

**МАТЕМАТИЧКА ГИМНАЗИЈА**

# **МАТУРСКИ РАД**

**из предмета Рачунарство и информатика**

**Андроид игрица и Lego Mindstorms**

**Ученик**

**Давид Виденовић 4б**

**Ментор**

**Јелена Хаџи-Пурић**

**Београд, јун 2014.**

## САДРЖАЈ

Садржај .....	2
Увод .....	3
Вештачка интелигенција .....	3
Роботи и роботика .....	4
Мобилни телефони .....	4
Компјутерске игре.....	5
Андроид игре као подврста компјутерских игара .....	5
Теорија игара .....	6
Дефиниције.....	6
АЛАТКЕ КОРИШЋЕНЕ У ОВОМ РАДУ .....	7
ТЕМА ОВОГ РАДА .....	7
АНДРОИД ПРОГРАМ - КОМПЈУТЕРСКА ИГРИЦА .....	8
Опис игре.....	8
Екрани игре .....	8
Графика .....	8
Gamerplay (на телефону) .....	9
Мод за једног играча .....	10
Мод за два играча.....	10
Профили и чување игре .....	10
Интернет део апликације .....	11
Структура интернет дела и објашњење .....	12
Механика игре (на телефону) .....	13
Катапулти.....	13
Кугле и бонуси .....	15
Мапе.....	15
Колизација .....	16
АИ противници - пример вештачке интелигенције .....	17
Остале одлике игре .....	18
Структура телефонског дела апликације .....	19
Robotic Play мод - примена робота.....	21
Лего робот.....	21
Лего NXT Сет.....	21
Дизајн Лего робота .....	21
Програмирање Robotic Play мода .....	22
Део апликације на роботу .....	22
Део апликације на телефону.....	22
Структура роботског дела апликације.....	23
На телефону .....	23
На роботу.....	23
Закључак .....	24
Литература.....	25

## УВОД

Вештачка интелигенција, роботика, мобилни телефони. У развој ове три области се последњих година улаже много, и сходно томе, ове области су доста напредовале и очекује се да ће и још више напредовати. Узмимо, на пример, развој мобилних телефона. Пре тек десет година, екрани осетљиви на додир су нам се чинили недостижним, а сада не можемо да замислимо телефон без њих. Такође, по неким прорачунима, до 2025-те године, роботи и вештачка интелигенција ће заменити људе на 50% послова у услужном сектору. Ове три области су несумњиво важне за будућност човечанства, и овај рад их повезује преко компјутерске игре.

### Вештачка интелигенција

Вештачка интелигенција (Скраћено АИ - Artificial Inteligence) представља испољавање интелигенције налик људској од стране машина или софтвера. У информатици, поље вештачке интелигенције проучава начине прављења интелигентних машина и интелигентних компјутерских програма. Овај термин је 1955. године саставио и дефинисао Американац John McCarthy.

Појам АИ нас углавном асоцира на роботе из дела научне фантастике који су у стању да у потпуности обављају све когнитивне функције које и човек може да обавља. Међутим, АИ је знатно шири појам. У АИ спадају роботи-телефонски оператери, компјутерски-навођени противници у играма, софтвери за банкомате... Све ово наведено научници називају "Слабом АИ". Уопштено, у слабу АИ спадају сви компјутерски програми који немају сопствени разум, и углавном су фокусирани на уску област деловања. Насупрот слабој АИ, постоји, бар теоретски, и јака АИ, која је разумна и обавља различите функције. Научници нису најсложнији око тога да ли је могуће направити јаку АИ.

Да би нека вештачка интелигенција била сматрана јаком, мора проћи Тјурингов тест. Овај тест, осмишљен од стране научника Алана Тјуринга који се због својих достигнућа у овој области често назива "оцем вештачке интелигенције", се састоји у томе да судија - човек - комуницира путем природног језика са још једним човеком и тестираном АИ машином. Учесници у тесту су међусобно удаљени, и судија на почетку не зна ко је човек а ко машина. Ако на крају разговора судија препозна машину, онда тест није успешан. Ако је не препозна, машина је прошла тест и може се сматрати јаком АИ.

У овом раду, нажалост, у могућности сам да се бавим искључиво слабом АИ, и то у виду компјутерски генерисаног противника у игри.

## Роботи и роботика

РОБОТ је машина која је способна да аутоматски изврши низ комплексних акција. Сама реч "РОБОТ" потиче од црквенословенског "Работа", односно служити. Назив је смислио чешки писац Карел Чапек 1921. године за потребе његовог дела "Р.У.Р. (Росумови Универзални Роботи)". Управо како им и име каже, роботи су направљени да би људима окашали разне послове, и користе се за ефикаснију израду задатака који су људима напорни или опасни. Данас су роботи углавном вођени компјутерским програмима, те је способност прављења софтвера којима се управљају роботи изузетно важна, а пошто ће број робота у будућности расти, та способност ће бити све важнија. Роботика је део информатике и инжињерства који се бави дизајнирањем, конструкцијом, функционисањем и применом робота и компјутерским програмима који њима управљају.

Први робот који се примењује у индустрији био је "Unimate", направљен 1961. године. Међутим, аутомати и машине са сличним концептима као данашњи роботи су се правиле још од античког доба. У трећем веку п.н.е. у Кини је направљена механичка фигура величине човека која је могла да помера руке и ноге. У античкој Грчкој су постојали описи за преко 100 машина и аутомата. Леноардо да Вини је чувен по својим визионарским нацртима машина. Никола Тесла је 1898. направио први даљински контролисан објекат... Од наноробота до АСИМА, данас постоји много различитих робота, па је и роботика веома широка област.

У овом раду биће демонстрирано коришћење и програмирање робота, и његово управљање путем Bluetooth везе и Андроид телефона.

## Мобилни телефони

Иако су мобилни телефони релативно нови изуми (први комерцијални телефон, Моторола ДупаТАС 8000х, у продају је пуштен 1983. године), данас је живот без њих незамислив. Од првог телефона па до данас, њихов изглед и карактеристике су много напредовале. Први телефон је био тежак 790г, без антене је био дугачак 25цм и није имао екран. Савремени телефони, нпр. Samsung Galaxy С4, има тезину 130г, дужину 13.6 цм и екран осетљив на додир.

Што се софтвера тиче, данас је свакако једна од најпопуларнијих платформа за мобилне телефоне Андроид. Први телефон са андроид платформом је HTC Dream, који је изашао 22. октобра 2008. године. Андроид платформа је веома брзо освојила тржиште захваљујући својој једноставности и отвореном стандарду. Ова платформа, поред корисника, привукла је и програмере. Управо због те отворености и једноставности коју нуди, Андроид је доживео невероватну популарност.

Главни део овог рада сачињаваће управо апликација прављена за Андроид уређаје.

## Компјутерске игре

Игре су саставни део живота. Свако од нас је од најраније младости био укључен у игре. У детињству, имитирањем одраслих (краљева, ратника...) у играма смо се учили разним ситуацијама у одраслом животу. У зрелијем добу, игре нам служе као одмор од посла, и пружају нам забаву и опуштање.

Несумњиво, како је човечанство напредовало кроз историју, тако су и игре напредовале. Са појавом компјутера, дошла је и потреба да се ова нова технологија искористи и за играчке сврхе. Свакако, компјутери су омогућили потпуно нов начин играња. Године 1958. нуклеарни физичар Вилијам Хигинботам, који је познат и по томе што је радио на првој атомској бомби, направио је прву компјутерску игру "Тенис за двоје" (Tennis for two). Што су компјутери постајали јачи, и игре су постајале све напредније. Индустрија видео игара је толико ојачала, и на њој се толико зарађује да данас постоје софтвери за прављење видео игара (Unreal Engine, Crytek ...) који коштају чак 700 000 долара.

### Андроид игре као подврста компјутерских игара

Као што је већ напоменуто, Андроид је једна од најнапреднијих и најпопуларнијих платформа за мобилне телефоне, и привлачи велики број програмера видео игара, како самосталних (инди), тако и оних из великих корпорација. Сам податак да је до јула 2013. године, дакле за мање од 5 година постојања платформе, на Гугловом тржишту постојало преко милион апликација, и да су оне преузете преко 50 милијарди пута је задивљујуће.

Оваква популарност Андроид платформе је засигурно због њеног отвореног стандарда. Идеја Андроид платформе је да она буде доступна свима, и да сваки програмер може да је користи. Као резултат те идеје, развој игара, и апликација уопште, и њихово објављивање за ову платформу су

практично бесплатни. Поред тога, популарност Андроид платформе омогућава приступ широком тржишту, што значи и велику могућност зараде.

Програмирање игара за Андроид уређаје је једноставно, бесплатно и забавно, и због тога је овај рад рађен управо за ову платформу.

## Теорија игара

Теорија игара је грана математике која проучава стратегију доношења одлука под одређеним склопом правила, односно игром. Теорија игара проучава сценарије у којима су укључена најмање два сукобљена играча. Ову област су први поменули Џон фон Нојман и Оскар Моргенштерн у њиховој књизи "Теорија игара и економско понашање" из 1944. године. Изузетан допринос овој области дао је и Џон Неш.

Пошто се теорија игара бави пројектовањем и планирањем развоја догађаја, и тражењем најоптималнијег исхода, повезана је са многим областима, од економије и политике, до еволуционе биологије и, наравно, информатике. Тражење најбоље стратегије је кључно за победу у било којој компјутерској игри, па и овој.

Принципи теорије игара су у овом раду често применљивани, поготово приликом прављења стратегија компјутерски генерисаних противника.

## Дефиниције

**Activity** - Одређена фокусирана активност коју играч може да врши. Како сваки activity интерагује са корисником, Activity ствара кориснички прозор који програмер може да уређује и због овога је веома значајан.

**View** - Основни градивни елемент програмирања за Андроид уређаје. Представља квадратну структуру која се исцртава и са којом се може интераговати.

**ImageView** - View који садржи слику и посебне процедуре везане за операције са сликама.

**WebView** - View који приказује интернет странице. Садржи специфичне процедуре.

## АЛАТКЕ КОРИШЋЕНЕ У ОВОМ РАДУ

У овом раду коришћени су следећи програми:

- Eclipse v3.8.2 са ADT-ом и LeJOS NXJ плугином
- Adobe Dreamweaver CS3
- Adobe Photoshop CS3

## ТЕМА ОВОГ РАДА

У овом раду се укратко обрађују основе АИ и роботике кроз примену комбинације игре за андроид мобилне уређаје и LEGO MINDSTORMS робота. Користи се:

- оригинални програм за андроид платформу,
- 2 мобилна телефона и
- 2 Lego Mindstorms NXT

## АНДРОИД ПРОГРАМ - КОМПЈУТЕРСКА ИГРИЦА

Апликација има три одвојена дела: телефонски, интернет и роботски. Главни део апликације је на телефону, и преко њега се приступа интернет и роботском делу. У овом поглављу ће бити више речи о прва два дела.

### Опис игре

Игра је направљена за Андроид оперативни систем и подржава три различита мода, односно режима рада, за играње - мод за једног играча, мод за два играча путем bluetooth-а, и "Robotic play" мод у коме се два играча надмећу у контроли робота. Поред тога, постоји и део апликације на интернету који формира и приказује базу података са поенима играча.

### Екрани игре

На почетку игре се појављује главни мени. На овом менију се налазе дугмићи који воде на остале меније: Мени за једног играча, за два играча, за "Robotic Play", за ранг-листу, за профиле и за информације. Прва три менија даље воде на модове игре. Мени за профиле омогућава руковођење профилима, о којима ће бити речи касније. Информације пружају помоћ и текст о апликацији. Ранг Листа даје ранг листу свих играча ове апликације, и у себи садржи WebView који показује вебсите, о коме ће такође касније бити речи.

Сваки мени има свој Activity који позива одређени layout. Layout-и за меније су сачувани у .xml фајлу.

Код за активирање layout-а: `setContentView(R.layout.main);`

### Графика

За развој игре било је потребно направити немали број цртежа. За потребе игре нацртано је 4 различита катапулта (за сваки једна "кашика" и једно "тело"), 6 различитих кугли (од којих две имају додатне 4 слике за анимацију репова), 5 позадинских мапа, 15 зидића који ограничавају кретање катапулта, 3 птице са по 4 слике за анимацију, мета са својом анимацијом, две муње са анимацијама, две ватрене кугле, леденица, 5 облака итд.



Графика у игри је оригинална, па самим тим и није савршена, али је у њу уложен труд. Рађена је у Adobe Photoshop CS3. Како је већина објеката у игри инстанца Image View-а, то је било потребно нацртати велики број слика, а у неким случајевима било је важно нацртати и додатне цртеже за анимацију.



### Gameplay (на телефону)

Пре почетка игре, играчи бирају свој катапулт од броја понуђених. Ови катапулти се разликују по одређеним карактеристикама. Играчи управљају својим катапултом користећи три команде - лево, десно и пуцај. Сваки катапулт има ограничен простор кретања; један катапулт игру почиње на левом делу екрана, други на десном, а између њих је зид који не могу да пређу. Циљ игре је спуштање животних поена противничког катапулта на нулу. Да би се тај циљ остварио, потребно је погодити противнички катапулт куглом. При том, постоје помагала - бонуси - које играч може искористити да би победио противника. Ови бонуси ће се, у виду мете, појављивати повремено на мапи, и освајају се погађањем мете куглом.

Ефекат једног од бонуса:

```
if(e_chill==true)
{
    e_chillTime-=50;
    if(e_chillTime%1000==0)
    {
        e_hp-=2;
        pb2.setProgress(e_hp);
    }
    if(e_chillTime==0)
    {
        e_chill=false;
        e_chillTime=5000;
    }
}
```

## Мод за једног играча

Мод за једног играча примењује овај gameplay. Мод је сачињен од 15 нивоа подељених у 5 секција. Свака секција има засебну мапу са засебним правилима, и играч мора да прилагоди своју стратегију свакој новој мапи. Секције се састоје од 3 нивоа, на којима играча чекају АИ противници, од једноставних ка комплексним. Прелажењем свих 15 нивоа играч добија награду у виду тапета који се аутоматски преузимају са веб сајта, и још једног катапулта за избор.

## Мод за два играча

Попут мода за једног играча, мод за два играча има 5 мапа на којима се може играти. Што се тиче gameplay-а, нема никакве разлике у односу на мод за једног играча. Једина разлика је у томе што противник није АИ, већ други играч. Веза која се користи за успостављање игре је Bluetooth. Пре почетка игре, играчу се излистају доступни уређаји против којих ће играч да се такмичи. Један од два повезана уређаја хостује конекцију, а други је клијент.

## Профили и чување игре

На једном телефону може се имати више од једног профила. Сваки профил се одликује са две ствари, достигнутом нивоом у моду за једног играча, и бројем поена оствареним играјући било који мод.

Када се први пут покрене игра, апликација прави један текстуални фајл у коме чува све профиле који постоје, као и њихов напредак у виду нивоа који су достигнули (број 1 до 15), и број освојених поена за ранг листу. Први пут нема профила, па се аутоматски прави "default профил" који је на 1. нивоу, и који има 0 поена. Нови профили се могу додавати, а постојећи мењати и брисати, у менију за профиле.

Код за ажурирање резултата неког профила после пређеног нивоа:

```
if(e_hp<=0)
{
    try
    {
        String[] textfile = new String[100];
        FileInputStream fis = openFileInput("catusers.txt");
        BufferedReader reader = new BufferedReader(new
InputStreamReader(fis));
        String s = "";
        int i = 0;
```

```

while((s = reader.readLine()) != null)
{
    textfile[i] = s;
    i++;
}
int chosen = Integer.parseInt(textfile[i-1]);
reader.close();
String profile = textfile[chosen];
fis = openFileInput(profile + ".txt");
reader = new BufferedReader(new InputStreamReader(fis));
s = "";
i = 0;
while((s = reader.readLine()) != null)
{
    textfile[i] = s;
    i++;
}
int current_points = Integer.valueOf(textfile[0]);
current_points+=15;
textfile[0] = String.valueOf(current_points);
int progress = Integer.valueOf(textfile[1]);
if(progress<2)
{
    progress++;
}
textfile[1] = String.valueOf(progress);
reader.close();

FileOutputStream fos = openFileOutput(profile + ".txt",
Context.MODE_PRIVATE);
BufferedWriter writer = new BufferedWriter(new
OutputStreamWriter(fos));
writer.write(textfile[0] + "\r\n");
writer.write(textfile[1] + "\r\n");
writer.flush();
writer.close();
}catch (IOException ioe) {
    ioe.printStackTrace();
}

t.cancel();
Intent intent = new Intent(Level1_1.this,WinScreen1.class);
Level1_1.this.startActivity(intent);
finish();
e_hp=1;
}

```

## Интернет део апликације

За потребу ове апликације направљена је посебна интернет страница. Она се састоји од базе података, једне странице и неколико фајлова окачених на интернет. Фајлови се, како је већ

речено, скидају на телефон приликом успешно завршеног мода за једног играча и они се не могу видети ни на који други начин (не рачунајући уношење тачног урл-а слике у претраживачу).

Веб страници се једино може приступити преко андроид апликације. Ова страница садржи ранг листу свих играча која се приликом приступа са телефона ажурира (додаје се корисников резултат и саставља се листа најбољих), што се ради путем Java и PHP кода и SQL упита. Приступ кода преко претраживача је такође онемогућен путем .htaccess фајла. Апликација на телефону отвара интернет страницу у Web View објекту.

### Структура интернет дела и објашњење

**.htaccess** - Онемогућава преглед структуре веб сајта у интернет претраживачима и подешава линк за Грешку 404 (непостојећа страница). У сваком фолдеру се налази један овакав фајл.

**missing.html** - Страница која се отвара приликом грешке 404.

**index.html** - Главни HTML фајл који одређује основне особине странице (нпр. уређење)

**images/** - Фолдер у коме се налазе слике које се користе на самом сајту (позадински градијент који се понавља по у-оси и позадинска слика за табелу)

**css/catapults.css** - CSS фајл који одређује визуалне карактеристике сајта (нпр. боја слова и особине позадинских слика)

**download/walpapers/** - Фолдер у коме се налазе слике које се скидају на телефон у случају преласка игре.

**js/work.js** - JS фајл у коме се дефинише функционалност сајта (првенствено функција која позива php код за повезивање са базом података)

**php/db.php, find.php, load.php, write.php** - php фајлови који служе за повезивање са базом података. Сваки од ових позива одређени SQL упит, осим првог који чува приступну информацију за базу података.

## Механика игре (на телефону)

Како се игра одвија у реалном времену, у њеној основи мора бити тајмер. На откуцај тајмера се извршавају све провере (колизије, животни поени, време до следећег бонуса...) и све кретње (катапулти, кугле...).

Код за тајмер:

```
t = new Timer(false);
t.scheduleAtFixedRate( new TimerTask() {
    private Handler updateUI = new Handler(){
        private Object setId;

        @Override
        public void dispatchMessage(Message msg) {
            super.dispatchMessage(msg);
            ...
        }
    };

    public void run() {
        try {
            updateUI.sendMessage(0);
        } catch (Exception e) {e.printStackTrace(); }
    }
}, 1000, 50);
```

Уместо (...) се ставља код који се извршава на откуцај тајмера.

## Катапулти

Како је већ речено, пре игре се бира између пар катапулта: Кедар Дрвени, Јавор Дрвени, Ледени и Царски (који се накнадно откључава). Они се разликују према карактеристикама, првенствено у брзини и броју животних поена, али и неким другим стварима. Тако, на пример, ледени катапулт је веома гломазан и спор, па је лака мета противнику. Међутим, издржљивији је од осталих и испаљује ледене кугле уместо обичних.



Брзина катапулта се одражава у померају на екрану по откуцају тајмера. Кретање се одвија тако што се на откуцају тајмера проверава да ли је притиснуто неко дугме за кретање (лево или десно). Испаливање се извршава повлачењем прста по екрану. У зависности од дужине повлака, и угла повлака са хоризонталом, катапулт испалије куглу под одређеним углом и интезитетом. Док је екран притиснут, појављује се потенцијална трајекторија кугле, тј. пут под којим би лопта летела да је у том тренутку испалиена. Овом механиком испаливања се искоришћава додирни елемент екрана телефона.

У самој игри, катапулт је састављен од два Image View-а, један за кашику катапулта, а један за тело. Анимације за кретање кашике катапулта су испрограмиране, не користи се вишак слика чиме се смањује величина апликације.

Кретање кугле:

```
        if(ball_exist==true)
        {
            vy+=1;
            boolean ok = true;
            ImageView cannonball = (ImageView)
findViewById(R.id.cannonball1);
            RelativeLayout.LayoutParams params = new
RelativeLayout.LayoutParams(RelativeLayout.LayoutParams.WRAP_CONTENT,
RelativeLayout.LayoutParams.WRAP_CONTENT);
            cbx+=vx*scrwidth/600;
            cby+=vy*scrwidth/600;

            if ((cbx>scrwidth)|| (cbx<0))
            {
                cbx=0;
                vx = 0;
                cby=0;
                vy = 0;
                ok = false;
            }
            if ((cby>scrheight)|| (cby<0))
            {
                cbx=0;
                vx = 0;
                cby=0;
                vy = 0;
                ok = false;
            }
            if(ok==false)
            {
                ball_exist = false;
                RelativeLayout main = (RelativeLayout)
findViewById(R.id.main);
                main.removeView(cannonball);
            }
        }
```

```
params.leftMargin = (int)cbx;  
params.topMargin = (int)cby;  
cannonball.setLayoutParams(params); ...
```

Овај код се извршава на откуцај тајмера. Прво се повећава брзина кугле по вертикали (чиме се симулира гравитација), онда се положај кугле мења према брзини. Затим се врши провера да ли је кугла и даље у оквиру екрана и, ако није, брише се из игре. Приликом помераја кугле,  $scrwidth/600$  је константа која служи да уједначи брзину кретања кугле на различитим телефонима.

### Кугле и бонуси

Бонус мета, која се у случајно генерисаном интервалу појављује на средини мапе (са изузетком планинске мапе) служи да донесе играчу специјалну куглу или бонус животне поене. Бонуси се ређају у инвентар и позивају по жељи. Специјалне кугле које се добијају су кугла-скочица, ватрена, ледена, камена и отровна кугла. Свака од ових има посебан ефекат. Скочица одскаче од земље 3 пута, па чак иако је противнички катапулт промашен испрва, кугла ће се одбити до њега. Ватрена на противничком пољу испаљује мале вертикалне кугле итд.

Кугле су у апликацији представљене једним Image View-ом. Углавном су то кугле димензија 16x16, али неке кугле имају реп. Анимација за ове друге је изузетно сложена, и састоји се из комбинације већег броја слика и програмског ротирања Image View-а.

### Мапе

Разлика између мапа није само у позадинској слици. Наиме, свака мапа има одређене елементе које је чине различитом у односу на остале. Мапа има пет: Пољана, Планине, Ледена пећина, Облаци, Замак. Разлике између њих су велике и умногоме утичу на победничку стратегију играча. Разлике између мапа су свакојаче, од промене начина кретања катапулта до посебних елемената који могу да оштете или заштите катапулте. На пример, у Леденој пећини постоји проклизавање катапулта, у Планинама сваки катапулт има заштиту над делом свог поља, из замка се испаљују ватрене кугле на играче...

Сви ови елементи имају кретање и ефекте дефинисано у горе наведеном Тајмеру, и углавном су инстанце класе Image View.



## Колизија

У овој игри је посебна пажња посвећена алгоритму колизије између објеката. Користе се две функције за одређивање колизије, једноставнија и комплекснија. Једноставнија колизија гледа да ли се оквир сличица објеката за које се проверава колизија преклапа, и не узима у обзир транспарентност пиксела. Ова колизија је добра само у одређеним случајевима, за колизију између зидића и катапулта.

Међутим, за колизију између катапулта и кугле, и кугле и зидића, ово није довољно и може довести до оштећења играча. Зато је било потребно увести комплекснији алгоритам колизије, који проверава по пикселима. Овај алгоритам за колизију не сматра оне случајеве где се на местима на којима се слике преклапају налазе транспарентни пиксели.

Слабији алгоритам колизије:

```
public boolean isNormalCollision(ImageView sprite1, ImageView sprite2)
{
    Rect bounds1 = new Rect(sprite1.getLeft(), sprite1.getTop(),
    sprite1.getRight(), sprite1.getBottom());
    Rect bounds2 = new Rect(sprite2.getLeft(), sprite2.getTop(),
    sprite2.getRight(), sprite2.getBottom());
    if( Rect.intersects(bounds1, bounds2) ){
        return true;
    }
    return false;
}
```

Јачи алгоритам колизије:

```
public boolean isCollisionDetected(ImageView sprite1, ImageView sprite2){
```



```

    Rect bounds1 = new Rect(sprite1.getLeft(), sprite1.getTop(),
sprite1.getRight(), sprite1.getBottom());
    Rect bounds2 = new Rect(sprite2.getLeft(), sprite2.getTop(),
sprite2.getRight(), sprite2.getBottom());

    if( Rect.intersects(bounds1, bounds2) ){
        Rect collisionBounds = getCollisionBounds(bounds1, bounds2);
        for (int i = collisionBounds.left; i < collisionBounds.right; i++) {
            for (int j = collisionBounds.top; j < collisionBounds.bottom; j++) {
                int sprite1Pixel = getBitmapPixel(sprite1, i, j);
                int sprite2Pixel = getBitmapPixel(sprite2, i, j);
                if( isFilled(sprite1Pixel) && isFilled(sprite2Pixel)) {
                    return true;
                }
            }
        }
    }
    return false;
}

private static int getBitmapPixel(ImageView sprite, int i, int j) {
    sprite.buildDrawingCache();
    Bitmap bmap = sprite.getDrawingCache();

    return bmap.getPixel(i-(int)sprite.getLeft(), j-(int)sprite.getTop());
}

private static Rect getCollisionBounds(Rect rect1, Rect rect2) {
    int left = (int) Math.max(rect1.left, rect2.left);
    int top = (int) Math.max(rect1.top, rect2.top);
    int right = (int) Math.min(rect1.right, rect2.right);
    int bottom = (int) Math.min(rect1.bottom, rect2.bottom);
    return new Rect(left, top, right, bottom);
}

private static boolean isFilled(int pixel) {
    return pixel != Color.TRANSPARENT;
}
}

```

### АИ противници - пример вештачке интелигенције

АИ противника представља примену вештачке интелигенције у овој игри, па је у њих уложено доста пажње. АИ противника је посебан за сваки од 15 нивоа. Нивои се разликују у тежини, и "ранији" нивои су лакши од "каснијих". Главна разлика у тежини нивоа је свакако напреднији алгоритам АИ противника у "каснијим" нивоима, мада то није и једина разлика.

АИ противници имају различит начин понашања, и различит степен развијености на различитим нивоима. Неки су мање, неки више прецизни. Неки се мање, неки више крећу. Неки боље, а неки лошије избегавају којима их играчи гађају. Иако се скоро сви труде да узимају бонусе (са

изузетком првог), неки њима боље рукују од других. Такође, АИ противници морају, као и играчи, имати засебну стратегију за засебне нивое, и за засебне катапулте којима се служе. Тако, на пример, АИ са спорим, леденим катапултом ће се мање кретати, и гледаће да буде прецизнији у односу на АИ са неким другим катапултом, а АИ на Облацима ће урачунати кретање облака по вертикали приликом циљања играча.

Један од једноставнијих АИ алгоритама за избегавање играчеве кугле:

```
public boolean AI_avoid(double x, double y)
{
    boolean moved = false;
    ImageView cata2_body = (ImageView) findViewById(R.id.cata2_body);
    ImageView cata2_throw = (ImageView) findViewById(R.id.cata2_throw);
    double xd = cata2_body.getLeft() - x;
    double yd = cata2_body.getTop() - cata2_throw.getHeight() - y;
    double t = xd/vx;
    double yp = 0;
    double ty = 0;
    if(vy<0)
    {
        ty = Math.abs(vy*2);
        yp = 0 - vy*(t-ty);
    }
    else
    {
        yp = Math.abs(vy*t);
    }
    yp += y;
    if((yp >= cata2_body.getTop() + cata2_body.getHeight() -
cata2_throw.getHeight())&&(yp <=cata2_body.getTop() + cata2_body.getHeight()))
    {
        moved = true;
        if(yp<=(cata2_body.getTop() + cata2_body.getHeight() -
(cata2_throw.getHeight()/2)))
        {
            AI_right();
        }
        else
        {
            AI_left();
        }
    }
}
```

## Остале одлике игре

Поред ових, постоји и још неколико одлика игра које вреди поменути. То су:

- оригинална прича која се развија током напретка игре,
- избор катапулта у моду за једног играча
- графика и програм су подешени за 4 стандардне густине андроид екрана (ldpi, mdpi, hdpi и xdpi)
- апликација је вишејезична са избором језика (Српски као главни и Енглески)
- позадински звук за сваки ниво и звучни ефекти
- инплементиран оригинални фонт Catapults.ttf
- тапети за андроид телефоне, са сликама из игре за преузимање као награда после победе (7 рализитих тапета, за ldpi, mdpi, hdpi и xdpi)...

### Структура телефонског дела апликације

#### **com.catapults:**

**ChooseLang.java** - Ово је Activity који се појављује на почетку игре и служи за избор језика апликације (српски или енглески).

**DotView.java** - Класа која дефинише View којим се црта трајекторија катапултске кугле.

**DrawView.java** - Класа која дефинише View којим се црта линија која представља угао и интензитет којим ће се избацити кугла из кашике катапулта.

**First/Second/Third/Fourth/Fifth/SixthActivity.java** - Activity-ји за различите изборне меније у игри.

**LevelX\_Y.java** - Ови Activity-ји одређују све што се дешава на нивоима игре, од AI-ја до тајмера. X иде од 1 до 5 (за сваку секцију по 1) а Y иде од 1 до 3 (за сваки подниво на секцији).

**mainActivity.java** - Главни мени.

**Multiplayer/Multiplayer2-5.java** - Исто што и LevelX\_Y.java само за мод за два играча. Како нема AI-ја, није било потребно правити 15 мапа за мод за више играча, већ само пет - за сваку секцију по један.

**MultiplayerBTControl.java** - Класа која контролише Bluetooth везу између два телефона у режиму за два играча

**MultiplayerDeviceList.java** - Изборни Activity који се јавља приликом остваривања конекције између два телефона. Омогућава играчу да бира телефон са којим ће да се повеже.

**MultiplayerMain.java** - Изборни мени за мод за више играча.

**Story1-5.java** - Ових 5 activity-ја се појављују у моду за једног играча и садрже текст са развојем приче.

**Winscreen1-5/Winscreen\_final.java** - Екран који се отвара по остваривању победе у игри, било у моду за једног или два играча. Последњи победнички екран садржи код за преузимање награде са веб сајта.

**res->layout:**

**custom\_title.xml** - Дефинише изглед назива апликације који се на појединим деловима апликације јавља у врху екрана.

**device\_list.xml** - Дефинише layout, тј. уређење екрана за одабир телефона за повезивање путем bluetooth везе.

**device\_name.xml** - Дефинише изглед опција за одабир телефона на истом екрану.

**first/second/third/fourth/fifth/sixth.xml** - Дефинише уређење различитих менија у игри.

**lang.xml** - Дефинише уређење менија за избор језика.

**main/level2-5.xml** - Дефинише уређење и изглед нивоа у игрици.

**main\_menu.xml** - Дефинише уређење главног менија.

**multiplayer\_main.xml** - Дефинише уређење менија за мод за више играча.

**story/story2-5.xml** - Дефинише уређење екрана на којим се појављује прича.

**win1-5/win\_final.xml** - Дефинише уређење победничких екрана.

У res-> menu постоји и options\_menu.xml фајл који се јавља У моду за два играча и даје кориснику могућност претраживања bluetooth уређаја и укључивања bluetooth-а. У res-> drawable се налазе слике и .xml фајлови који дефинишу анимације. У res->raw се налазе музички фајлови .aac формата који се користе у игри. У res->values се чувају константе, стрингови и ИД-ови.

## ROBOTIC PLAY МОД - ПРИМЕНА РОБОТА

У овом моду се уводи роботика у ову апликацију. Идеја иза овог мода је да исти gameplay који се користи у телефону применимо и у реалности, тј. да се игра унапреди тако што ће се виртуелни сегменти апликације заменити физичким, реалним објектима - роботима. Роботи који се користе су Lego Mindstorms NXT са уграђеним LeJOS NXJ firmware-ом који обезбеђује Јава Виртуелну Машину на роботу. Имплементација овог firmware-а је омогућила израду дела апликације намењеног роботу у Јави.

### Лего робот

#### Лего NXT Сет

За изградњу робота у овој апликацији се користи NXT сет. То је програмабилни роботски сет који је, због слободе у дизајну и једноставности израде робота (изградња робота је ништа друго до слагање Лего коцкица), изузетно цењен и користи се као почетнички сет на многим универзитетима. Сет се састоји од једне Лего NXT коцке ("мозак" робота) која има 4 дугмета и LCD екран, 3 Серво мотора, и 4 сензора: додирни, гласовни, ултраљубичасти (за мерење растојања) и сензор за боје. Поред ових делова који су кључни за програмирање робота, уз сет долази и 519 градивних елемената различитог облика и функције, и од њих се састављају роботи.

#### Дизајн Лего робота

Лего робот је дизајниран налик нацртаним катапултима из мода за једног играча. Тело катапулта чини Лего NXT блок, који представља "мозак" робота, испод кога се налазе три мотора - два за кретање (један за леви, други за десни точак) и један за кашику катапулта (тј. механизам испљивања кугле). Кашка катапулта се налази са задње стране робота. Лоптице које долазе уз Лего сет представљају кугле. Са предње стране робота налази се трећи, помоћни точак. Изнад њега се налази додирни сензор и систем за колизију, који, у случају поготка, завршава игру.



## Програмирање Robotic Play мода

Robotic Play мод се у потпуности разликује од осталих модова игре. Програм је подељен у два дела, за телефон и за робота. Веза између телефона и робота се успоставља bluetooth-ом, где телефон шаље, а робот прима информације.

### Део апликације на роботу

Као што је већ речено, робот чека на информацију коју му шаље телефон. У почетку, робот чека да се оспособи bluetooth веза. Затим се са телефона шаље број, и у зависности од броја који је послат робот извршава неку од функција - кретање унапред, лево, десно, уназад или испаливање кугле. Апликација се гаси, и bluetooth веза се прекида у случају активације додирног сензора - тј. када противничка кугла погоди систем за колизију.

### Део апликације на телефону

Притиском дугмета "Robotic play" на почетном екрану отвара се овај мод. У почетку, играч бира NXT робота са којим ће да се повеже и телефон противника од листе доступних bluetooth уређаја. После овога се успоставља конекција и на екрану се појављују команде за контролу робота, пет дугмета, за напред, назад, лево, десно и испаливање. Робот извршава одређену команду све док је дугме за ту команду притиснуто. Да би се смањила количина послате информације (пошто bluetooth веза има одређену брзину слања информације, око 800 бита у секунди), преко bluetooth везе се шаље код на робота кад се дугме притисне, и кад се дугме пусти (стоп код).

Кад један од робота бива погођен, он гаси своју апликацију и прекида bluetooth везу. По томе што је веза прекинута, апликација зна да је робот погођен, и да је тај играч изгубио. Тада се bluetooth-ом шаље сигнал другом телефону да је игра готова, и да је тај телефон победник.

## Структура роботског дела апликације

### На телефону

**BroadcastReceiver.java** - Овај фајл прати информације Bluetooth система.

**ButtonHolder.java** - Овај фајл води рачуна о притиску и пуштању команди које управљају роботом.

**Control.java** - Главни фајл који повезује све остале фајлове овог дела апликације на телефону.

**Message.java** - Класа која дефинише особине порука које се шаљу роботу.

**MessageTypes.java** - Класа која дефинише врсте порука које се шаљу роботу и њихово значење.

**SixthActivityMain.java** - Изборни мени на коме се бира робот и противнички телефон са којим се повезује.

**sixth.xml** дефинише изглед овог дела апликације.

### На роботу

**ButtonListener.java** - Класа која чека, прима и тумачи поруке са телефона.

**Main.java** - Главна класа на телефону, дефинише понашање робота.

**Quiter.java** - Одређује процедуру приликом краја игре.

## ЗАКЉУЧАК

Сврха ове апликације је повезивање три изузетно компетитивне области - роботике, вештачку интелигенцију и мобилне телефоне - кроз једну видео игру. Кроз игру, неозбиљну и безбрижну творевину се демонстрирају основни принципи ових области. Технике које су примењене у овом раду се на сличан начин могу применити у другим пројектима, са далеко озбиљнијом сврхом, па чак и са утицајем на развој науке и човечанства.



## ЛИТЕРАТУРА

1. Joel Ouaknine - "Algorithmic probabilistic game semantics: Playing games with automata"
2. Alan Turing - "Computing Machinery and Intelligence"
3. Ian Darwin - "Android Cookbook Problems and Solutions for Android Developers"
4. [www.wikipedia.org](http://www.wikipedia.org)