Bomberman

Borek Požár

Letní semestr 2018/19
Programování II. NPRG031

Anotace

Program spouští známou hru Bomberman, kde cílem hráče je pomocí pokládání bomb postupně uvolnit mapu pro průchod a především zneškodnit monstra. Na výběr jsou varianty pro jednoho a pro dva hráče.

Ovládání

Na začátku hry si uživatel tlačítkem zvolí variantu pro jednoho nebo dva hráč. První hráč poté ve hře ovládá pohyb šipkami a bombu enterem, případný druhý hráč klávesami WASD a bombu mezerníkem. Po dokončení úrovně si tlačítkem ve vyskakovacím okně zvolí, jestli chce pokračovat do další, nebo skončit. Při prohře si tlačítkem zvolí, jestli chce zkusit úroveň znovu, nebo skončit.

Algoritmus

Hra začíná stisknutím jednoho z tlačítek. V tu chvíli se z textových souborů načte vzhled mapy a počáteční bonusy, z obrázku se načtou ikonky jednotlivých věcí na mapě. Vše je nadále uchováváno a zpracováváno v polích, pohyblivé prvky jsou navíc uloženy do seznamu. Po spuštění hry se spustí `timer`, který následně řídí celou hru – dokud hra běží, tak v každém jeho kroku je pohnuto s každým pohyblivým prvkem v seznamu (případně jsou některé smazány), a potom je celá mapa vykreslena na obrazovku. Když hráč vyhraje/prohraje úroveň, změní se stav hry a na základě toho je zobrazen `MessageBox`, kde má hráč další volby.

Program

Základem prvků na mapě je abstraktní třída `MovingElement`, která předepisuje abstraktní funkci `MakeStep()` přepisovanou každou odvozenou třídou, protože každý pohyblivý prvek musí být schopen se pohnout. Od této třídy je odvozena další abstraktní třída `Man` určená pro tvorbu hráčů, a pak přímo odvozené třídy dalších dvou pohyblivých prvků na mapě – `Bomb` a `Monster`. Třída `Man` má specifikované funkce platné pro všechny hráče – například co se s hráčem stane, když sebere bonus, třídy pro jednotlivé hráče `Player1` a `Player2` jsou od ní odvozovány z důvodu potřeby odlišného ovládání. Každá třída má specifikované proměnné používané každou instancí této třídy, jako `bombMax` pro každého hráče nebo `range` pro každou bombu.

Pohyb hráčů

```
if (Keyboard.IsKeyDown(x))
```

Je rozhodovací podmínka pro každého hráče, podle které se upraví jeho nové souřadnice pro pohyb. Vyžaduje `using Windows.Input` a reference `PresentationCore` a `WindowsBase`.

`timer`

Tato proměnná je na začátku nastavena na `2`, čímž zaručuje, že hráč se pohne jen každé dva tiky. Klávesnice je naproti tomu snímána každý tik, protože aby se uživatel trefil stiskem do každého druhého tiku je příliš nepravděpodobné, a ovládání by tak bylo velmi nekomfortní. Po sebrání bonusu se sníží na `1` a hráč se tak pohybuje každý tik.

Příšery

Příšery se pohybují každý třetí tik, aby před nimi měl hráč šanci utéct – to zajišťuje proměnná `monsterTimer`. Jinak jsem k jejich pohybu využil kód z jednoho z předcházejících domácích úkolů. Pokud navíc příšera vstoupí na pole, kde stojí hráč, zavolá funkci `map.KillPlayer(newX, newY)` s parametry souřadnic a hráče zabije.

Bomby

Při pokládání bomb volá hráč funkci `map.PutBomb(x)` s parametry souřadnic, dosahu, času do výbuchu a vlastníka. Tato funkce je funkcí mapy, protože bombu je třeba přidat do seznamu pohyblivých prvků. Třída bomby pak řeší výbuchy – má funkci `SolveBomb(x)` s parametry souřadnic, které podle obsazenosti explodujícího políčka zabije hráče/příšeru, případně vytvoří bonus, pokud tam byla hlína. Nakonec políčko vyprázdní pomocí `map.EmptySpace(x)`, protože v mapě na něm po výbuchu nic nezbude (jediné, co se může na tomto políčku objevit, je bonus, ale ten zůstane v poli nevyprazdňovaném poli `bonuses`). Druhou funkcí bomby je `ExplodeBomb(x)`, která má parametry souřadnic a dosahu. Bomba se při výbuchu odečte z bomb vlastněných hráčem `this.owner.bombNow--`, aby ji mohl znovu použít, a pak vybuchne ve všech čtyřech směrech – v každém zvlášť, protože pokaždé se zastaví o první zeď, první hlínu, nebo první konec mapy. Na každé políčko, kam vybuchuje, pak zvlášť volá zmíněnou funkci `SolveBomb(x)`. Nakonec tuto bombu odstraní z pole bomb a seznamu pohyblivých prvků `map.DeleteBomb(x)`.

Poslední funkcí je přepisovací funkce `MakeStep()`, která jen každé bombě odečte časovač a když dosáhne nuly, zavolá na ni výbuch.

Mapa

Největší třídou celého programu je `Map`. Jejími hlavními proměnnými jsou 4 dvourozměrná pole:

- `char[,] board` – uchovává samotnou mapu
- `bool[,] bombs` – kde jsou bomby – mohou být i na stejném poli, jako hráč v mapě
- `char[,] bonuses` – kde jsou bonusy – mohou skrz ně procházet monstra
- `bool[,] explosions` – aktuálně vybuchující místa, jen pro zobrazení

První volanou funkcí při spuštění hry je funkce `Map()`, která v parametrech dostane cestu ke třem zdrojovým souborům, na každý zvlášť zavolá speciální funkci (`LoadMap`, `LoadIcons` nebo `LoadBonus`), která je načte do požadovaných polí a proměnných. Dále změní stav hry na `State.running`. Funkce `PutBomb(x)` přidá bombu do pole bomb a vytvoří nový prvek bomby, který pak přidá do seznamu pohyblivých prvků.

`DeleteBomb(x)` najde bombu na hledaných souřadnicích, přidá ji do fronty `ToDelete` pro odebrání ze seznamu pohyblivých prvků a odstraní ji z pole bomb.

`Move(x)` dostane parametry, odkud a kam prvkem pohybovat, a následně ho posune v poli mapy, pokud je to možné. Poté tomuto prvku změní souřadnice, aby odpovídaly novým. `ChangeDir(x)` je pro třídu monstra, aby v poli mapy změnila jeho směr na odpovídající. `DeleteMovingElement(x)` najde prvek z daného místa, přidá ho do fronty k odstranění `ToDelete` a smaže ho z pole mapy.

`CreateBonus(x)` na dané pole s 10% pravděpodobností vloží nový bonus jedné ze čtyř kategorií. `DrawMap()` je volána každý tik `timeru` a vykreslí na obrazovku mapu podle aktuálního pole mapy. `MoveAll()` zavolá pro každý prvek v seznamu `MakeStep()`, a potom ještě zvlášť pro každého hráče, pokud je živý. Pokud žádný hráč není živý, změní stav hry na `State.lost`.

`DeleteGarbage()` je volána až po dokončení všech kroků, protože jinak by bylo do seznamu pohyblivých prvků saháno v průběhu jeho procházení. Potupně bere prvky z fronty `ToDelete`, dokud v ní něco je. Ve frontě je uloženo, na kterém místě seznamu prvek je. Odstraněním každého prvku se seznam zkrátí. Díky tomu, že seznam je procházen vzestupně, stačí od každého dalšího pořadí odečíst počet prvků v tomto průběhu funkce odstraněných.

`KillPlayer(x)` zabije hráče nacházejícího se na daném místě změněním proměnné `isAlive` na `false`. `WhatsThere(x)`, `IsClay(x)`, `ForwardFree(x)`,… je několik funkcí, které se jen ptají na aktuální stav mapy pro vyhodnocování podmínek jinde.

Vstupní data

Vstupními daty jsou na začátku výběr módu pro jednoho nebo dva hráče stisknutím daného tlačítka. Při běhu hry jsou to stisky kláves pro pohyb, po dokončení úrovně nebo prohře pak tlačítka **Z** `MessageBoxu`.

Výstupní data

Výstupními daty je hra zobrazená na obrazovce.

Průběh práce

Na začátku jsem napsal základní pohyb hráčů, a pak postupně přidával další funkce – bomby, příšery,… Delší dobu jsem strávil s vymýšlením, jak odstranit prvek ze seznamu prvků, protože to nemohlo být uděláno v rámci kroku. Poté jsem dopsal přecházení mezi úrovněmi. Posledními drobnostmi bylo použití ikonek (ze stránky thenounproject.com) a zprovoznění zobrazení výbuchů. Pro načítání a vykreslování mapy jsem použil kód připravený ze cvičení.

Co zlepšit

Možná by bylo lepší na začátku si nakreslit graf, jaké funkce budu potřebovat a s čím budou muset komunikovat. Přidávání celých tříd postupně bylo pohodlné, ale funkce občas mohou být trochu zmatené – ať už protože mají víc parametrů, než by bylo potřeba, nebo jejich seřazením. Celkově přehlednost kódu asi není nejlepší.

Ukázka ze hry