

Gymnázium, Praha 6, Arabská 14

Obor programování, vyučující Mgr. Jan Lána



Slepá mapa na města

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 hra, která při spuštění načte mapu České republiky nebo světa (dle výběru hráče) a bude načítat otázky typu "Najdi na mapě Prahu!". Při hodně přesném zásahu hráč dostane 10 bodů, za méně přesný 5 bodů, a když je příliš vedle, tak nedostane nic. Na konci hra vypíše kolik bodů získal hráč a z kolika. Na programování jsem použila jazyk Java a různé knihovny.

Abstract:

The result of this project should be a game, where the player chooses a map between the world and Czech Republic and then it asks questions of the type "Find Prague on the map!". If the player hits the exact place, he gets 10 points, if he is a little off, he gets 5 p. and if he is too far, he doesn't get any. In the end of the game, the program will show how many points has the player gotten from the maximum. For the programming I used language Java and different platforms.

Zadání práce:

Zadání ročníkové práce je vytvořit hru či program v Javě. Můj projekt je tedy slepá mapa na města. Program by měl dát možnost vybrat mapu. Poté hráč umísťuje požadovaná města a program určí, jak daleko je od cíle a podle toho přiřadí 0, 5 nebo 10 bodů. Program by měl sčítat body z předchozích kol a také jejich celkový součet, který pak vypíše po skončení hry. Hráč se může kdykoliv v průběhu hry vrátit do původního menu.

Obsah

1. Úvod	2
2. Popis tříd a metod	3
2.1 Logika programu	3
2.1.1 Třída <i>HelloController</i>	3
2.1.2 Třída <i>MenuController</i>	5
2.1.3 Třída <i>MapData</i>	6
2.1.4 Třída <i>Question</i>	7
2.2 Grafické rozvržení	7
2.2.1 Třída <i>hello-view.fxml</i>	7
2.2.2 Třída <i>menu-view.fxml</i>	8
3. Nezávislé komponenty programu	9
4. Použité nástroje	10
5. Komplikace a řešení	10
6. Nápady na budoucí zlepšení	11
7. Závěr	11
Reference	12
Seznam obrázků	13

1. Úvod

Původní myšlenka tohoto projektu vycházela z klasické zeměpisné slepé mapy, kde uživatel musí doplnit požadované město, stát, řeku, pohoří, jezero a tak dále. Elektronická verze by vypadala tak, že si uživatel v hlavním menu zvolí mapu území, které si chce procvičit, a program na něj bude házet otázky typu „Označ na mapě Londýn.“.

Začala jsem přemýšlet, jak by se dal takový program naprogramovat v Javě a jako nejlepší řešení mi přišlo přes zeměpisné souřadnice. To by ale znamenalo, že to bude fungovat nejlépe na města, a proto jsem se rozhodla se v tomto projektu spíše zaměřit na logiku a podstatu programu a udělat slepou mapu pouze na města. Samozřejmě je mnoho prostoru pro přidávání zeměpisných dat a v budoucnosti se tomu budu věnovat, ale prozatím to bude vypadat následujícím způsobem.

Po spuštění programu si musí uživatel zvolit buď mapu světa nebo České republiky. Poté se mapa spustí a nad ní se vypisují otázky a aktuální počet bodů. Uživatel má 10 sekund na označení požadovaného města a pak podle vzdálenosti mezi cílem a jeho odhadem se mu přiřadí body. Hráč může kdykoliv hru ukončit, vrátit se do menu a začít od znova. Po skončení hry program vypíše celkový zisk bodů hráče z maximálního počtu a poté se vrátí do hlavního menu.

2. Popis tříd a metod

2.1 Logika programu

V této kapitole popíšu podrobně funkce jednotlivých metod a tříd. Celkově se program skládá z pěti konzolových a ze dvou FXML – tříd. Dále ze dvou textových dokumentů, ve kterých jsou uloženy data pro generování otázek, a ze dvou map (světa a České republiky).

2.1.1 Třída *HelloController*

Nejvíce herní logiky se nachází ve třídě s názvem *HelloController*. Právě tam jsou vytvořeny všechny komponenty hry a propojují se všechny třídy. V této třídě se také vytváří hrací plocha pomocí *Canvas*, což představuje prázdný obdélník, kde program může zobrazovat samotný průběh hry a vstupy od uživatele.

Funkce první metody – *onNextButtonClick* je, že pokud program vyčerpal všechny otázky, tak se vrátí zpátky do hlavního menu. Pokud ne, tak vynuluje předchozí otázku a odpověď a zavolá metodu *askQuestion*, která vygeneruje novou otázku.

Dále máme metodu *goToMainMenu*, která restartuje hru a vrací jí do původního stádia.

Následuje metoda *askQuestion*, která byla již zmíněna. Ta volá metodu *generateQuestion*, která vytváří a pokládá otázku. Úkolem metody *askQuestion* je také časomíra. Prvně kontroluje, zda je otázka právě aktivní. Jestli ano, tak čas běží, jestli ne, tak přestane. Po položení otázky se ze seznamu jedna odečte a program běží do té doby, než dosáhne maximálního počtu otázek, který je předem zadáný.

Další metoda je *OnClick*, která kontroluje, zda je otázka právě aktivní. Pokud ano, tak hráč může kliknout do pole a na požadovaném místě se objeví černý puntík. Pomocí matematických funkcí, program vypočítá rozdíl mezi odhadem hráče a správnou lokací a podle toho mu přiřadí body. Pokud je uživatelův odhad blíže než 10 pixelů od cíle, tak dostane 10 bodů, pokud mezi 10 a 20 pixely, tak 5 bodů, a při moc nepřesném zásahu nedostane žádné body. Program vždy vypíše body po zásahu pomocí labelu *currentStatusLabel* a poté je přičte k celkovému součtu bodů, který je uložen v *pointsLabel*. Nakonec se zavolá metoda *showResult*. Toto je kód metody *OnClick*.

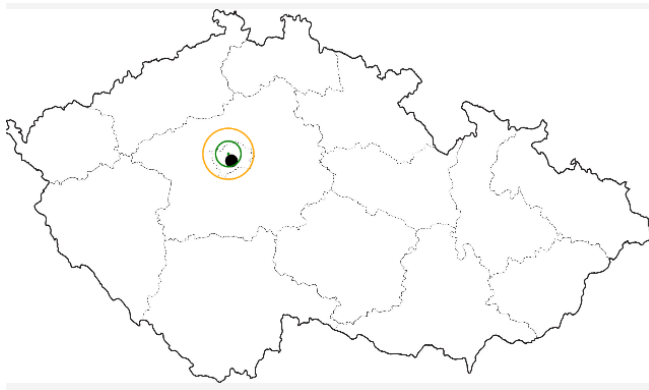
```

public void onClick(MouseEvent mouseEvent) {
    if (isQuestionActive == true) {
        timeCounterThread.interrupt();
        GraphicsContext gc = canvas.getGraphicsContext2D();
        gc.setFill(Color.BLACK);
        gc.fillOval(mouseEvent.getX() - 5, mouseEvent.getY() - 5, 10,
10);

        double distance = Math.sqrt(Math.pow(mouseEvent.getX() - x, 2)
+ Math.pow(mouseEvent.getY() - y, 2));
        int accuracy = 0;
        if (distance < 10) {
            points += 10;
            pointsLabel.setText("You have " + points + " points");
            currentStatusLabel.setText("You earned 10 points!");
        } else if (distance < 20) {
            points += 5;
            pointsLabel.setText("You have " + points + " points");
            currentStatusLabel.setText("You earned 5 points!");
        }
        else{
            currentStatusLabel.setText("Too far, you don't get any
points!");
        }
        showResult();
    }
}

```

Dále byla zavolaná metoda *showResult*, která ukáže správnou odpověď pomocí barevných koleček viz. Obrázek 1. V rozmezí zeleného kolečka dostane



Obrázek 1 - Správné označení Prahy

uživatel 10 b. a mezi žlutým a zeleným - 5 b. Na konci této metody je podmínka, pokud došly otázky, tak se ukáže tlačítko *Back to main menu* a vypíše získaný počet bodů. Poslední příkaz této metody je, že pokud otázka není právě aktivní, tak hra nedovolí uživateli klikat.

Dalších několik metod se týká položení otázek. V první metodě *setQuestions* se nastaví

počet otázek a podle toho se spočítá maximální počet bodů. Program čerpá otázky ze seznamu, který je vytvořen pomocí dokumentů, kde jsou sepsaná města a jejich zeměpisné souřadnice. Tady je ukázka kódu této metody.


```

public void setQuestions(List<Question> questions, int
numberOfQuestions) {
    this.questions = questions;
    this.numberOfQuestions = numberOfQuestions < questions.size() ?
numberOfQuestions : questions.size();
    this.maxPoints = this.numberOfQuestions*10;
}

public void setConvertWidth(Function<Double, Double> convertWidth) {
    this.convertWidth = convertWidth;
}

public void setConvertLength(Function<Double, Double> convertLength) {
    this.convertLength = convertLength;
}

private void generateQuestion() {
    Question question = questions.remove(0);
    x = convertWidth.apply(question.getX());
    y = convertLength.apply(question.getY());
    cityNameLabel.setText(question.getCityName());
}

```

Metody *setConvertWidth* a *setConvertLength* mají funkci převést zeměpisné souřadnice na pixely. K tomu jsem využila lambda výrazy, které se ale nachází v jiné třídě, takže budou podrobněji popsány později.

Metoda *generateQuestion* položí otázku tím, že vezme data z textového dokumentu, a zároveň ji odebere ze seznamu všech otázek, aby se již neopakovala.

Následuje metoda *runGame*, která spouští hru. Na začátku vynuluje body a poté je přičítá po každé otázce a následně volá metodu *askQuestion*.

Poslední komponent třídy *HelloController* je *getTimeCounterThread*, což je vlákno, které běží paralelně s programem a odpočítává čas na zodpovězení dané otázky. Do té doby, než je otázka stále aktivní se odečítají sekundy a po kliknutí uživatele na požadované místo se čas zastaví. Pokud se dostane k nule, tak program vypíše, že už není čas, a ukáže správné město.

2.1.2 Třída *MenuController*

Tato třída se zabývá převážně logikou hlavního menu hry a správným zobrazením map. Jak jsem již zmínila, použila jsem lambda výrazy pro převedení zeměpisných souřadnic na pixely. Lambda výrazy jsou elegantní způsob, jak zapsat funkci, která jinak zabere více řádků pomocí seznamu parametrů oddělených čárkou, šipky (->) a těla funkce. V následujících dvou úsecích kódu jsou zobrazeny cesty k jednotlivým mapám v programu a použité lambda výrazy pro převod na pixely. U České republiky to funguje tak, že nejvýchodnější bod má západní délku přibližně 12 stupňů, poté je to vydělené 6.7675, protože to je vzdálenost mezi nejvýchodnějších a nejzápadnější bodem Česka (opět uvedená ve stupních).

U zeměpisné šířky je stejný postup. Nejsevernější bod se nachází v 51.1 stupních severní šířky a rozdíl mezi ním a nejižnějším bodem je 2.5 stupně.

```
private MapData mapDataCz = new
MapData("src/main/resources/com/example/rp_slepamapa/CZ.txt",
"SlepaMapaCr.jpg", 294, 511, x -> (x - 12.09138889)/6.7675 * 511, y ->
(51.05583333 - y)/2.503333333 * 294);
```

U mapy světa je to jednodušší na pochopení, protože víme, že je na mapě 180 poledníků a 90 rovnoběžek.

```
private MapData mapDataWorld = new
MapData("src/main/resources/com/example/rp_slepamapa/World.txt",
"SlepaMapaWorld3.jpg", 483, 960, x -> 480 + (x/180*480), y -> 242 -
(y/90*242));
```

Následují metody *runCZMap* a *runWorldMap*, které mají za úkol spustit příslušnou slepou mapu. Argumenty těchto tříd jsou *MapData.CZ* a *MapData.World*, ze kterých dostanou parametry k tomu, jak spustit mapu.

Metoda *runMap* je hlavní spouštějící metoda. Je volána oběma mapami a její parametr je *MapData*. To je samostatná třída, ve které jsou uložena nastavení pro zobrazení mapy. Následně otevírá třídu *hello-view.fxml*, kde je Scene Builder s podkladem pro otevření mapy. Poté si vyhledá cesty k mapě a otázkám, nastaví velikost okna a spustí hru.

Poslední metoda v této třídě je *readQuestions*. Její funkce je, že dokud jsou objekty v seznamu, tak bude načítat otázky. Má tři parametry – název města, souřadnice x a souřadnice y, které načte z příslušného textového dokumentu. Na konci promíchá otázky pomocí *shuffle*.

2.1.3 Třída *MapData*

V této třídě se nachází všechny data a parametry, týkající se vzhledu oken a map. S těmito daty pracuje třída *MenuController*, jak bylo již zmíněno v předchozí kapitole. Jsou tu také uloženy např. cesty k dokumentům s městy a souřadnicemi, a také k samotným obrázkům map. Toto je ukázka kódu z této třídy, který obsahuje zmíněné parametry.

```
private String pathToQuestions;
private String pathToMapImage;
private int height;
private int width;
private Function<Double, Double> convertWidth;
private Function<Double, Double> convertLength;
```

2.1.4 Třída *Question*

Tato třída se zabývá podklady pro tvorbu otázek. Každá otázka má tři atributy, a to název města, souřadnice zeměpisné délky a souřadnice zeměpisné šířky. Tyto údaje poté používá metoda *generateQuestion* ve třídě *HelloController* k vyhotovení otázek.

```
private String cityName;  
private double x;  
private double y;
```

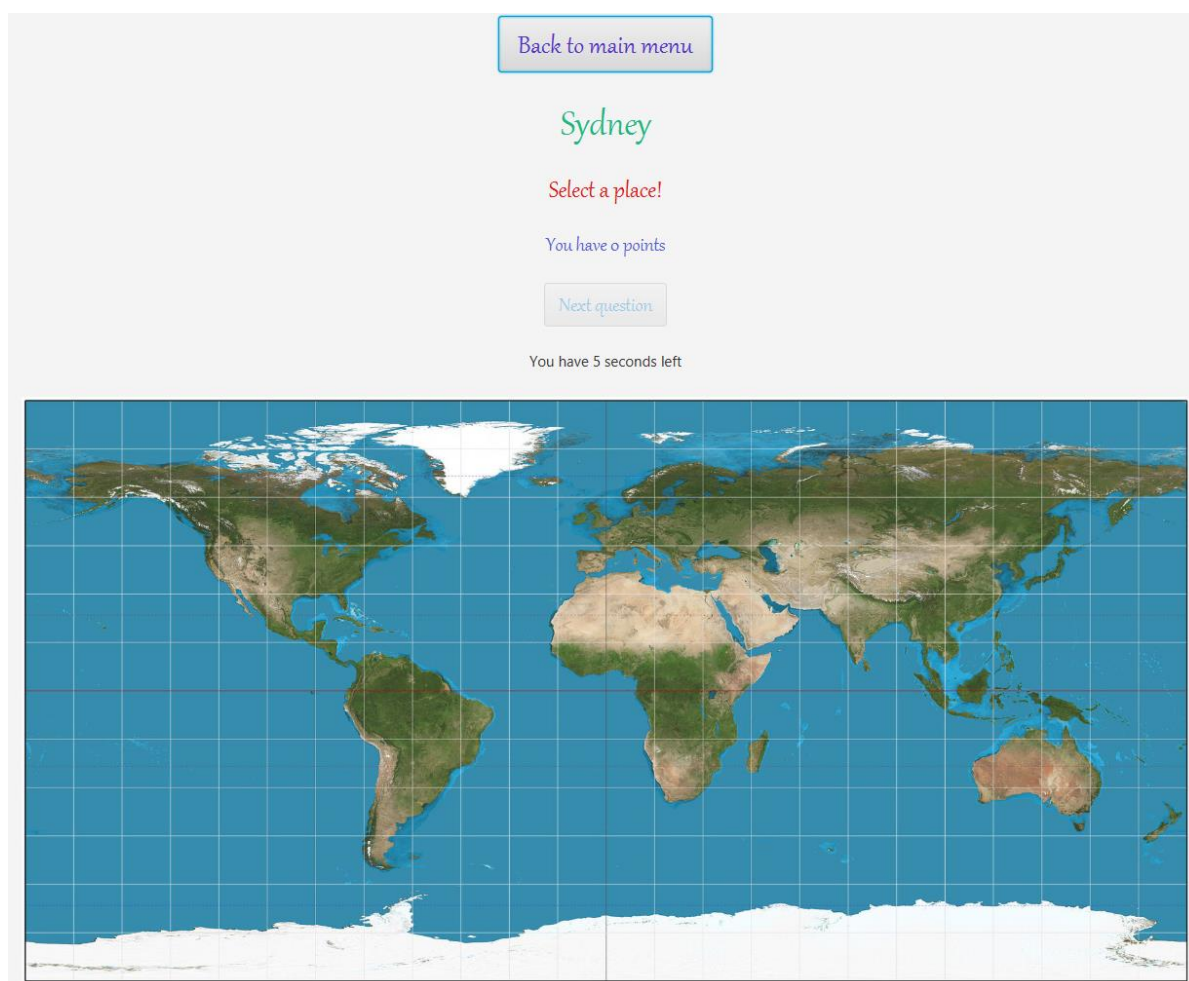
2.2 Grafické rozvržení

Pro grafickou stránku programu jsem použila *FXML* soubory, tedy *Scene Builder*. V programu se vyskytují dvě třídy tohoto typu a každá má jinou funkci. V následující podkapitole je popíšu detailněji.

2.2.1 Třída *hello-view.fxml*

Součástí této třídy je *Scene Builder*, který uchovává různá tlačítka, označení atd. Příklady některých prvků jsou *VBox* (vertikální box), což je kontejner pro ovládací prvky, který uspořádává objekty vertikálně a určuje jejich vnitřní rozestupy. Další elementy jsou různé ovládací prvky (tlačítka, popisky a obrázky), které jsou umístěny uvnitř *VBox*. Každý prvek má přidělený identifikátor (*fx:id*), který může být použit pro ovládání prvku z kódu. Například tlačítko *MenuButton* má přiřazenou událost *onAction*, která se spustí při kliknutí na tlačítko a volá metodu *onMenuButtonClick* ve třídě *HelloController*. Některé prvky mají také atributy, které určují jejich vlastnosti, jako je textový obsah, barva písma, font atd. V poslední části kódu je umístěn kontejner *StackPane*, který obsahuje *ImageView* a *Canvas*, které slouží k zobrazování a kreslení na plátno. *Canvas* má také přidělenou událost *onMouseClicked*, která se spustí při kliknutí na plátno a volá metodu *onClick* ve třídě *HelloController*. Na obrázku jsou vidět tlačítka a popisky, které byly vytvořeny právě v této třídě a poté s nimi pracují i jiné. Zkusila jsem si také pohrát i s fonty a barvami a na Obrázku 2 je zobrazena konečná verze pro variantu svět. Tady je ukázka kódu grafického upravení jednotlivých prvků. Toto je příklad pro název města.

```
<Label fx:id="cityNameLabel" textFill="#22b780">  
  <font>  
    <Font name="Gabriola" size="34.0" />  
  </font></Label>
```

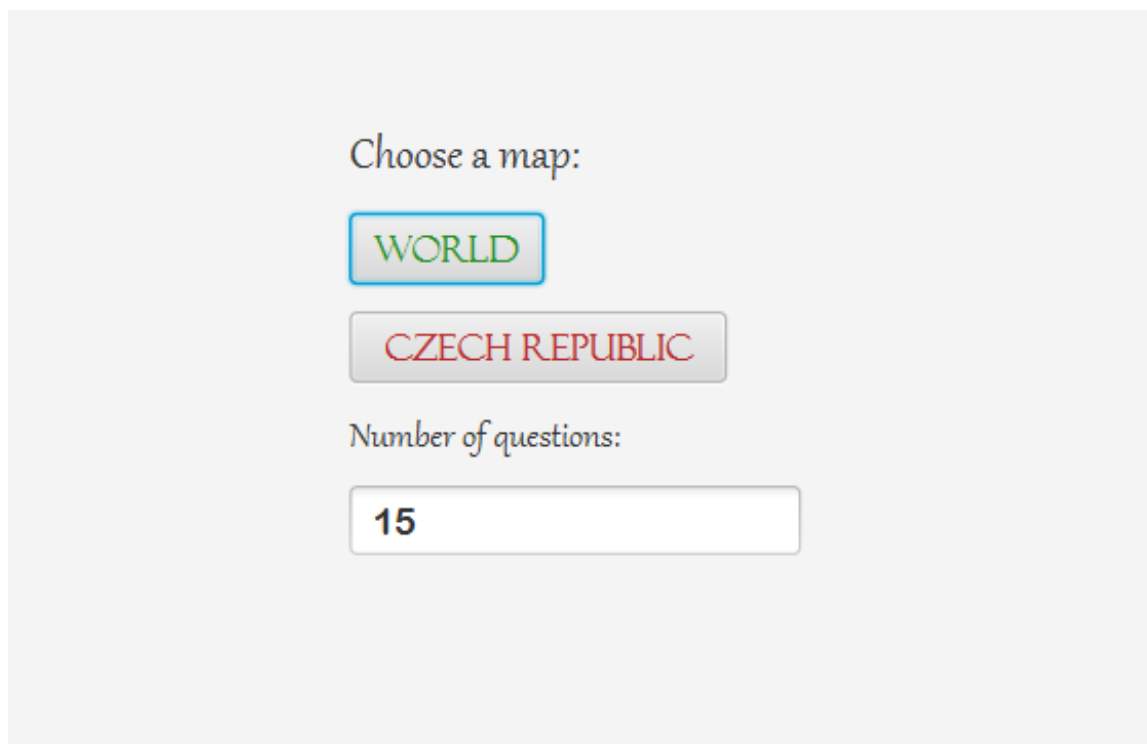


Obrázek 2 – Vzhled hry

2.2.2 Třída *menu-view.fxml*

Tato třída také využívá *Scene Builder*, ale tentokrát pro hlavní menu, na rozdíl od *hello-view.fxml*, která se zabývala hracím polem. Jeden z hlavních komponentů této třídy je kořenový prvek *Anchor pane* s parametry *prefHeight* a *prefWidth*, které určují rozměry rozhraní. Uvnitř kořenového prvku jsou další prvky, v tomto případě pouze jeden: *VBox*, což je kontejner pro umístění dalších prvků do sloupcového uspořádání. Tento prvek má několik atributů, jako je *layoutX*, *layoutY*, *prefHeight* a *prefWidth* určující pozici a velikost prvku. Uvnitř *VBox* jsou další prvky: dvě tlačítka s akčními funkcemi, *Label* a textové pole.

Tyto prvky slouží k vytvoření uživatelského rozhraní pro výběr mapy a počtu otázek pro hru. Každý prvek má vlastní atributy, jako je *onAction* pro zavolání metody při stisknutí tlačítka, *text* pro zobrazení textu a *textFill* pro nastavení barvy textu. Na obrázku 2 je finální vzhled hlavního menu hry.



Obrázek 3 - Vzhled hlavního menu

Na začátku hry si hráč zvolí mapu a případně si upraví počet otázek. Výchozí počet je 15. Poté se spustí příslušná mapa a hra začíná. Hráč se může kdykoliv vrátit do hlavního menu a případně začít hru od znova.

3. Nezávislé komponenty programu

Zde už se nejedná o třídy či metody, ale pouze o čtyři přílohy, které jsou nezbytnou součástí hry. Právě na nich se zakládá celá podstata hry. Jde konkrétně o dvě mapy (České republiky a světa) a dva textové dokumenty (v prvním jsou uložena data pro česká města a v druhém pro světová). S těmito prvky pracují všechny ostatní třídy.

Obě mapy jsou importovány jako obrázky ve formátu *jpg*. Měla jsem menší problém s nalezením správné mapy, protože ne každá by pak odpovídala souřadnicím. Rozhraní obrazu by mělo poměrově odpovídat rovnoběžkám a poledníkům na zeměkouli pro nejpřesnější označení daného města.

Dále obsahem obou textových dokumentů jsou podklady pro zformulování otázek a odpovědí. Skládají se z názvu města, zeměpisné délky a zeměpisné šířky vždy oddělené mezerami. To je důvod, proč jsou víceslovní názvy oddělené podtržítkem. Textový dokument pro Česko obsahuje 20 známých měst a pro svět 50. V případě, že uživatel zadá větší počet otázek, než je k dispozici, tak se promíchají a budou se opakovat.

4. Použité nástroje

Jako vývojové prostředí jsem použila IntelliJ IDEA Community Edition a jako programovací jazyk Javu společně s knihovnou JavaFX. Java je objektově orientovaný programovací jazyk, který se často používá pro vývoj desktopových, webových a mobilních aplikací. (1)

JavaFX je jedna ze známých knihoven, spojované s Javou. Tato knihovna umožňuje tvorbu grafického uživatelského rozhraní pro aplikace napsané v Javě. JavaFX nabízí širokou paletu grafických prvků a efektů, jako jsou například tlačítka, textová pole, obrázky a animace. (1)

JavaFX Scene Builder je nástroj pro vizuální rozvržení, které uživatelům umožňuje rychle navrhovat uživatelská rozhraní aplikace JavaFX bez kódování. Uživatelé mohou přetahovat komponenty uživatelského rozhraní do pracovní plochy, upravovat jejich vlastnosti, používat šablony stylů a kód FXML pro rozvržení, které vytvářejí, je automaticky generován na pozadí. Výsledkem je soubor FXML, který lze zkombinovat s projektem Java navázáním uživatelského rozhraní na logiku aplikace. (2)

5. Komplikace a řešení

Ze začátku jsem měla dost problémů, někdy i s drobnostmi, ale poté jsem se dostala do tempa a pak to šlo dobře. Poměrně dlouho mi trvalo vymyslet převody ze stupňů na pixely, ale po vyhledávání na internetu a konzultacích s různými lidmi jsem na to přišla. Dále jsem nevěděla, jak propojit dva *FXML* soubory ve stejném programu, ale to jsem nakonec udělala tímto způsobem:

```
FXMLLoader loader = new FXMLLoader(getClass().getResource("hello-view.fxml"));
```

Poté s tím třídou pracuji dál bez problému. Dlouho jsem byla zaseklá u toho, že mi mapa Česka házela světová města při opětovném spuštění, ale to taky nebyla závažná chyba. Časomíra nebyla úplně jednoduchá, ale pak jsem to vyřešila pomocí lambda výrazů.

Ve výsledku bych řekla, že jsem neměla příliš závažné problémy, ale program nebyl jednoduchý.

6. Nápady na budoucí zlepšení

Jak jsem již zmínila, je spousta prostoru pro zlepšení v různých ohledech. Například přidání zeměpisných dat či různých úrovní obtížnosti. Měla jsem nápad, že by si hráč mohl zvolit úroveň složitosti, ale poté mi došlo, že by se to nedalo udělat objektivně. Původní myšlenka představovala slepou mapu na více zeměpisných objektů, ale vzhledem k tomu, že metoda přes zeměpisné souřadnice mi přišla více efektivní a propracovaná, i přes to, že funguje nejlépe na města, tak jsem se rozhodla to prozatím nechat takhle. Rozhodně mám v plánu pracovat s tímto programem i nadále a vylepšit ho. Další nápad, který jsem měla, je, že uživatel bude dostávat např. mince za rychlé zodpovězení otázek a poté by je mohl použít na odemčení nových map, prvků atd. Dále bych nějak zakomponovala i vlajky jednotlivých států a přidala více území (Evropa, USA, Afrika atd.).

7. Závěr

Myslím si, že se mi podařilo splnit úkol úspěšně a program více méně odpovídá původnímu zadání. Povedlo se mi vytvořit dva *FXML* soubory, které by byly navzájem propojeny, respektive že by spolupracovaly. Jeden je hlavní menu a druhý je hrací pole (mapa), která navazuje na menu. Dále se mi podařilo vymyslet způsob, jak zkoordinovat zeměpisné polohy s pixely, tak, aby to vycházelo přesně. Povedlo se mi také nastavit časomíru (10 sekund na každou otázku) a vypočítat, jak daleko je uživatelův odhad od cíle. Poté mu přiřadit příslušný počet bodů. Vydařilo se mi také propojit ty dvě mapy tak, aby měl uživatel možnost si vybrat v hlavním menu, ale aby se taky mohl kdykoliv vrátit a přepnout mapu.

Celkově bych považovala projekt za úspěšný, i přes komplikace, které vznikly cestou. Řekla bych, že jsem si hodně obohatila znalosti, týkající se Javy a získala jsem nové zkušenosti. Tato práce mi také pomohla k lepšímu rozvržení času během roku. Počítám s budoucím zlepšením, ale prozatím jsem s prací velmi spokojená.

Reference

1. OpenAI. *Chat GPT*. [Online] [Citace: 20. 04 2022.] <https://chat.openai.com/>.
2. JavaFX Scene Builder. *Oracle*. [Online] [Citace: 22. 04 2023.] <https://www.oracle.com/java/technologies/javase/javafxscenebuilder-info.html>.

Seznam obrázků

Obrázek 1 - Správné označení Prahy [4]

Obrázek 2 – Vzhled hry [8]

Obrázek 3 - Vzhled hlavního menu [9]