

Hra Bomber Street

Anotace

Hra Bomber Street je koncipována pro jednoho hráče, jehož cílem je porazit všechny nepřátele. Hráč vítězí v případě, že všichni nepřátelé ztratili všechny své životy. Naopak pokud dříve hráč ztratí všechny své životy, hru prohrává. Po vítězství následuje postup do další úrovně s jinou herní plochou, zatímco při prohře lze pouze znovu spustit aktuální level. Postavy hry mají 2 základní režimy: režim pohybu a režim umísťování bomby. U hráče se mezi těmito režimy rozlišuje na základě uživatelského vstupu, zatímco u nepřátel jsou oba režimy (a taktéž samotný výběr režimu pro aktuální kolo hry) řízeny náhodným generátorem. Uživatel ovládá pohyb hráče prostřednictvím klávesnice – pohyb je řízen pomocí šipek a umísťování bomb stiskem klávesy mezerníku.

Algoritmus

Hra je realizována na principu programování řízeného událostmi. Zde za jednotlivé události považujeme uživatelský vstup, tedy stisknutí příslušných kláves, případně spuštění nové hry či ukončení programu prostřednictvím komponent tlačítek. Pokud není stisknuta žádná klávesa, postava hráče se nepohybuje, ani neprovádí jinou akci. V takovém případě ovšem není zbytek hry nečinný, neboť neustále iterujeme hlavní smyčkou (dokud je spuštěn časovač, který se vypíná při změně stavu z probíhající hry na výhru či prohru) a při každé iteraci kontrolujeme, zda některý z prvků hry nemění svůj stav. Tedy zda se nepohybuje na jiné místo na herní ploše, neumísťuje bombu, nemění obsah herní plochy ve svém okolí, případně jestli se na herní ploše neobjevuje nový prvek. Tyto změny se zaznamenávají do matice reprezentující herní plochu a po každé iteraci se tato matice nově vykreslí na základě textur odpovídajících aktuálním symbolům v matici.

Postavy nepřátel se pohybují v principu stejným způsobem jako samotný hráč, nicméně jejich pohyb není řízen uživatelem, ale náhodným generátorem. Pomocí něj se rozlišuje, kterým směrem (a zda vůbec) se nepřítel v daném kole pohne, případně jestli raději zkusí na herní plochu umístit bombu. Náhodný generátor je využit i v další části hry, která se stará o umísťování různých vylepšení na herní plochu. Opět se náhodně rozhoduje, jestli na herní plochu umístit nové vylepšení a pokud ano, tak na jakou pozici a o které vylepšení se bude jednat.

Program

Program je aplikací Windows Forms a je napsán v jazyce C#. Struktura programu je rozdělena do více jednotlivých souborů dle šablony aplikace Windows Forms. Ovládací komponenty typu tlačítek či časovače byly vytvořeny prostřednictvím režimu návrháře. Klíčovými soubory, v nichž byly prováděny manuální úpravy, jsou *Form1.cs* a *Bomber.cs*.

Form1.cs

V souboru *Form1.cs* jsou definovány metody třídy **Form1**, konkrétně metody udávající, co se má změnit při stisku tlačítek *button1* nebo *button2* (spuštění nové hry nebo ukončení aplikace), dále je zde metoda související s časovačem, která zohledňuje stavový diagram hry. Tato metoda mění viditelnost jednotlivých komponent na základě aktuálního stavu a stará se o hlavní herní smyčku. Ve stavu rozehrané hry volá metodu **AktualizujPlochu**, která simuluje 1 „kolo“ hry – zaznamenává změny na herní ploše, které proběhly ve stejný okamžik (pohyb hráče, nepřátel, sbírání vylepšení, výbuchy bomb a podobně). Bezprostředně po zavolání metody **AktualizujPlochu** je pak volána metoda **VykresliSvujObsah**, která aktualizovanou herní plochu vykreslí s využitím již dříve vytvořené instance třídy *Graphics* a její metody

DrawImage.

Také je zde předdefinována metoda *ProcessCmdKey*, pomocí které rozlišujeme, zda byla stisknuta některá z kláves (šipka nebo mezerník), či nikoli.

Ve výchozím stavu jsou šipky i mezerník označeny jako nestisknuté.

Kromě zmíněných metod se v třídě *Form1* také inicializují důležité proměnné: grafika třídy *Graphics*, dále samotná herní plocha třídy *HerniPlocha* a jednorázově se zde vytváří instance třídy *Random* pro části hry využívající náhodný generátor.

Bomber.cs

Soubor *Bomber.cs* obsahuje deklaraci třídy ***Prvek***, která je abstraktní, a dědí z ní třídy ***StatickyPrvek*** a ***DynamickyPrvek*** (obě jsou opět abstraktní). Důležitá je zejména metoda ***OdehrajTah*** třídy ***DynamickyPrvek***. Ta je totiž volána při každé iteraci hlavní herní smyčky a to pro každý objekt, který je aktuálně na herní ploše a je typu *DynamickyPrvek*. Metoda je později předdefinována dle potřeb konkrétnějšího typu dynamického prvku – z třídy *DynamickyPrvek* dědí třídy ***Hrac*** a ***Bomba***. Navíc pak třída ***Nepřítel*** dědí od třídy *Hrac*, neboť základní chování nepřátel ve hře odpovídá chování hráče. Mohou se stejně jako hráč pohybovat horizontálně či vertikálně po herní ploše, umisťovat bomby na pole o 1 před sebou v aktuálním směru otočení, sbírat vylepšení a získávat vlastnosti dle charakteru těchto vylepšení a podobně. Rozdíl oproti chování hráče je, že nepřítel nikdy sami od sebe nevstoupí do lávy vzniklé po výbuchu bomby (znamenalo by to ztrátu 1 života). Pokud ale v blízkosti nepřítel vybuchne bomba a zasáhne jej, život již ztrácí (stejně jako hráč, pokud je zasažen bombou), ať už je detonující bomba jeho vlastní, či umístěná hráčem.

Dále je v souboru obsažena deklarace tříd ***Vylepšení*** a ***Lava*** dědicích od třídy *StatickyPrvek*.

V nich je předdefinována metoda ***OdstranSe***.

V *Bomber.cs* je umístěna také deklarace třídy ***HerniPlocha***, která má na starost načtení herní plochy z textového souboru *plocha.txt* (část odpovídající aktuálnímu levelu), vytvoření této herní plochy v podobě matice znaků a také vytvoření instancí objektů třídy *Prvek*, pokud jsou identifikovány znakem, který se v načtené herní ploše nacházel. Dále má za úkol načíst textury ze souboru *texture.png*, provázat je s odpovídajícími znaky a podle toho vykreslit daný index herní plochy. Součástí třídy *HerniPlocha* jsou právě i klíčové metody ***AktualizujPlochu*** a ***VykresliSvujObsah***, které jsou opakovaně volány ve stavu rozehrané hry. V třídě *HerniPlocha* je také řešeno umisťování vylepšení na herní plochu (náhodný výběr typu vylepšení a indexu umístění), přesouvání prvků po herní ploše, odstraňování dynamických prvků a odebírání životů prvku typu *Hrac*. Kromě toho se zde nachází i sada metod vracejících hodnoty typu *bool*, které kontrolují, zda se na dotazovaném indexu na herní ploše nachází konkrétní prvek (např. volná pozice, zeď, nepřítel, některé z vylepšení a podobně).

Ovládání aplikace

Uživatelské rozhraní

Po spuštění aplikace jsou dostupná 2 tlačítka označená textem ***Nová hra*** a ***Konec hry*** pro spuštění hry či ukončení aplikace. V případě dohrání hry, ať už se jedná o výhru nebo prohru, se objeví zpráva o výsledku hry (výhra / prohra). Tuto zprávu je potřeba kliknutím potvrdit a následně je opět možné spustit novou hru, případně celý program ukončit. V případě výhry a možnosti postupu do další úrovně se text prvního tlačítka změní z „*Nová hra*“ na „*Nový level*“. Stisknutí tlačítek i potvrzení zprávy o výhře / prohře lze provést pomocí klávesy *Enter*, nebo stisku *levého tlačítka myši* (tlačítko *Konec hry* není ve výchozím stavu označeno, takže pro jeho stisknutí *Enter* nelze použít).

Během rozehrané hry není tlačítko *Konec hry* viditelné, ukončení aplikace je tak možné pouze klasickým zavřením programu.

Ovládání postavy hráče

Pohyb vlevo – šipka vlevo

Pohyb vpravo – šipka vpravo

Pohyb nahoru – šipka nahoru

Pohyb dolů – šipka dolů

Umístění bomby (o 1 pole před sebe v aktuálním směru pohybu) – mezerník

Herní logika

Cílem hry je zneškodnit všechny nepřátele, nepřátelé tedy musejí přijít o všechny své životy. Aktuální počet celkových životů nepřátel je zobrazován v levém horním rohu okna aplikace, spolu s aktuálním počtem životů hráče. Každý nepřítel má svou sadu životů, a jakmile ji vyčerpá, je zcela odstraněn z herní plochy. Aktuální počty životů jednotlivých nepřátel jsou hráči skryty – viditelný je pouze součet životů všech nepřátel. Pokud tento celkový počet životů nepřátel klesne na 0, hra končí výhrou. Naopak pokud počet životů hráče klesne na 0, nastává prohra.

Ztráta života

Nepřátelé mohou ztratit život pouze tak, že je zasáhne detonující bomba. K tomu může docházet i bez přičinění hráče, neboť nepříteli odebírá život i zásah jeho vlastní bombou.

Hráč může ztratit 1 život následujícími způsoby: byl zasažen detonující bombou, vstoupí do pole, na kterém je láva, nebo se střetne s nepřítelem (ať už hráč provede krok směrem na pole nepřítele, nebo naopak nepřítel na pole hráče).

Umísťování bomb

Ve výchozím stavu má každá postava (hráč / nepřítel) možnost umístit právě 1 bombu najednou. Tento počet lze dočasně navýšit sebráním příslušného vylepšení. Při umístění bomby se spustí časovač pro danou bombu, po jehož uplynutí bomba detonuje. Po dokončení detonace je počet umísťitelných bomb zvýšen o 1. Pokud byla ale bomba umístěna „nad limit“, tedy s využitím daného vylepšení, a toto vylepšení již není platné, počet umísťitelných bomb se o tuto bombu nenavýší (je potřeba resetovat maximální počet bomb opět na výchozí hodnotu 1). Bomba je umísťována o 1 pole dopředu v aktuálním směru pohybu. Aby tedy bylo možné umístit bombu na požadované pole, je potřeba být otočený ve směru tohoto pole a dále toto pole musí být zcela volné.

Detonace bomby

Po uplynutí nastaveného časovače bomby dochází k její detonaci. Design bomby se mírně změní z „klidového stavu“ na „detonující“. Dle aktuálně platného dosahu bomby se kontrolují pole ve všech směrech horizontálně a vertikálně se středem na pozici bomby. Pokud je pole volné, nebo je na něm „zničitelný“ prvek (dům, tráva, vylepšení apod.) a není mezi ním a bombou neprorazitelná zeď, pole je přeměněno na lávu. Jestliže na poli stojí hráč nebo nepřítel, je mu odebrán život (v případě aktivního vylepšení pro záchranu života není život odebrán a vylepšení je deaktivováno).

Význam jednotlivých vylepšení

Hra je rozšířena o následující vylepšení (náhodně se objevující na herní ploše):

Život – přidá postavě, která jej vzala, 1 život (a v případě sebrání některým z nepřátel je aktualizována také proměnná celkového počtu životů nepřátel).

Štít – jednorázově ochrání svého nositele před ztrátou 1 života (ať se jedná o jakýkoli způsob ztráty života).

Batoň – navýší maximální počet bomb, které je možné umístit najednou (během čekací doby na detonaci bomby) o danou konstantu.

Lektvar síly – umožní hráči / nepříteli tlačit před sebou bombu (přidá možnost posunovat bombu v horizontálním i vertikálním směru).

Krystal – přenastaví dosah bomby z 1 pole na 2 pole (v každém směru horizontálně i vertikálně).

Jednotlivá vylepšení jsou vytvářena s předem daným časovačem (ve hře použit stejný časovač pro všechna vylepšení). Po uplynutí tohoto časovače je vylepšení zrušeno (nositel ztrácí schopnost získanou sebráním tohoto vylepšení). Časovač se netýká vylepšení *Život*, neboť toto vylepšení jednorázově přidává 1 život a jeho přidání zůstává permanentní. O vylepšení *Štít* lze přijít i jiným způsobem – je možné jej „spotřebovat“, tedy jednorázově zabránit odebrání 1 života. Po této záchraně života štít přestává být aktivní.

Při sebrání vylepšení stejného typu během toho, kdy je toto vylepšení již aktivní, se resetuje časovač tohoto vylepšení na maximum (nevytváří se nová instance tohoto vylepšení). U vylepšení *Batoň* je s každým sebráním počet maximálních bomb navýšen o danou konstantu, v 1 okamžiku tento počet lze tedy navýšit několikrát (sebráním více vylepšení *Batoň* v průběhu běžícího časovače). Po konečném uplynutí časovače *Batoňu* je maximální počet umístitelných bomb snížen na výchozí hodnotu, tedy na 1.

Vstupní data

Kromě souborů samotné Windows Forms aplikace jsou v programu využívány následující soubory:

plocha.txt

Znárodnuje obsahy herní plochy pro jednotlivé úrovně ve výchozím stavu. Ty jsou zapsané pomocí odpovídajících znaků. Samotnému zápisu obsahu herní plochy předchází 2 řádky udávající (v tomto pořadí): výšku herní plochy, šířku herní plochy. Jednotlivé úrovně jsou pro přehlednost odděleny 2 prázdnými řádky.

textury.png

Obsahuje ikony pro jednotlivé znaky z herní plochy. Ty jsou umístěny v 1 řádku bezprostředně vedle sebe a každá ikona má čtvercový formát velikost 50 × 50.

Navíc byl využit soubor *bomba_ikona.ico* pro nastavení ikony aplikace.

Použité zdroje

Textury

<https://craftpix.net/freebies/>

<https://opengameart.org/content/moss-on-rocks-96x96-tilemap>

<https://opengameart.org/node/7580>

Struktura programu

Hra Boulder Dash (projekt *cshra*) demonstrována na cvičení Programování 2 (využita zejména jako vzor pro způsob vykreslování herní plochy s využitím dodaných textur a také pro celkovou organizaci programu).

Hra je inspirovaná existující hrou Bomb It.