

Gymnázium, Praha 6, Arabská 14

Obor programování, vyučující Mgr. Jan Lána



ROČNÍKOVÝ PROJEKT

Hra – Lodě

Gabriela Stamenova, 1.E

Duben 2022

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Praze dne

Podpis:

Anotace:

Výstupem tohoto projektu by měla být jednoduchá verze hry Lodě, kde program umožňuje hráči hrát proti jinému hráči na jednom zařízení. Cílem je vytvořit dvě pole 10 x 10 a deset náhodně rozmístěných lodí 1 x 1. Hráč si může zvolit pozice svých lodí. Pro samotné programování jsem použila jazyk Java a platformu Java FX.

Abstract:

The result of this project should be a primitive version of the game Battleship, where the program allows a player to play against another player on one device. The goal is to create two fields 10 x 10 and ten randomly located ships. The player can choose the positions of their ships. For the programming I used language Java and platform Java FX.

Zadání práce:

Zadání ročníkové práce je vytvořit jednoduchou hru či program v Javě. Můj projekt jsou tedy Lodě a zadání je naprogramovat dvě hrací pole 10 x 10 a deset lodí. Loď se nemůže stranou dotýkat jiné lodě. Cílem hry je potopit všechny soupeřovy lodě. Hráč 1 se střídá s hráčem 2 při střelbě jednotlivých lodí a při úspěšném zásahu má ještě jeden pokus. Program mezi tím škrtne dané pole. Na konci program vypíše výherce a po zmáčknutí tlačítka reset hra začíná od začátku.

Obsah

1. Úvod	6
1.1 Historie hry	6
1.2 Pravidla	6
2. Použité nástroje	7
2.1 Java	7
2.2 JavaFX	7
2.3 JavaFX Scene Builder	7
3. Práce se třídami	8
3.1 Hello-view	8
3.2 HelloController	8
3.3 GameLogics	9
Bibliografie	11
Seznam obrázků	11

1. Úvod

1.1 Historie hry

Hra byla původně desková a později i počítačová. Vznikla ve Francii, ale první komerční verze byla publikovaná v Anglii v roce 1931. „ V roce 1967 představil Milton Bradley verzi hry, která používala plastové desky a kolíky. V roce 1977 Bradley také vydal Elektronické Lodě (Electronic Battleship), následovanou v roce 1989 Elektronickými mluvícími Loděmi (Electronic talking Battleship).“ (Wikipedie - Lodě)



Obr. 1 – Desková verze Lodí (Wikipedie - Námořní bitva, 2011)

1.2 Pravidla

Pravidla hry jsou jednoduchá. Oba hráči si připraví dvě hrací pole ve tvaru čtverce, kde do prvního rozmístí své lodě a do druhého zapisuje výstřely na soupeře. Hráči se střídají ve střelbě a v případě úspěšného zásahu střílí znova. Hra končí, když jeden z hráčů potopí všechny soupeřovy lodě. (Wikipedie-Lodě-pravidla)

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

Obr. 2 – Lodě na papíře či počítači

(Wikipedie - Lodě)

2. Použité nástroje

V téhle kapitole krátce popíšu nástroje, které jsem použila při zhotovení tohoto projektu. Také uvedu stručné informace o jejich funkci a použití. Jako IDE (vývojové prostředí) jsem použila IntelliJ IDEA Community edition, což je jedno z nejpoužívanějších prostředí pro programování.

2.1 Java

Program je napsaný v jazyce Java. Java je objektově orientovaný programovací jazyk, který vznikl 23. května roku 1995. Je to jeden z nejrozšířenějších a nepoužívanějších jazyků, který se používá k vývoji mobilních aplikací či webových služeb. Můj program je tvořen ve verzi Java 8 update 311.

2.2 JavaFX

JavaFX je platforma Javy, která slouží k vývoji bohatých internetových aplikací. Další rozdíl oproti Javě je, že stránky nějakým způsobem komunikují s uživatelem, na rozdíl od těch statických. „ *Platforma byla navržena tak, aby se její uživatelé mohli místo kódování soustředit na kreativní část práce. Mimo jiné nabízí také velmi dobrou podporu pro používání multimediálních prvků, ať už jde o video, audio, animace, atd.* “ (Wikipedie-JavaFX-Vlastnosti)

2.3 JavaFX Scene Builder

„ *JavaFX Scene builder je vizuální nástroj podporující technologii drag&drop pro vytváření grafického uživatelského rozhraní v JavaFX aplikacích bez potřeby psaní kódu.* “ (Oracle JavaFX Scene builder) Použila jsem ho pro vytvoření základních polí a Label, který popisuje okamžitý stav hry. Tady vznikla první drobná komplikace. Když jsem vytvářela Buttony pro každý čtvereček hracího pole, tak jsem se snažila přijít na způsob jak nevytvářet každý Button zvlášť, ale např. nějakým cyklem. Bohužel se mi to nepodařilo a vypisování zabralo více času, ale teď je to v pořádku a funguje to. Také většina grafického rozhraní je právě pomocí Scene Builderu.

3. Práce se třídami

V této kapitole Vás seznámím s obsahem a funkcí různých tříd. Použila jsem čtyři základní třídy souborů fxml a k tomu pomocnou třídu GameLogics, o které se dozvíte více později.

3.1 Hello-view

Vzhledem k tomu, že soubor je typu fxml, tak hello-view.fxml je nezbytná část projektu. To je třída, ve které se nachází Scene Builder a její funkce je hlavně grafická. Tady jsem vytvořila Buttony, Labely a rozměry polí.

3.2 HelloController

Tahle třída se zabývá akcemi, které nastanou po zmáčknutí různých Buttonů a také řídí statusy a situace v průběhu hry. Zde nastal menší problém s propojením třídy GameLogics. Chtěla jsem to naprogramovat tak, že třída HelloController vezme data z GameLogics a pak vypíše, co se má v dané chvíli dít. Nakonec se to povedlo s využitím podmínek a větvení. Na konci nastal problém s tlačítkem Reset. Nevěděla jsem jak na to, tak jsem se pokusila, a nevyšlo to úplně jak jsem si to představovala, ale časem spravím. Teď po ukončení hra automaticky začíná znova, aniž by byla potřeba něco mačkat.

```
if (int_splitter[0] == 1) { // p 1
    if (HelloApplication.logics.getGameStatus() == 0) { // selecting
        int chooseKey = HelloApplication.logics.chooseShip(int_splitter[1], int_splitter[2], playerOne: true);
        switch (chooseKey) {
            case 1:
                label.setText("Ship chosen" + HelloApplication.logics.getPlayerOneAliveShips() + "/" + 10);
                ((Button) event.getSource()).setText("■");
                shipButtons.add((Button) event.getSource());
                break;
            case 2:
                label.setText("Player 1 placed their ships\n" +
                    "Please player 2 choose your ships");
                for (Button b : shipButtons) {
                    b.setText("0");
                }
                break;
            case -1:
                label.setText("There is already a ship at this location\n" +
                    "Please choose again");
                break;
            case -2:
                label.setText("Some ship is nearby your location\n" +
                    "Please choose again");
                break;
        }
    } else if (HelloApplication.logics.getGameStatus() == 3) { // shooting
        switch (HelloApplication.logics.shoot(int_splitter[1], int_splitter[2], playerOne: false)) {
            case 0:
                label.setText("Something went wrong.\nPlease try again");
                break;
        }
    }
}
```

Obr. 3 – Ukázka kódu ze třídy HelloController

3.3 GameLogics

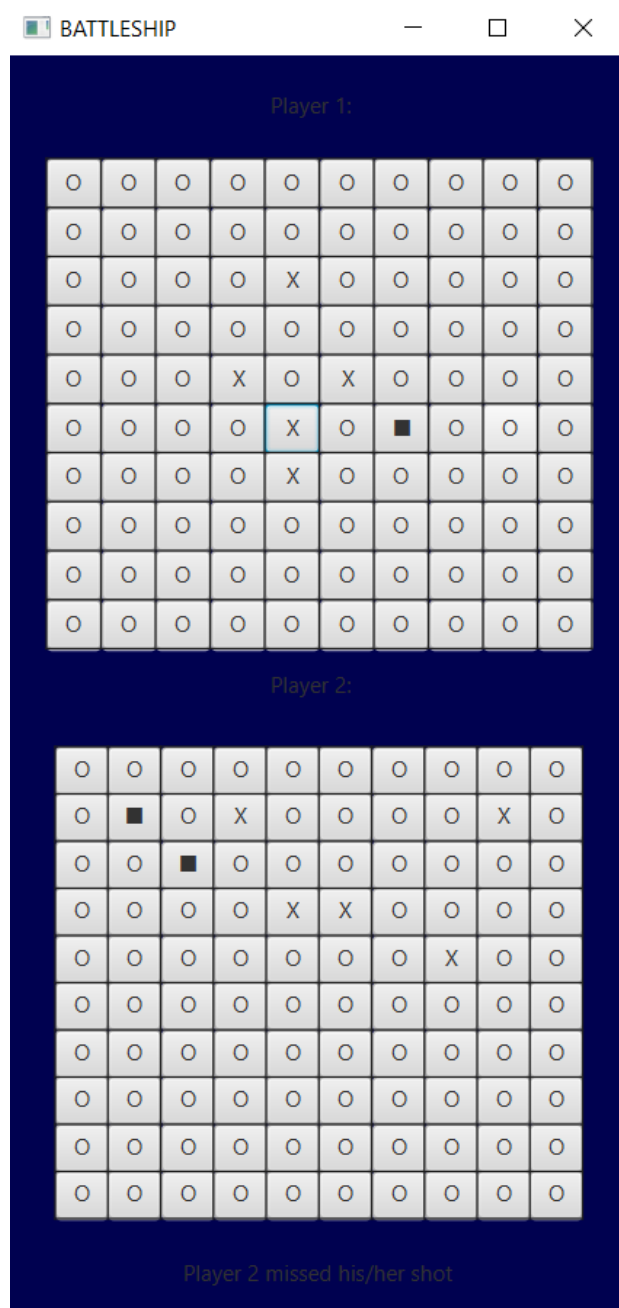
V téhle třídě se nachází podstata pro funkčnost programu. Zde je celá logika hry. Tahle třída mi zabrala nejvíce času z důvodu vymyšlení algoritmů atd. Nakonec jsem nepoužila nic příliš složitého a opět podmínky splnily většinu práce. Neměla jsem nějaké závažné problémy, jen se mi občas pletli hráči dohromady a ztrácela jsem se.

```
// CHOOSE RETURN KEYS
// 1 = SHIP ADDED
// 2 = PLAYER 1 CHOSE ALL SHIPS
// 3 = PLAYER 2 CHOSE ALL SHIPS
// -1 = SHIP IS ALREADY AT THIS LOCATION
// -2 = OTHER SHIP IS NEARBY
public int chooseShip(int x, int y, boolean playerOne) {
    if (playerOne) {
        if (this.playerOneMap[x][y] == 1)
            return -1;
        if ((x == 0 ? 0 : this.playerOneMap[x - 1][y]) == 1 ||
            (x == 9 ? 0 : this.playerOneMap[x + 1][y]) == 1 ||
            (y == 0 ? 0 : this.playerOneMap[x][y - 1]) == 1 ||
            (y == 9 ? 0 : this.playerOneMap[x][y + 1]) == 1)
            return -2;
        this.playerOneMap[x][y] = 1;
        this.playerOneAliveShips++;
    } else {
        if (this.playerTwoMap[x][y] == 1)
            return -1;
        if ((x == 0 ? 0 : this.playerTwoMap[x - 1][y]) == 1 ||
            (x == 9 ? 0 : this.playerTwoMap[x + 1][y]) == 1 ||
            (y == 0 ? 0 : this.playerTwoMap[x][y - 1]) == 1 ||
            (y == 9 ? 0 : this.playerTwoMap[x][y + 1]) == 1)
            return -2;
        this.playerTwoMap[x][y] = 1;
        this.playerTwoAliveShips++;
    }
}
```

Obr. 4 – Ukázka kódu ze třídy GameLogics

Závěr

Celkově si myslím, že se projekt vydařil. V budoucnu bych rozhodně chtěla vylepšit některé věci jako třeba grafiku. Také bych chtěla hru obohatit různými mody např. hráč proti počítači atd. Jinak program splňuje zadání a hlavně funguje, takže bych ho zhodnotila jako úspěšný. Je hodně prostoru pro vylepšení, ale na to, že programuji prvním rokem si myslím, že to je dostačující. Nepodařilo se mi úplně propracovat grafiku, ale časem se to pokusím zlepšit. Také nevyšlo tlačítko Reset. Jsem ráda, že přes všechny problémy je hra funkční a použitelná.



Obr. 5 – Program v průběhu hry

Bibliografie

Oracle JavaFX Scene builder. (n.d.). Retrieved from Oracle:

<https://www.oracle.com/java/technologies/javase/javafxscenebuilder-info.html>

Wikipedie - Lodě. (n.d.). Retrieved from Wikipedie: <https://cs.wikipedia.org/wiki/Lod%C4%9B>

Wikipedie - Lodě. (n.d.). Retrieved from Wikipedie:

https://cs.wikipedia.org/wiki/Lod%C4%9B#/media/Soubor:Battleship_game_board.svg

Wikipedie - Námořní bitva. (2011). Retrieved from Wikipedie:

[https://cs.wikipedia.org/wiki/Lod%C4%9B#/media/Soubor:Hra_n%C3%A1mo%C5%99n%C3%AD_bitva_\(1\).jpg](https://cs.wikipedia.org/wiki/Lod%C4%9B#/media/Soubor:Hra_n%C3%A1mo%C5%99n%C3%AD_bitva_(1).jpg)

Wikipedie-JavaFX-Vlastnosti. (n.d.). Retrieved from Wikipedie:

<https://cs.wikipedia.org/wiki/JavaFX>

Wikipedie-Lodě-pravidla. (2021, May 27). Retrieved from Wikipedie:

<https://cs.wikipedia.org/wiki/Lod%C4%9B>

Seznam obrázků

Obr. 1 – Desková verze Lodí	6
Obr. 2 – Lodě na papíře či počítači	6
Obr. 3 – Ukázka kódu ze třídy HelloController	8
Obr. 4 – Ukázka kódu ze třídy GameLogics	9
Obr. 5 – Program v průběhu hry	10