Mobile App Development Report: The Dungeon Run

Charlie Coyle
403225029@napier.ac.uk
Edinburgh Napier University - Module Title (SET08114)

1 Introduction

1.1 Project Description

The app allows a user to create a profile and play trough a game with the profiles attributes being integral. The game its self consists of 30 predetermined assailants(Loaded from a CSV File) that the user must defeat using on screen buttons which represent actions, the Foe will pick an action based on an algorithm which takes into account the Foes properties. Upon completing the game the user is issued the name and there score which is represented by the time elapsed during the play session.

1.2 Project Goal

To create an application that can entertain a user for a short burst of time, through the use of careful planing and execution to insure ample quality.

1.3 Scope

- Use of input from files
- Validation
- A user friendly design/aesthetic.
- Use of multidimensional arrays to store data.
- Displays all relevant information in an obvious manner.

1.4 Boundaries

- The user will have no direct access to Files containing game relevant data.
- The user will be unable to share contents of application to external sources.

2 Software Design

2.1 UML Activity Diagram

A UML(unified Modeling language) activity diagram was created as screen in Figure 1: the diagram shows the actions taken by the user to complete certain tasks for which the application will be used and from this, I learned the tasked to be completed.

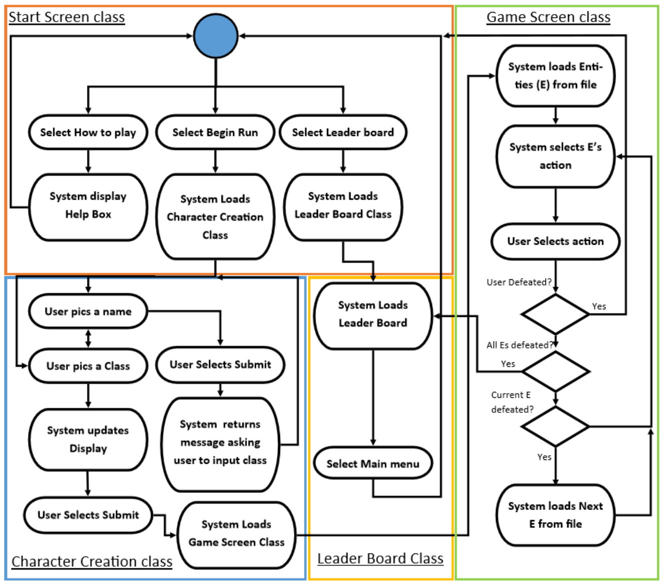


Figure 1: **UML Activity Diagram** UML Diagram of Mobile Application

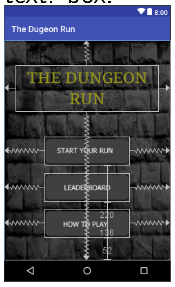
2.2 Pseudocode

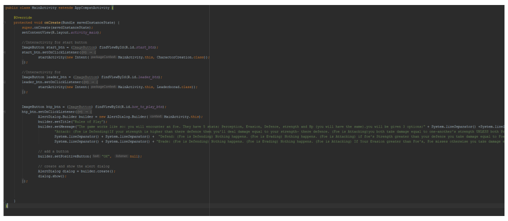
```
Foes AI: EAction is Global variable
EAction = new random number(between 1 and 4)
if EAction == 4 then
    if Entity's strength is largest Stat then
        Set EAction = 1
    else
        if Entity's defence is largest stat then
            Set EAction = 2
        else
            Set EAction = 3
        end
    end
end
switch EAction do
    Case 1 Display Foe is Attacking
    Case 2 Display Foe is Defending
    Case 3 Display Foe is Evading
end
```

3 Implementation

3.1 Main Screen Class

The implementation of the main screen was simple and only took around 2hours all that was needed was some simple XML and code to change to different classes or display a text. box.

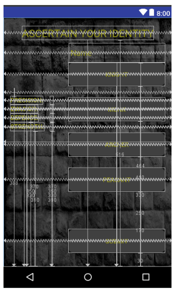


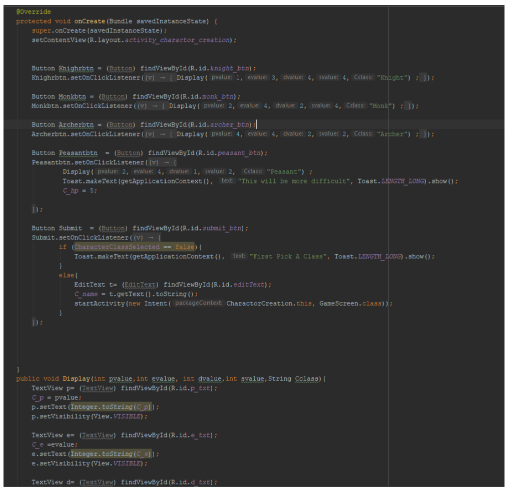


```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //Interactivity for start button
        ImageButton start_btn = (ImageButton) findViewById(R.id.start_btn);
        start_btn.setOnClickListener((v) → {
                startActivity(new Intent( packageContext: MainActivity.this, CharactorCreation.class));
        });

        //Interactivity for
        ImageButton leader_btn = (ImageButton) findViewById(R.id.leader_btn);
        leader_btn.setOnClickListener((v) → {
                startActivity(new Intent( packageContext: MainActivity.this, Leaderborad.class));
        });


        ImageButton htp_btn = (ImageButton) findViewById(R.id.how_to_play_btn);
        htp_btn.setOnClickListener((v) → {
                AlertDialog.Builder builder = new AlertDialog.Builder( context: MainActivity.this);
                builder.setTitle("Rules of Play");
                builder.setMessage("The game works like so: you will encounter an foe. They have 5 stats; Perception, Evasion, Defence, strength and Hp (you will have the same).you will be given 3 options;" + System.lineSeparator() +System.line
                        "Attack: (Foe is Defending)If your strength is higher than there defence then you'll deal damage equal to your strength- there defence. (Foe is Attacking)you both take damage equal to one-another's strength UNLESS both Pr
                        System.lineSeparator() + System.lineSeparator() +  "Defend: (Foe is Defending) Nothing happens. (Foe is Evading) Nothing happens. (Foe is Attacking) if foe's Strength greater than your defence you take damage equal to Foe
                        System.lineSeparator() + System.lineSeparator() + "Evade: (Foe is Defending) Nothing happens. (Foe is Evading) Nothing happens. (Foe is Attacking) If Your Evasion greater than Foe's, Foe misses otherwise you take damage e

                // add a button
                builder.setPositiveButton( text: "OK",  listener: null);

                // create and show the alert dialog
                AlertDialog dialog = builder.create();
                dialog.show();
        });


    }
}
```

3.2 Character Creation Class

In character creation class require the allocation of global variables and input validation to insure the user didn't enter the game without first selecting a class. Overall the implementation of the character creation class was smooth only taking a few hours to complete, encountered issue with XML re-sizing but fixed the issue by adding additional constraints.

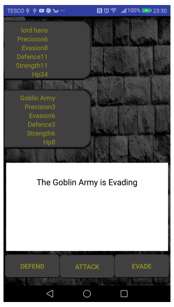




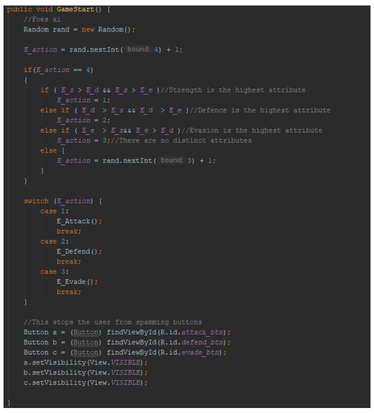
```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_charactor_creation);


    Button Knighrbtn = (Button) findViewById(R.id.knight_btn);
    Knighrbtn.setOnClickListener((v) → { Display( pvalue: 1, evalue: 3, dvalue: 4, svalue: 4, Cclass: "Knight") ; });

    Button Monkbtn = (Button) findViewById(R.id.monk_btn);
    Monkbtn.setOnClickListener((v) → { Display( pvalue: 2, evalue: 4, dvalue: 2, svalue: 4, Cclass: "Monk") ; });

    Button Archerbtn = (Button) findViewById(R.id.archer_btn);
    Archerbtn.setOnClickListener((v) → { Display( pvalue: 4, evalue: 4, dvalue: 2, svalue: 2, Cclass: "Archer") ; });

    Button Peasantbtn  = (Button) findViewById(R.id.peasant_btn);
    Peasantbtn.setOnClickListener((v) → {
            Display( pvalue: 2, evalue: 4, dvalue: 1, svalue: 2,  Cclass: "Peasant") ;
            Toast.makeText(getApplicationContext(),  text: "This will be more difficult", Toast.LENGTH_LONG).show();
            C_hp = 5;

    });

    Button Submit  = (Button) findViewById(R.id.submit_btn);
    Submit.setOnClickListener((v) → {
            if (CharacterClassSelected == false){
                Toast.makeText(getApplicationContext(),  text: "First Pick A Class", Toast.LENGTH_LONG).show();
            }
            else{
                EditText t= (EditText) findViewById(R.id.editText);
                C_name = t.getText().toString();
                startActivity(new Intent( packageContext: CharactorCreation.this, GameScreen.class));
            }
    });




}
public void Display(int pvalue,int evalue, int dvalue,int svalue,String Cclass){
    TextView p= (TextView) findViewById(R.id.p_txt);
    C_p = pvalue;
    p.setText(Integer.toString(C_p));
    p.setVisibility(View.VISIBLE);

    TextView e= (TextView) findViewById(R.id.e_txt);
    C_e =evalue;
    e.setText(Integer.toString(C_e));
    e.setVisibility(View.VISIBLE);

    TextView d= (TextView) findViewById(R.id.d_txt);
```

3.3 Game Screen Class

This Class Game gave me the most trouble at first but after the CSV file import section was complete the implementation was rarely impeded. Then the Entity AI was coded following from my Pseudocode this was mainly painless. Both user interaction and the display update were coded with minimal issue. The game checks were implemented as part of the Button listeners and were simple to set up and run. This section of code was the longest section of code and took around 16 hours to complete

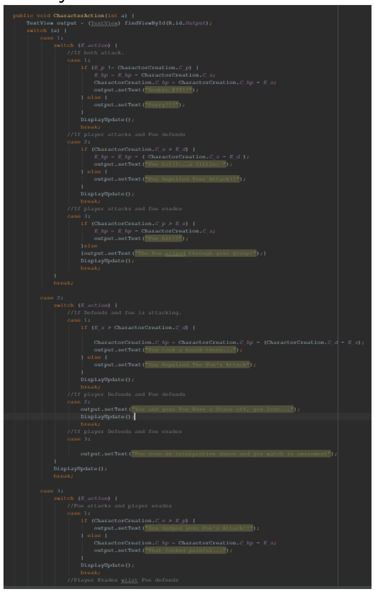


```
lord hans
Precision6
Evasion8
Defence11
Strength11
Hp34

Goblin Army
Precision3
Evasion6
Defence3
Strength6
Hp8

The Goblin Army is Evading
```



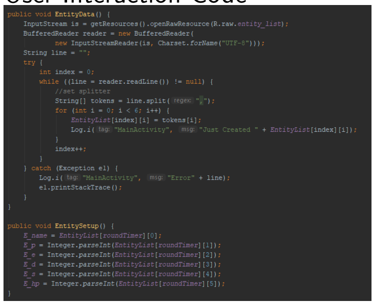
```
public void GameStart() {
    //Foes ai
    Random rand = new Random();

    E_action = rand.nextInt( bound: 4) + 1;

    if(E_action == 4)
    {
        if ( E_s > E_d && E_s > E_e )//Strength is the highest attribute
            E_action = 1;
        else if ( E_d  > E_s && E_d  > E_e )//Defence is the highest attribute
            E_action = 2;
        else if ( E_e  > E_s&& E_e > E_d )//Evasion is the highest attribute
            E_action = 3;//There are no distinct attributes
        else {
            E_action = rand.nextInt( bound: 3) + 1;
        }
    }

    switch (E_action) {
        case 1:
            E_Attack();
            break;
        case 2:
            E_Defend();
            break;
        case 3:
            E_Evade();
            break;
    }

    //This stops the user from spamming buttons
    Button a = (Button) findViewById(R.id.attack_btn);
    Button b = (Button) findViewById(R.id.defend_btn);
    Button c = (Button) findViewById(R.id.evade_btn);
    a.setVisibility(View.VISIBLE);
    b.setVisibility(View.VISIBLE);
    c.setVisibility(View.VISIBLE);

}
```

Entity AI Code



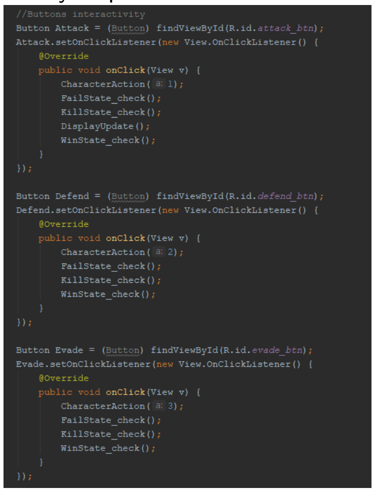
```
public void CharacterAction(int a) {
    TextView output = (TextView) findViewById(R.id.Output);
    switch (a) {
        case 1:
            switch (E_action) {
                //If both attack.
                case 1:
                    if (E_p != CharactorCreation.C_p) {
                        E_hp = E_hp - CharactorCreation.C_s;
                        CharactorCreation.C_hp = CharactorCreation.C_hp - E_s;
                        output.setText("Double HIT!!");
                    } else {
                        output.setText("Parry!!!");
                    }
                    DisplayUpdate();
                    break;
                //If player attacks and Foe defends
                case 2:
                    if (CharactorCreation.C_s > E_d) {
                        E_hp = E_hp - ( CharactorCreation.C_s - E_d );
                        output.setText("Foe hit!!...a little. ");
                    } else {
                        output.setText("Foe Repelled Your Attack!!");
                    }
                    DisplayUpdate();
                    break;
                //If player attacks and foe evades
                case 3:
                    if (CharactorCreation.C_p > E_e) {
                        E_hp = E_hp - CharactorCreation.C_s;
                        output.setText("Foe hit!!");
                    }else
                    {output.setText("The Foe sliped through your grasp!");}
                    DisplayUpdate();
                    break;
            }
            break;

        case 2:
            switch (E_action) {
                //If Defends and foe is attacking.
                case 1:
                    if (E_s > CharactorCreation.C_d) {

                        CharactorCreation.C_hp = CharactorCreation.C_hp - (CharactorCreation.C_d - E_s);
                        output.setText("You took a knock there...");
                    } else {
                        output.setText("You Repelled The Foe's Attack");
                    }
                    DisplayUpdate();
                    break;
                //If player Defends and Foe defends
                case 2:
                    output.setText("You and your Foe Have a Stare off, you lost...");
                    DisplayUpdate();
                    break;
                //If player Defends and foe evades
                case 3:

                    output.setText("Foe dose an interpretive dance and you watch in amazement");
            }
            DisplayUpdate();
            break;

        case 3:
            switch (E_action) {
                //Foe attacks and player evades
                case 1:
                    if (CharactorCreation.C_e > E_p) {
                        output.setText("You dodged your Foe's Attack!!");
                    } else {
                        CharactorCreation.C_hp = CharactorCreation.C_hp - E_s;
                        output.setText("That looked painful...");
                    }
                    DisplayUpdate();
                    break;
                //Player Evades wilst Foe defends
```

User Interaction Code



```
public void EntityData() {
    InputStream is = getResources().openRawResource(R.raw.entity_list);
    BufferedReader reader = new BufferedReader(
            new InputStreamReader(is, Charset.forName("UTF-8")));
    String line = "";
    try {
        int index = 0;
        while ((line = reader.readLine()) != null) {
            //set splitter
            String[] tokens = line.split( regex: ",");
            for (int i = 0; i < 6; i++) {
                EntityList[index][i] = tokens[i];
                Log.i( tag: "MainActivity",  msg: "Just Created " + EntityList[index][i]);
            }
            index++;
        }
    } catch (Exception e1) {
        Log.i( tag: "MainActivity",  msg: "Error" + line);
        e1.printStackTrace();
    }
}

public void EntitySetup() {
    E_name = EntityList[roundTimer][0];
    E_p = Integer.parseInt(EntityList[roundTimer][1]);
    E_e = Integer.parseInt(EntityList[roundTimer][2]);
    E_d = Integer.parseInt(EntityList[roundTimer][3]);
    E_s = Integer.parseInt(EntityList[roundTimer][4]);
    E_hp = Integer.parseInt(EntityList[roundTimer][5]);
}
```

Entity Import Code

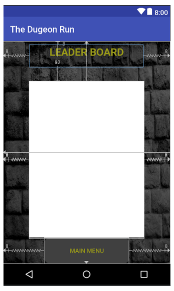


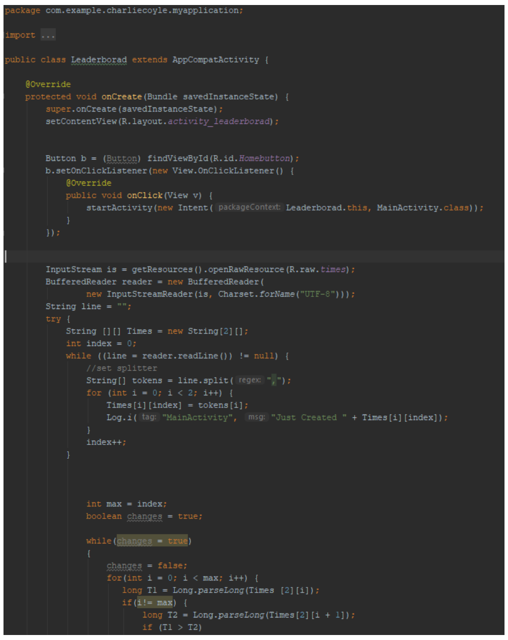
```
//Buttons interactivity
Button Attack = (Button) findViewById(R.id.attack_btn);
Attack.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        CharacterAction( a: 1);
        FailState_check();
        KillState_check();
        DisplayUpdate();
        WinState_check();
    }
});

Button Defend = (Button) findViewById(R.id.defend_btn);
Defend.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        CharacterAction( a: 2);
        FailState_check();
        KillState_check();
        WinState_check();
    }
});

Button Evade = (Button) findViewById(R.id.evade_btn);
Evade.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        CharacterAction( a: 3);
        FailState_check();
        KillState_check();
        WinState_check();
    }
});
```

Button Listeners

3.4 Leader board Class

The Leader board class was by far the simplest class it displays the leader board and allows the user to return to the main screen.The implementation of this class took around 2hours.





```
package com.example.charliecoyle.myapplication;

import ...

public class Leaderborad extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_leaderborad);


        Button b = (Button) findViewById(R.id.Homebutton);
        b.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent( packageContext: Leaderborad.this, MainActivity.class));
            }
        });


        InputStream is = getResources().openRawResource(R.raw.times);
        BufferedReader reader = new BufferedReader(
                new InputStreamReader(is, Charset.forName("UTF-8")));
        String line = "";
        try {
            String [][] Times = new String[2][];
            int index = 0;
            while ((line = reader.readLine()) != null) {
                //set splitter
                String[] tokens = line.split( regex: ",");
                for (int i = 0; i < 2; i++) {
                    Times[i][index] = tokens[i];
                    Log.i( tag: "MainActivity",  msg: "Just Created " + Times[i][index]);
                }
                index++;
            }



                int max = index;
                boolean changes = true;

                while(changes = true)
                {
                    changes = false;
                    for(int i = 0; i < max; i++) {
                        long T1 = Long.parseLong(Times [2][i]);
                        if(i!= max) {
                            long T2 = Long.parseLong(Times[2][i + 1]);
                            if (T1 > T2)
```

4 Critical Evaluation

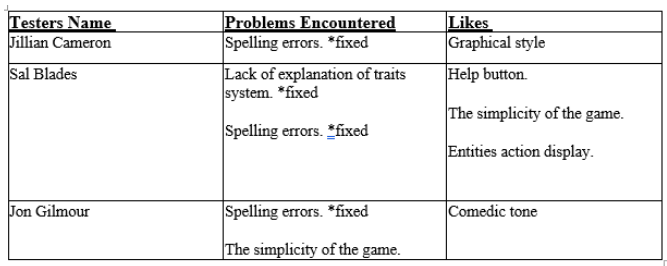
4.1 Comparison to Concept

The game runs the way it was designed, all features were implemented and function like intended.

4.2 Comparison to other Applications

The application; functions well in comparison to other applications, is User friendly in design and follows a consistent aesthetic through out which isn't always the case for apps. It doses however lack comparable graphics when compared to bigger apps.

4.3 User Feedback/ Possible Changes



Testers Name	Problems Encountered	Likes
Jillian Cameron	Spelling errors. *fixed	Graphical style
Sal Blades	Lack of explanation of traits system. *fixed Spelling errors. *fixed	Help button. The simplicity of the game. Entities action display.
Jon Gilmour	Spelling errors. *fixed The simplicity of the game.	Comedic tone

From this its reasonable to assume that the game could use more features like; more graphics, more game play elements and a custom stat editor for the character creator class.

5 Personal Evaluation

Considering the factor of not having previous experience working in java or mobile app development I am entirely happy with the outcome BUT if i were to start the app development now with the comparable wealth of knowledge i now have, the outcome would be entirely different. Learning Java from scratch allowed me to grow as an individual and broadens my understanding of mobile app development and its perils. My performance for the most part was good but i lack discipline in my time keeping ability. Overall i fell that the development of my application was a success.