

Универзитет у Београду - Електротехнички факултет  
Катедра за рачунарску технику и информатику



# Развој 2Д видео-игре гађања покретних мета

дипломски рад основних академских студија

Ментор

др Игор Тартаља, в.проф.

Студент

Кристина Бабић  
2017/0245

Београд, 2021.

## Апстракт

Игра „Мете“ представља надградњу домаћег задатка из предмета Рачунарска графика на Електротехничком факултету Универзитета у Београду. Циљ игре јесте да тестира концентрацију, прецизност и рефлексе играча. Конкретно, играч треба да за што мањи интервал времена и са што већом прецизношћу погоди покретне мете које се појављују на екрану. Такмичарски дух подстиче се поређењем освојених бодова на крају игре са бодовима 10 најбољих играча. У намери да буде уписан на ову ранг листу, играча могу спречити и бројне препреке које отежавају погодак мета или имају улогу да одвуку његову пажњу. У посебној, едитор, апликацији корисник може да креира нови изглед сваке од стрељана и да након тога игра нову, самостално дизајнирану, верзију игре.

Надаље, у овом документу биће описан развој игре, њен концепт, интеракција са играчем, али и проблеми који се јављају приликом рада са *JavaFX* библиотеком, која је коришћена за израду пројекта.

## Садржај

1. Увод .....	4
2. Проблем .....	5
3. Функционална спецификација .....	6
4. Алати за развој и коришћене библиотеке.....	9
5. Пројекат софтвера .....	10
6. Имплементација игре.....	13
7. Закључак .....	15
8. Литература .....	16

## 1. Увод

Рачунарска графика је област рачунарства која проучава методе за синтезу слике, манипулацију визуелним садржајем и интеракцију са корисником. Она је данас, уз фотографију и телевизију трћи доминантан начин производње слика [1]. Визуелизацијом података рачунарска графика кориснику пружа потпуно природан начин комуникације са рачунаром због чега има велику примену, како у сфери мултимедије и видео игара, тако и у науци, индустрији, архитектури и дизајну.

Игра „Мете“ представља такозвану пуцачку видео-игру за једног играча (енг. *single player shooter video game*). Фокус игара из овог жанра јесте на потпуном поразу непријатеља користећи оружје које је на располагању [2]. Оне тестирају просторну оријентацију, рефлексе и снажљивост играча. Конкретно, игра „Мете“ направљена је по узору на игре често присутне на вашарима, где играч осваја награде прецизним погађањем покретних мета пуцањем из ваздушне пушке.

Једна од *shooter* игара која је послужила као инспирација, не толико за бодовање, колико за изглед интерфејса игре, јесте *Moorhuhn* [3], једна од најпознатијих Немачких видео игара за рачунар која датира још из раних двехиљадитих година и која је достигла велику славу у то време.

У наредном поглављу представљена је скица пројекта, ставке које је требало имплементирати и изложени су проблеми који су решавани приликом израде пројекта. Треће поглавље описује функционалну спецификацију односно ток игре и интеракцију са играчем. У четвртом делу наводи се који су алати коришћени за имплементацију игре. У петом делу дати су архитектура софтвера, кратак преглед пакета, класа и њихових релација. Неки од детаља имплементације и техничке карактеристике дати су шестом делу. Документација игре „Мете“ поред овог рада обухвата и прилог А – упутство за коришћење.

## 2. Проблем

Основни проблем који се јавља приликом развоја игре јесте осмишљавање концепта игре. На основу ставки и захтеваних функционалности које игра треба да оствари да би од пројекта димензије домаћег задатка била надограђена до дипломског рада, треба осмислити концепт и ток игре тако да игра буде што занимљивија, интуитивнија и да држи пажњу играча на дуже стазе.

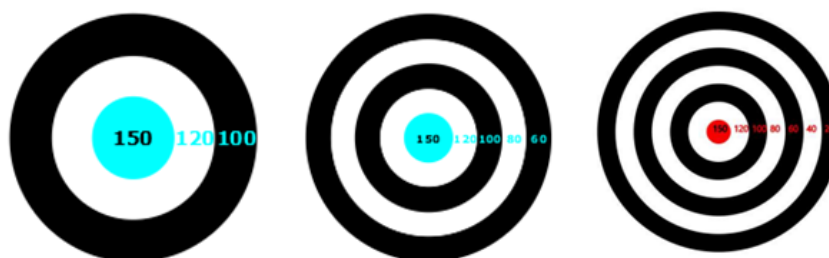
У наставку су дати неки од захтева игре:

- Креирање мета и осмишљавање начина на који ће се вршити бодовање поготка
- Приказ и обрада информација о поготку
- Осмишљавање стрељање
- Убацавање менија за лакшу комуникацију са апликацијом
- Осмишљавање начина рангирања и приказ ранг-листе
- Додавање музике и звучних ефеката
- Паузирање игре
- Конверзија облика и путање у SVG путању и обрнуто
- Осмишљавање начина на који ће корисник интераговати са едитором стрељање и његова имплементација
- Учитавање слике позадине и њено чување у JSON фајлу
- Омогућавање кориснику да може да доцрта основне геометријске облике и да на што интуитивнији начин може накнадно да помера и врши геометријске трансформације над објектима и њиховим крајњим тачкама.
- Имплементирање менија за бојења објеката и текста, као и менија за анимацију објеката препрека и мета различитим прелазима
- Снимање креираних стрељања у JSON фајл и њихово накнадно учитавање у жељени ниво.

### 3. Функционална спецификација

Целокупна функционална спецификација описана је у Прилогу А – Упутство за употребу. У наставку поглавља дати су њен преглед и кратак опис тока игре.

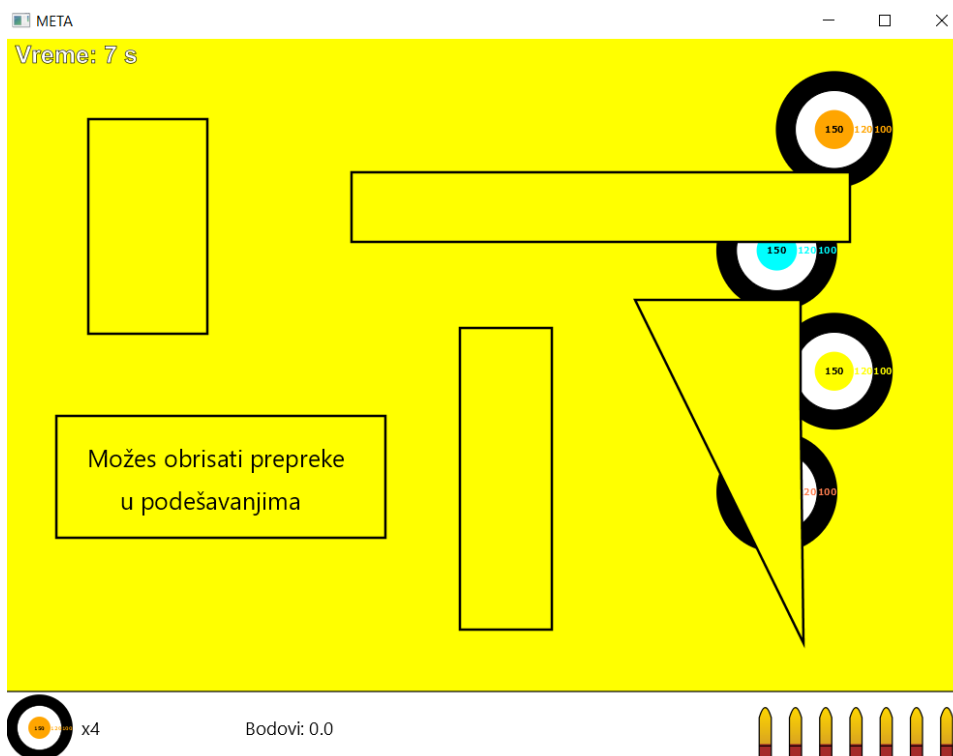
Циљ сваког од нивоа игре јесте погодити што више покретних мета (по могућству све) пуцањем из ваздушне пушке. Мете су кружног облика тако да се прецизност поготка мери удаљеношћу од њеног центра. У те сврхе постоје три врсте мета са 3, 5 и 7 прстенова који се различито бодују (Слика 3.1 – изглед мета).



Слика 3.1 – изглед мета

Мете се повремено удаљавају од играча. У тим тренутцима, њеним поготком играч остварује додатне (бонус) бодове који зависе од броја прстенова али и од удаљености мете. С обзиром да се крајњи резултат формира као разлика освојених бодова и десетоструког броја секунди проведених у игри у случају када је сигуран да ће погодити мету са највећом прецизношћу док је најудаљенија, играч може остварити бонус и до 100 бодова што премашује број бодова утрошених у време док се мета одаљи (2-3 секунде максимално) и доноси му добит од 70-80 бодова.

У току игре постоје препреке попут непробојног зида, тунела, ветрењача и других покретних и непокретних објеката, које за циљ имају да поремете концентрацију играча и отежају погодак мета њиховим делимичним или потпуним прекривањем (Слика 3.2 – приказ стрељање са препрекама које прекривају мете). Притом, поред препрека које га ометају, играч има и ограничен број метака у шаржеру (већи од броја мета на нивоу) које може да искористи на сваком од нивоа. Уколико истроши сву муницију пре него што уништи све мете, ниво се зауставља, а играчу пружа шанса да исти ниво покуша поново да одигра или да пређе на наредни ниво.



Слика 3.2 – приказ стрелјане са препрекама које прекривају мете

Целокупна интеракција између корисника и игре обавља се преко корисничког интерфејса који садржи следеће меније и прозоре:

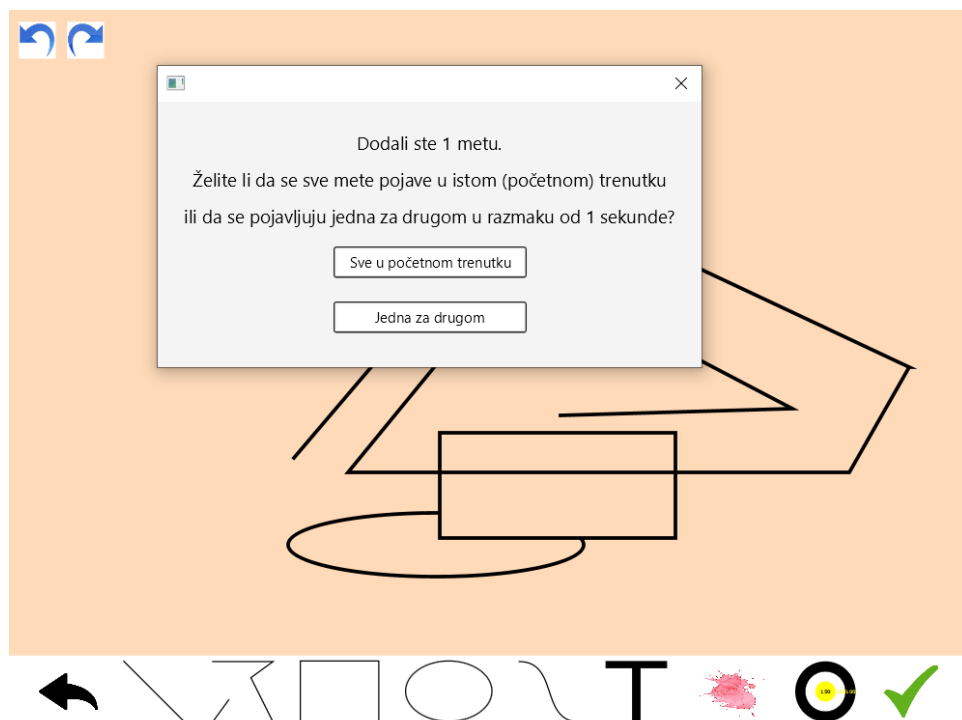
- Почетни мени – мени који се појављује приликом покретања апликације
- Подешавања игре – мени преко кога играч мења јачину позадинске музике и звучних ефеката, одабира да ли жели да игру настави у прозорском или режиму целог екрана и обавља конфигурације везане за препреке стрелјане
- Резултати (Рекорди) – мени који садржи листу 10 најбољих играча и њихових резултата поређаних по опадајућем поретку
- Мени паузе – зауставља ток игре (кретање мета), омогућава поновно покретање нивоа, повратак на почетни мени или излазак из игре

Међуниво – мени који се појављује по преласку са једног на виши ниво игре; Садржи опцију за поновно покретање истог у случају да играч није задовољан резултатом. Апликација „Едитор стрелјане“ покреће се из почетног менија и представља посебан додатак игри преко ког играч има могућност да сам осмисли своје стрелјане и тиме направи нове верзије игре.

Интеракција између едитора и корисника обавља се преко низа искачућих прозора у које је потребно унети неопходне параметре попут броја метака на нивоу, броја обртаја приликом ротације и слично, као и следећих менија:

- За одабир нивоа чију стрелјану жели да едитује

- За одабир позадине – играч може да одабере једну од боја из постојеће палете или да учита произвољну слику у JPG, PNG или GIF формату
- За цртање облика (препрека) – облици се одабирају из траке алата која се налази уз доњу ивицу прозора. Могуће је нацртати праву и изломљену линију, многоугао, правоугаоник, елипсу, круг, Безјеову криву или унети текст
- За бојење облика – отвара се кликом миша на дугме за бојење из менија за цртање чиме се у траци алата појављује палета боја
- За анимирање облика(препрека) – приликом селектовања неког од објеката из менија за цртање облика у траци алата се појављује опција за анимирање истог. Овај мени садржи опције за анимирание translације објекта, ротацију за произвољан угао и мењање провидности објекта. Поред тога постоји и опција за приказ направљене анимације
- За додавање и анимирање мета



Слика 3.3 – приказ менија за цртање облика и прозора за одабир конфигурације приликом повратка из менија за додавање мета



## 4. Алати за развој и коришћене библиотеке

За израду дипломског рада коришћен је објектно-оријентисани програмски језик *Java* развијен од стране компаније Sun Microsystems. За развој 2Д компонената игре коришћена је *JavaFX* [4] библиотека која представља скуп пакета за графику, мултимедију и веб [1]. *JavaFX* има подршку за следеће елементе 2Д графике:

- геометријски облици (примитиве, путање, комбиновани облици),
- атрибути стила примитива,
- трансформације,
- анимација,
- интеракција,
- текст,
- ефекти.

Све остале елементе неопходно је самостално имплементирати.

Екстерне библиотеке

- `org.apache.batik.parser`
- `org.apache.batik.util`

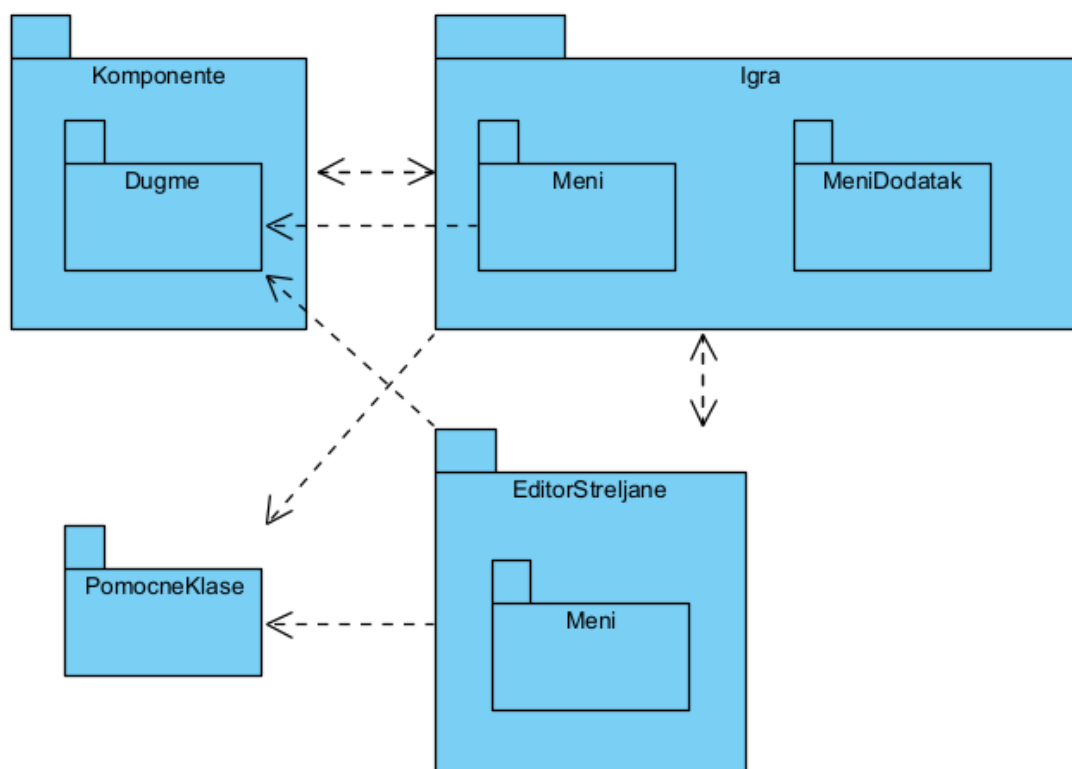
увезене су у пројекат у сврхе парсирања и манипулације са SVG путањама.

За читање, упис и прављење конфигурационих фајлова коришћена је екстерна библиотека `org.json.simple`.

Пројекат је реализован у *IntelliJ IDEA* [5] развојном окружењу.

## 5. Пројекат софтвера

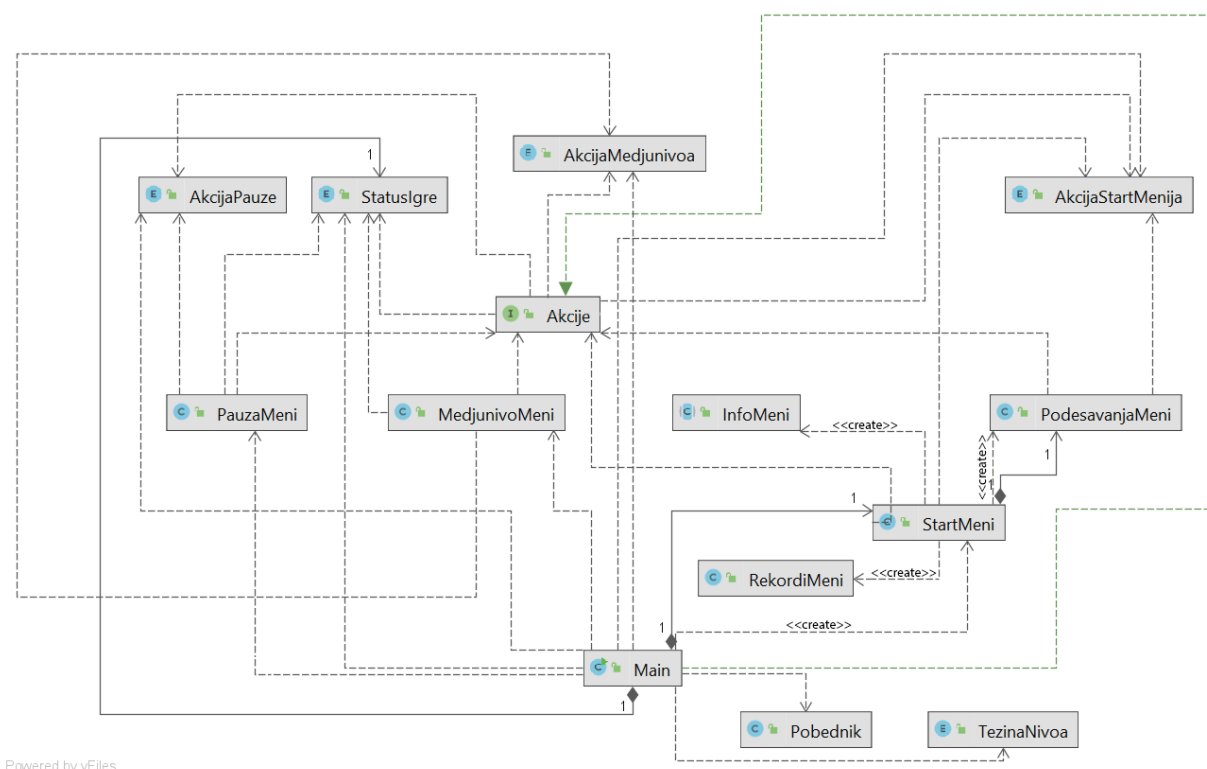
У овом поглављу биће дат преглед архитектуре пројекта и украто бити описани најважнији пакети и класе. Дијаграм свих пакета и начина на који су они повезани дат је на Слика 5.1.



Слика 5.1 – дијаграм пакета игре Мете

Пакет `Igra` садржи главне класе апликације за конфигурацију и контролу тока игре. Он садржи главну класу `Main` која имплементира интерфејс `Akcije` и представља својеврсни сервер игре док потпакет `Meni` садржи меније игре. Комуникација између менија и главне класе обавља се уз помоћ метода овог интерфејса. Игра може бити у једном од следећих стања: `OPCIJE`, `PAUZA`, `IGRA`, `PONOVLJEN_NIVO` или у једном од стања која означавају различите исходе на крају нивоа `USPESNO_UNISTENO`, `POTROSENA_MUNICIJA` или `ISTEKLO_VREME`. У складу са тим у класи `Main` се приликом покретања стварају два бројача која се пале и гасе наизменично и која имају улогу да детектују промене стања игре и да у складу са тим предузму наредне акције. Пакет `MeniDodatak` садржи више типова акција које сваки од менија може да врати као одговор и на основу којих

главна класа `Main` треба да одреагује (Слика 5.2 – Класни дијаграм пакета `Igra`).

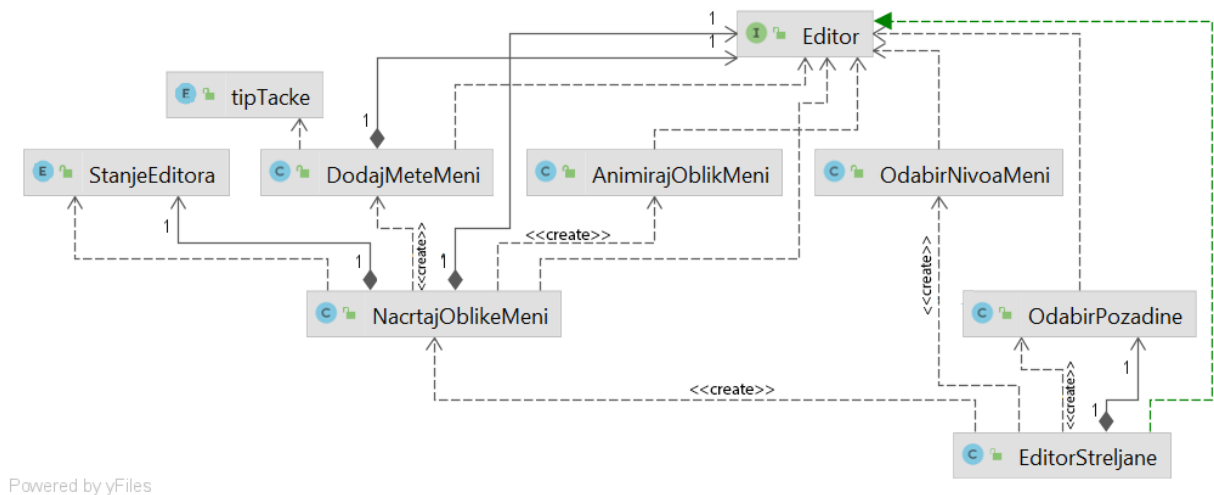


Слика 5.2 – Класни дијаграм пакета *Igra*

Пакет `Komponente` садржи класе стилизованих графичких компоненти посебно дизањираних за игру, попут дугмади која мењају облик, боју и слику и која се налазе у класи `Dugme`. Такође, овај пакет садржи класе `Meta`, `Metak`, `Brojac` који служи за исцртавање целобројних и децималних типова подата на екрану и њихове накнадне манипулације и класу `Stoperica` која уз помоћ тајмера из *JavaFX* пакета рачуна број протеклих секунди.

Пакет EditorStreljane, попут класе Igra, садржи главну класу EditorStreljane која имплементира интерфејс Editor и комуницира са менијима едитора уз помоћ метода поменутог интерфејса. Едитор покреће приказ низа менија почевши од одабира позадине до менија за цртање облика а касније се по потреби из менија за цртање облика могу позивати менији за додавање мета и анимирање облика. Управо због тога, мени за цртање облика може бити у једном од 4 стања, а то су ANIMACIJA, BOJENJE, CRTANJE и PRAVLJENJE METE.

Концептуални дијаграм класа пакета EditorStreljane налази се на Слика 5.3.



Powered by yFiles

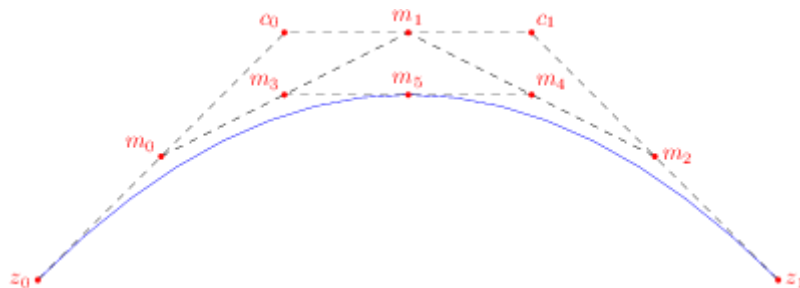
Слика 5.3 – Класни дијаграм пакета *EditorStreljane*

Пакет *PomocneKlase* садржи класе за читање и уписивање у JSON фајл, затим интерфејс *ObradaOblika* који садржи методе за конверзију облика у низ карактера и обрнуто, потом интерфејс *SkaliranjePozornice* и на крају интерфејс *BezjeovaKriva* за израчунавање средишње тачке Безјеове криве.

## 6. Имплементација игре

У овом поглављу биће описан један од детаља имплементације као и техничке карактеристике развијеног софтвера.

С обзиром да *JavaFX* библиотека не садржи функцију за одређивање централне тачке Безјеове криве, а како је за потребе едитора и опције која би омогућила дохватање и померање криве превлачењем њене централне тачке, она била неопходна, функција је имплементирана ручно и то у склопу интерфејса *BezjeovaKriva* која се налази у склопу пакета *PomocneKlase*. На Слика 6.1 – поступак израчунавања централне тачке Безјеове криве је приказан начин на који се централна тачка израчунава а након тога дат је и изглед интерфејса са имплементацијом функције.



Слика 6.1 – поступак израчунавања централне тачке Безјеове криве

У првом пролазу неопходно је одредити средишта дужи  $Z_0C_0$ ,  $C_0C_1$ , и  $C_1Z_1$  где је  $Z_0$  почетна тачка,  $C_0$  и  $C_1$  контролне тачке а  $Z_1$  крајња тачка Безјеове криве. Означимо их са  $M_0$ ,  $M_1$  и  $M_2$ , респективно. Затим је потребно одредити средишта дужи  $M_0M_1$  и  $M_1M_2$  која ћемо означити са  $M_3$  и  $M_4$  и коначно, централну тачку израчунавамо као средиште дужи  $M_3M_4$ .

```
public interface BezjeovaKriva {
    static Point2D findMiddle(CubicCurve curve){
        return helpFUNC(new ArrayList<> (Arrays.asList(
            new Point2D(curve.getStartX(), curve.getStartY()),
            new Point2D(curve.getControlX1(), curve.getControlY1()),
            new Point2D(curve.getControlX2(), curve.getControlY2()),
            new Point2D(curve.getEndX(), curve.getEndY()))));
    };

    public static Point2D helpFUNC(ArrayList<Point2D> list){
        if (list.size()==1) return list.get(0);
        ArrayList<Point2D> list1= new ArrayList<>();
        for (int i=1; i<list.size();i++) list1.add(
            new Point2D((list.get(i).getX()+
                list.get(i-1).getX())/2,
                (list.get(i).getY()+list.get(i-1).getY())/2));
        return helpFUNC(list1);
    }
}
```

У наставку су дате техничке карактеристике апликације.

Метрика	Вредност
Број фајлова	58
Број класа	42
Број метода	232
Број линија кода	5641
Величина ресурса	14.3 MB
Величина IntelliJ пројекта	43.8 MB
Величина извршног(JAR) фајла	14.5 MB

*Табела 1 – Техничке карактеристике*

## 7. Закључак

Мотивација за израду оваквог вида апликације јесте овладавање рада са *javaFX* библиотеком и концептима 2Д рачунарске графике, затим упознавање са проблематиком прављења видео игара и стицање искуства које би помогло при одабиру будуће професије и даљег опредељења. Осмишљене су и реализоване функционалности цртања различитих 2Д облика, додавања трансформација и анимација, конвертовања облика и путања у низове карактера и њихово чување и читање из JSON фајлова. Развијена је игра која буди такмичарски дух и која кориснику нуди могућност импровизације и неограниченог прављења нових верзија игре. Те особине могле би се искористити у сврхе организовања неких мањих надметања, те додељивања награде најбољем играчу.

Игра поседује могућност додавања великог броја облика, међутим могло би се увести комбиновање облика и селекција више од једног облика чиме би се омогућило лакше манипулисање објектима.

Још пар ствари које би се могле додати у будућности и које би унапредиле игру јесу:

- увођење неограниченог броја нивоа са могућностима брисања и едитовања сваког посебно
- увођење више облика за анимирање објеката и мета
- унапређење визуелне атрактивности игре и корисничког интерфејса по узору на различите „*Paint*“ [6] апликације или неке њима сличне са пространим и сликовитим тракама алата
- увођење више тежина нивоа попут супер лаког, лаког, средњег и тешког
- манипулација колекцијама нивоа (чување, отварање, модификација), са сачуваним рекордима.

Ово би такође могле бити ставке неког од будућих радова или имплементације у случају мастер рада.

## 8. Литература

- [1] Тартаља И., „Материјали за предавања из предмета Рачунарска графика“, 2020.
- [2] Дефиниција shooter видео игара, Википедија страница, [https://en.wikipedia.org/wiki/Shooter\\_game](https://en.wikipedia.org/wiki/Shooter_game)
- [3] *Moorhuhn* видео игра, Википедија страница: <https://en.wikipedia.org/wiki/Moorhuhn>
- [4] JavaFX званична страница: Општи преглед, <https://openjfx.io/>
- [5] IntelliJ IDEA развојно окружење, званична веб страница: <https://www.jetbrains.com/idea>
- [6] Paint апликација, званична веб страница: <https://blogs.windows.com/windowsexperience/2018/05/21/windows-10-tip-a-guide-to-the-basic-tools-in-paint-3d/#1b7jOcsJTRXXsH5R.97>