

Gymnázium, Praha 6, Arabská 14



Ročníkový Projekt

Adam Rubeš

Vědomostní Kvíz

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 _____

Adam Rubeš _____

Anotace

Cílem tohoto ročníkového projektu bylo vytvoření vědomostního kvízu. Za použití kvízu si může hráč vyzkoušet své znalosti z několika tematických okruhů a snažit se kvíz vyplnit na co nejlepší score za co nejkratší čas. Čas a score si může porovnat s jiným spoluhráčem na stejném zařízení.

Zadání

Hra kterou jsem si vybral je vědomostní Kviz.

Hráč bude mít na výběr ze 3 tematických oddělených kategorií otázek mezi kterými si bude moci vybrat z jakého tématu chce otázky dostávat.

Jedno kolo Kvizu sestává z 10 otázek které se náhodně vybírají. Poté co hráč zodpoví 10 otázek bude mu hrou vyměřena úspěšnost.

Za správnou odpověď +1b za špatnou +0. Na konci kola bude si hráč moci uložit svůj výsledek včetně času který potřeboval k jeho dosažení.



Obsah

1. Úvod	6
2. Nástroje	7
3. Grafické a multimediální rozhraní	8
4. Řetězce otázek a odpovědí	10
5. Aplikační logika	10
a. Třída „VedomostniQuiz“	10
b. Třída „FXMLDocumentControler“	10
I. Metoda „zmacknutoStart“	10
I. Metoda „novaOtazka“	12
II. Metoda „posuvNaDalsiOtazku“	13
III. Metoda „Tlačítka odpovědí“	13
IV. Metoda „zmenaTematu“	14
V. Metody „zmacknutoUlozit0...3“	14
VI. nahodnaDalsiOtazka	14
VII. Metoda „konecHry“	15
VIII. Zvuk a Animace	16
6. Jak spustit	17
7. Závěr	17
7. Obrázky	17
8. Zdroje	17

1.Úvod

Vědomostní Kvíz je program testující hráčovy znalosti a rychlost myšlení. Po spuštění hry stisknutím tlačítka start je hráči postupně ukázáno deset otázek a u každé musí vybrat jednu správnou odpověď ze čtyř možných. Na konci hry program ukáže vyhodnocení a čas za jaký hráč stihl zodpovědět všechny otázky. Své výsledky je možné si uložit přímo ve hře a tak umožnit jejich vzájemné porovnání.

2.Nástroje

K tvorbě svého ročníkového projektu jsem využil:

- Programovací jazyk Java
- Vývojové prostředí NetbeansIDE
- GUI editor SceneBuilder
- Grafickou knihovnu JavaFX

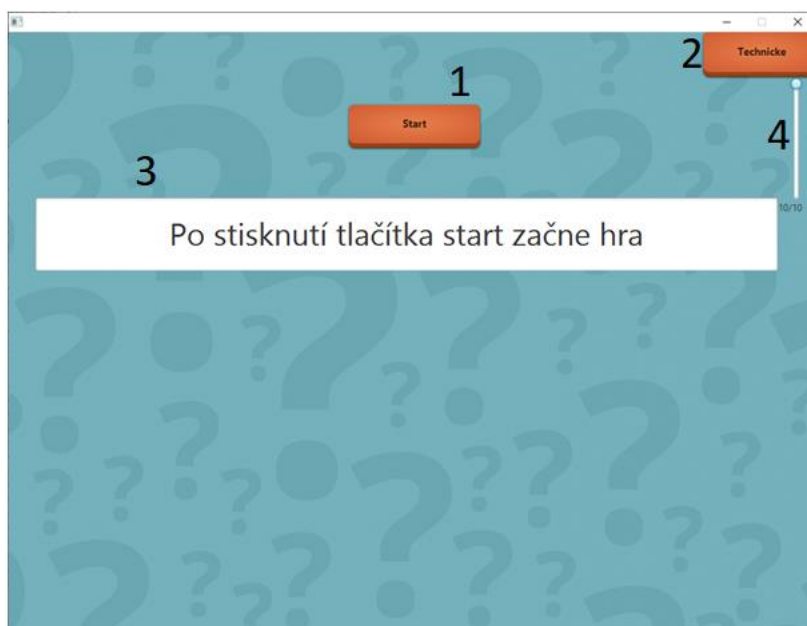


NetBeans.IDE

3. Grafické a multimediální rozhraní

Pro tvorbu grafického a zvukového rozhraní jsem využil knihovny Javy Javafx. Program si v souboru XML udržuje informace o tom jak má výsledná scéna vypadat. Soubor CSS nadále upravuje vzhled některých prvků scény. Soubor XML byl vytvořen za pomoci SceneBuilderu.

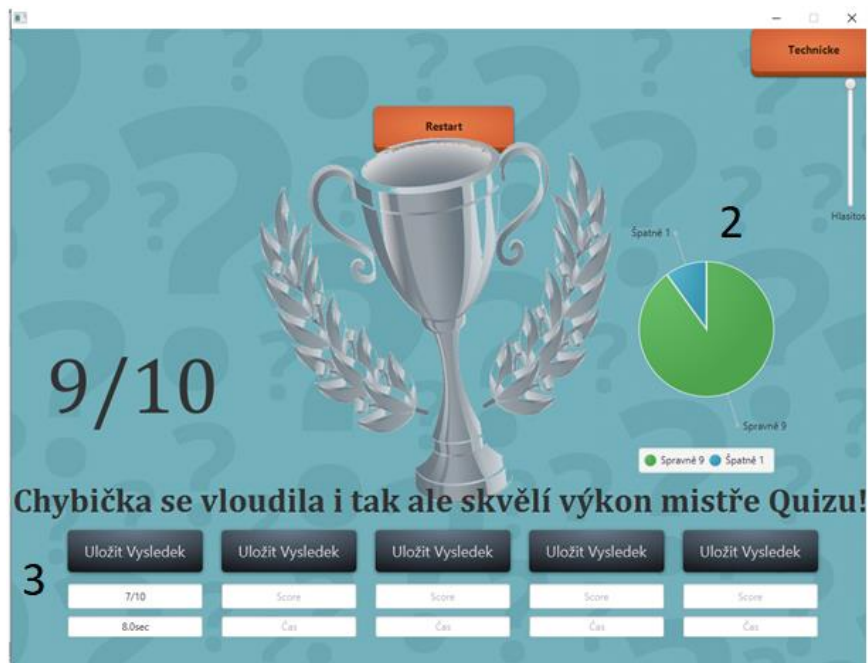
- Po spuštění hry se hráči zobrazí úvodní obrazovka s několika interaktivními prvky:
 - Tlačítko „Start“ spustí jedno kolo herního kvízu
 - Výběr tematického okruhu otázek
 - Textové pole ve kterém se zobrazují základní instrukce a po spuštění kvízu otázky.
 - Ovladač hlasitosti zvuků které program vydává



- Po stisknutí tlačítka start se zobrazí GUI kvízu samotného:
 - Tlačítko „Start“ se změní na tlačítko Restart
 - Zobrazí se čtyři tlačítka nesoucí v sobě potencionální odpovědi na otázky
 - Zobrazí se postupová lišta ukazující postup ke konci hry
 - Zobrazí se label s průběžným ukazatelem score



- Po zodpovězení desíti otázek se zobrazí GUI s vyhodnocením
Zobrazí se GUI konce hry.
 1. Ve středu scény se zobrazí rotující obrázek podle výsledků kvízu.
 2. Zobrazí se graf úspěšnosti
 3. Zobrazí se políčka pro uložení výsledku



4. Řetězce otázek a odpovědí

Otázky a odpovědi jsou v programu uloženy prostřednictvím pole řetězců. V případě otázek se jedná o jednoduchý jednorozměrné pole řetězců s indexem od [0] do [16]. Tímto indexem se pak program nadále řídí při přiřazování otázek a vyhodnocování správných odpovědí. V případě otázek se jedná o dvourozměrné pole s indexem „výšky“ od [0] do [16] a „šířky“ [0] až [3]. Index výšky odpovídá indexu otázek, tedy otázka s indexem [5] bude mít čtyři odpovědi na pozici otázky [5] x [0]...[3].

```
String[] poleOtazekHistorie = new String[]{
    "Jak se jmenoval první československý prezident?",//[0]
    "Ve kterém roce skončila druhá světová válka?",//[1]
    "Kolik manželek měl Karel IV.?",//[2]
    "Kdo byl upálen v Kostnici?",//[3]
    "Kdo přivedl Slované na naše území?",//[4]
    ...
};

String[][] poleOdpovediHistorie = new String[][]{
    {"T.G. Masarik", "Eduard Beneš", "Václav Havel", "Václav Klaus"},//spravne [0]
    {"1948", "1945", "1918", "1939"},//spravne [1]
    {"Žádnou", "6 manželek", "3 manželky", "4 manželky"},//spravne [3]
    {"Tycho De Brahe", "Jan Žižka", "Jan Hus", "Jan Kepler"},//spravne [2]
    {"Praotec Čech", "Kněžna Libuše", "Kníže Krok", "Kučec Sám"},//spravně [0]
    ...
};
```

5. Aplikační logika

Program se skládá ze dvou tříd „FXMLDocumentControler“ a „VedomostniQuiz“ a následně ze souborů XML a CSS.

a. Třída „VedomostniQuiz“

- Obsahuje metodu „main“
- Vykresluje grafické rozhraní
- Využil jsem předem vygenerovaný template od Netbeans

b. Třída „FXMLDocumentControler“

Tato třída obstarává veškerou logiku kvízu. Jsou zde uloženy všechny metody a proměnné které program nadále využívá.

I. Metoda „zmacknutoStart“

Tato metoda je volána při stisku tlačítka „Start/Reset“. Jejím úkolem je měnit viditelnost grafických prvků především na začátku a na konci kvízu. Při každém stisknutí zavolá metodu „nahodnaDalsiOtazka“ která promíchá pořadí otázek. Následně v závislosti na to v jaké je hra fázi (na začátku nebo v průběhu/na konci) se odehraje jeden ze scénářů.

- Možnost 1 (začátek hry)

V případě že začíná hra tak metoda zobrazí všechny grafické prvky které jsou zapotřebí k samotnému průběhu kvízu (Tlačítka odpovědí, score..) a zavolá metodu „novaOtazka“ která dosadí do pozic pro otázku a odpovědi odpovídající řetězce z polí řetězců. Na konec se tlačítko start přejmenuje na „Restart“ a zaznamená se aktuální čas který se použije ke zjištění celkového času.

- Možnost 2 (restart běžící nebo ukončené hry)

V druhé možnosti která nastane pouze když kolo kvízu již započalo nebo skončilo se restartují proměnné klíčové pro běh kvízu. Následně se skryjí všechny grafické prvky které byly vytvořeny po konci nebo při startu hry. Tlačítko restart se tedy znovu přejmenuje na tlačítko „Start“ a je program připraven na nové kolo kvízu.

```
@FXML
void zmacknutoStart(ActionEvent event) {
    nahodnaDalsiOtazka();
    prehrajClick();

    if (kolikataOtazka == 0) { // začátek hry
        poleOtazky.setVisible(true);
        odpoved0.setVisible(true);
        odpoved1.setVisible(true);
        odpoved2.setVisible(true);
        odpoved3.setVisible(true);
        postupovaLista.setVisible(true);
        grafOdpovedi.setVisible(false);
        tlacitkoTematu.setMouseTransparent(true);
        scoreText.setVisible(true);

        novaOtazka();
        cisloOtazky = intArray[kolikataOtazka];

        scoreText.setText(scoreString + "/" +olikataOtazka);
        tlacitkoStart.setText("Restart");
        start = System.currentTimeMillis();
    } else if (kolikataOtazka > 0) { // restart hry - obnova proměnných
        pocetOtazek = poleOtazekTechnika.length;
        cisloOtazky = 0;
        score = 0;
        scoreString = String.valueOf(score);
        cisloOtazkyDouble = 0;
       olikataOtazka = 0;

        poleOtazky.setText("Po stisknutí tlačítka start začne hra");
        scoreText.setText(scoreString + "/" +olikataOtazka);
        postupovaLista.setProgress(cisloOtazkyDouble);
        grafOdpovedi.setVisible(false);
        odpoved0.setVisible(false);
        odpoved1.setVisible(false);
        odpoved2.setVisible(false);
        odpoved3.setVisible(false);
        tlacitkoTematu.setMouseTransparent(false);
        obrazek.setVisible(false);
        labelVysledku.setVisible(false);
        labelVelkeVysledky.setVisible(false);
        HBoxVysledku.setVisible(false);

        tlacitkoStart.setText("Start");
    }
}
```



I. Metoda „novaOtazka“

Metoda využívaná pouze k nastavení textů otázek a odpovědí do textového pole otázky a tlečitek pro odpovědi. Metoda nastavuje otázky a odpovědi podle toho jaký tematický okruh je zrovna vybrán. V případě že je vybrán tematický okruh náhodný tak má metoda integer „randomNum“ podle kterého poté vybere jednu ze sad otázek.

odpoved0.setText(**poleOtazekTechnika**[**intArray**[**kolikataOtazka**]]);

Žlutá- pole řetězců (otázek) ze kterého bude otázka vybrána.

Modrá- pole náhodně promíchaných čísel od [0] [16] které slouží jako náhodně vygenerovaný index pro pole „poleOdpovediTechnika“ ukazující na jednu z otázek.

Zelená- Proměnná která slouží jako index pro „intArray“ která se zpravidla zvětšuje o jednu po každé otázce tedy vždy vybere náhodné další číslo z „intArray“.

```
public void novaOtazka() {
    if (vyberSady == 0) {
        poleOtazky.setText(poleOtazekTechnika[intArray[kolikataOtazka]]);
        odpoved0.setText(poleOdpovediTechnika[intArray[kolikataOtazka]][0]);
        odpoved1.setText(poleOdpovediTechnika[intArray[kolikataOtazka]][1]);
        odpoved2.setText(poleOdpovediTechnika[intArray[kolikataOtazka]][2]);
        odpoved3.setText(poleOdpovediTechnika[intArray[kolikataOtazka]][3]);

    } else if (vyberSady == 1) {
        poleOtazky.setText(poleOtazekHistorie[intArray[kolikataOtazka]]);
        odpoved0.setText(poleOdpovediHistorie[intArray[kolikataOtazka]][0]);
        odpoved1.setText(poleOdpovediHistorie[intArray[kolikataOtazka]][1]);
        odpoved2.setText(poleOdpovediHistorie[intArray[kolikataOtazka]][2]);
        odpoved3.setText(poleOdpovediHistorie[intArray[kolikataOtazka]][3]);

    } else if (vyberSady == 2) {
        poleOtazky.setText(poleOtazekVseobecne[intArray[kolikataOtazka]]);
        odpoved0.setText(poleOdpovediVseobecne[intArray[kolikataOtazka]][0]);
        odpoved1.setText(poleOdpovediVseobecne[intArray[kolikataOtazka]][1]);
        odpoved2.setText(poleOdpovediVseobecne[intArray[kolikataOtazka]][2]);
        odpoved3.setText(poleOdpovediVseobecne[intArray[kolikataOtazka]][3]);

    } else if (vyberSady == 3) {
        int randomNum = ThreadLocalRandom.current().nextInt(0, 2 + 1); //https://stackoverflow.

        if (randomNum == 0) {
            poleOtazky.setText(poleOtazekTechnika[intArray[kolikataOtazka]]);
            odpoved0.setText(poleOdpovediTechnika[intArray[kolikataOtazka]][0]);
            odpoved1.setText(poleOdpovediTechnika[intArray[kolikataOtazka]][1]);
            odpoved2.setText(poleOdpovediTechnika[intArray[kolikataOtazka]][2]);
            odpoved3.setText(poleOdpovediTechnika[intArray[kolikataOtazka]][3]);

        } else if (randomNum == 1) {
            poleOtazky.setText(poleOtazekHistorie[intArray[kolikataOtazka]]);
            odpoved0.setText(poleOdpovediHistorie[intArray[kolikataOtazka]][0]);
            odpoved1.setText(poleOdpovediHistorie[intArray[kolikataOtazka]][1]);
            odpoved2.setText(poleOdpovediHistorie[intArray[kolikataOtazka]][2]);
            odpoved3.setText(poleOdpovediHistorie[intArray[kolikataOtazka]][3]);

        } else if (randomNum == 2) {
            poleOtazky.setText(poleOtazekVseobecne[intArray[kolikataOtazka]]);
            odpoved0.setText(poleOdpovediVseobecne[intArray[kolikataOtazka]][0]);
            odpoved1.setText(poleOdpovediVseobecne[intArray[kolikataOtazka]][1]);
            odpoved2.setText(poleOdpovediVseobecne[intArray[kolikataOtazka]][2]);
            odpoved3.setText(poleOdpovediVseobecne[intArray[kolikataOtazka]][3]);

        }

    }
}
```

II. Metoda „posuvNaDalsiOtazku“

Tato metoda se stará o to aby se po každém správném či špatném tipu odpovědi kvíz přesunul na další otázku. Na začátku zvýší proměnnou „kolikataOtazka“ o jedna. Aktualizuje proměnné pro vyhodnocení otázek a pro postupovou lištu, hned poté co se zaktualizují proměnné se zaktualizují i na nich závislé prvky. Následně volá metodu „novaOtazka“. (Metoda „novaOtazka“ nemusí být nutně samostatnou metodou ale pro orientaci v kódu jsem je rozdělil od sebe)

```
public void posuvNaDalsiOtazku() {  
  
    nekolikataOtazka++;  
  
    cisloOtazky = intArray[kolikataOtazka];  
    cisloOtazkyDouble = cisloOtazkyDouble + 0.1;  
    novaOtazka();  
    postupovaLista.setProgress(cisloOtazkyDouble);  
    scoreString = String.valueOf(score);  
    scoreText.setText(scoreString + "/" + nekolikataOtazka);  
}
```

III. Metoda „Tlačítka odpovědí“

Každé tlačítko má předem vymezené pro které otázky je spávnou volbou a pro které ne. Každé z tlačítek má svou vlastní on action metodu s názvy zmacknuto0 zmacknuto3.

Struktura těchto metod je v základu stejná, jediný rozdíl zde tvoří podmínky. Ty vytváří celkem 3 hlavní scénáře které mohou nastat

- Vybrali jste správnou odpověď

V takovém případě se proměnná score zvýší o jedna. Poté pokud se nejedná o desátou otázku v kole tak zavolá metodu „posuvNaDalsiOtazku“ v opačném případě zavolá metodu „konecHry“.

Vybrali jste špatnou odpověď, Pokud jste vybrali špatnou odpověď pak v závislosti na tom, jestli se jedná o 10 otázku kvízu se zavolá metoda „konecHry“ nebo „posuvNaDalsiOtazku“.

```
@FXML  
void zmacknuto1(ActionEvent event) {  
    if (cisloOtazky == 1 || cisloOtazky == 5 || cisloOtazky == 13) {  
        score++;  
        prehrasSpravne();  
        if (kolikataOtazka != 9) {  
            posuvNaDalsiOtazku();  
        } else if (kolikataOtazka == 9) {  
            konecHry();  
        }  
    } else if (kolikataOtazka == 9) {  
        konecHry();  
        prehrasSpatne();  
    } else if (cisloOtazky != 0 || cisloOtazky != 4 || cisloOtazky != 9) {  
        posuvNaDalsiOtazku();  
        prehrasSpatne();  
    }  
}
```

IV. Metoda „zmenaTematu“

Tato metoda je volána při zmáčknutí tlačítka pro výběr sady otázek. Upravuje proměnou „vyberSady“ a podle ní se pak řídí například metoda „novaOtazka“ při výběru otázek. Série „Ifů“ se dá nahradit „switchem“.

```
@FXML
void zmenaTematu(ActionEvent event) {
    prehrayClick();
    if (vyberSady == 0) { //0
        vyberSady++;
        tlacitkoTematu.setText("Historicke");

    } else if (vyberSady == 1) { //1
        vyberSady++;
        tlacitkoTematu.setText("Vseobecne");

    } else if (vyberSady == 2) { //2
        vyberSady++;
        tlacitkoTematu.setText("Nahodne");

    } else if (vyberSady == 3) { //3

        tlacitkoTematu.setText("Technicke");
        vyberSady = 0;

    }
}
```

V. Metody „zmacknutoUlozit0...3“

Metoda zmacknutoUlozit(0...3) je metoda volaná při stisku tlačítka pro uložení výsledku. Umožňuje na konci hry uložit své score a čas za který jste herní kolo kvízu vyplnili.

```
@FXML
void zmacknutoUlozit0(ActionEvent event) {
    ulozeneScore0.setText(scoreString + "/10");
    ulozenyCas0.setText(ubehlyCasString + "sec");
}
```

VI. nahodnaDalsiOtazka

Tato metoda prohází pořadí otázek kvízu. Tato metoda náhodně prohází prvky v poli čísel od 0 do 16.

```
public void nahodnaDalsiOtazka() { https://www.journaldev.com/32661/shuffle-array-java

    List<Integer> intList = Arrays.asList(intArray);

    Collections.shuffle(intList);

    intList.toArray(intArray);

    //System.out.println(Arrays.toString(intArray));
}
```

VII. Metoda „konecHry“

Metoda „konecHry“ je metoda volaná na konci jednoho kola kvízu. Její účel je vykreslit závěrečnou obrazovku s vyhodnocením. Její první operace je zaznamenání času a jeho zapamatování. Následně začne měnit viditelnost jednotlivých grafických prvků na scéně

```
public void konecHry() {
    end = System.currentTimeMillis();
    sec = (end - start) / 1000;
    poleOtazky.setText("Konec Hry");
    ubehlyCasString = String.valueOf(sec);

    cisloOtazkyDouble = cisloOtazkyDouble + 0.1;

    postupovaLista.setProgress(cisloOtazkyDouble);

    scoreString = String.valueOf(score);
    scoreText.setText(scoreString + "/10");

    pocetSpatnych = 10 - score;

    grafOdpovedi.setVisible(true);

    ObservableList<PieChart.Data> pieChartData //https://docs.oracle.com/javafx/2/
        = FXCollections.observableArrayList(
            new PieChart.Data("Správně " + score, score),
            new PieChart.Data("Špatně " + pocetSpatnych, pocetSpatnych));

    grafOdpovedi.setData(pieChartData);
    grafOdpovedi.setStartAngle(90);

    odpoved0.setVisible(false);
    odpoved1.setVisible(false);
    odpoved2.setVisible(false);
    odpoved3.setVisible(false);
    postupovaLista.setVisible(false);
    poleOtazky.setVisible(false);
    labelVelkeVysledky.setVisible(true);
    labelVelkeVysledky.setText(scoreString + "/10");
    scoreText.setVisible(false);
    HBoxVysledku.setVisible(true);

    if (score == pocetOtazek) {
        obrazek.setVisible(true);
        labelVysledku.setVisible(true);
        zobrazObrazek(koruna);
        animaceToceni(obrazek);
        labelVysledku.setText("Úžasný výkon bezchybný králi Quizů!");
    } else if (pocetSpatnych == 1) {
        obrazek.setVisible(true);
        labelVysledku.setVisible(true);
        zobrazObrazek(stribrnyPohar);
        animaceToceni(obrazek);
        labelVysledku.setText("Chybička se vloudila i tak ale skvělí výkon mistře Quizu!");
    } else if (pocetSpatnych == 2 || pocetSpatnych == 3 || pocetSpatnych == 4) {
        obrazek.setVisible(true);
        labelVysledku.setVisible(true);
        zobrazObrazek(bronzovyPohar);
        animaceToceni(obrazek);
        labelVysledku.setText("Průměrný výkon, cvič ať se staneš Mistrem Quizu!");
    } else if (pocetSpatnych >= 5) {
        obrazek.setVisible(true);
        labelVysledku.setVisible(true);
        zobrazObrazek(knihy);
        animaceToceni(obrazek);
        labelVysledku.setText("Chabý výkon, vrat se až si oprášíš znalosti!");
    }
}
```


VIII. Zvuk a Animace

a. Metody „přehrajSpravne“/“Spatne“/“Click“

Tyto tři metody fungují prakticky stejně, jediné v čem se liší je soubor MP3 který přehrávají. Tyto metody by se daly předělat na jednu metodu a dodávat jí název souboru v podobě parametru.

```
public void prehrajSpravne() {
    AudioClip zvuk1 = new AudioClip(this.getClass().getResource("spravneZvonek1.mp3").toString());
    zvuk1.setVolume(hlasitost / 10);
    zvuk1.play();
}
```

b. Metoda „animacetoceni“

Tato metoda se stará o animaci trofeje na vyhodnocovací obrazovce kvízu. Skládá se ze dvou hlavních pohybů: Rotace a zvětšování a zase návrat do původní velikosti.

```
public void animaceToceni(ImageView obraz) {

    rotaceTam = new RotateTransition();
    rotaceTam.setNode(obraz);
    rotaceTam.setDuration(Duration.millis(1500));
    //rotaceTam.setInterpolator(Interpolator.LINEAR);
    rotaceTam.setCycleCount(TranslateTransition.INDEFINITE);
    rotaceTam.setByAngle(360);
    rotaceTam.setAxis(Rotate.Z_AXIS);
    rotaceTam.playFromStart();

    zmenaVelikosti = new ScaleTransition();
    zmenaVelikosti.setNode(obraz);
    zmenaVelikosti.setDuration(Duration.millis(750));
    zmenaVelikosti.setCycleCount(TranslateTransition.INDEFINITE);
    //zmenaVelikosti.setInterpolator(Interpolator.LINEAR);
    zmenaVelikosti.setByX(0.2);
    zmenaVelikosti.setByY(0.2);
    zmenaVelikosti.setAutoReverse(true);
    zmenaVelikosti.playFromStart();

}
```

c. Ovládání hlasitosti a přehrávání hudby

Ovládání hlasitosti a přehrávání hudby je napsáno v metodě initialize. Tato metoda je spuštěna hned po vykreslení scény. Za využití „Listeneru“ který hlídá pohyb na „Slideru“ se pak upravuje hlasitost přehrávaného audia.

```
@Override
public void initialize(URL arg0, ResourceBundle arg1) { // https://www.youtube.com/watch?v=X9mEBGXX3dA&t=302s překonvert

    String path = new File("src\\testvedomostniquiz\\hudbanapozadi2.mp3").getAbsolutePath();
    me = new Media(new File(path).toURI().toString());
    mp = new MediaPlayer(me);
    nastaveniZvuku.setValue(mp.getVolume() * 100);
    mp.play();
    nastaveniZvuku.valueProperty().addListener((java.beans.Observable observable) -> {
        mp.setVolume(nastaveniZvuku.getValue() / 100);
    });

    nastaveniZvuku.valueProperty().addListener((ObservableValue<? extends Number> arg2, Number arg3, Number arg4) -> {
        hlasitost = (int) nastaveniZvuku.getValue();
        hlasitostInt = (int) hlasitost;
        ukazatelHlasitosti.setText(hlasitostInt + "/10");
    });
}
```


6.Jak spustit

Program se spouští v IDE po zkompilování.

7.Závěr

Se závěrem své tvorby tohoto ročníkového projektu jehož cílem bylo vytvořit vědomostní kvíz jsem spokojen. Zadáním bylo vytvořit vědomostní kvíz ve kterém bude možnost vybrat si z několika druhů otázek a na konci si porovnat výsledky s ostatními hráči. Tohoto cíle jsem myslím dosáhl. Zároveň jsem měl možnost se seznámit s knihovnou Javy JavaFX se kterou se velmi intuitivně pracuje a určitě jí budu využívat i nadále v projektech zahrnující GUI. Vymyslet samotnou logiku programu nebylo příliš obtížné a je velmi jednoduché hru rozšířit například o další sady otázek nebo o uživatelem nastavitelnou délku herních kol. Těchto cílů jsem bohužel nestihl dosáhnout kvůli nedostatku času. V projektu budu pokračovat v rámci soukromého cvičení s grafickým rozhraním.

7.Obrázky

Logo Javy	7
Logo NetBeansIDE	7
Logo JavaFX	7
Logo SceneBuilder	7

8.Zdroje

Release: JavaFX 2.2. (2013, December 06). Retrieved April 29, 2021, from <https://docs.oracle.com/javafx/2/charts/pie-chart.htm>

ProgrammingKnowledge (Director). (2016, January 26). *JavaFx tutorial for BEGINNERS 33 - adding volume slider TO MEDIAPLAYER* [Video file]. Retrieved April 29, 2021, from <https://www.youtube.com/watch?v=X9mEBGXX3dA&t=302s>

Java platform, standard Edition (Java se) 8. (n.d.). Retrieved April 29, 2021, from <https://docs.oracle.com/javase/8/javase-clienttechnologies.htm>

Java game ENGINE TUTORIAL #17 sound volume controls [Video file]. (2016, December 03). Retrieved April 29, 2021, from <https://www.youtube.com/watch?v=gZzhJz3BS0&t=144s>

Thenewboston (Director). (2015, April 26). *JavaFX Java GUI tutorial - 26 - CSS inline styles and selectors* [Video file]. Retrieved April 29, 2021, from <https://www.youtube.com/watch?v=MAiKpkQqb6Q>

Thenewboston (Director). (2015, April 29). *JavaFX Java GUI tutorial - 31 - introduction to FXML* [Video file]. Retrieved April 29, 2021, from https://www.youtube.com/watch?v=K7BOH-Ll8_g&t=201s

Java tutorial (how to make animation in Javafx) 01 [Video file]. (2017, January 19). Retrieved April 29, 2021, from <https://www.youtube.com/watch?v=MgD2Fxm7AA&t=438s>

P. (2019, September 02). How to shuffle an array in Java. Retrieved April 29, 2021, from <https://www.journaldev.com/32661/shuffle-array-java>

User42155, & Greg Case. (1957, September 01). How do I generate random integers within a specific range in Java? Retrieved April 29, 2021, from <https://stackoverflow.com/questions/363681/how-do-i-generate-random-integers-within-a-specific-range-in-java>