Lukáš Kondiolka

Dokument popisující můj projekt Prehistorik do PJ1

Prehistorik

Lukáš Kondiolka KON0355

Ovládání

W – skok

A – doleva

D – doprava

SPACE – útok

Popis hry

Cílem je nasbírat co nejvíce bodů a přežít. Body hráč získává za zabíjení opic a sbírání mincí s obrázkem jídla. Pokud hráč nasbírá všechny možné body v daném herním levelu, hra se sama ukončí a zapíše jeho skóre a případně i jméno (defaultně „player“) do textového souboru s názvem leaderboards. Hráč má na začátku hry 10 životů, které se mu odčítají, pokud ho trefí opice. Opice mohou být nebezpečné, někdy hráči uberou jeden život, někdy třeba všech 10. Hráč získává 5 životů navíc pokud sebere minci.

Funkčnost hry

Po spuštění hry se načte z textového souboru level, který lze i modifikovat, jednotlivé objekty jsou popsány v souboru levelEditorTutorial. Skóre po ukončení hry je zapsáno do souboru leaderboards vzestupně. Celou logiku hry řeší třída Game a její metody. Vytvoření instance třídy Game a samotné spuštění je řešeno ve třídě PlatformerGame.

Bugy

Hra vypíše do konzole chybové hlášení, pokud je jakákoliv opice zabita, kvůli manipulování s daty během procházení kolekce. Hra zde ale nepadá a pokračuje dál. Dále útok hráče není nějak vizualizován, hráč musí přijít blízko k opici a stačí několikrát pomačkat SPACE a opice zemře. Bohužel se mi nepovedlo nějak rozumně vytvořit animaci.

Popis jednotlivých tříd a metod

Neuvádím zde gettery a settery pokud nemají nějakou další funkčnost.

Třída PlatformerGame

Tato třída řeší spuštění samotné hry a vytvoření objektů které potřebuje třída Game pro vytvoření svojí instance.

Třída Game

Tato třída řeší veškerou logiku hry.

**FUNKCE**

**1)initContent**

V této třídě se inicializuje veškerý obsah nutný pro hru. Obrovský switch zde řeší jednotlivé znaky z instance třídy LevelData a přidává podle nich do jednotlivých kolekcí dané entity. Dále se zde inicializuje kamera a prvky UI.

**2)initLostEndGame**

Inicializuje prvky, které se zobrazí, pokud hráč prohrál.

**3)initWinEndGame**

Inicializuje prvky, které se zobrazí, pokud hráč vyhrál.

**4)initEndGame**

Inicializuje prvky, které se zobrazí po skončení hry i pokud hráč vyhrál nebo prohrál. Dále řeší kliknutí na button pro ukončení hry, při kterém volá metodu writeScore.

**5)update**

Řeší jednotlivé USE CASY – logika hry, co se stane, pokud hráč stiskne to nebo to. Dále řeší gravitaci hráče a opic.

**6)movePlayerX**

Řeší pohyb a kolizi hráče po ose X.

**7)movePlayerY**

Řeší pohyb a kolizi hráče po ose Y.

**8)collectCollectible**

Řeší sebrání mince hráčem. Volá se při kolizi hráče s mincí.

**9)jumpPlayer**

Řeší vyskočení hráče.

**10)moveMonkeyY**

Řeší pohyb opic po ose Y.

**11)monkeyJump**

Řeší vyskočení opice a rotaci při skoku.

**12)attackPlayer**

Řeší útok hráče jak zleva, tak i zprava.

**13)monkeyDie**

Řeší smrt opice, pokud jí dojdou životy.

**14)writeScore**

Řeší zapisování výsledného skóre hráče po ukončení hry.

**15)getAllNames**

Tato funkce vypíše do konzole všechna jména hráčů z leaderboards. Funkce není nějak zvlášť používána. Je zde spíše pro pokrytí požadavku na práci s agregovanými operacemi.

Třída GameObject

Třída popisující základní vlastnosti všech objektů ve hře.

Třída RectangleObject

Tato třída dědí ze třídy GameObject a přidává vlastnosti nutné pro práci s objekty obdélníkového tipu (platformy, hráč, monkeys, …).

Třída CollectibleObject

Tato třída dědí ze třídy GameObject a přidává vlastnosti nutné pro práci s objekty kruhového tipu (collectibles).

Třída LevelData

Instance této třídy reprezentuje level.

**FUNKCE**

**1)levelFromFile**

Funkce řešící nahrání dat o levelu ze souboru.

Třída LeaderboardsLine

Instance této třídy reprezentují jednotlivé řádky v souboru leaderboards.

Třída LeaderboardsLine

Tato třídá řeší inicializování a práci z UI. Její funkce nastavují text ve ScoreBaru a HealthBaru.