

## 1 Uživatelská příručka

### 1.1 Spustenie

Aplikácia bola vyvíjaná vo Visual Studio 2019. Využíva MonoGame framework verzie 3.7.1, ktorý sa dá stiahnuť na oficiálnej stránke: <http://www.monogame.net/downloads/>. Po nainštalovaní MonoGame 3.7.1 stačí otvoriť súbor Bomberman.sln vo Visual Studio a spustiť pomocou klávesy F5.

### 1.2 Hra

Bombberman je hra jedného hráča. Ústrednou postavou je hrdinka, ktorá sa pohybuje po mriežke. Cieľom je dostať sa k cieľovej zástavke a prekonať všetky úrovne.

#### 1.2.1 Hrdinka



Hrdinka sa počas hry nachádza v strede obrazovky. Pohybuje sa stlačením šípok na klávesnici. Pri stlačení medzerníka položí bombu na mieste, kde sa nachádza.

#### 1.2.2 Mriežka

Dej hry prebieha na mriežke. Sú 3 typy blokov.

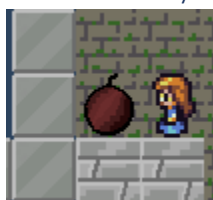


Podlaha – tu sa môže pohybovať hrdinka, príšery a pokladať bomby.

Stena – zastaví explóziu bomby, nič cez ňu neprejde.

Tehly – správajú sa ako stena. Môžu byť zničené bombou, potom sa zmenia na podlahu.

#### 1.2.3 Bomby



Bomba sa pokladá stlačením medzerníka. Najprv sa objaví pod hrdinkou, ktorá môže prejsť na susedné políčko. Následne už cez bombu prejsť nemôže. Po niekoľkých sekundách bomba vybuchne. Explózia sa šíri v 4 smeroch. Dosah explózie je zozatiaku 1 políčko vo všetkých smeroch, môže byť zvýšený bonusom. Explóziu zastaví stena alebo tehly, ktoré sa následne zmenia na podlahu a so šancou 20% zanechajú náhodný bonus. Explózia zoberie hrdinke a príšerám 1 život. Počet bômb, ktoré môžu byť položené v jednom momente, je obmedzený. Môže byť navýšený bonusom.

#### 1.2.4 Stavový riadok



Stavový riadok obsahuje 3 informácie. Každá ikonka vyplneného srdca predstavuje jeden život hrdinky. Počet prázdnych srdiec značí koľko životov si môže doplniť pomocou bonusov. Počet ikoniek bomby znázorňuje, koľko bômb je v súčasnosti k dispozícii. Položením bomby sa počet zvýši o 1 a vybuchnutím bomby sa zvýši o 1. Zobrať modrého bonusu sa taktiež zvýši o 1.

### 1.2.5 Příšery



Existujú dva druhy príšer – Skeleto a Spider. Pohybujú sa samostatne, s podobnými pravidlami ako hrdinka. Nedokážu pokladať bomby. Skeleto sa pohybuje pomalšie ako hrdinka ale, rýchlejšie ako Spider. Príšery majú 1 život. Stret s hrdinkou spôsobí zranenie hrdinke aj príšere. Pohyb Spidera závisí od okolitej mriežky a bomby. Skeleto sa snaží ísť rovno, inak zatáča vľavo. Ak vidí na hrdinku, ide za ňou.

### 1.2.6 Bonusy



Bonusy sa môžu objaviť po zničení tehál. Konkrétny efekt bonusu sa aplikuje pri strete hrdinky s ikonou bonusu. Efekty bonusov sú nasledujúce:

- Modrý: zvýšenie množstva bômb o 1
- Červený: pridanie 1 života
- Biely: zvýšenie dosahu explózie bomby o 1
- Zelený: zvýšenie rýchlosti pohybu

Bonusy trvajú po dobu jedného levelu alebo kým hrdinka nezomrie.

### 1.2.7 Pribeh hry



Hra pozostáva z levelov. Cieľom je úspešne ukončiť každý level. Level je úspešne ukončený pri dotyku hrdinky s cieľovou zástavkou. Potom sa okamžite načíta nasledujúci level. Level je neúspešne ukončený, keď počet životov hrdinky klesne na nulu. Vtedy je level reštartovaný. Príšery sa objavia na pôvodných pozíciach, hrdinka má 3 životy a stratí všetky bonusy.

## 2 Programátorská príručka

Vstupným bodom hry je trieda **Game1**. Na začiatku sa volá metóda **LoadContent**, potom sa opakovane volajú metódy **Update** a **Draw** (to zabezpečuje MonoGame). V metóde **Update** sa vykonáva herná logika, metóda **Draw** hru vykresľuje. Väčšina ostatných komponent používa rovnaký princíp **Update** a **Draw** metódy, pričom je zodpovedná za volanie **Update** a **Draw** podriadených komponent.

**Game1** načíta informácie o leveloch z konfiguračného súboru *Config/levels.txt* pomocou modulu **Parser**, zvukové efekty a textúry. Vytvorí inštancie základných častí hry – **World**, **StatusBar** a **Background**.

### 2.1 World

**World** predstavuje herný svet. Jeho základné komponent tvoria **Charactor** (hrdinka), **Monsters** (príšery), **Effects** (efekty), **Grid** (mriežka). Úlohou triedy **World** je zabezpečiť volanie **Update** a **Draw** metód, zabezpečiť niektoré interakcie medzi objektami, premazávať mŕtve príšery a efekty a poskytuje metódy pre vkladanie týchto objektov do sveta.

#### 2.1.1 Grid

Mriežka má obdĺžnikový tvar. Skladá sa z blokov. Každý blok zapíňa jeden **Sector**. **Grid** je dátová štruktúra pre mriežku navyše so schopnosťou vykresliť ju do herného sveta. Trieda **Sector** popisuje súradnice

jedného sektoru – má X a Y zložku. Navyše obsahuje ďalšie pomocné metódy na výpočet susedných sektorov, porovnanie sektorov a prevod sektoru do herných súradníc a naspäť.

### 2.1.2 Actor

Trieda **Actor** predstavuje postavu v hre – príšeru alebo hrdinku. Jej základné komponenty sú **WalkingSprite** a **Controller**. Odvodené triedy ako **Character**, **Spider** a **Skeleton** len nastavujú alebo pridávajú niektoré vlastnosti.

#### 2.1.2.1 AnimatedSprite

Vykresľuje jednu postavu na zvolené miesto. Zavádza koncept pohybu – orientácie postavy a snímky animácie. V metóde **Update** počíta počet volaní, z toho odvodzuje ktorý snímok animácie sa má vykresliť.

#### 2.1.2.2 WalkingSprite

Dedí z **AnimatedSprite**, rozširuje ho o pohyb v mriežke. V každom momente je postava buď v pokoji alebo je procese pohybu medzi zdrojovým sektorom a nejakým jeho susedným pohybom. Metódou **Walk** sa začne pohyb v smere danom parametrom. Túto metódu využíva najmä **Controller**.

#### 2.1.2.3 Stat

Pomocná dátová štruktúra, ktorá uchováva jednu hodnotu s jej základnou hodnotou, minimálnou hodnotou, maximálnou hodnotou a veľkosťou kroku.

#### 2.1.2.4 Charactor

Hrdinka. Dedí z triedy **Actor**. Rozširuje ju o nejaké vlastnosti ako polomer explózie, počet dostupných bômb a kapacita.

#### 2.1.2.5 Skeleton

Nastavuje vlastnosti **Skeletona**.

#### 2.1.2.6 Spider

Nastavuje vlastnosti **Spidera**.

### 2.1.3 Controller

**Controller** je abstraktná trieda, z ktorej dedia konkrétne ovládače príšer a hrdinky. Jej úlohou je čítať vstup alebo informácie z mriežky a na základe toho v metóde **Update** riadiť pohyb postáv, prípadne kladenie bomby. Taktiež je to miesto, kde sa logicky prepája mriežka s postavami – rozhoduje o tom, kam sa daná postava môže pohybovať a kam nie.

#### 2.1.3.1 PlayerController

Ovládanie hrdinky – čítanie vstupu s klávesnice.

#### 2.1.3.2 ChaseCharactorController

Ovládanie **Skeletona**. Algoritmus pohybu je nasledujúci:

- 1 Ak sa hýbe bezprostredne smerom k bombe, otočí sa o 180 stupňov.
- 2 Ak má voľný výhľad hrdinku, začne sa pohybovať smerom k nej.
- 3 Inak: Ak je pred ním voľno, tak ide dopredu.
  - Inak: Otočí sa vľavo. Ak je pred ním voľno, ide dopredu. Ak nie je pred ním voľno a neskúsil všetky smery, skočí na krok 3.

### 2.1.3.3 *WalkByWallsController*

Ovládanie **Spidera**. Algoritmus pohybu je nasledujúci:

- 1 Ak sa hýbe bezprostredne smerom k bombe, otočí sa o 180 stupňov.
- 2 Ak je napravo voľno, choď vpravo.
- 3 Inak: Ak je pred ním voľno, tak ide dopredu.
  - Inak: Otočí sa vľavo. Ak je pred ním voľno, ide dopredu. Ak nie je pred ním voľno a neskúsil všetky smery, skočí na krok 3.

### 2.1.4 *Effect*

**Effect** je objekt v hre, ktorý sa môže v priebehu hry objaviť v hernom svete a spôsobiť niečo po uplynutí časového intervalu alebo pri kolízii s hrdinkou alebo príšerou. Nasledujúce triedy dedia z triedy **Effect**.

#### 2.1.4.1 *Bomb*

Položená bomba. Jej pridanie do herného sveta spôsobí **PlayerController**. Po uplynutí časového intervalu vytvorí explózie a prípadne rozbije tehly. Postavy cez neho nemôžu prechádzať – správa sa ako stena.

#### 2.1.4.2 *Explosion*

Explózia po vybuchnutí bomby. Postavy ňou môžu prejsť ale pri kontakte s postavou jej uberie 1 život. Ak touto explóziou bola zničená bomba, zavolá metódu **World.MaybeSpawnRandomPickup**.

#### 2.1.4.3 *Pickup*

Bonus. Pri kontakte s hrdinkou sa aplikuje a zmizne. Na príšery sa neaplikuje.

#### 2.1.4.4 *Finish*

Cieľová zástavka. Pri kontakte s hrdinkou zmení stav levelu na úspešne ukončený. Príšery cez neho môžu prechádzať.

## 2.2 *UI*

### 2.2.1 *StatusBar*

Stavový riadok. V každom volaní **Update** si z inštancie **World** aktualizuje hodnoty. Pomocou **Draw** vykreslí celý stavový riadok.

### 2.2.2 *Background*

Vykreslí hviezdnu oblohu a mesiac v pozadí. Ako parameter berie vektor posunutia sveta, ktorý vynásobí rôznymi hodnotami pre mesiac a oblohu, čím vytvorí dojem hĺbky scény.

## 2.3 *Parser*

Jednotlivé levely sú načítané z konfiguračného súboru. Ten pre každý level obsahuje rozmery mriežky, konkrétne bloky mriežky, miesto začiatku hrdinky, miesto cieľovej zástavky, príšery, a sprievodné texty.

### 2.3.1 *ConfigReader*

Trieda na čítanie zo súboru. Obsahuje pomocné metódy na porovnávanie riadkov s regexom.

### 2.3.2 *ParsedMonster*

Dátová štruktúra - predstavuje jednu načítanú príšeru – teda jej typ a miesto, kde sa objaví.

### 2.3.3 ParsedLevel

Dátová štruktúra - predstavuje jeden načítaný level. Zahŕňa mriežku, načítané príšery, sektor začiatku, sektor cieľovej zástavky a sprievodné texty.

### 2.3.4 Texts

Obsahuje sprievodné texty načítané pre nejaký level a metódu na ich vykreslenie.

### 2.3.5 LevelLoader

Sprostredkuje výsledné rozhranie na načítanie levelov a vytvorenie konkrétnych herných objektov.

#### 2.3.5.1 Konfiguračný súbor

Konfiguračný súbor sa nachádza v relatívne k spustiteľnému súboru na mieste *./Config/levels.txt*. Obsahuje postupne popisy prvého a n-tého levelu. Je načítaný raz pri spustení hry.

#### 2.3.5.2 Formát konfiguračného súboru

Popis levelu začína riadkom:

*LEVEL n*

Kde *n* je konkrétne číslo levelu.

Nasleduje riadok:

*GRID w h*

Kde *w* je šírka mriežky a *h* je výška mriežky.

Nasleduje *h* riadkov po najvyšš *w* znakov obsahujúcich znaky 0, 1, 2 a *medzera*. Popisujú jednotlivé riadky mriežky. Význam znakov je nasledujúci:

- 0 – podlaha
- 1 – tehly
- 2 – stena
- *medzera* – prázdne miesto

Ak má riadok menej ako *w* znakov, efektívne sa doplní medzerami.

Nasleduje 0 a viac riadkov popisujúcich príšery:

*TYP\_PRISERY x y*

Kde *TYP\_PRISERY* je buď *SPIDER* alebo *SKELETON*. *x,y* sú súradnice sektoru, na ktorom sa daná príšera objaví.

Nakoniec nasleduje 0 a viac riadkov popisujúcich texty:

*TEXT x y CONTENT*

Kde *x, y* sú súradnice sektoru, kde sa začne vypisovať text. *CONTENT* je konkrétny text, ktorý sa vypíše. Povolené sú takmer akékoľvek znaky okrem znaku nového riadku.

Poznámka k obrázkom:

Autorkou obrázkov postáv a pozadia je Lanea Zimmerman (<https://opengameart.org/users/sharm>).