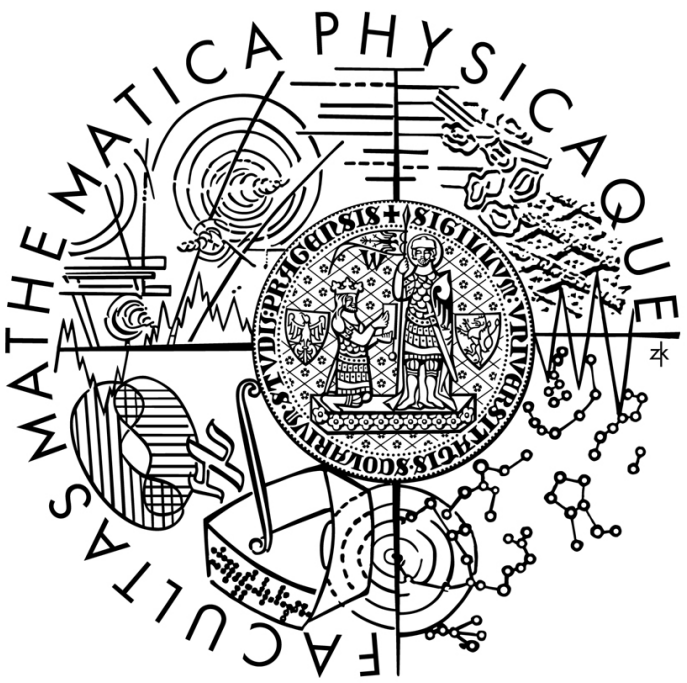

Univerzita Karlova
Matematicko-fyzikální fakulta

Objektově orientované programování - ZS 2013/2014

MathBOY

Dokumentace k zápočtovému programu

Vypracovali: Ondřej Bouchala
Martin Petrla

20. ledna 2014

Obsah

1 Specifikace

V jazyce C# naprogramovat počítačovou 2D plošinovku pro jednoho hráče obsahující:

- úvodní menu (nová hra, nastavení, highscore tabulka, exit)
- animovaná postavička ovládaná pomocí klávesnice
- animované nepřátelské postavičky se ZÁKLADNÍ umělou inteligencí
- dlaždicově navrhované levely načítané ze souboru
- základní fyzikální prostředí, ve kterém se hra odehrává (předměty těžší než vzduch budou padat k zemi), řešení kolizí
- počítání score
- zvukové efekty

2 Popis hry

Hráč ovládá postavičku jménem „MathBOY“. Cílem každé úrovně je dostat se na její pravý okraj. Po cestě je třeba se vypořádat s různými nástrahami, mezi které patří „Apolloniovy kruhy“, zákeřní „Trisecktoři“ nebo „Iracionální zrůdy“, ale také „nepřeskočitelné dálky“ nebo „příliš vysoké plošiny“. Za každou zabitou příšeru je hráč patřičně odměněn. Stiskem escape se hráč dostane do menu, kde je možné přepnou fullscreen, vypnout/zapnout hudbu nebo se podívat na highscore.

3 Program (popis tříd a objektů)

3.1 Game1

`Game1` je základní třídou celého programu. Jejím základem je pár tříd, které volá samotné XNA, a sice `Update` a `Draw`.
Tato třída má vcelku dost veřejných proměnných, které jsou společné pro celou hru, jako například ve kterém levelu jsem, jaké je současné rozlišení a podobně.

3.1.1 LoadContent

Na začátku hry (a pak manuálně při začátku nového levelu) se spustí funkce `LoadContent`. V té se vytvořím „já“ (to jest načtou se všechny textury), vytvoří se pozadí (vytvoří se instance třídy a načtou se textury), a konečně se vytvoří třída `zoo`, do které přibudou všechny příšery daného levelu.

3.1.2 Update

Tuhle třídu volá samotné XNA, a je to posun o jeden krok dopředu. Tato funkce se volá (velmi zhruba) 50 krát za sekundu (v závislosti na výkonu a vytíženosti počítače). Jako parametr dostanu `gameTime`, ze kterého vyčtu, kolik milisekund uběhlo (a tedy o kolik se mám pohnout). Tato třída (pokud nejsem v menu) pohlídá, zdali nebyla zmáčknuta klávesa `Esc` (jinak by se přešlo do menu), pokud ne tak zavolá funkce `Update` postavičce, všem příšerkám (prostřednictvím třídy `zoo`). Pokud jsem v menu, pak zavolá funkci třídy `Menu`, a sice `Update`.

3.1.3 Draw

Tuto třídu opět volá XNA, a slouží k překreslení obrazovky. Buď zavolá `Draw` třídy `Menu` (pokud jsem v menu), a nebo třídy `Draw` třídy pro postavičku, třídy s pozadím a třídy `zoo`.

3.1.4 newgame

Tahle třída slouží pro start daného levelu od začátku, a náhodnou volbu písničky.

3.2 Menu

Základem menu je třída `Menu`. Nejdůležitějším prvkem je výčtový typ `KtereMenu`, a veřejná proměnná tohoto typu (ve třídě `Menu`) `ktereMenu`. Ta říká, ve které části menu se právě uživatel nachází a tedy která tlačítka se mají zobrazit.
Třída `Menu` je vytvořená ve výchozí třídě `Game1`, a sama využívá tříd `Button` a `label`.

3.2.1 Menu

Konstruktor `Menu` bere za parametr pointer na svého předchůdce (to jest třídu `Game1`. To je hlavně pro to, že informace o rozlišení jsou uloženy právě v této třídě, a mohou se v průběhu běhu programu změnit. Dále konstruktor vytvoří spousty tlačítek (instancí třídy `Button`). Každé dostane pointer na menu (aby mohlo například zavolat funkci po zmáčknutí tlačítka), ID (které jim řekne které v pořadí je (a tedy kde se má vykreslit), dále dostane informaci kdy se má vykreslit (pomocí `KtereMenu`) a konečně string s tím, co má na něm být napsané (přesněji název souboru s obrázkem s texturou, která obsahuje ten text).
Třída `Menu` má, stejně jako skoro všechny naše třídy, funkci `update`. Ta se volá z výchozí třídy (pokud se menu má zobrazit). Ona funkce dělá několik věcí. Zkontroluje, zdali uživatel náhodou nezmáčkl šipku nebo enter, v kterémžto případě by změnila vybrané tlačítko (popřípadě by ho stiskla). A konečně zavolá funkci `Update` všech tlačítek a popisků.
Dále má třída `Menu` funkci `clicked`. Tu zavolá tlačítko, když bylo stisknuto. Na základě toho, která obrazovka menu je zobrazena a které tlačítko bylo stisknuta se vykoná daná akce.
A konečně je tam funkce `Draw`. Ta vykreslí pozadí menu a zavolá funkci `Draw` všech tlačítek.

3.2.2 Button

Třída pro tlačítko je veskrze jednoduchá. Ve funkci update se zkontroluje, zdali není myš nad tímto tlačítkem (pak by se změnila textura), popřípadě jestli ono tlačítko nebylo stisknuto. A ve funkci `Draw` se vykreslí (pro pozici a velikost si šáhne do třídy menu a navíc ví, které v pořadí v daném menu je)

3.2.3 Label

Třída `Label` slouží pro zobrazení highscore. Na rozdíl od třídy `Button` není třeba ověřovat polohu myši. Ale jména hráčů nejsou natvrdo napsaná v texturách, takže je na ně použita ještě třída `Text`.

3.2.4 Text

Tato třída slouží pro uložení a zobrazení textu na obrazovku. Jejím hlavním důvodem je, aby se text mohl zobrazit v různých rozlišeních fontem různé velikosti. Konstruktor dostane kromě pozice, velikosti a chtěného textu také pointer na hlavní třídu hry, aby mohl načítat obrázky (pozadí pod text).

3.3 Textury postaviček

Postavičky (to jest já nebo nepřítel) mají za texturu sérii obrázků, jsou tedy animované (nyní je jedno, že jich může být několik druhů v závislosti na tom, zdali jsem například ve vzduchu nebo na zemi). Na to slouží třída `AnimatedSprite` a `AnimatedSpriteHead`. Rozdíl mezi nimi je ten, že „já“ mám ve hře hlavu a tělo animované zvlášť (z několika důvodů, například aby bylo snažší měnit hlavu v závislosti na levelu, nebo aby hlava a tělo mohly mít jiný počet snímků). Nyní budu popisovat jen `AnimatedSprite`, druhá třída je zcela analogická s přidaným bonusem hlavy, která je tam zvlášť (je tam navíc ještě vrstva s rukou, jelikož třeba za letu musí být ruka nad hlavou a hlava nad tělem).

3.3.1 AnimatedSprite

Konstruktor této třídy dostane texturu, což je tabulka `rows`×`columns` obrázků (její rozměry konstruktor dostane taky).
Z funkcí stojí za zmínku `Update`, která přejde k dalšímu snímku, `stop`, která se vrátí na začátek posloupnosti snímků (když se zastavím, tak mám stát a ne být v půlce kroku)[1], a dále klasická dvojice funkcí `Update` a `Draw`. V `Update` se pouze přejde k dalšímu obrázku v posloupnosti (a je na tom, kdo vyrobil instanci této třídy aby se rozhodl kdy to chce dělat). A v `Draw` se vybere z `Texture` správný obdélník který se vykreslí na dané souřadnice a danou velikostí (dané jako parametry funkce `draw`).

3.4 Postavička

Objekt „ME“ reprezentuje MathBOYe, v jeho funkci Update jsou implementovány instrukce pro chůzi vpravo, vlevo a výskok.

3.5 Enemy

Tato třída je odvozena od třídy `Postavicka`. Má jinou funkci Update, která rozlišuje mezi jednotlivými typy nepřátel a obsahuje instrukce pro jejich chování. Současně všechny objekty na tuto třídu vidí a mohou se skrz ni dívat na ostatní objekty ve hře.

3.6 Kolize

Rozlišujeme mezi dvěma typy kolizí

- kolize MathBOYe s prostředím,
- kolize MathBOYe s nepřáteli.

První zmíněné obstarává sám MathBOY. MathBOY vidí na objekt `Background`, tedy pozadí. Při každém volání své funkce `Update` se podívá na dlaždice, které se nachází pod ním. Rozlišuje mezi dlaždicemi, na kterých může a nemůže stát. Pokud se nachází nad dlaždicí, na které nemůže stát, potom je ve skoku a řídí se instrukcemi pro skok (ty zahrnují i pád).

Druhé zmíněné kolize obstarávají nepřátelské postavičky. Ty vidí na objekt MathBOYe a porovnávají jeho polohu s polohou svou. Pokud se obdélníky reprezentující tyto objekty překrývají, vykonají se instrukce pro kolizi MathBOY - nepřítel. Obecně se dá říci, že pokud MathBOY skočí na nepřítele shora, pak umírá nepřítel, ve všech ostatních případech umírá MathBOY. To platí pro všechny nepřátelské jednotky.

[1] Ve třídě `AnimatedSpriteHead` se vrátí na začátek pouze tělo, hlava se pohybuje dál i když panáček stojí.

3.7 Soubory

V souborovém systému jsou dvě hlavní složky, `The Game` a `The GameContent`. V první jsou soubory s kódem, který je do těchto souborů rozdělen víceméně dle tříd.
A ve složce `The GameContent` se nachází převážně textury a jiné mediální soubory. Ty jsou opět rozděleny do podsložek, podle toho kdy se mají zobrazit, jestli například v menu nebo v některém z levelů.
Z těchto souborů si zaslouží komentář soubor `l1.txt`, popřípadě `l?.txt`, kde `?` je číslo levelu. V tomto souboru je uložena informace o tom, jak bude daný level vypadat, to jest kde budou různé plošiny a různí nepřátelé. Tento soubor se při kompilaci zkopíruje do adresáře kde je příslušný binární soubor, a není tedy nutné po změně tohoto souboru kompilovat hru znovu.
K jeho struktuře:
Předpokládá se, že každý řádek bude mít stejně symbolů, na základě tohoto počtu se určí šířka dané úrovně. V kódu je napevno zadané, že tento soubor má mít šest řádlů, a tedy že i level bude mít šest „řádků“. Pro určení významu symbolů slouží tato tabulka:

.	prázdné pole
X	Dolní cesta, např. most
L	Levý konec dolní cesty (mostu)
R	Pravý konec dolní cesty (mostu)
x	plošina
l	levý konec plošiny
r	pravý konec plošiny
1, 2, ...	příšera typu 1, 2, ...

4 Testování

Hra byla řádně testována jejími tvůrci a jejich kamarády na systému Windows 7 Ultimate 64, na kterém projevila plynulý chod a slušné vychování, tedy ovládání se ukázalo jako intuitivní a hra za žádných okolností nepadala.

5 Závěr

Ondřej `Bouchala`: Tuto hru považuji za vcelku vydařený projekt, bylo fajn naučit se o úlohách a nástrahách 2D her myslet objektově, což se ukázalo jako účinné.
Bylo také zajímavé zkusit si programování ve více lidech.

Martin `Petrla`: Na programování MathBOYe jsem se naučil pracovat s platformou XNA Game Studio a shledávám ji jako perfektní vývojový nástroj 2D her, se kterým si ve volném čase užiji ještě hodně zábavy.