player.....	9
5.4.1 Pohyb.....	9
5.4.2 Kolize.....	10
5.4.3 Animace.....	10
5.5 Game.....	10
5.5.1 Ukládání.....	11
6 Vzhled.....	12
6.1 Význam a inspirace.....	12
6.2 Hlavní postava.....	14
7 Závěr.....	16
8 Použitá literatura.....	17

2 Úvod

Tento dokument se zabývá vysvětlením ročníkového projektu Melinoe, což je tzv. platform hra, skákací 2D hra. Jednotlivé kapitoly popisují princip projektu, použité technologie, strukturu a algoritmy aplikace. První kapitola vysvětluje pravidla a princip hry.

Hra se bude skládat z několika úrovní, které se hráči odemknou po splnění úrovně předchozí. Cílem jednotlivé úrovně bude dostat se ze startu do cíle, v tom mu budou bránit různé překážky, kterým se hráč musí vyhnout. Některé překážky hráče, pokud se jich dotkne, vrátí zpět na start. Při své cestě by hráč měl také sesbírat tzv. collectibles, předměty nacházející se někde v úrovni. Jejich sebrání, ale není nutné pro dokončení úrovně.

Zajímavost hry, která ji bude odlišovat od ostatních, bude funkce 2 dimenzí. Hráč bude pomocí klávesy shift přepnout mezi jednotlivými dimenzemi. Každá dimenze bude mít jiné rozložení a vzhled. Pro úspěšné řešení úrovně bude muset hráč nejen úspěšně přeskákat přes jednotlivé plošinky, ale nejprve bude muset vymyslet způsob, jak se dostat do cíle pomocí přepínání mezi 2 světy.

3 Použité technologie

V této kapitole jsou popsány technologie použité k vytvoření hry, což jsou programovací jazyk Python, ve kterém je celá hra napsaná, a knihovna Pygame, použitá pro usnadnění vývoje. Další technologie, které stojí za zmínění je vývojové prostředí Visual Studio Code a grafický software Krita, použitý na grafické zpracování hry.

3.1 Python

Python je vysokoúrovňový objektově orientovaný programovací jazyk, který se vyznačuje jednoduchou syntaxí a dobrou čitelností. Python také nabízí rozsáhlou škálu různých knihoven, například knihovnu Pygame, což je jeden z hlavních důvodů proč byl tento programovací jazyk zvolen. Další důvod byla dostupnost informací, Python je jeden z nejpobulárnějších programovacích jazyků dnes, což znamená rozsáhlá dokumentace a podpora pro vývoj.[1]

3.2 Pygame

Pygame je open-source knihovna v programovacím jazyce Python, která usnadňuje vytváření počítačových her. Umožňuje načítání a manipulaci se obrázky a zvuky, kreslení na obrazovku, práci s geometrickými tvary, obdélníky, kruhy, linie apod. Také umožňuje jednoduché pracování s událostmi generovanými uživatelem např. stisknutí kláves nebo kliknutí myši. Tyto vlastnosti pomohly při vývoji k soustředění na tvorbu samotné hry, místo řešení složitostí spojených s nízkoúrovňovou manipulací s grafikou.[2][3]

4 Architektura

Projekt lze rozdělit několika způsoby. V této kapitole se podíváme princip fungování počítačové hry. Hra, velmi zjednodušeně, je několik jednotlivých obrázků daných za sebou tak rychle, že vytváří plynulý obraz, který my jako hráči můžeme nějakým způsobem ovlivňovat. Všechny metody, které zajišťují toto obnovování obrazovky a reakci na stisknutí kláves, se nazývají event loop, neboli smyčka událostí. Jsou to metody s danou strukturou, probíhají neustále dokola a zajišťují interaktivní prostředí pro uživatele.[4]

Před začátkem smyčky se provedou inicializace všech proměnných a tříd v smyčce potřebných. Poté začíná smyčka událostí, kde se nejdříve zkontroluje jakékoliv stisknutí kláves od hráče, provedou se řádné operace a zavolají se metody, které přemístí všechny prvky. Všechny tyto změny se aplikují a vykreslí se na obrazovku a vytvoří 1 snímek. Těchto snímků musí být alespoň 24 za vteřinu, aby to lidské oko vnímalo za plynulé. Tato struktura je použita v několika metodách v programu a zajišťuje tak interaktivnost.[5]

Problém, který nastává v souvislosti s touto metodou, je rozdíl ve výpočetní síle jednotlivých počítačů. Pokud budeme postavu ve hře v každé smyčce pohybovat o 10 pixelů, hráč, jehož počítač zvládne zpracovat 120 snímků za vteřinu, se bude pohybovat 2krát rychleji než hráč, jehož počítač zvládne pouze 60 snímků za vteřinu. Tento problém řeší knihovna Pygame, pomocí třídy *Clock*, které lze nastavit maximální počet obnovení za sekundu. Minimální mez nelze vyřešit tak snadno, jestli počítač nemá dostatečnou kapacitu, nemůžeme mu nařídít vytvářet určitý počet snímků. Řeší se to optimalizací hry, aby jednotlivé operace byly co nejméně náročné.

5 Třídy

Další způsob třídění projektu je podle tříd, kdy lze rozdělit program na 6 různých tříd, každou s jiným účelem. Tato kapitola se věnuje jejich popisu a vysvětlení. Dále také vysvětluje jejich metody a algoritmy.

5.1 Button

Třída `Button` je třída reprezentující obecné tlačítko, ale funguje i jako třída pro nápisy. Její konstruktor potřebuje informace a pozici, barvu a výšku textu a text samotný. Dále obsahuje metody, které jsou schopny všechny tyto atributy změnit. Poslední metoda týkající se pouze jednotlivého tlačítka, je na malování tlačítka na obrazovku.

Metoda `activate()`, která se aktivuje právě tehdy, když je zmáčknuto tlačítko je velmi snadná díky knihovně `Pygame`, problém vznikl v potřebě obecné a zároveň konkrétní metody. Každé tlačítko by mělo něco dělat, ale pro každé tlačítko by to mělo dělat něco jiného. Problém byl vyřešen, že metoda vrací hodnotu `True`, pokud je tlačítko zmáčknuto a co se má stát se definuje v kódu, kde je metoda použita.

5.1.1 Statické metody `Button`

Druhá skupina metod této třídy zahrnuje metody spojené s listy tlačítek. Všechny tyto metody jsou statické a jdou tedy použít bez instance třídy. První z nich, `create_butt_list()` vrací list tlačítek pomocí parametrů stejných jako pro obyčejné tlačítko, až na text, místo něj bere list textů a navíc bere vzdálenost mezi jednotlivými tlačítky.

Metody `create_level_select_buttons()` a `add_level_select_button()` jsou vytvořené pro jednu obrazovku ve hře, vybírání úrovně, jelikož na UI na této obrazovce nelze použít obecnou metodu pro vytvoření listu tlačítek.

5.2 Scene

Třída `Scene`(scéna), reprezentuje 1 část úrovně, konkrétně jednu obrazovku v úrovni. Obsahuje informace o platformách, pastích a sběratelských předmětech. Všechny tyto atributy jsou vytvořeny pomocí `Pygame` třídy `Rect` a jsou přidávány v seznámech. Konstruktor třídy potřebuje pouze seznam platforem a proměnou `next_scene_index`, která

může nabývat hodnot `up`, `down`, `left`, `right` a `none` a určuje směr, ve kterém se nachází další scéna.

Poslední atribut, který třída `Scene` má je *`anim_index`*, který souvisí s metodou *`draw()`*. Tato metoda vykreslí scénu na obrazovku, pozadí a sběratelské předměty, pokud je hráč ještě nesebral. *`Anim_index`* zde slouží k animacím sběratelských předmětů. Princip animaci bude vysvětlen detailně u třídy hráč.

5.2.1 Úprava scén

Třída má také skupinu metod související s vytvářením a upravováním scén, jelikož přidávání a upravování všech atributů scény tvořených třídou *`Rect`* je v programu velmi náročné existují metody, které jsou pro uživatele neviditelné a slouží pro vývojáře k vytváření a editování scén. Třída `Level` má také jednu metodu určenou pouze pro vývoj *`add_scene()`*, která umožňuje přidání scény k existující úrovni.

5.3 Level

Třída `level`, úroveň, se skládá z 2 seznamů scén, kde každý seznam obsahuje scény pro 1 dimenzi. Další atributy, které úroveň má, jsou *`level_unlock`*, proměnná typu `boolean`, která určuje, jestli je úroveň odemknutá, *`dimension`*, která určuje aktivní dimenzi, začínající pozici hráče a vyhrávající pozici hráče.

5.3.1 Skóre a jeho počítání

Každá úroveň má také 2 atributy určující skóre hráče, čas, za který dokončil úroveň, a kolik *`collectibles`*(sběratelských předmětů) sebral. Program si pamatuje 10 nejlepších dvojic pro každou úroveň, a to i po ukončení hry. Skóre jsou v kódu řazené nejdříve podle lepšího času a poté podle počtu získaných sběratelských předmětů. To znamená, že skóre s více sesbíranými předměty a horším časem bude v tabulce umístěno výše než skóre s lepším časem ale méně sesbíraných předmětů. Toto řazení by mělo hráče nutit soustředit se převážně na *`collectibles`* místo času dokončení.[6]

5.4 Player

Třída hráč je potomek třídy *`Sprite`* z knihovny `Pygame`, což je třída, která kombinuje třídy *`Rect`* a *`Image`*. Umožňuje tak pracovat s geometrickým tvarem a obrázkem třídy najednou. Protože je potomek znamená to také, že

musí mít implementovanou metodu *update()*, která zkontroluje kolize, aktualizuje pozici hráče na obrazovce a provede animace postavy.

5.4.1 Pohyb

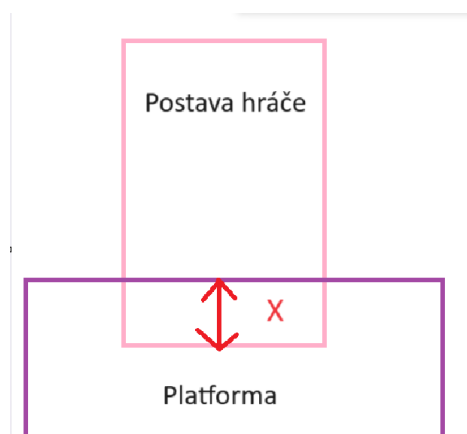
K pohybu má hráč k dispozici 2 způsoby. Chůze zleva a doprava a skok. Pohyb řeší metoda *input()*. V každé smyčce pokud hráč drží klávesu a nebo d posune postavu o 6 pixelů na jednu stranu. Také v tomto případě mění proměnnou *face_side*, pokud by měla hodnotu jinou než aktuální stranu pohybu.

Druhý způsob pohybu, který má hráč k dispozici je skok pomocí mezerníku. Pro tento pohyb jsou důležité proměnné *gravity*, *jump* a *double_jump*. V každé smyčce, kdy hráč nestojí na pevné zemi, se *gravity* zmenší o 0,85 a následně hodnota této proměnné je odečtena od souřadnice y postavy. Skočit hráč může právě tehdy, kdy hodnota *jump* je False a stiskne mezerník. Pak se nastaví hodnota *gravity* na kladných 14 a postava se začne pohybovat nahoru. Stejný princip funguje i pro druhý skok, akorát navíc existuje určitá doba po 1. skoku, kdy hráč nemůže skočit znovu.

5.4.2 Kolize

Třída *Player* má 2 zajímavé metody související s kolizemi *collisions()* a *harsh_collisions()*. Obě metody používají metodu z knihovny *Pygame* pro vyhodnocení kolizí mezi různými geometrickými tvary. Tato metoda je schopná zkontrolovat kolize mezi obdélníkem hráče a seznamem obdélníků. Návrátová hodnota této metody je seznam všech obdélníků, které se s prvním překrývají.

Metoda *collisions()* zajišťuje správné interakce mezi hráčem a podlahou a stěnami. Nejdříve posune postavu o gravitační hodnotu a zkontroluje, zda nedosáhla maximální hodnoty. Pak zkontroluje, jestli existují nějaké kolize mezi hráčem a seznamem pastí, pokud ano, navrátí True hodnotu, v druhém případě zjistí kolize mezi hráčem a platformami. Při kolizi existují 2 možnosti pro kolizi



Obrázek 2: Maximální vzdálenost hráče a platformy

závislou na souřadnici y. Hráč padá na platformu s maximální rychlostí gravitační proměnné nebo hráč skáče do platformy s maximální rychlostí udělené při skoku, z toho vyplývá, že maximální hodnota vzdálenosti x nabývá těchto hodnot. Stačí tedy zkontrolovat vzdálenost x a posunout hráče nad platformu nebo pod platformu. Dále znova zkontrolujeme, jestli existují kolize, a pokud ano provedeme samý proces pro kolize zprava a zleva. Zde maximální vzdálenost x je rychlost pohybu postavy do stran.

Druhá metoda související s kolizemi je metoda *harsh_collisions()*, která je zavolána právě tehdy, když chce hráč změnit dimenzi. Aby byl hráč schopný změnit dimenzi, nesmí se jakákoliv překážka, včetně pastí, z druhé dimenze překrývat s obdélníkem hráče. Metoda vrátí False pokud lze změnit dimenzi.

5.4.3 Animace

Poslední skupina metod důležitých pro třídu Player kontrolují animaci hráče. Postava má ve hře 4 stavy, stání, skákání, padání a chození a pro každý stav 4 snímky(jednotlivé animace). Všechny obrázky se vloží do dvourozměrného pole při vytváření instance třídy. Při každé smyčce se nastaví obrázek zobrazený na pozici v poli určenou proměnnými *anim_index* a *anim_list_index*. Kde *anim_list_index* určuje stav a mění se v reakci na stisknutí kláves od hráče a *anim_index* se v každé smyčce zvětší o 0,1 a do pole se bere jeho int hodnota.

5.5 Game

Třída Game je nejrozsáhlejší ze všech tříd projektu, avšak obsahově není příliš zajímavá. Funguje hlavně jako rozcestník. Vykresluje jednotlivé obrazovky a menu a čeká na další vstupy od hráče. Převážná většina metod má výše zmíněnou strukturu event loop a reprezentuje stav programu například *level_select_menu()*, což je metoda která, vykresluje menu vybírání úrovně a čeká na vstup od hráče.

5.5.1 Ukládání

Nejdůležitější část této třídy jsou metody pro ukládání a načítání hry. K tomuto účelu byl použit modul pickle, který je schopen převést objekty do posloupnosti bajtů a to lze uložit do binárního souboru. Tento soubor jde následovně pomocí modulu

pickle zase načíst a převést na objekt. Při inicializaci této třídy se načte soubor levels, který obsahuje seznam úrovní, a vždy pokud hráč správně ukončí hru se zase uloží zpět do souboru.[7][8]

6 Vzhled

Grafický styl, ve kterém je celý vzhled hry vytvořený, se nazývá pixel art. Jedná se o styl digitální malby, kde se maluje pomocí jednotlivých pixelů, kostiček, na obrazovku. Pochází z 80. a 90. let 20. století, kdy byly velmi limitované možnosti výkonu a paměti počítačů. Byl velmi rozšířený ve starších hrách, a i dnes je často používán.[9]

6.1 Význam a inspirace

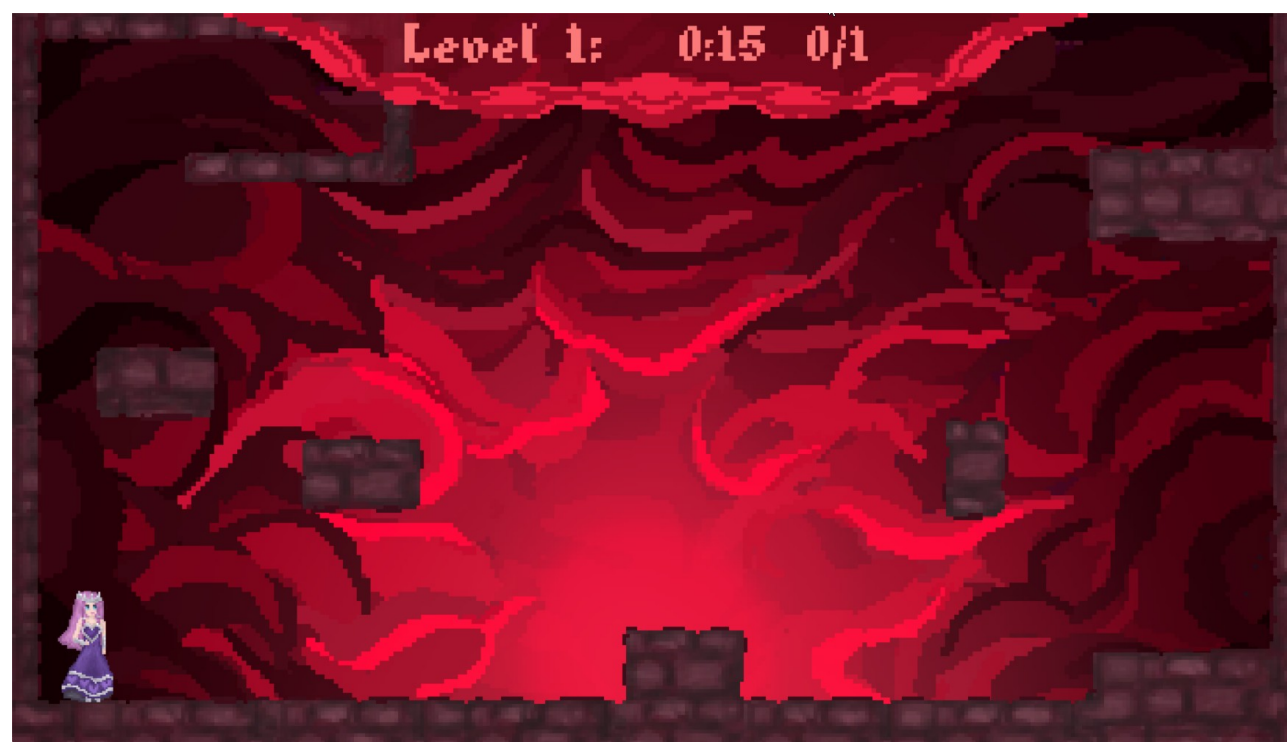
Jméno Melinoe pochází z řecké mytologie. Je to jméno méně známé bohyně, která je označována jako nositelka nočních můr a šílenství. Do dnes nebylo dochováno moc záznamů a informací existuje velmi málo, i přesto známe její rodiče a něco málo o ní samotné.

Její rodiče jsou Zeus, hlavní bůh z Olympu a Persefona, královna podsvětí. Melinoe svým vzhledem reprezentuje jejich dualitu. Polovina jejího těla je černá a druhá polovina je bílá. Ukazuje tak svoji nebeskou část, spojenou s otcem, a pekelnou část, spojenou s matkou a podsvětím, a kontrast svého původu. Často je také spojována s bohyní Hekaté, bohyně čarodějnictví, noci a Měsíce. Melinoe je bohyně, nejen nočních můr a podsvětí, ale i Měsíce, snů a tmy. [10]

Toto vše je zobrazeno ve vzhledu a designu hry. Koncept 2 dimenzí symbolizuje dualitu Melinoe. První dimenze spojená s nocí, sny, spánkem a magií je v kontrastu s druhou dimenzí podsvětí, nočních můr, pekla. Barevně je první dimenze pastelová, příjemná a druhá tmavá, krvavě červená.



Obrázek 3: První dimenze



Obrázek 4: Druhá dimenze

6.2 Hlavní postava

Hlavní postava není Melinoe, ale čarodějnice, která má speciální schopnosti přecházet z jednoho světa do druhého. Při své cestě sbírá bílé kočky, které svým vzhledem ladí k snové dimenzi. Hlavní postava je bezejmenná a její příběh je nedůležitý pro hru. Existuje pouze jako průvodce 2 protikladných světů reprezentující Melinoe.



Obrázek 5: Hlavní postava



Obrázek 6: Bílá kočka

7 Závěr

Cílem projektu bylo vytvořit jednoduchou 2D skákací hru. Toto zadání bylo úspěšně splněno, až na vytvoření několika úrovní. Program samotný je hotový, zbývá ale dodělat chybějící úrovně, na které nezbyl čas, kvůli špatnému plánování. I když hra nebyla úplně dodělána, obsahuje funkce, které nebyly součástí zadání. Například hra má 2 dimenze, mezi kterými hráč musí přepínat, aby dokončil hru, nebo na své cestě může hráč sbírat collectibles. Celkově program je v dobrém stavu, ale určitě není ve finálním stavu. Plány do budoucna obsahují dodělat úrovně, optimalizovat hru a přidat hudbu.

8 Použitá literatura

Bibliografie

- 1: Wikipedie, Python [online], [cit: 01. 04. 2024], <https://cs.wikipedia.org/wiki/Python#Odkazy>
- 2: Wikipedie, Pygame [online], [cit: 01. 04. 2024], <https://cs.wikipedia.org/wiki/Pygame>
- 3: Pygame, Dokumentace Pygame [online], [cit: 01. 04. 2024], <https://www.pygame.org/docs/#>
- 4: Clear Code, Pygame tutorial, [cit: 01. 04. 2024], <https://www.youtube.com/watch?v=AY9MnQ4x3zk&t=272s>
- 5: Wikipedie, Snímková frekvence [online], [cit: 01. 04. 2024], https://cs.wikipedia.org/wiki/Snímková_frekvence
- 6: roippi, Sort multidimensional array by column, [cit: 02.04.2024], <https://stackoverflow.com/questions/20183069/how-to-sort-multidimensional-array-by-column>
- 7: Al Lukaszewski, How to Use Pickle to Save Objects in Python, [cit: 02. 04. 2024], <https://www.thoughtco.com/using-pickle-to-save-objects-2813661>
- 8: Dionysia Lemonaki, How to Check if a File Exists in Python with isFile() and exists(), [cit: 02.04. 2024], <https://www.freecodecamp.org/news/how-to-check-if-a-file-exists-in-python/>
- 9: Wikipedie, Pixel art, [cit: 01. 04. 2024], https://cs.wikipedia.org/wiki/Pixel_art
- 10: Wikipedie, Melinoë, [cit: 01. 04. 2024], <https://en.wikipedia.org/wiki/Melinoë>
- 11: PSF , Python documentation, [cit:02. 04. 2024], <https://docs.python.org/3/>