

Bombberman

Závěrečná maturitní práce

Vedoucí práce:
Lukáš Rýdlo

David Beneš

Brno 2019

Chtěl bych poděkovat Lukáši Rýdlovi za pomoc s mojí prací a za úsilí, které věnoval při seznamování mne s jazykem Java a teorií programování.

Prohlašuji, že jsem tuto práci vyřešil samostatně s použitím literatury, kterou uvádím v seznamu.

V Brně dne 29. května 2019

.....

Abstrakt

BENEŠ D., Bomberman. Brno, 2019.

Práce pojednává o online hře Bomberman, o problematice programování této hry a o prvcích obsažených ve hře.

Obsah

1	Úvod	9
1.1	Programovací jazyk Java	9
1.2	Cíl a záměr práce	9
1.3	O hře	9
2	Obecně o vývoji programu	10
2.1	Komunikace po síti	10
2.2	Běh programu	10
3	Třídy	13
3.1	Třídy na straně klienta	13
3.2	Třídy na straně serveru	13
3.3	Datové a výčtové třídy	16
4	Komunikace mezi serverem a klientem	18
4.1	Komunikace klient–server	18
4.2	Komunikace server–klient	18
5	Výsledek práce	19
5.1	Shrnutí	19
5.2	Možné doplňky do hry	19
5.3	Závěr	20

1 Úvod

1.1 Programovací jazyk Java

Jazyk Java se řadí mezi objektově orientované programovací jazyky. Jde o jeden z nejpoužívanějších jazyků na světě, podle některých zdrojů je Java nejpopulárnějším programovacím jazykem vůbec [1].

Objektově orientované programování pracuje s tzv. objekty, což jsou jakési balíčky, obsahující různé atributy a metody, které mohou vykonávat [2].

1.2 Cíl a záměr práce

Jako cíl jsem si vytyčil naprogramovat v Javě online hru. Tedy hru, která by se dala hrát přes síťové spojení na dvou různých počítačích.

Mým záměrem bylo rozšířit si své znalosti v programování zábavního softwaru a v programování síťových aplikací. Dokumentaci, kódy i grafiku ke hře nechávám volně šířitelné.

1.3 O hře

Hra samotná, patřící mezi klasické hry z 80. let (spolu například se hrou Pac-Man), byla poprvé vydána roku 1985 na nejúspěšnější 8bitovou herní konzoli Nintendo Entertainment System (NES) firmou Hudson Soft [3]. Od té doby se dočkala mnoha různých přepracování vydaných i pod jinými názvy (například Dyna Blaster, Bomb It nebo Playing With Fire) [4].

Podrobná pravidla

Hráč hraje za robota, který se snaží uprchnout z továrny na výbušniny. Herní plán je rozdělen na tabulku o určitém množství polí (specifikuje se při zakládání serveru). Cílem hry je zabít ostatní hráče a zůstat ve hře jako poslední. Aby se k ostatním hráčům vůbec dostal, musí překonat dřevěné krabice, které leží po celém plánu a blokují cestu. K ničení krabic a hráčů využívá Bomberman bomby, kterých má u sebe dostatek. Na herním plánu jsou i kovové bloky, které zničit nelze. V dřevěných krabicích bývají občas i ukryté nějaké speciální bonusy – například život navíc nebo silnější typ bomby. Zajímavým bonusem jsou speciální boty, které Bombermanovi umožní chodit i po dřevěných krabicích.

2 Obecně o vývoji programu

2.1 Komunikace po síti

U síťové komunikace vystupují dva aktéři – klient a server. Server zprostředkovává komunikaci mezi jednotlivými klienty. Klient je software na straně uživatele zpracovávající data ze serveru.

Veškerá logika hry musí být zpracována a vyhodnocena na straně serveru, aby se zamezilo podvádění. Klienti tedy pouze posílají serveru požadavky na provedení nějaké akce (v našem případě například položení bomby). Server po vyhodnocení situace informuje všechny klienty o výsledku a celkovém stavu hry (například hráč číslo 1 položil bombu).

2.2 Běh programu

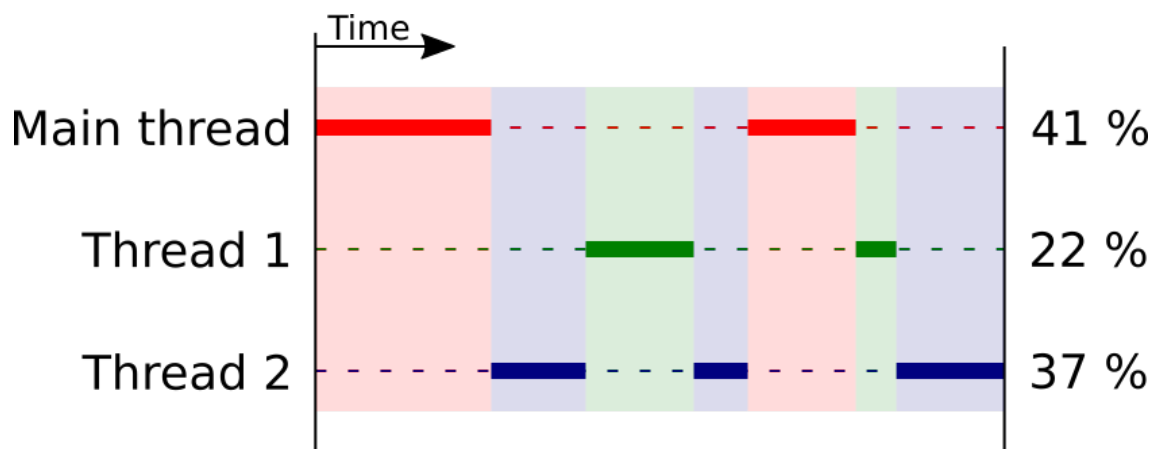
Paralelní výpočty

Paralelní výpočty je označení pro výpočty, které jsou řešeny souběžně [5]. Je k tomu tedy potřeba několik jader procesorů, přičemž každé řeší jeden výpočet. V našem případě by se hodilo si výpočty tímto způsobem rozdělit. Například by se hodilo, aby se informace o stavu hry odeslala všem klientům ve stejnou chvíli. Něčeho takového ale nejsme schopni dosáhnout, proto musíme využívat sekvenčních výpočtů (tedy instrukce jdou za sebou a řeší se jedna po druhé). Výpočty jsou tzv. serializované.

Vláknové programování

Ve vláknovém programování se určitý problém může rozdělit na několik tzv. vláken („threadů“). Při běhu programu se pak vlákna střídají ve výpočtech. Každé vlákno tedy dostane určitý časový prostor pro provádění svých výpočtů.

V našem případě si vytvoříme na straně serveru pro každého klienta jedno vlákno, které bude obstarávat komunikaci s tímto právě jedním klientem. Na straně klienta zase vytvoříme jedno vlákno na odposlouchávání zpráv ze serveru.



Obrázek 1: Znázornění běhu několika vláken na jednom jádře procesoru.

Problémy vláknového programování

Vláknové programování s sebou přináší i řadu problémů. Jedním je například nerovnoměrné rozložení poskytnutého prostoru pro výpočty u jednotlivých vláken (jak jde vidět na obrázku). Dalším problémem je, že může být výpočet některého vlákna přerušen v situaci, kdy potřebujeme určitou operaci řešit v celku. Rozložení prostoru pro jednotlivá vlákna řeší plánovač operačního systému, který se může při každém běhu rozhodovat různě. Program se pak tedy může chovat pokaždé jinak. Uvedu příklad – máme globální proměnnou *first*, do které chceme uložit instanci prvního vytvořeného vlákna. Vlákno si po spuštění zkontroluje, že je proměnná *first* prázdná. Pokud ano, vypíše OK a přiřadí do proměnné odkaz na sebe. Pokud ne, vypíše PLNO.

Běh programu při jednom spuštění:

- Vlákno A je vytvořeno.
- Vlákno A zjistí, zda je proměnná *first* prázdná (ano).
- Vlákno B je vytvořeno.
- Vlákno A vypíše OK.
- Vlákno A přiřadí do proměnné *first* odkaz na sebe.
- Vlákno B zjistí, zda je proměnná *first* prázdná (ne).
- Vlákno B vypíše PLNO.

Běh programu při jiném spuštění:

- Vlákno A je vytvořeno.
- Vlákno B je vytvořeno.
- Vlákno B zjistí, zda je proměnná *first* prázdná (ano).
- Vlákno B vypíše OK.
- Vlákno A zjistí, zda je proměnná *first* prázdná (ano).
- Vlákno A vypíše OK.
- Vlákno A přiřadí do proměnné *first* odkaz na sebe.
- Vlákno B přiřadí do proměnné *first* odkaz na sebe.

Jak můžeme vidět, při druhém spuštění je v proměnné *first* odkaz na vlákno B, přestože bylo vytvořeno až jako druhé. Toto chování může způsobovat nečekané chování programu nebo chyby, konkrétně v Javě je to chyba *ConcurrentModificationException*, kdy například chceme grafickým widgetem vykraslit textový řetězec, který je ale v polovině vykreslování změněn jiným aktivním vláknem. Toto se dá částečně obejít tak, že onen textový řetězec nejprve zklonujeme, poté vykreslíme klona a nevadí nám tudíž, že je původní řetězec mezitím přepsán. Toto řešení není ovšem úplně korektní.

Správně se tento problém řeší přes tzv. semaforey. Pomocí semaforů lze ochránit určitou část kódu, aby nebyla přerušena, dokud není celá provedena. Semaforey ovšem kvůli nedostatku času nemám v projektu implementovány.

3 Třídy

Window.java

Třída obsahující metodu *main*. Zároveň poskytuje grafické prostředí pro hlavní menu aplikace, obsahující možnosti *Join Server* (pro připojení k běžícímu serveru), *New Server* (pro založení nového serveru) a *Help* (pro přehled ovládání hry).

HelpWindow.java

Třída pro grafické rozhraní okna s pomocí (například ovládání hry).

3.1 Třídy na straně klienta

Client.java

Třída zpracovávající data ze serveru a obsahující metody pro posílání zpráv serveru.

ClientWindow.java

Třída obsahující grafické rozhraní pro klienta. Obsahuje grafický widget pro vykreslování herního plánu, dále panely jednotlivých hráčů s informacemi o jméně, barvě, životech a bonusech.

ClientOptionsWindow.java

Třída obsahující grafické rozhraní pro možnosti klienta. Jde zde nastavit například adresa serveru, číslo portu nebo jméno hráče.

ClientCallback.java

Rozhraní pro zpětnou komunikaci mezi klientem (*Client.java*) a grafickým rozhraním (*ClientWindow.java*), například změna stavu herní mapy nebo počtu životů některého hráče.

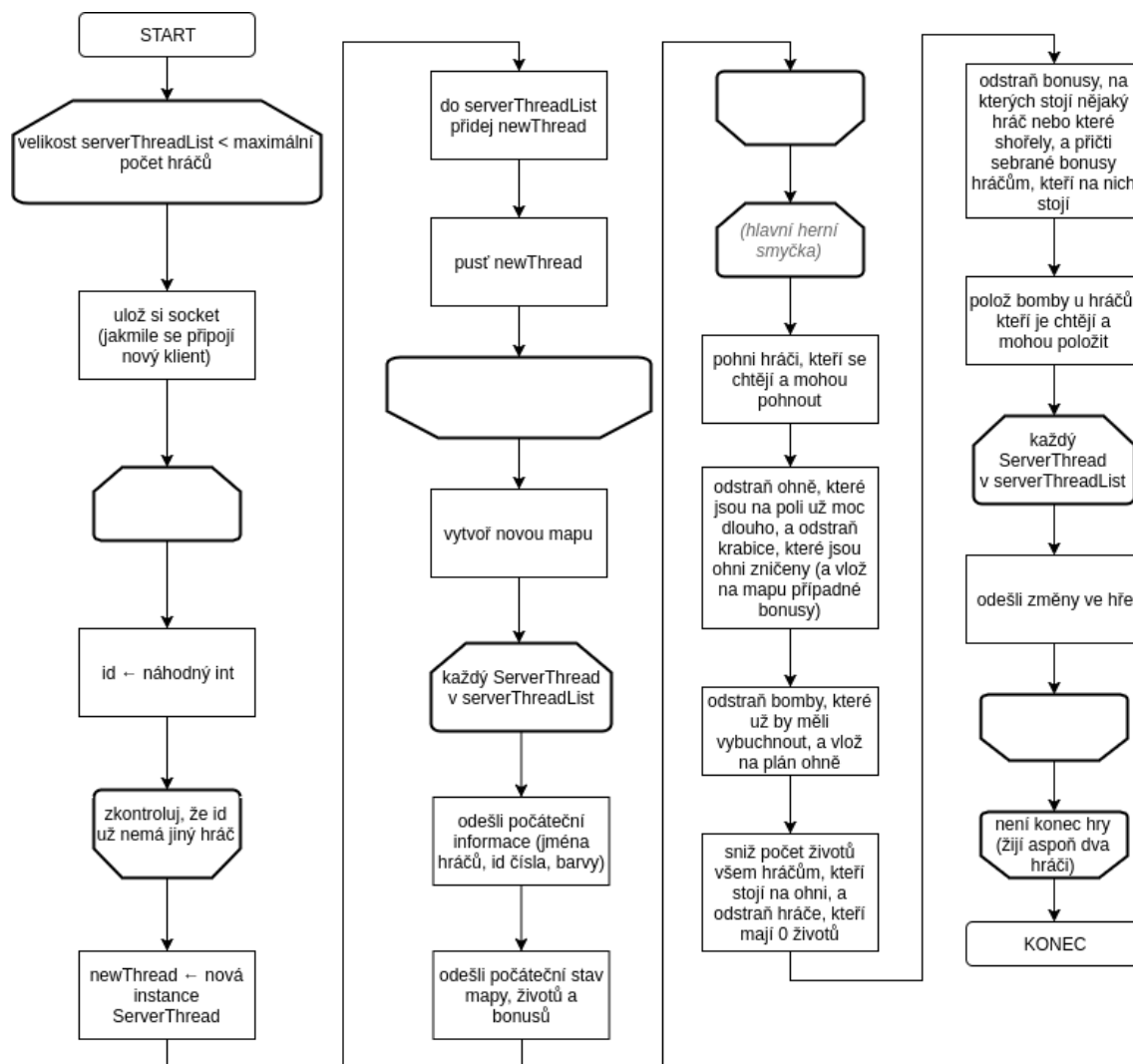
GraphicWidget.java

Třída vykreslující herní plán.

3.2 Třídy na straně serveru

Server.java

Třída operující na straně serveru. Obsahuje hlavní herní smyčku.



Obrázek 2: Zjednodušený vývojový diagram herní smyčky v třídě Server.java.

ServerThread.java

Tato třída se pouští jako samostatné vlákno. Stará se o komunikaci s právě jedním klientem. Sama zpracovává zprávy od klienta a přes tuto třídu posílá server data klientovi.

ServerWindow.java

Třída obsahující grafické rozhraní pro server. Vypisuje například číslo portu, počet připojených klientů nebo stav hry.

ServerOptionsWindow.java

Třída obsahující grafické rozhraní pro možnosti serveru. Nastavuje se zde například číslo portu, počet hráčů nebo velikost herní mapy.

ServerCallback.java

Rozhraní (interface) pro zpětnou komunikaci mezi serverem (*Server.java*) a grafickým rozhraním (*ServerWindow.java*), například změna stavu hry nebo počtu připojených hráčů.

Jméno	Popis	Hodnota
PLAYER START LIFES	počáteční životy hráčů	3
BOMB TIME	počet projití cyklu než vybuchne položená bomba	30000
FIRE TIME	počet projití cyklu než zmizne oheň	třetina BOMB TIME
PLAYER MOVE TIME	počet projití cyklu než se hráč může znovu pohnout	500
PLAYER LIFE LOST TIME	počet projití cyklů než může hráč znovu ztratit život	FIRE TIME + 15000
BOMB RANGE	počet polí v okolí bomby, na kterých se objeví ohně	3
LARGE BOMB RANGE	počet polí v okolí velké bomby, na kterých se objeví ohně	6
PLAYER MAX BOMB COUNT	maximální počet bomb, které může mít jeden hráč položený na plánu	2
BONUS PROBABILITY	šance na výskyt bonusu (v každých X krabicích by měli být 3 bonusy)	250
WOODENBOX PROBABILITY	šance na výskyt krabice (na každých X polích by mělo být X-1 krabic)	8

GameLogic.java

Třída obsahující metody pro logické operace ve hře, volané třídou *Server.java*. Dále také důležité konstanty (tabulka viz výše).

GameMap.java

Třída uchováající veškerá data o herní mapě. Obsahuje množiny všech instancí jednotlivých herních objektů (potomci třídy *MapObject.java*).

MapObject.java

Třída uchováající pozici objektu. Seznam potomků:

- *Bomb.java*
- *Bonus.java*
- *Fire.java*
- *IronBox.java*
- *Player.java*
- *WoodenBox.java*

3.3 Datové a výčtové třídy

BombType.java

Výčtová třída typů bomb (základní, velká).

BonusType.java

Výčtová třída typů bonusů (život navíc, speciální boty, velká bomba).

Direction.java

Výčtová třída možných směrů (nahoru, dolů, doprava, doleva).

MapObjectType.java

Výčtová třída všech možných typů objektů včetně možných variant a otočení.

PlayerColor.java

Výčtová třída možných barev hráče (modrá, zelená, červená, bílá).

Position.java

Třída pro určení pozice na mapě. Obsahuje dva atributy – pozice na ose X a pozice na ose Y.

4 Komunikace mezi serverem a klientem

4.1 Komunikace klient–server

Jakmile klient obdrží první zprávu od serveru („HELLO“), odpoví zprávou „HI_MY_NAME_IS“, za kterou ještě dodá hráčem zvolené jméno. Když se hra rozjede, může klient poslat několik možných zpráv:

- „MOVEUP“ – pro pohyb nahoru (šipka nahoru)
- „MOVEDOWN“ – pro pohyb dolů (šipka dolů)
- „MOVERIGHT“ – pro pohyb doprava (šipka doprava)
- „MOVELEFT“ – pro pohyb doleva (šipka doleva)
- „DROPBOMB“ – pro položení bomby (mezerník)

Tyto zprávy se odešlou po stisknutí specifikované klávesy (v závorce). Na konci hry se po stisku libovolné klávesy odešle zpráva „BYE“.

4.2 Komunikace server–klient

Jakmile se připojí nový klient, nově spuštěné vlákno serveru jako první pošle svému klientovi zprávu „HELLO“ a přidá k ní i klientovo identifikační číslo (například tedy „HELLO 154618944“). Když se připojí všichni hráči a hra je připravena, dorazí ke klientovi dlouhá zpráva začínající slovem „INFO“, následuje trojice dat pro každého hráče – první je identifikační číslo hráče, druhé je jméno hráče a třetí je barva hráče (například tedy „INFO 154618944_David_blue 254564005_Tomas_green“).

Následně během hry chodí klientovi dlouhé zprávy začínající slovem „DATA“. Tyto zprávy informují o stavu hry. Následují další řetězce:

- „L“ – pro změny v počtu životů (doplněno o identifikační číslo hráče a počet životů).
- „B“ – pro změny v získaných bonusech (doplněno o identifikační číslo hráče a získané bonusy).
- „M“ – pro změny na mapě (doplněno o souřadnice a objekty).

Například tedy „DATA L_154618944_2 B_254564005_false_true M_1_2_fire M_1_3_fire_woodenbox M_1_1_player-blue-left_fire M_14_12_player-green-left M_14_13“.

Jakmile hra skončí, pošle server klientovi poslední zprávu „END“, doplněnou o identifikační číslo vítěze (například tedy „END 254564005“). Po libovolné odpovědi od klienta spojení skončí (proto je na konci hry jedno, kterou klávesu hráč stiskne).

5 Výsledek práce

5.1 Shrnutí

Od základů jsem naprogramoval jednoduchou online hru během asi 6 měsíců. Má stále nějaké chyby a dalo by se do ní přidat hromada dalších možností, které shrnu v další sekci. Celý projekt byl vytvořen ve vývojovém prostředí NetBeans IDE 8.2, které mi hodně zjednodušilo práci při tvorbě grafických rozhraní. Veškerou grafiku ke hře jsem pak tvořil v programu GIMP. Tato dokumentace je vytvořena programem Texmaker.

5.2 Možné doplňky do hry

Semaforey

Využití semaforů v projektu by výrazně snížilo chybovost hry (především by zamezila *ConcurrentModificationException*). Jejich implementace by tedy byla logickým postupem v příští verzi hry.

Zvýraznění hráče

Pohodlnost uživatelů by se určitě zvýšila, pokud by se nějakým způsobem každému hráči zvýraznil v *ClientWindow.java* panel s jeho postavou, aby tak měl jasný přehled, kolik má životů a jaké bonusy sebral. Implementace není tak složitá a efekt by byl velký.

Hra více hráčů na jednom počítači

Ne vždy je možnost zahrát si hru po síti. Nabízí se tedy varianta hrát hru na jednom počítači, přičemž by třeba hráč 1 používal k ovládání šipky (pohyb) a mezerník (bomba) a hráč 2 zase klávesy W, S, A, D (pohyb) a třeba Q (bomba). Na jednu klávesnici by se vešel i třetí hráč, který by mohl používat klávesy na numerické klávesnici 8, 5, 4, 6 (pohyb) a 0 (bomba). V grafickém rozhraní je už pro tuto možnost nachystaná i nabídka, která ale v současné verzi nemá žádný efekt.

Hudba a zvuky

Správná hra by měla mít i hudbu a zvukové efekty. Zde se nabízí zvuky při výbuchu bomby a sebrání bonusu. Využít by se dalo volně šiřitelných zvukových efektů bez autorských práv, kterých se dá najít na webu spoustu. V grafickém rozhraní *Window.java* jsou opět nachystána tlačítka pro vypnutí a zapnutí zvuku a hudby.

Ikonka a logo

S designovými prvky by se dalo ještě hodně vyhrát. Nápis Bomberman v hlavním menu by mohl být napsán nějakým stylovým písmem a hra samotná by mohla mít ikonku. Kromě toho by se samozřejmě dalo pohrát i třeba s pozadím všech oken nebo se vzhledem tlačítek, což by rovněž přidalo na atraktivitě.

Více balíčků s grafikou

Kdyby se někomu nelíbila grafika hry, jednoduše by se dal vytvořit další balíček. Dřevěné krabice, tráva i bomby by tak mohly vypadat úplně jinak. Toto se dá aplikovat na současnou verzi tak, že si uživatel vytvoří svoji grafiku (všechny obrázky musí mít rozměry 32x32 pixelů) a pojmenuje ji stejně, jako je pojmenovaná ta současná (tedy například *woodenbox.png* nebo *player-blue-right.png*).

Další prvky do hry

Do hry by se dalo přidat spoustu dalších vylepšení. Nabízí se například další typ bomby. Hráč by bombu položil jedním stiskem tlačítka pro položení a následným stiskem stejného tlačítka by bomba vybuchla. Bonusů by se ale dalo vymyslet spousta.

Animace

Aby postavy při přechodu na sousední pole jen tak neskočily, mohla by se do hry přidat animace přechodu. Pohyb by působil více dynamicky a plynule.

Jiné

Takto bych mohl pokračovat dál. Možných doplňků do hry je velmi velké množství. Můžu zde ještě zmínit třeba tzv. Single Player, tedy možnost hrát proti botům (hráčům ovládaným umělou inteligencí). Pěkně by se v *ClientWindow.java* vyjímal i chatovací box, kde by si mohli hráči posílat zprávy.

5.3 Závěr

Tato práce plně splnila své cíle i záměry. Rozhodně jsem si rozšířil své znalosti a schopnosti v programování, a to především objektově orientovaném. Kostra komunikace mezi klientem a serverem se navíc dá využít i pro jiné, budoucí projekty.

Tato práce je volně dostupná na webu včetně dokumentace, grafiky i zdrojového kódu na adrese github.com/beny258/bomberman.

Reference

- [1] *TIOBE Index for May 2019*. [online] Dostupné z: tiobe.com/tiobe-index [cit. 2019-05-20]
- [2] ROUSE, Margaret. *What is object-oriented programming (OOP)*. [online] Dostupné z: searchmicroservices.techtarget.com/definition/object-oriented-programming-OOP [cit. 2019-05-20]
- [3] *Retrogames – Bomberman*. [online] Dostupné z: retrogames.cz/play_085-NES.php [cit. 2019-05-20]
- [4] *Raketka – Bomberman*. [online] Dostupné z: raketka.cz/t/bomberman [cit. 2019-05-20]
- [5] *Parallel Computing*. [online] Dostupné z: techopedia.com/definition/8777/parallel-computing [cit. 2019-05-20]