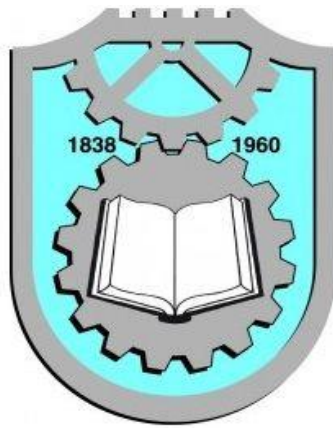


Факултет инжењерских наука Универзитета у Крагујевцу



Рачунарска графика

Тема: Игрица Centipede

Студент:

Јелена Глишић 566/2015

Професор:

др Ненад Филиповић

Крагујевац, 2019.

Садржај

1. Увод.....	3
2. Реализација пројектног задатка	5
3. Литература	20

1. Увод

Циљ овог пројектног задатка је креирање апликације игрице „Centipede“. Centipede је вертикално оријентисана, пуцачка аркадна игра коју је креирао Atari, Inc., у јуну године 1981. Atari је америчка компанија у већинском власништву француске компаније Infogames Entertainment SA, и синоним је за рачунарске, конзолне и аркадне игре и рачунаре (Zemljagljupiranja, 2018). Игру су дизајнирали Dona Bailey и Ed Logg. Logg је био супервизор и радио је дизајн а Bailey је урадила око половине саме апликације. Циљ игре је био да се привуку играчи женског пола јер су дотадашње игрице биле борбене или спортске. Успели су. Игрица је заиста била једна од првих аркадних игара коју су жене радо играле, нека истраживања тврде да је 60% играча било женског пола. Свакако, била је једна од најуспешнијих комерцијалних игра из златног доба видео аркада. Концепт игрице је клониран у многим игрицама: Bug Attack, Arachnoid, Arthropod, Bug Off!, Decipede итд (Spiitzak, 2007).

Играч се бори са стоногама, пауцима, шкорпионима и бувама, завршавајући рунду након елиминисања стоноге која се спушта на простор за игру. Према упутству за игру, играч преузима улогу вртног патуљка, наоружаног чаробним штапићем који мора да брани своју шуму од печурака и од инвазије циновских стонога, паукова, бува и шкорпиона.

Играч је постављен на дну екрана и помера се по доњем делу екрана помоћу миша и лансира пуцње ка сегментираној стоноги која напредује од врха ка дну екрана. Играч може симултано највише пет метака да испали. Сваки уништени сегмент стоноге постаје гљива; погодак једног сегмената стоноге раздваја исту на два дела. Сваки комад затим наставља независно на свом путу кроз екран. Ако је глава сегмент који је погођен, сегмент иза њега постаје следећа глава. Стонога почиње на врху екрана и путује лево или десно. Када додирне гљиву или досегне ивицу екрана, спушта се за један ниво и мења правац. Играч може уништити печурке пуцањем у њих, али тек након испуцаних четири метака. На вишим нивоима, екран може постати све гушћи са гљивама због акција играча због чега се стонога брже спушта.

Када стонога стигне до дна екрана, она се креће лево и десно а могу и њени делови са њом који су постали независне стоноге сада. Она може да се пење али максимално пет редова. Ово ће се настављати све док играч не елиминише и оригиналну стоногу и све главе. Када су сви сегменти стоноге уништени, играч прелази на следећи ниво. Почетна стонога је дуга 10 или 12 сегмената, укључујући и главу; свака следећа стонога је за један сегмент краћа што јој даје већу брзину. Играч такође, сусреће и друга створења поред стоноге. Бове (црвени круг) се повремено појављују и скачу по екрану и једу печурке ако се додирну са њима. Пауци се повремено спушта преко екрана и оставља за собом траг печурки. Шкорпиони се крећу хоризонтално преко екрана, претварајући сваку печурку коју додирну у отровну. Стонога у додиру са отровном печурком ће појурити ка дну екрана и када стигне, вратиће се на нормално понашање. Ова "отрована" стонога може бити веома изазовна да се избегне ако се појављује као више одвојених сегмената.

Играч ће бити уништен када га погоди било који непријатељ. Када играч пређе у нови ниво, боје свега осим играча се мењају и додаје се додатна глава и зато је стонога краћа али и бржа. Поени се додељују за сваку регенерисану гљиву.

2. Реализација пројектног задатка

Пројекат је реализован у развојном окружењу Eclipse IDE. За развој апликација је коришћен програмски језик Java.

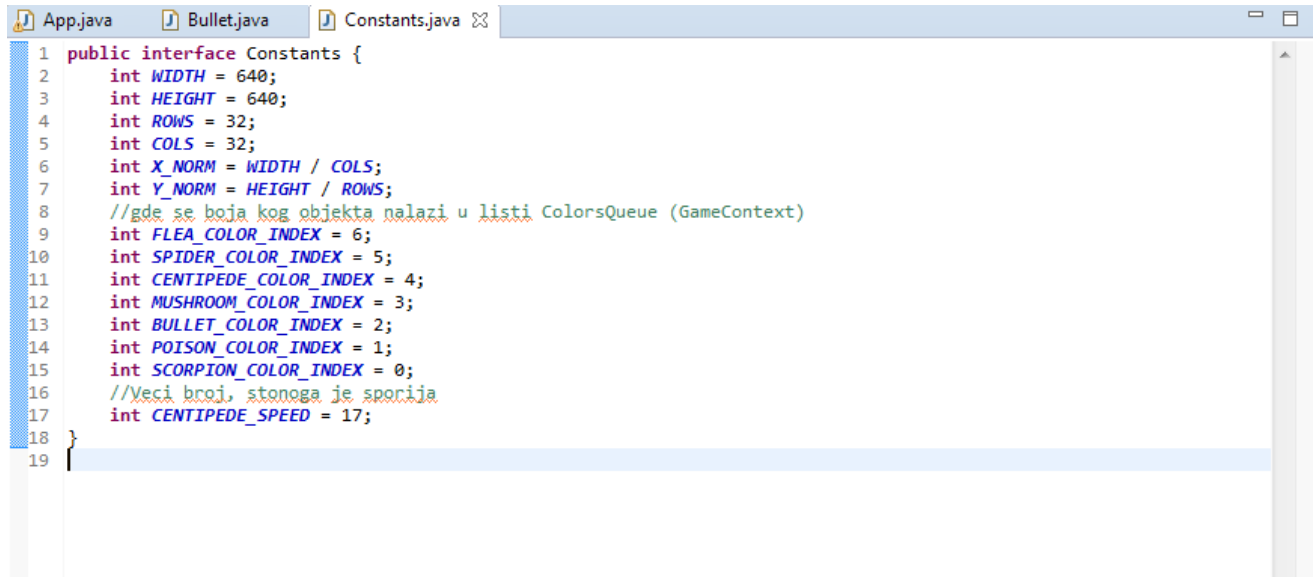
Пројекат садржи два интерфејса и једанаест класа. Интерфејси су:

- Constants.java
- GameObject.java

Класе су:

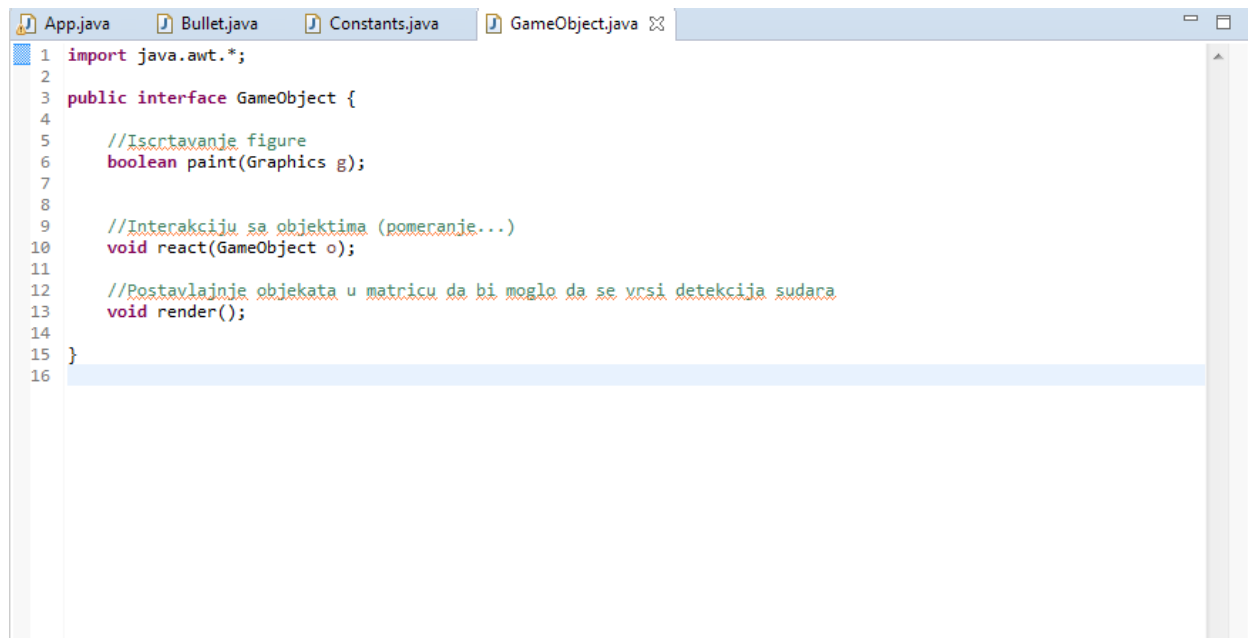
- App.java
- Bullet.java
- Centipede.java
- CentipedeSegment.java
- Flea.java
- GameContext.java
- GameField.java
- Mushroom.java
- Player.java
- Scorpion.java
- Spider.java

Интерфејс Constants.java садржи променљиве које се користе током целе апликације као на пример колоне, редове, боје објеката, ширину и дужину екрана... На слици 1 је дат изглед интерфејса Constants.java.



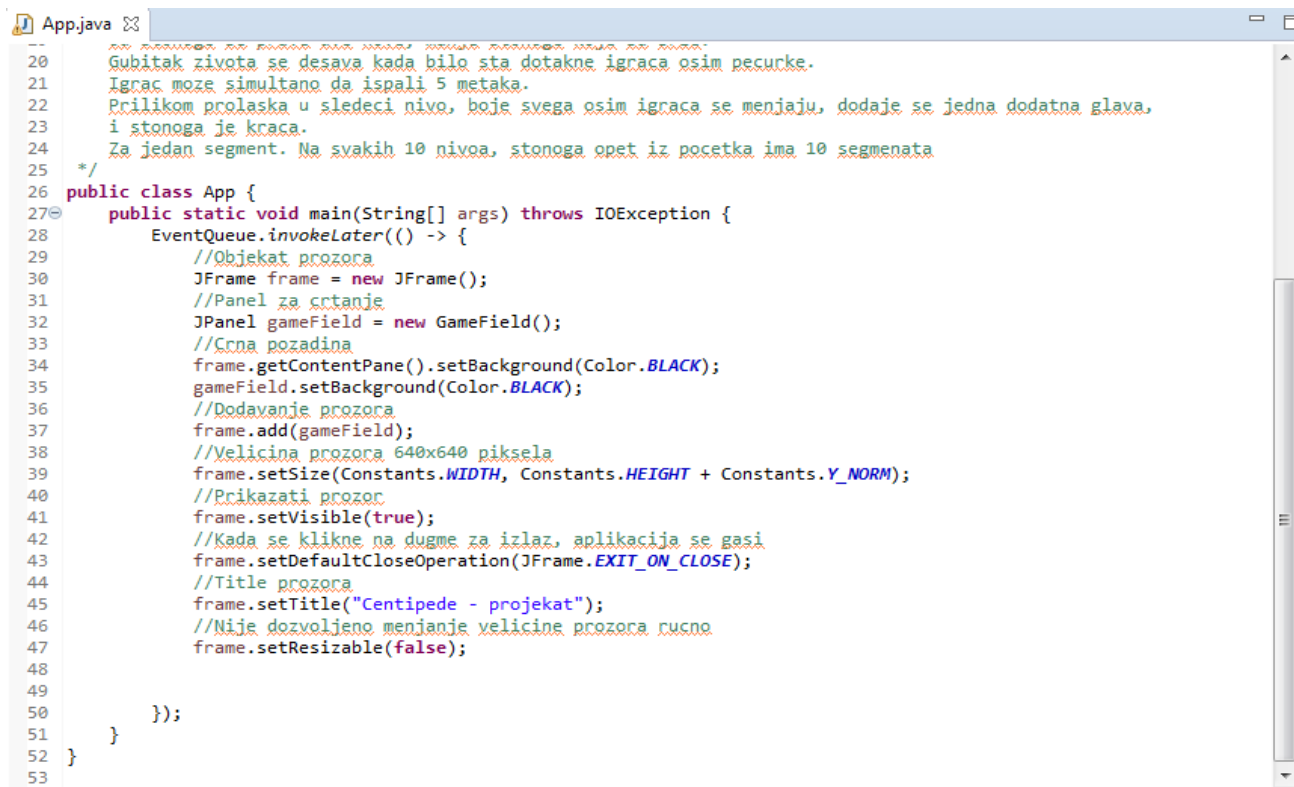
Слика 1. Изглед интерфејса Constants.java

Интерфејс GameObject.java садржи рутину за исцртавање фигура, рутину за интеракцију са објектима и поставља све објекте преко методе render() у матрицу да би могла да се изврши детекција судара. На слици 2 је дат изглед интерфејса GameObject.java.



Слика 2. Изглед интерфејса GameObject.java

Класа App.java служи за постављање прозора, боје позадине, постављање и исцртавање панела али је панел имплементиран у класи GameField.java. На слици 3 је дата класа App.java.



```
20  Gubitak zivota se desava kada bilo sta dotakne igraca osim pecurke.
21  Igrac moze simultano da ispali 5 metaka.
22  Prilikom prolaska u sledeci nivo, boje svega osim igraca se menjaju, dodaje se jedna dodatna glava,
23  i stonoga je kraca.
24  Za jedan segment. Na svakih 10 nivoa, stonoga opet iz pocetka ima 10 segmenata
25  */
26  public class App {
27      public static void main(String[] args) throws IOException {
28         .EventQueue.invokeLater(() -> {
29              //Objekat prozora
30              JFrame frame = new JFrame();
31              //Panel za crtanje
32              JPanel gameField = new GameField();
33              //Crna pozadina
34              frame.getContentPane().setBackground(Color.BLACK);
35              gameField.setBackground(Color.BLACK);
36              //Dodavanje prozora
37              frame.add(gameField);
38              //Velicina prozora 640x640 piksela
39              frame.setSize(Constants.WIDTH, Constants.HEIGHT + Constants.Y_NORM);
40              //Prikazati prozor
41              frame.setVisible(true);
42              //Kada se klikne na dugme za izlaz, aplikacija se gasi
43              frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
44              //Title prozora
45              frame.setTitle("Centipede - projekat");
46              //Nije dozvoljeno menjanje velicine prozora rucno
47              frame.setResizable(false);
48
49          });
50      }
51  }
52
53  }
```

Слика 3. Класа App.java

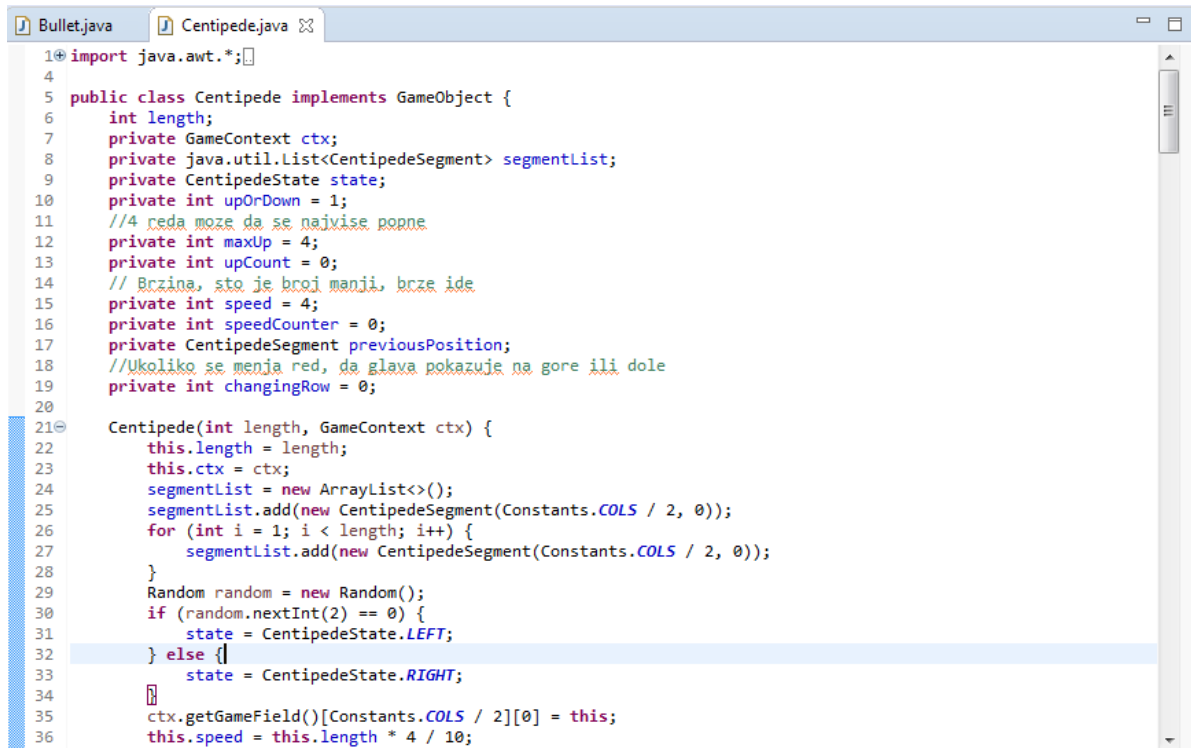
Класа Bullet.java је класа у којој је дефинисан метак, односно зрно које играч испуцава. Играч може највише пет узаступних метака непрекидно да испали. На слици 4 је дата класа Bullet.java.



```
1 import java.awt.*;
2
3 public class Bullet implements GameObject {
4     int x;
5     int y;
6     private GameContext ctx;
7
8     Bullet(int x, int y, GameContext ctx) {
9         this.x = x;
10        this.y = y;
11        this.ctx = ctx;
12    }
13
14    @Override
15    public boolean paint(Graphics g) {
16        g.setColor(ctx.getColorsQueue().get(Constants.BULLET_COLOR_INDEX));
17        g.drawLine((int) ((x + 0.5) * Constants.X_NORM), y * Constants.Y_NORM,
18            (int) ((x + 0.5) * Constants.X_NORM), (y + 1) * Constants.Y_NORM);
19        return true;
20    }
21
22    @Override
23    public void react(GameObject o) {
24        if (y <= 0) {
25            // Ako je van ekrana, samo se unisti
26            ctx.getGameObjectsMap().get(Bullet.class).remove(this);
27            return;
28        } else {
29            // Pomeranje metka na gore
30            y--;
31        }
32        GameObject go = ctx.getGameField()[x][y];
33        // Da moze da prodje kroz igraca
34        if (go != null && !(go instanceof Player)) {
```

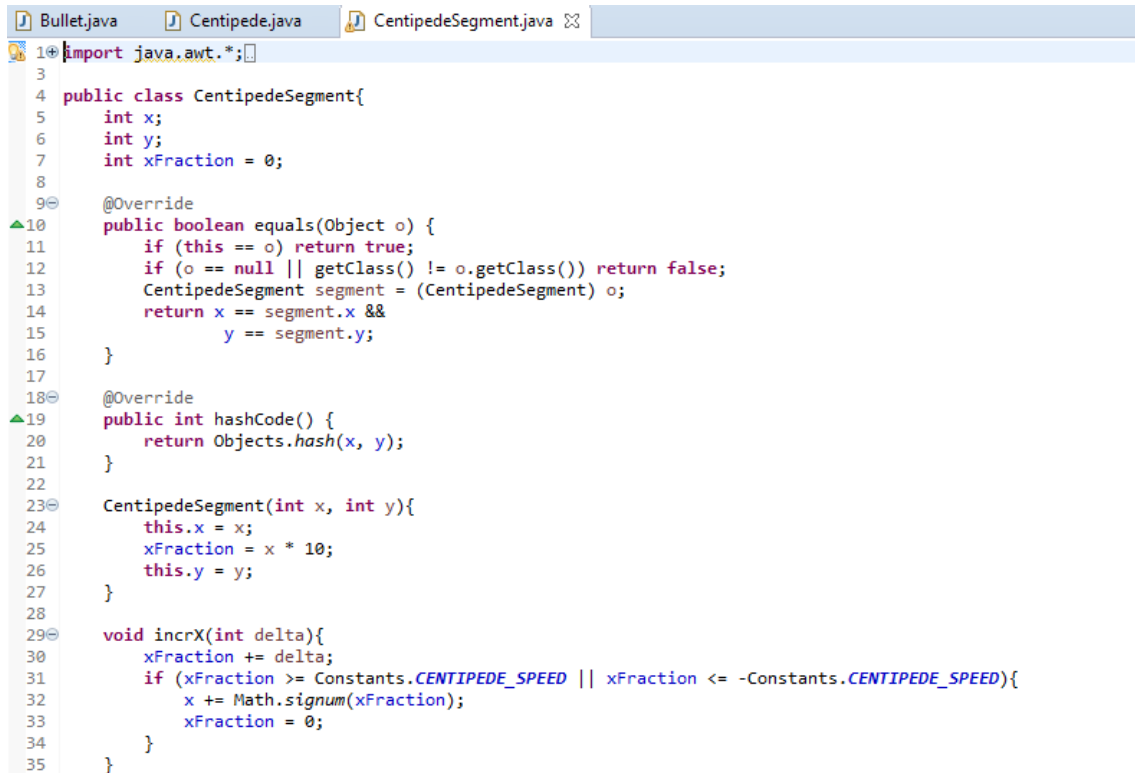
Слика 4. Класа *Bullet.java*

Класа *Centipede.java* имплементира концепт стоноге коју играч треба да погоди и да убије јер је то циљ игрице. Највише четири реда може да се пење стонога навише када се спусти до краја. У овој класи су дефинисана и правила кретања стоноге и провера да ли је дошла до краја екрана, да ли је наишла на отровну печурку. Такође, дефинише се и да уколико стоногу играч погоди у реп (последњи сегмент), само је скраћује, односно стонога се не дели што јесте теже за играча али може бити врло ефикасно за лакше постизање циља. На слици 5 је дата класа *Centipede.java*.



Слика 5. Класа Centipede.java

Класа CentipedeSegment.java говори о сегментима стоноге и даје правила по којима се ови сегменти креирају, увећавају, деле... На слици 6 је дата класа CentipedeSegment.java.



```
1 import java.awt.*;
2
3
4 public class CentipedeSegment{
5     int x;
6     int y;
7     int xFraction = 0;
8
9     @Override
10    public boolean equals(Object o) {
11        if (this == o) return true;
12        if (o == null || getClass() != o.getClass()) return false;
13        CentipedeSegment segment = (CentipedeSegment) o;
14        return x == segment.x &&
15            y == segment.y;
16    }
17
18    @Override
19    public int hashCode() {
20        return Objects.hash(x, y);
21    }
22
23    CentipedeSegment(int x, int y){
24        this.x = x;
25        xFraction = x * 10;
26        this.y = y;
27    }
28
29    void incrX(int delta){
30        xFraction += delta;
31        if (xFraction >= Constants.CENTPEDE_SPEED || xFraction <= -Constants.CENTPEDE_SPEED){
32            x += Math.signum(xFraction);
33            xFraction = 0;
34        }
35    }
```

Слика 6. Класа *CentipedeSegment.java*

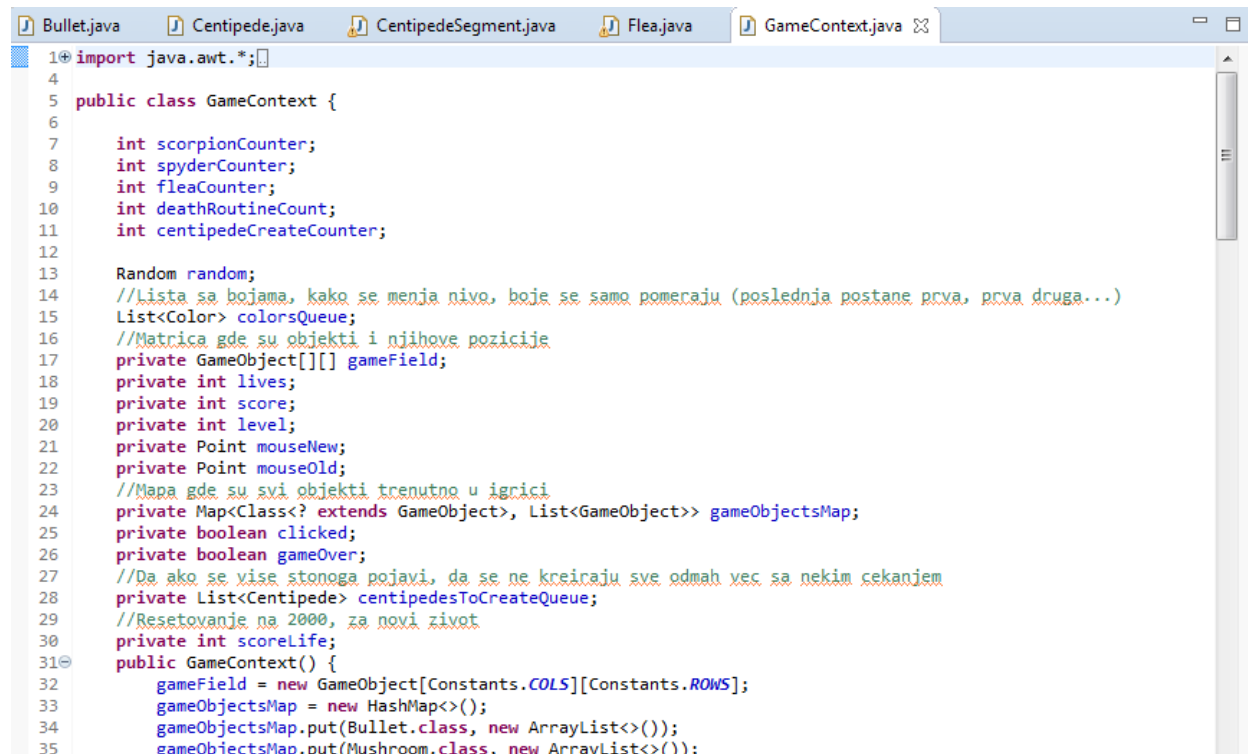
Класа *Flea.java* дефинише буву коју играч треба да заобиђе јер у супротном, може изгубити живот. Бува се може јавити у последњих тринаест редова, минимално од трећег. Њено кретање је кретање цик-цак и ограничено по у-оси. На слици 7 је дата класа *Flea.java*.



```
1 import java.awt.*;
2
3
4 public class Flea implements GameObject {
5     private GameContext ctx;
6     private int x;
7     private int y1;
8     private int y;
9     private int counter;
10    private int incr;
11    Flea(GameContext ctx) {
12        counter = 0;
13        this.ctx = ctx;
14        this.x = Constants.COLS - 1;
15        //Moze da se pojavi u donjih 13 redova, minimalno od treceg
16        this.y1 = Constants.ROWS - 3 - ctx.random.nextInt(10);
17        this.y = y1;
18    }
19
20    @Override
21    public boolean paint(Graphics g) {
22        g.setColor(ctx.getColorsQueue().get(Constants.FLEA_COLOR_INDEX));
23        g.fillOval(x * Constants.X_NORM, y * Constants.Y_NORM + 10, Constants.X_NORM, Constants.Y_NORM - 10);
24        return true;
25    }
26
27    @Override
28    public void react(GameObject o) {
29        if (o == this) {
30            x--;
31            if (x < 0) {
32                ctx.getGameObjectsMap().get(Flea.class).remove(this);
33                return;
34            }
35        }
36    }
37}
```

Слика 7. Класа Flea.java

Класа GameContext.java је класа која садржи концепт игрице. Овде је дефинисана листа са бојама, како се мења ниво, боје се само померају, односно шифтују (последња постаје прва, прва друга, итд...). У њој је дефинисана и матрица где су садржани објекти и њихове позиције која служи за детекцију колизије (ниво, животи, поени). Постоји и мапа која садржи све објекте који су тренутно у игрици, такође, правила да се не креира више од једне стоноге, односно да се не креирају у истом треутку, већ са неком малом паузом. На слици 8 је дата класа GameContext.java.



Слика 8. Класа GameContext.java

Класа GameField.java је класа која даје изглед екрана. У њој су дефинисани догађаји за тастере са тастатуре односно ако се притисне SPACE то је сигнал да игрица може да се покрене, уколико се притисне P поставља се пауза. Дефинисан је и рад миша, односно пуцњи. На сваких две хиљаде прикупљених поена, играч добија нови живот. У овој класи је урађен и пребацај у нови ниво уколико је стонога уништена. За сваки погодак печурке, играч добија десет поена, четири погодака су му потребна да је уништи скроз. Ова класа провера да ли играча није додирнуло ништа друго сем печурке јер само њоме не губи живот. Шкорпија и бува се могу појавити са 50% вероватноће а паук са 75%. На слици 9 је дата класа GameField.java.

```
1 import javax.swing.*;
12
13 public class GameField extends JPanel {
14
15     private GameContext ctx;
16     private Timer timer;
17
18     GameField() {
19         timer = new Timer(50, listener -> mainLoop());
20         //Panel da bude u fokusu
21         setFocusable(true);
22         requestFocus();
23         //event handleri za mis i tastaturu
24         addMouseListener();
25         addKeyListeners();
26     }
27
28     private void addKeyListeners() {
29         addKeyListener(new KeyAdapter() {
30             @Override
31             public void keyPressed(KeyEvent e) {
32                 super.keyPressed(e);
33                 switch (e.getKeyCode()) {
34                     //Pocetak igre - SPACE
35                     case KeyEvent.VK_SPACE:
36                         if (ctx == null || ctx.isGameOver()) {
37                             initializeGame();
38                         }
39                         break;
40                     //Pauza - P
41                     case KeyEvent.VK_P:
42                         if (timer.isRunning()) {
43                             timer.stop();
44                         } else {
```

Слика 9. Класа GameField.java

Класа Mushroom.java дефинише изглед печурке, њено тело и глава. Погодак сваког дела тела доноси играчу по десет поена а погодак главе јер је то њено уништење двадесет поена. На слици 10 је дата класа Mushroom.java.

```
1 import javax.swing.*;
12
13 public class GameField extends JPanel {
14
15     private GameContext ctx;
16     private Timer timer;
17
18     GameField() {
19         timer = new Timer(50, listener -> mainLoop());
20         //Panel da bude u fokusu
21         setFocusable(true);
22         requestFocus();
23         //event handleri za mis i tastaturu
24         addMouseListener();
25         addKeyListeners();
26     }
27
28     private void addKeyListeners() {
29         addKeyListener(new KeyAdapter() {
30             @Override
31             public void keyPressed(KeyEvent e) {
32                 super.keyPressed(e);
33                 switch (e.getKeyCode()) {
34                     //Pocetak igre - SPACE
35                     case KeyEvent.VK_SPACE:
36                         if (ctx == null || ctx.isGameOver()) {
37                             initializeGame();
38                         }
39                         break;
40                     //Pauza - P
41                     case KeyEvent.VK_P:
42                         if (timer.isRunning()) {
43                             timer.stop();
44                         } else {
```

Слика 10. Класа *Mushroom.java*

Класа *Player.java* дефинише белог играча на дну екрана који може максимално пет редова навише да се креће, односно по у-оси је ограничен. Ту су и методе за проверу да ли је још у игри, тј. да ли је жив и метода да може максимално пет меткова да испали одједном. Играч не може кроз печурку да пролази. На слици 11 је дата класа *Player.java*.

```
1 import java.awt.*;
2
3 public class Player implements GameObject {
4     GameContext ctx;
5     int x;
6     int y;
7
8     Player(GameContext ctx) {
9         this.ctx = ctx;
10        //Na pola ekrana, na donjoj ivici
11        y = Constants.ROWS - 1;
12        x = Constants.COLS / 2;
13    }
14
15    @Override
16    public boolean paint(Graphics g) {
17
18        g.setColor(Color.WHITE);
19        Polygon p = new Polygon();
20        p.addPoint((int) ((x + 1 / 2.) * Constants.X_NORM), (int) ((y + 2 / 3.) * Constants.Y_NORM));
21        p.addPoint((int) ((x + 1 / 2.) * Constants.X_NORM), y * Constants.Y_NORM);
22        p.addPoint(((x + 1) * Constants.X_NORM), (int) ((y + 2 / 3.) * Constants.Y_NORM));
23        p.addPoint((int) ((x + 2 / 3.) * Constants.X_NORM), (y + 1) * Constants.Y_NORM);
24        p.addPoint((int) ((x + 1 / 3.) * Constants.X_NORM), (y + 1) * Constants.Y_NORM);
25        g.fillPolygon(p);
26
27        return false;
28    }
29
30
31    @Override
32    public void react(GameObject o) {
33        if (o == this) {
34            checkIsDead();
35        }
36    }
37}
```

Слика 11. Класа Player.java

Класа Scorpion.java дефинише шкорпију која се повремено јавља и пада хоризонтално. При судару са печурком, прави је отровном. На слици 12 је дата класа Scorpion.java.

```
import java.awt.*;

public class Scorpion implements GameObject {
    private GameContext ctx;
    private int x;
    private int y;

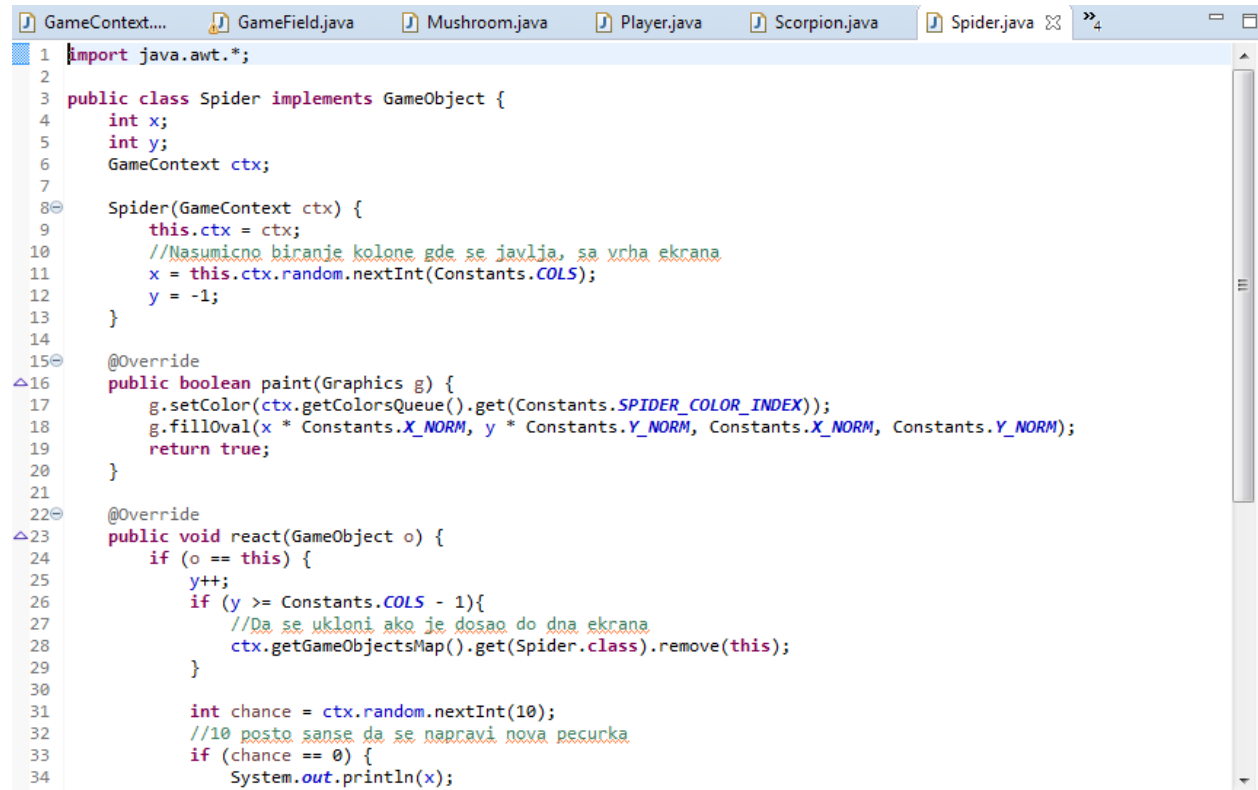
    Scorpion(GameContext ctx) {
        this.ctx = ctx;
        this.x = 0;
        this.y = 5 + ctx.random.nextInt(Constants.ROWS - 10);
    }

    @Override
    public boolean paint(Graphics g) {
        g.setColor(ctx.getColorsQueue().get(Constants.SCORPION_COLOR_INDEX));
        g.fillRect(x * Constants.X_NORM, y * Constants.Y_NORM, Constants.X_NORM, Constants.Y_NORM);
        return true;
    }

    @Override
    public void react(GameObject o) {
        if (o == this) {
            if (++x > Constants.COLS - 1) {
                ctx.getGameObjectsMap().get(Scorpion.class).remove(this);
                return;
            }
            GameObject go = ctx.getGameField()[x][y];
            if (go != null) {
                go.react(this);
            }
        } else if (o instanceof Bullet) {
            ctx.setScore(ctx.getScore() + 20);
            ctx.setScoreLife(ctx.getScoreLife() + 20);
        }
    }
}
```

Слика 12. Класа Scorpion.java

Класа Spider.java креира паука који наилази насумићно хоризонтално, са врха екрана и иза себе оставља печурку. Када стигне до дна, он нестаје. На слици 13 је дата класа Spider.java.



```
1 import java.awt.*;
2
3 public class Spider implements GameObject {
4     int x;
5     int y;
6     GameContext ctx;
7
8     Spider(GameContext ctx) {
9         this.ctx = ctx;
10        //Nasumično biranje kolone gde se javlja, sa vrha ekrana
11        x = this.ctx.random.nextInt(Constants.COLS);
12        y = -1;
13    }
14
15    @Override
16    public boolean paint(Graphics g) {
17        g.setColor(ctx.getColorsQueue().get(Constants.SPIDER_COLOR_INDEX));
18        g.fillOval(x * Constants.X_NORM, y * Constants.Y_NORM, Constants.X_NORM, Constants.Y_NORM);
19        return true;
20    }
21
22    @Override
23    public void react(GameObject o) {
24        if (o == this) {
25            y++;
26            if (y >= Constants.COLS - 1){
27                //Da se ukloni ako je dosao do dna ekrana
28                ctx.getGameObjectsMap().get(Spider.class).remove(this);
29            }
30
31            int chance = ctx.random.nextInt(10);
32            //10 posto sanse da se napravi nova pecurka
33            if (chance == 0) {
34                System.out.println(x);
```

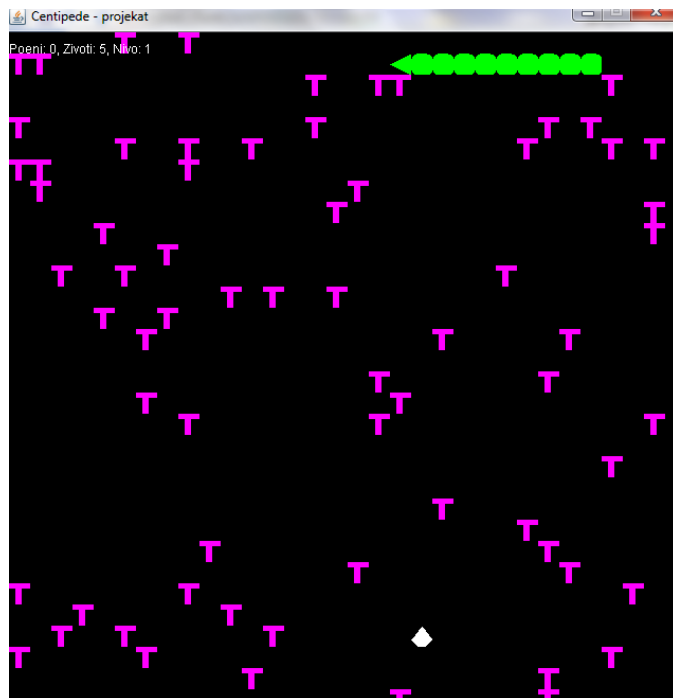
Слика 13. Класа Spider.java

Када се апликација покрене, почетни прозор изгледа као на слици 14.



Слика 14. Изглед почетног прозора апликације

Корисник треба да притисне SPACE и игра може да почне. Увек када пожели, кликом на Р на тастатури, може паузирати игрицу. На слици 15 је дат сам почетак игрице. Када корисник погоди стоногу, она се дели и сада постоје две, брже које су дате на слици 16.



Слика 15. Почетак апликације



Слика 16. Стонога погођена и подељена на две исте

3. Литература

- Zemljaglupiranja. (2018, Maj 17). *Atari*. Преузето Jul 1, 2019, са Wikipedija:
<https://sr.wikipedia.org/wiki/%D0%90%D1%82%D0%B0%D1%80%D0%B8>
- Spiitzak. (2007, Septembar 2). *Centipede*. Преузето Jul 1, 2019, са Wikipedia:
[https://en.wikipedia.org/wiki/Centipede_\(video_game\)](https://en.wikipedia.org/wiki/Centipede_(video_game))