Description: PMF Cirilica.wmf

**Институт за математику и информатику**

**Природно-математички факултет**

**Универзитет у Крагујевцу**

Семинарски рад

**Lavirint**

Ментор: Студент:

Ана Капларевић-Малишић Алекса Савић 102/2016

Јануар 2020.

**Садржај**

[1. Увод 3](#_Toc493024191)

[1.1 О игри 3](#_Toc493024192)

[1.2 Спецификације захтева 3](#_Toc493024193)

[2. Софтверска реализација 4](#_Toc493024194)

[2.1 UML дијаграм класа 4](#_Toc493024195)

[2.2 Графички кориснички интерфејс 5](#_Toc493024196)

[2.2.1 Класа GUI 5](#_Toc493024197)

[2.2.2 Ток игре 7](#_Toc493024198)

[2.2.3 Начин заустављања игре 8](#_Toc493024200)

[2.3 Логика игре 8](#_Toc493024201)

[2.3.1 Иницијализација 9](#_Toc493024202)

[2.3.2 Померање играча](#_Toc493024203) 9

[3. Финалне напомене 1](#_Toc493024205)1

[Литература 1](#_Toc493024206)1

# **1. Увод**

Овај семинарски рад представља пратећи документ уз апликацију израђену као део испитних обавеза на предмету Објектно-оријентисано програмирање. Софтверско решење је реализовано употребом Јава програмског језика и стандардних Јавиних библиотека. Резултат развоја је игрица **Lavirint**.

* 1. **О игри**

Игра је сачињена од 4 нивоа. Сваки ниво је представља посебан лавиринт тј. посебну годину на факултету. Циљ игре је помоћи Алекси да прође кроз све лавиринте у што мање корака, и да при томе што мање удара у ивице лавиринта, јер то доноси негативне поене (5 додатних корака за сваки ударац о ивицу). Уколико се играч изгуби, може започети нову игру кликом на дугме “Zapocni igru ponovo”, или нови ниво, кликом на дугме “Zapocni nivo ponovo”. Игра се завршава на крају четвртог нивоа, Алексиним полагањем завршног испита.

* 1. **Спецификација захтева**

Софтверско решење је реализовано тако да испуњава следеће захтеве:

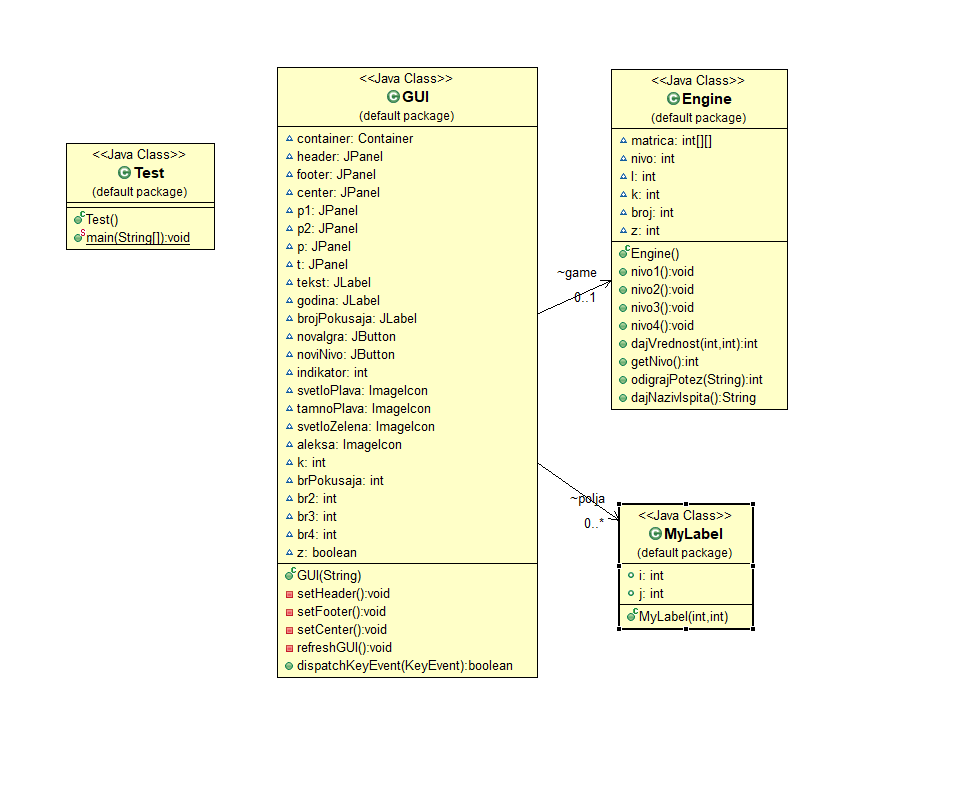
* Реализовати игру **Lavirint** за једног играча.
* Омогућити контролисање играча помоћу тастатуре.
* Имплементирати систем за бројање пређених корака.
* Мењати боју header-a I footer-a у зависности од поља на које је играч стао.

# **2. Софтверска реализација**

Решење је реализовано коришћењем класа **Engine, GUI, MyLabel** и **Test**.

## 

## **2.1 UML дијаграм класа**



## 

## **2.2 Графички кориснички интерфејс**

Графички кориснички интерфејс је реализован помоћу класе **GUI**. Ова класа је пројектована тако да омогући:

* Приказ корисничког интерфејса.
* Интеракцију са корисником.

У класи се налазе потребне методе за покретање играча.

### **2.2.1 Класа GUI**

Kласа *GUI* садржи дефиниције главног прозора*.* Она наслеђује класу *JFrame* и имплементира *KeyEventDispatcher* али садржи и инстанцу класе *Engine*. Класа *GUI* садржи методу за постављање почетног изгледа централног панела – *setCenter* која поставља изглед централног дела прозора тако да је он подељен на 625 делова у облику матрице 25х25. Такође, класа *GUI* садржи и методу за постављање почетне вредности северног панела, који се састоји од информације о томе на којој је години тј. на ком се нивоу на игрици налази играч и информације колико је корака употребио играч да би стигао до позиције на којој се тренутно налази. Постоји и метода за постављање вредности јужног панела који се састоји од два дугмета, прво дугме јесте дугме за повратак на почетак игре, а друго је за повратак на почетак нивоа на коме се играч налази. Ту су и метода *refreshGUI* која на основу вредности матрице у логичком делу игрице (Еngine) поставља позадину централног дела игре. Преко методе*DispatchKeyEvent*се врши контролисање играча као и замена боја и текста северног панела и јужног панела на основу вредности које се враћају из логичког дела игрице (Engine) – уколико играч удари у тамно плави квадратић северни и јужни панел се боје у црвену боју. Такође, боја ова два панела је иста уколико играч покуша да се врати назад, кроз поље на коме се налази испит који је већ положио. Боја северног и јужног панела је зелена уколико играч стане први пут на поље на коме се налази неки испит .

|  |
| --- |
| Изворни код – GUI.java – метода **DispatchKeyEvent,** део са бојењем header-a и footer-a |
|  |
|  |
|  |

### 

### **2.2.2 Ток игре**

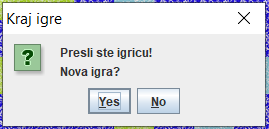
# На почетку игре играч се налази на левој страни. Играч се креће само по светло плавим пољима. Потребно је да играч прође кроз лавиринт и стигне до крајњег десног поља, које означава прелазак на нови ниво. Играч у току проласка кроз лавиринт не сме да удара у тамно плава поља јер она доносе додатне кораке. Играч је на правом путу ако с времена на време нагази поље на коме се налази неки испит, у супротном играч није на правом путу.

# 

# Циљ игре је да играч у што мање корака прође кроз четири нивоа – четири, од којих сваки засебно представља годину на факултету. То неће бити једноставно, а ту је и проблем ударања тамно плавих поља која доносе 5 додатних корака.

### **2.2.3 Начин заустављања игре**

# Начини за заустављање игре је дат прозором након успешног завршетка игре, где је постоји и могућност поновног покретања игре, приказано на *Слика1*. Такође игра се може прекинути и кликом на Х, који се налази у горњем десном углу прозора.



*Слика1*

## **2.3 Логика игре**

Логика игре смештена је у класи **Engine**. Док се приказ и само покретање играча одвија кроз класу **GUI**.

### **2.3.1 Иницијализација**

На почетку игре се врши почетно иницијализовање вредности за први ниво игре (иницијализовање вредности матрице на основу које се исцртава централни панел), након чега се позивањем одговарајућих метода те вредности мењају:

* У потпуности, ако се прелази на следећи ниво игре .
* Појединачно, ако се играч помера само за једно поље у односу на претходно место на ком се налазио.

У овој класи, **Engine** постоје и методи за враћање вредности одређеног поља матрице, нивоа на ком се налази играч и назива испита.

### **2.3.2 Померање играча**

Померање играча се одвија притиском на стрелицу на тастатури. Играч ће се померити само ако му је то дозвољено (креће се по светло плавим пољима). Ако му је дозвољено, он се помера тако што се вредност играча обрише са поља на ком је био и упише се на следеће поље, након чега се поново исцртавају лабеле централног панела.

У наредном блоку дат је део кода који поставља вредности матрице на основу које се исцртава централни панел, када се притисне десна стрелица.

**public** **int** odigrajPotez(String s)

{

**int** ind=0;

**boolean** n = **false**;

/\*uzima se trenutna pozicija igraca\*/

**for**(**int** i=0;i<25;i++)

**for**(**int** j=0;j<25;j++)

**if**(matrica[i][j]==2)

{

l=i;

k=j;

}

/\*ako je pritisnuta desna strelica \*/

**if**(s=="R")

{

/\*ako je trenutna ispit a desno je ispit ili obicna \*/

**if**(matrica[l][k]==2 && z==5 && (matrica[l][k+1]==0 || matrica[l][k+1]==5))

{

matrica[l][k]=broj;

broj++;

n=**true**;

}

/\*ako je desno od igraca kraj\*/

**if**(matrica[l][k+1]==4)

{

/\*ako prosla nije ispit, ako je ispit vec je postavljena u prvom if-u\*/

**if**(n==**false**)

matrica[l][k]=3;/\*trenutna se boji u zelenu\*/

matrica[l][k+1]=2;/\*desno od igraca se boji u sliku igraca\*/

ind=4; /\* da javi GUI-ju da je kraj\*/

}

/\*ako je desno od igraca ispit ili obicno polje\*/

**else** **if**(matrica[l][k+1]==0 || matrica[l][k+1]==5)

{

z=0;

/\*ako je ispit\*/

**if**(matrica[l][k+1]==5)

{

ind=5; /\*da javi GUI-ju da je ispit\*/

z=5; /\*ako je ispit z se postavlja na 5 kao indikator da bi se u sledecem koraku znalo \*/

}

/\*trenutna se boji u svetlo zelenu ako nije ispit\*/

**if**(n==**false**)

matrica[l][k]=3;

/\*sledeca se boji u sliku igraca\*/

matrica[l][k+1]=2;

}

/\*ako je desno vec zelena\*/

**else** **if**(matrica[l][k+1]==3)

{

/\*trenutna se boji u svetlo plavu ako nije ispit\*/

**if**(n==**false**)

matrica[l][k]=0;

/\*sledeca se boji u igraca\*/

matrica[l][k+1]=2;

}

/\*ako je desno ispit na kom je vec bio\*/

**else** **if**(matrica[l][k+1]>5)

{

ind=6;

}

**else**

{

ind=1;

}

}

}

Метода враћа индикатор који у класи ***GUI*** служи за одређивање садржаја и боје header-а и footer-а.

## **Финалне напомене**

* Игра просто захтева известан програмерски рад на **GIU**-у**.**
* За израду ове игре, коришћене су стандардне Јавине библиотеке.
* Да би се игра покренула, потребно је да на рачунару постоји инсталирана Јава Виртуелна Машина (**Java Runtime Environment**).[[1]](#footnote-1)
* Планирана унапређења игре:
  + Почетни избор играча на почетку игре.
  + Мерење времена од почетка до краја игре.
  + Приказивање ранг листе најбољих играча на крају игре.

# 

# **Литература**

[1] D. Poo, D. King, S. Ashok, *Object-oriented programming in Java*, Springer-verlag, 2008.

[2] <https://imi.pmf.kg.ac.rs/moodle/course/view.php?id=33>, 15.1.2020.

[3] <https://www.javatpoint.com/java-swing>, 15.1.2020.

[4] <https://docs.oracle.com/javase/tutorial/uiswing/components/dialog.html>, 15.1.2020

1. Оперативни систем се подразумева [↑](#footnote-ref-1)