# Team 57-
# Blackbox Studios

# The System, the Process and the Final Printed Report

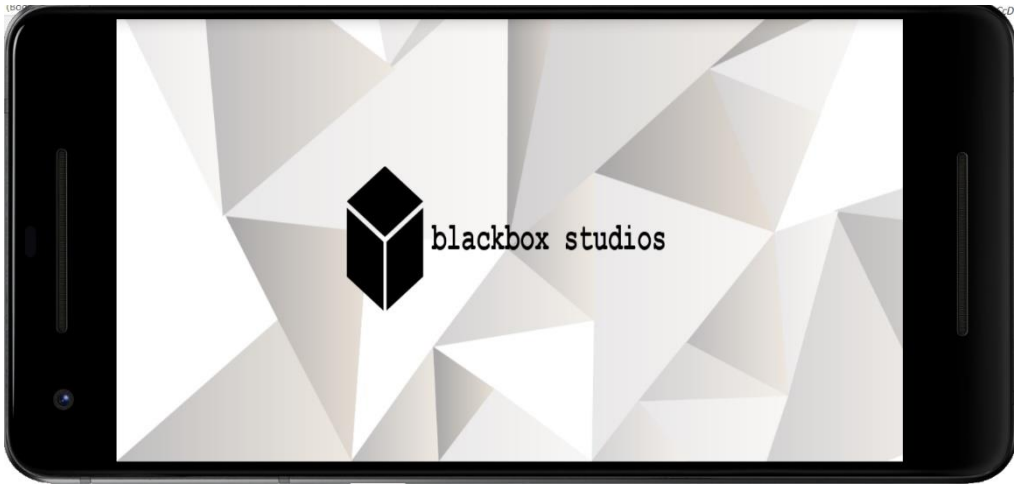

| Name | Student Number | QUB Email Address |
|---|---|---|
| Brandon Smylie | 40178070 | bsmylie02@qub.ac.uk |
| Conor Clarke | 40184143 | cclarke44@qub.ac.uk |
| Edward Muldrew | 40176081 | emuldrew03@qub.ac.uk |
| Grace Turkington | 40172213 | gturkington05@qub.ac.uk |
| Jack McNaughton | 40179912 | jmcnaughton01@qub.ac.uk |
| **GitLab Project** | csc-lvl2-1718-57 (Team 57) | |

# Contents

# Documented interface design

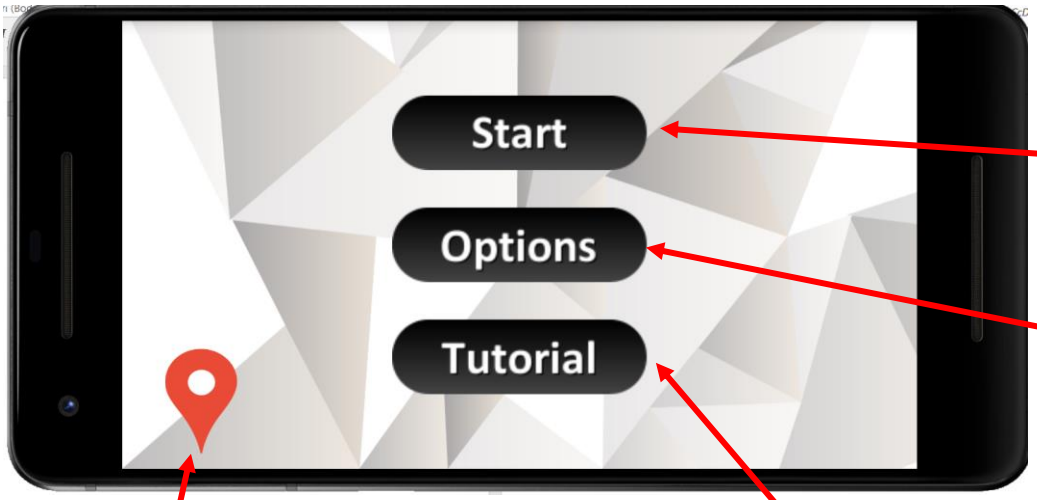
## Splash Screen

The splash screen is the first screen the user will see and shows the user the developers of the game. The user simply clicks on this screen to get brought to the options screen.




## Menu Screen

The Menu screen carries design consistency from the splash screen. The menu screens provides visible options for the user to alter the course of the game.




When clicked enters user in to game play.

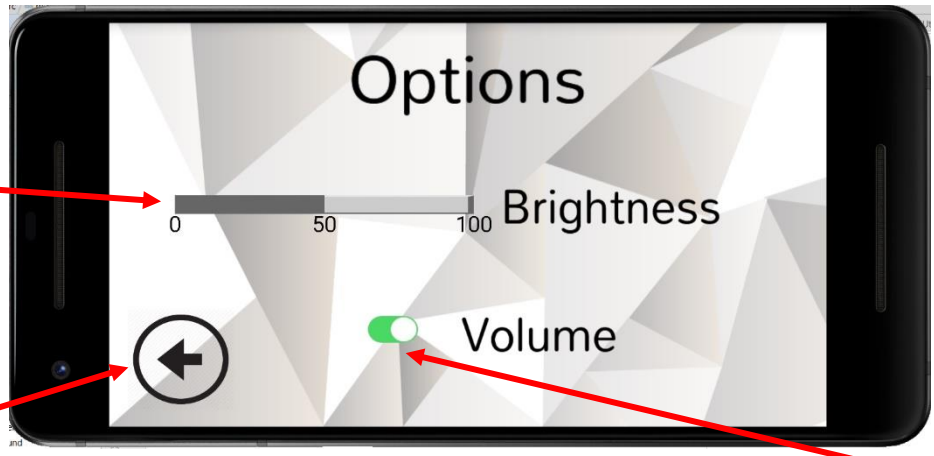
When clicked enters user to the options screen which allows the user to alter the games settings

Overworld Feature:
Currently in development but will allow the user to walk a sprite around an “overworld” map and enter in to different games.

When clicked enters user in to a step by step tutorial of the game

## Options Screen

The options screen carries design consistency from the menu screen. The user has the option to adjust the brightness via a slider or turn on and off the volume via a toggle switch. The interface design also consists of clear readable text.




Brightness slider to alter the brightness of the game screen

When clicked returns the user back to the options screen.

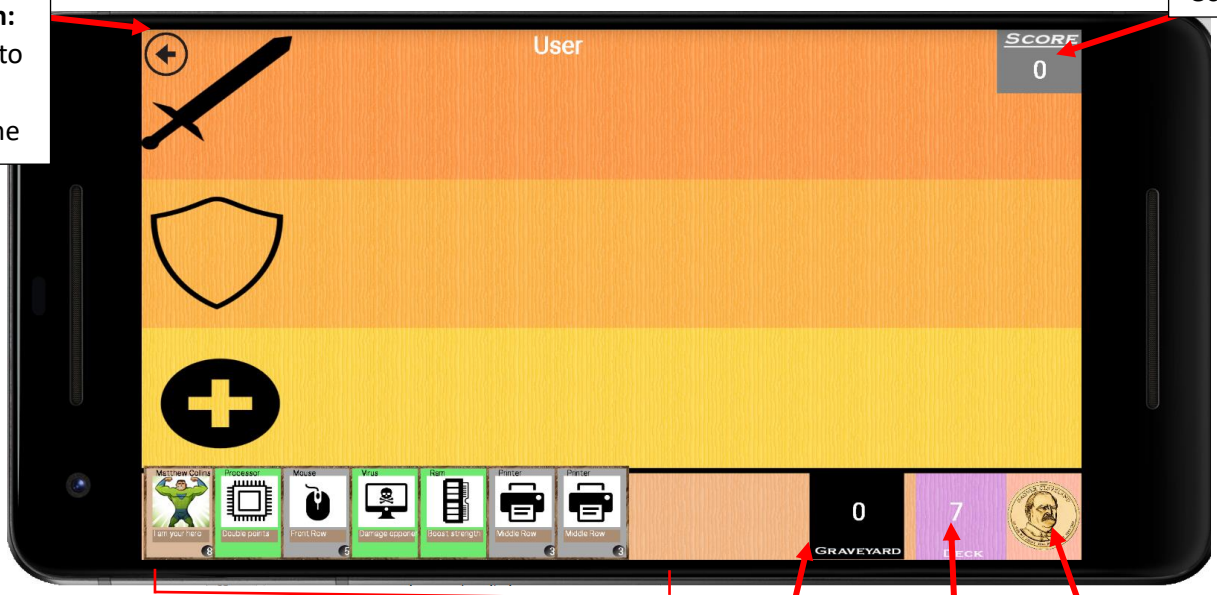
Volume toggle switch.

## User Game Play Screen

This screen is the screen the game is actually played on. The user can choose to pass goes, place cards all on this screen. Multiple goes and iterations can take place here depending on the AI's response to the users go.

**Back Button:** Offers user to quit the current game

**Users Score**




**Users Hand:**
This shows the users hand and available cards the user has to play. User has options of Minion, Spell and Hero cards to play.

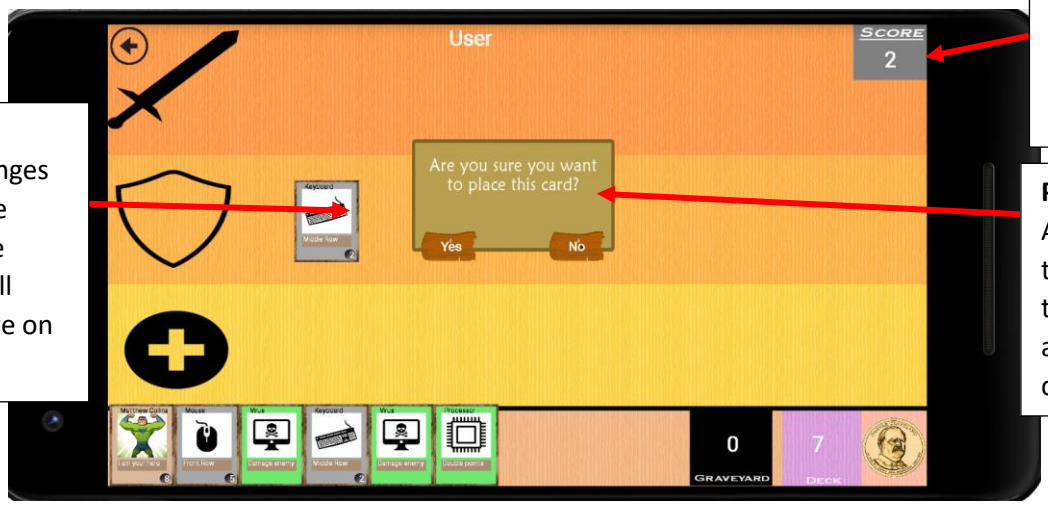
Number of cards left in the graveyard are shown here

Number of cards left in the deck are shown here

**Pass coin**:
User clicks here to confirm to pass their go.

## User Place Card

This screen represents what the user will see once they place their card. The score will be updated and the user will be asked if they want to confirm their selection and placement of card.



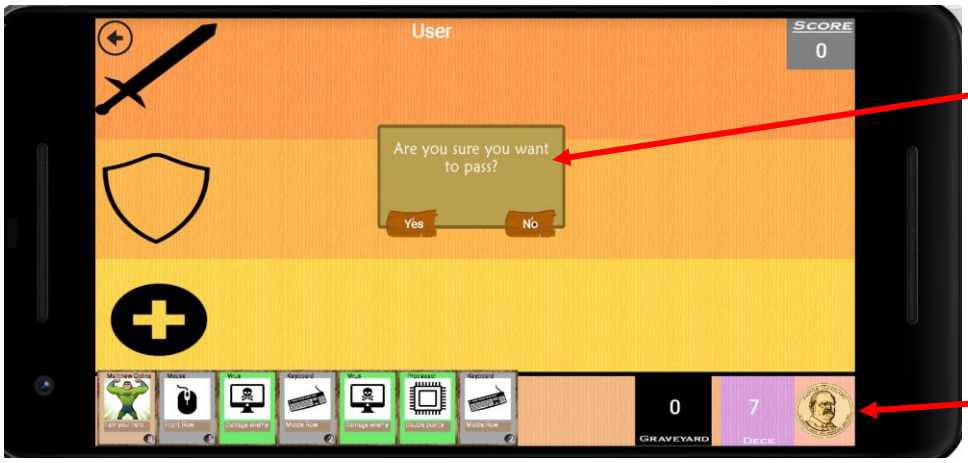

**Users Score:**
Score will change accordingly to the strength number of card

**Placed Card:**
The card changes position to be placed on the board and will become active on the board.

**Prompt:**
A prompt appears with two buttons which allows the user to either choose another card to play or to confirm their selection.

## User Passes Turn

This screen shows what happens when the user chooses to click the pass coin to pass their go. This will mean the AI can choose to either play or pass if the user confirms their pass.

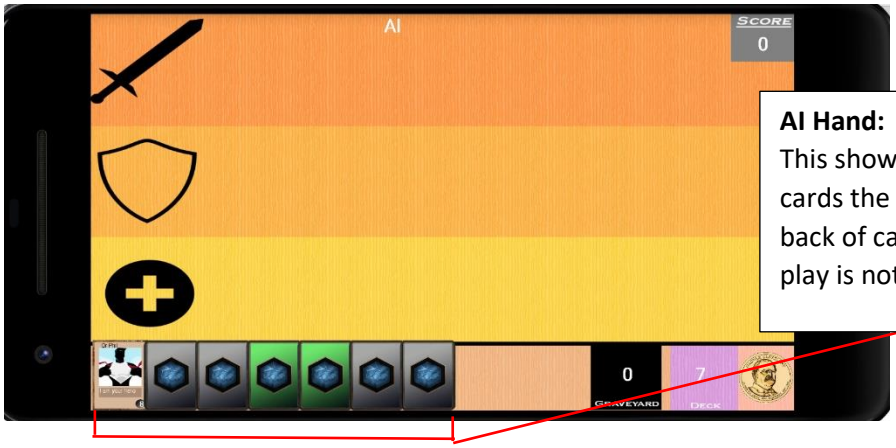



**Prompt:**
A prompt appears with two buttons which allows the user to either confirm their pass or decline and choose and alternative course.

**Pass coin:**
Pass coin must be clicked to allow the user to pass

## AI Game Play Screen

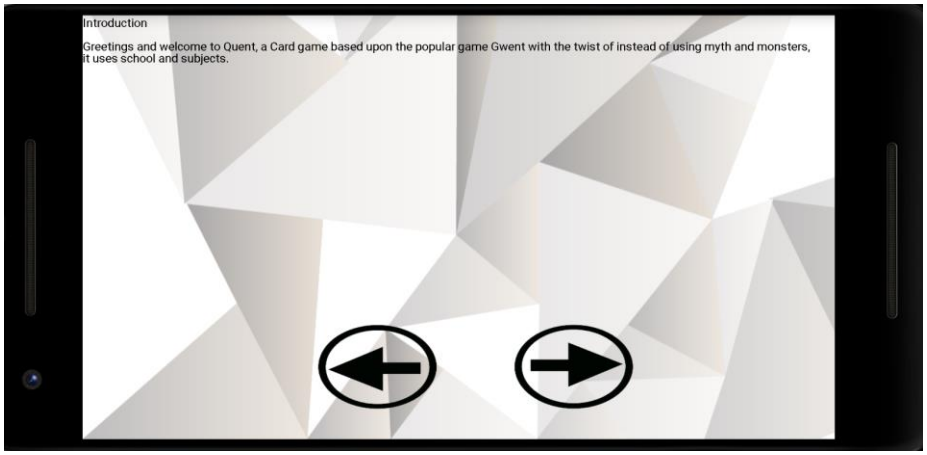
This screen is the AI screen the user will be able to see only the AI's hero card but not the AI's hand. They can only see the backs of the cards.-




**AI Hand:**
This shows the users hand and available cards the ai has to play. User only sees back of cards and hero card so turn of play is not altered.

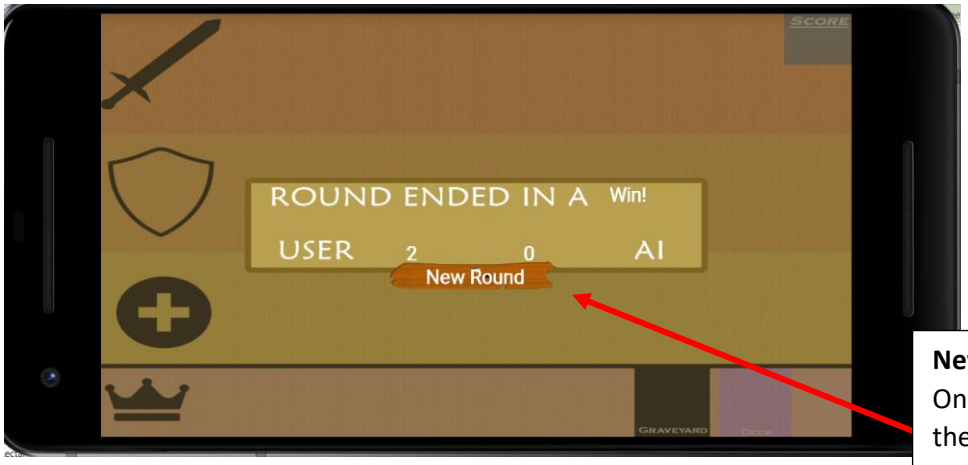
## Tutorial Screen

This screen is a tutorial the user can go through by clicking on the right and left arrows. It outlines the rules of the game how to play.




## Round ending screen

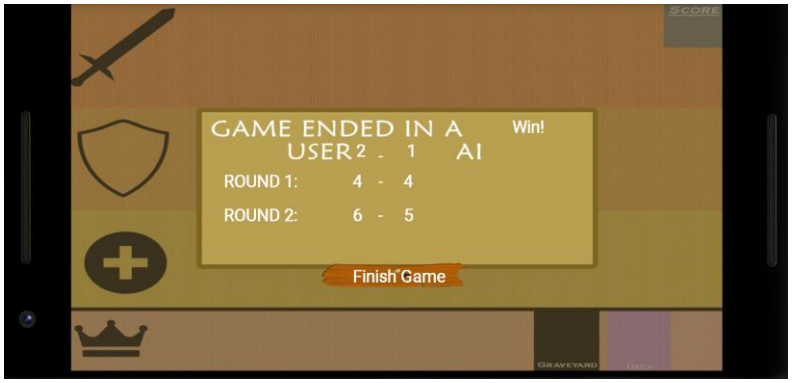
This screen appears once a round has finished displaying feedback to the user about who won the round and the score of the round. A button is also made available to the user to allow them to create a new round.



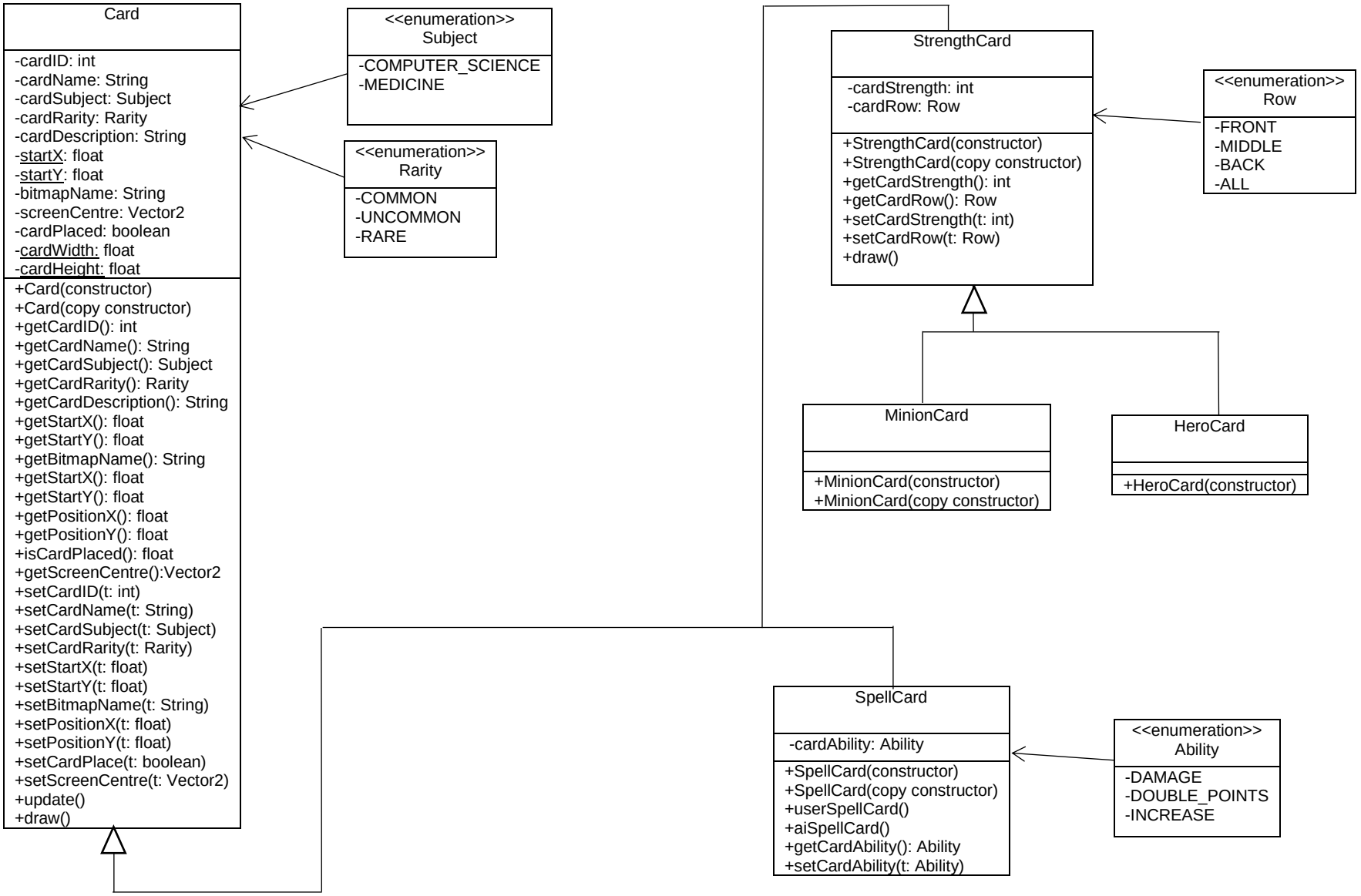

**New round button:**
Only available selection for the user so that either a new round is created or the game will be ended.

## Game Ending Screen

The game ending screen will appear when either the user or the AI has a lead to which the opponent can not come back from. As there is a max possibility of three rounds the User has won in this instance. The game ending screen summarises the scores in each round and the overall score.



# Class Relationship Models



This class relationship model represents the various classes involved in the Card superclass and its various subclasses. There are two immediate subclasses: SpellCard and the StrengthCard.

The SpellCard class is intended for cards that do not sit on the board of the game, rather they cast their effect(s) and then they are moved to the graveyard, and as such has an enum called ability attached to it for the time being although this may change to an interface.

The StrengthCard subclass on the other hand is intended for cards that stay on the board when they are played, i.e. they have a strength attribute. This class contains both the strength of the card as an integer and the row that it may be played it (dictated by the "Row" enumeration). The two subclasses of StrengthCard are MinionCard and HeroCard respectively. MinionCard is a class that represents normal minions that can be found in the deck while the HeroCard class represents heroes which exist outside of the deck and instead on their own in the hero card slot on the board.

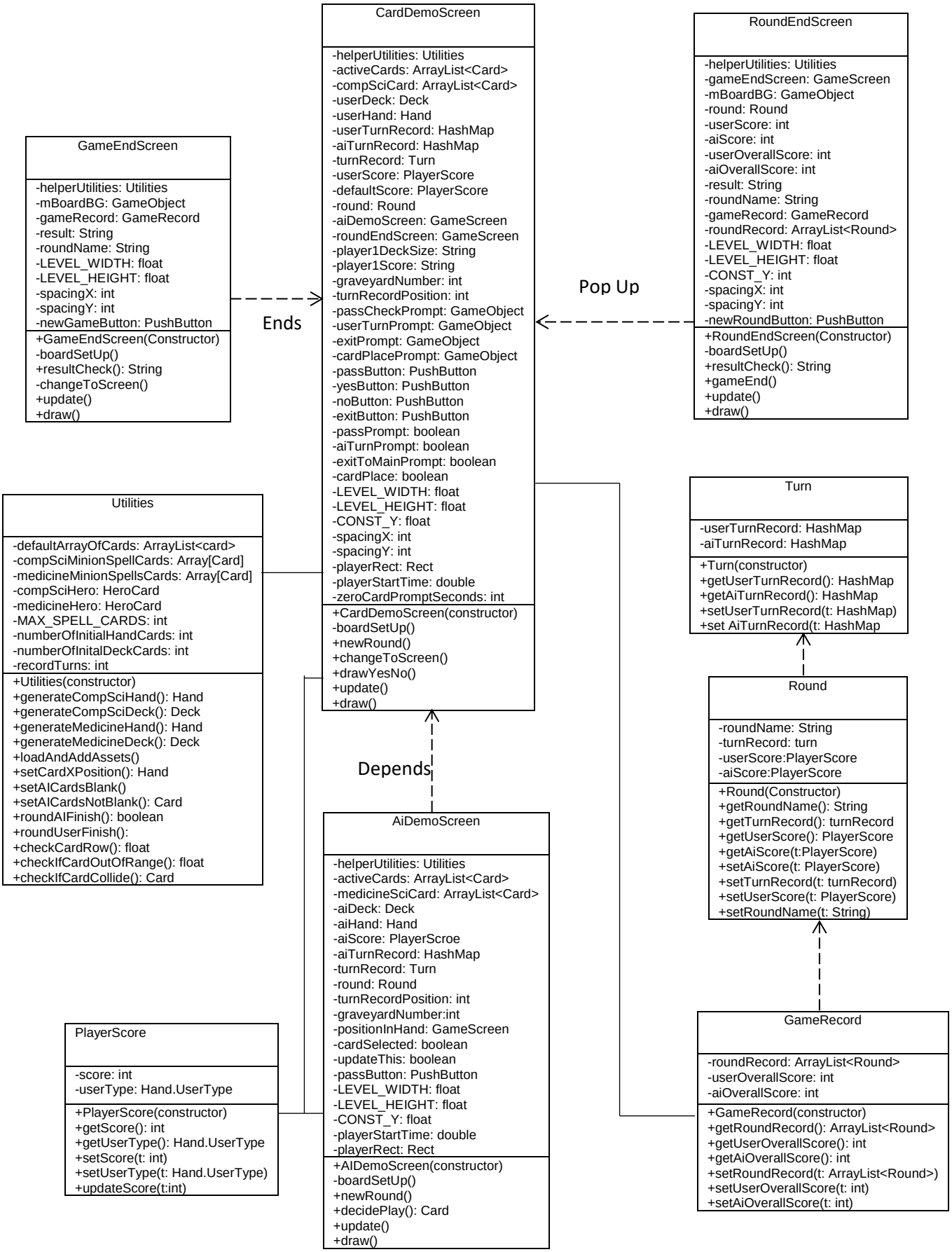
---

The next class relationship model represents the various screens for our actual game. This is a complex model showing the interactions between the game classes.

On the CardDemoScreen, the actual game is displayed to the player, with the board, their new hand and their hero being displayed to them. Much of the game's interaction and change comes from this class. Various screens and classes are entirely dependent on this screen, most prominently being the AiDemoScreen, which handles the Ai' board and behaviour. Other screens such as the RoundEndScreen and the GameEndScreen are also dependent on the CardDemoScreen, ending a round and the game respectably when their conditions are met. The PlayerScore class updates the CardDemoScreen's and AiDemoScreen's scores.

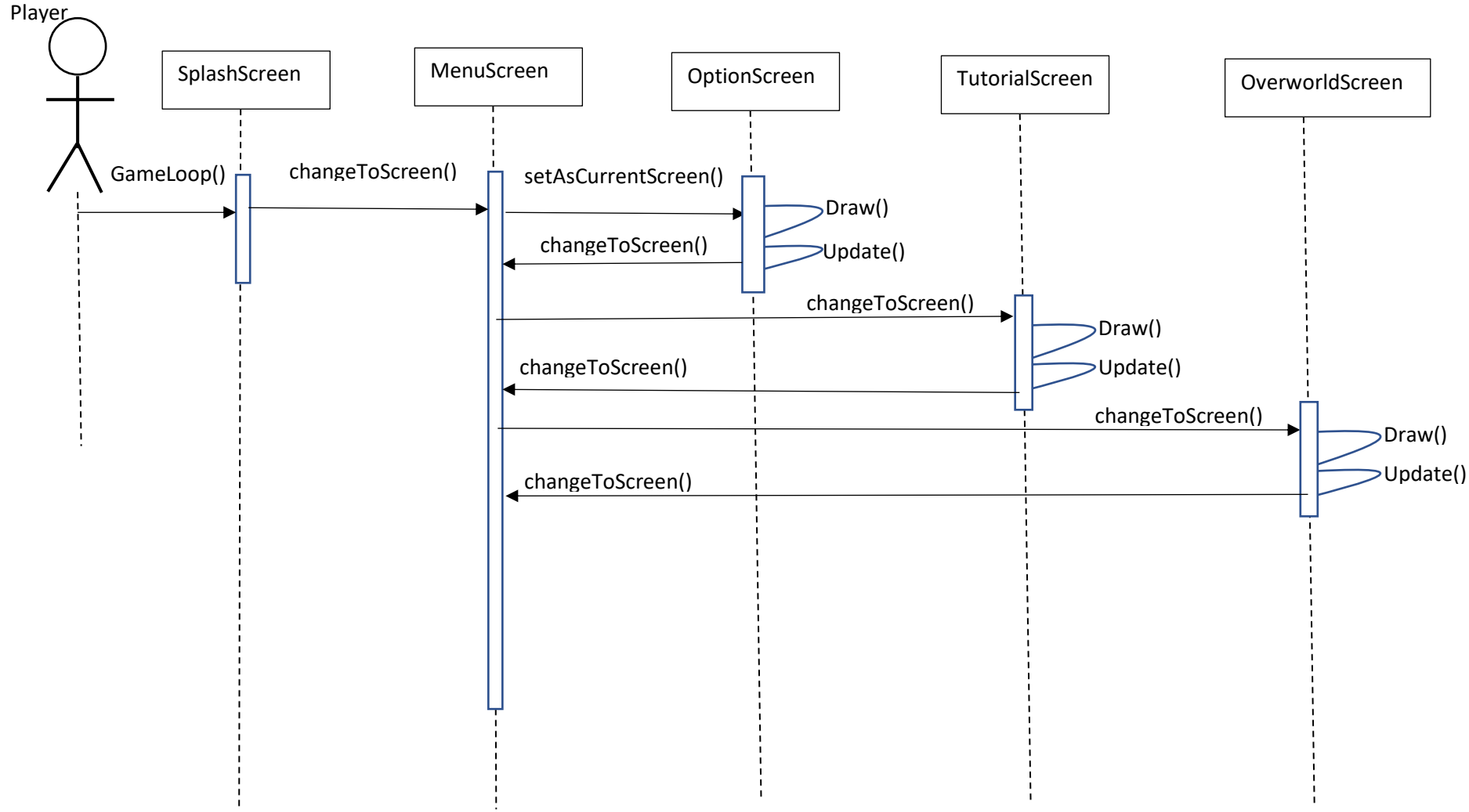
The Utilities class is heavily linked to the CardDemoScreen but is linked to most classes providing useful methods to be used.

Lastly, the GameRecord class, which is dependent on the Round class which in turn is dependent on the Turn class, handles how turns are carried out and calculates whose turn it should be and thus which board should be showing.


CardDemoScreen
-helperUtilities: Utilities
-activeCards: ArrayList<Card>
-compSciCard: ArrayList<Card>
-userDeck: Deck
-userHand: Hand
-userTurnRecord: HashMap
-aiTurnRecord: HashMap
-turnRecord: Turn
-userScore: PlayerScore
-defaultScore: PlayerScore
-round: Round
-aiDemoScreen: GameScreen
-roundEndScreen: GameScreen
-player1DeckSize: String
-player1Score: String
-graveyardNumber: int
-turnRecordPosition: int
-passCheckPrompt: GameObject
-userTurnPrompt: GameObject
-exitPrompt: GameObject
-cardPlacePrompt: GameObject
-passButton: PushButton
-yesButton: PushButton
-noButton: PushButton
-exitButton: PushButton
-passPrompt: boolean
-aiTurnPrompt: boolean
-exitToMainPrompt: boolean
-cardPlace: boolean
-LEVEL_WIDTH: float
-LEVEL_HEIGHT: float
-CONST_Y: float
-spacingX: int
-spacingY: int
-playerRect: Rect
-playerStartTime: double
-zeroCardPromptSeconds: int
+CardDemoScreen(constructor)
-boardSetUp()
+newRound()
+changeToScreen()
+drawYesNo()
+update()
+draw()
GameEndScreen
-helperUtilities: Utilities
-mBoardBG: GameObject
-gameRecord: GameRecord
-result: String
-roundName: String
-LEVEL_WIDTH: float
-LEVEL_HEIGHT: float
-spacingX: int
-spacingY: int
-newGameButton: PushButton
+GameEndScreen(Constructor)
-boardSetUp()
+resultCheck(): String
-changeToScreen()
+update()
+draw()
Ends
RoundEndScreen
-helperUtilities: Utilities
-gameEndScreen: GameScreen
-mBoardBG: GameObject
-round: Round
-userScore: int
-aiScore: int
-userOverallScore: int
-aiOverallScore: int
-result: String
-roundName: String
-gameRecord: GameRecord
-roundRecord: ArrayList<Round>
-LEVEL_WIDTH: float
-LEVEL_HEIGHT: float
-CONST_Y: int
-spacingX: int
-spacingY: int
-newRoundButton: PushButton
+RoundEndScreen(Constructor)
-boardSetUp()
+resultCheck(): String
+gameEnd()
+update()
+draw()
Pop Up
Utilities
-defaultArrayOfCards: ArrayList<card>
-compSciMinionSpellCards: Array[Card]
-medicineMinionSpellsCards: Array[Card]
-compSciHero: HeroCard
-medicineHero: HeroCard
-MAX_SPELL_CARDS: int
-numberOfInitialHandCards: int
-numberOfInitalDeckCards: int
-recordTurns: int
+Utilities(constructor)
+generateCompSciHand(): Hand
+generateCompSciDeck(): Deck
+generateMedicineHand(): Hand
+generateMedicineDeck(): Deck
+loadAndAddAssets()
+setCardXPosition(): Hand
+setAICardsBlank()
+setAICardsNotBlank(): Card
+roundAIFinish(): boolean
+roundUserFinish():
+checkCardRow(): float
+checkIfCardOutOfRange(): float
+checkIfCardCollide(): Card
Turn
-userTurnRecord: HashMap
-aiTurnRecord: HashMap
+Turn(constructor)
+getUserTurnRecord(): HashMap
+getAiTurnRecord(): HashMap
+setUserTurnRecord(t: HashMap)
+set AiTurnRecord(t: HashMap
Round
-roundName: String
-turnRecord: turn
-userScore:PlayerScore
-aiScore:PlayerScore
+Round(Constructor)
+getRoundName(): String
+getTurnRecord(): turnRecord
+getUserScore(): PlayerScore
+getAiScore(t:PlayerScore)
+setAiScore(t: PlayerScore)
+setTurnRecord(t: turnRecord)
+setUserScore(t: PlayerScore)
+setRoundName(t: String)
Depends
AiDemoScreen
-helperUtilities: Utilities
-activeCards: ArrayList<Card>
-medicineSciCard: ArrayList<Card>
-aiDeck: Deck
-aiHand: Hand
-aiScore: PlayerScroe
-aiTurnRecord: HashMap
-turnRecord: Turn
-round: Round
-turnRecordPosition: int
-graveyardNumber:int
-positionInHand: GameScreen
-cardSelected: boolean
-updateThis: boolean
-passButton: PushButton
-LEVEL_WIDTH: float
-LEVEL_HEIGHT: float
-CONST_Y: float
-playerStartTime: double
-playerRect: Rect
+AIDemoScreen(constructor)
-boardSetUp()
+newRound()
+decidePlay(): Card
+update()
+draw()
PlayerScore
-score: int
-userType: Hand.UserType
+PlayerScore(constructor)
+getScore(): int
+getUserType(): Hand.UserType
+setScore(t: int)
+setUserType(t: Hand.UserType)
+updateScore(t:int)
GameRecord
-roundRecord: ArrayList<Round>
-userOverallScore: int
-aiOverallScore: int
+GameRecord(constructor)
+getRoundRecord(): ArrayList<Round>
+getUserOverallScore(): int
+getAiOverallScore(): int
+setRoundRecord(t: ArrayList<Round>)
+setUserOverallScore(t: int)
+setAiOverallScore(t: int)

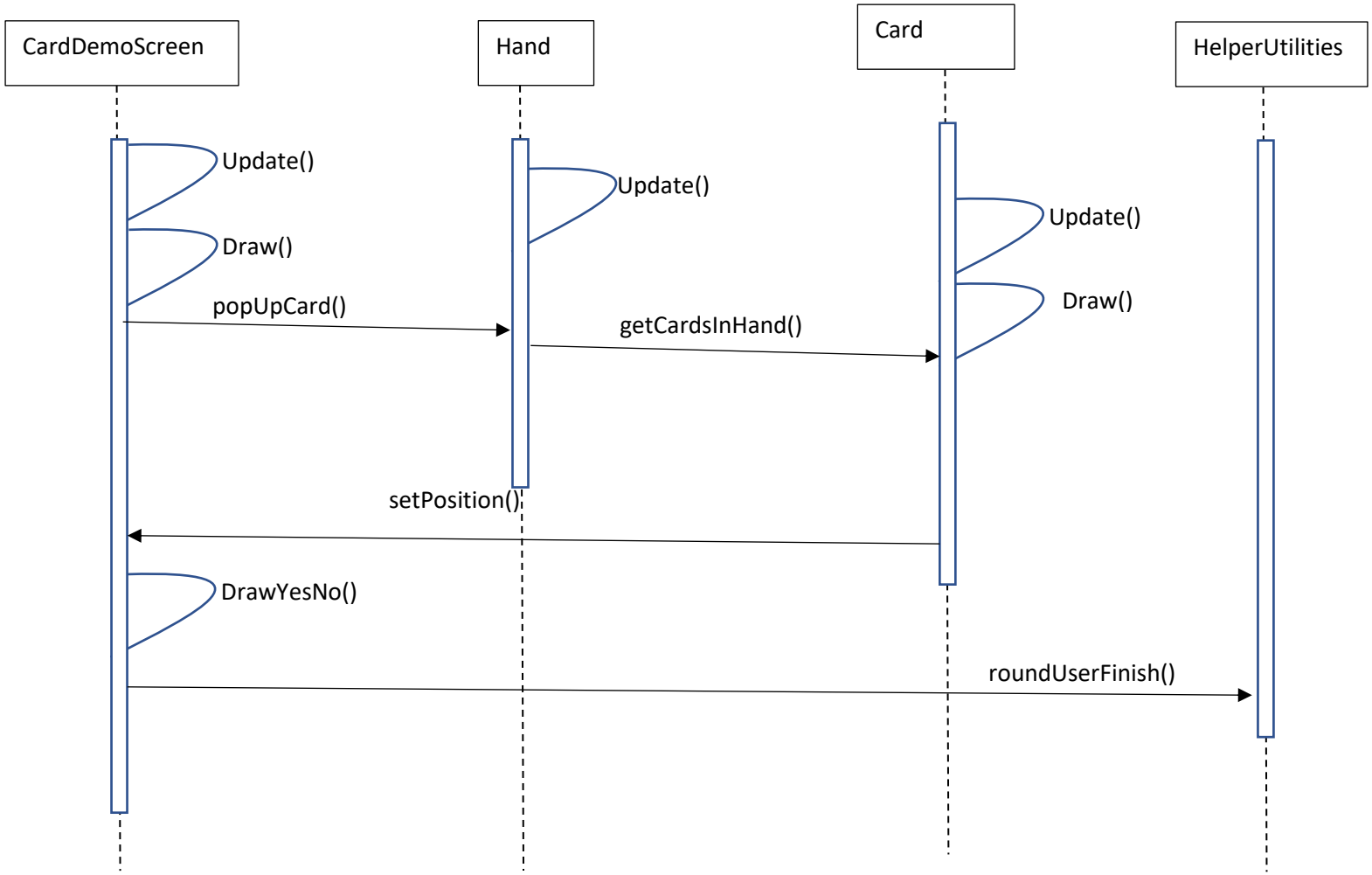
# Use Case Realisations

## Screen Select



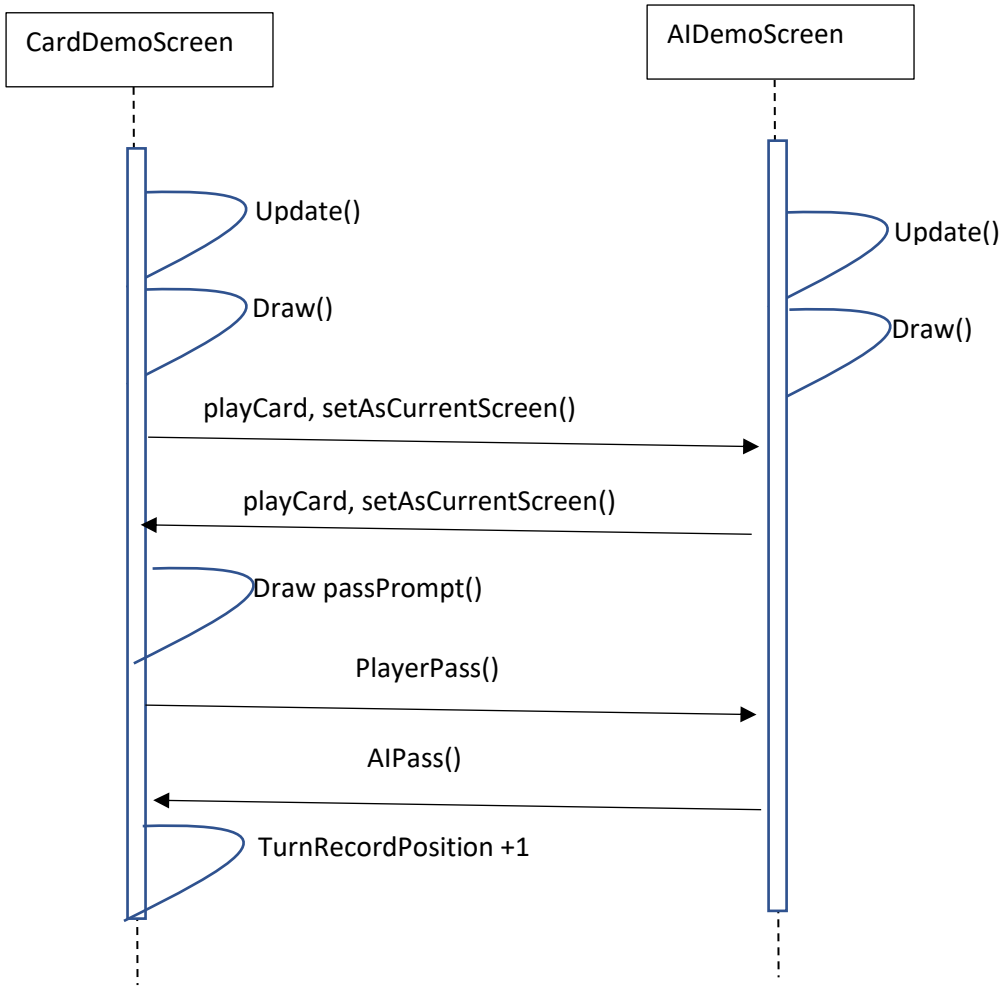
## Playing a card

A player clicks on a card, then clicks where they want to place the card, removing the card from the hand and ending the turn if they select Yes when the Yes No screen is drawn by the DrawYesNo section, which is handled by the roundUserFinish method in the helper utilites.



## Game Progression

When a card is placed or if the player presses the pass button their turn ends, when both sides pass a round ends, above diagram abbreviated to playCard in below realisation



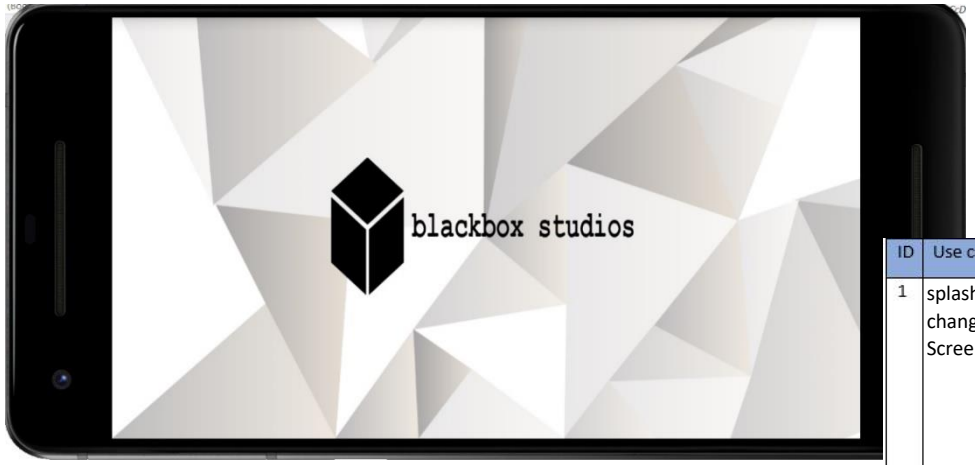
# Test Plan

## User Interface Tests

## Splash Screen

This test plan is used to test if the navigation to the menu screen from the splash screen is correct. This is black box acceptance/unit testing. Unit tests for this class have also been written, as can be seen within the SplashScreenTest class in our project.

A screenshot of the splash screen is shown below, followed by the test plan for this screen.

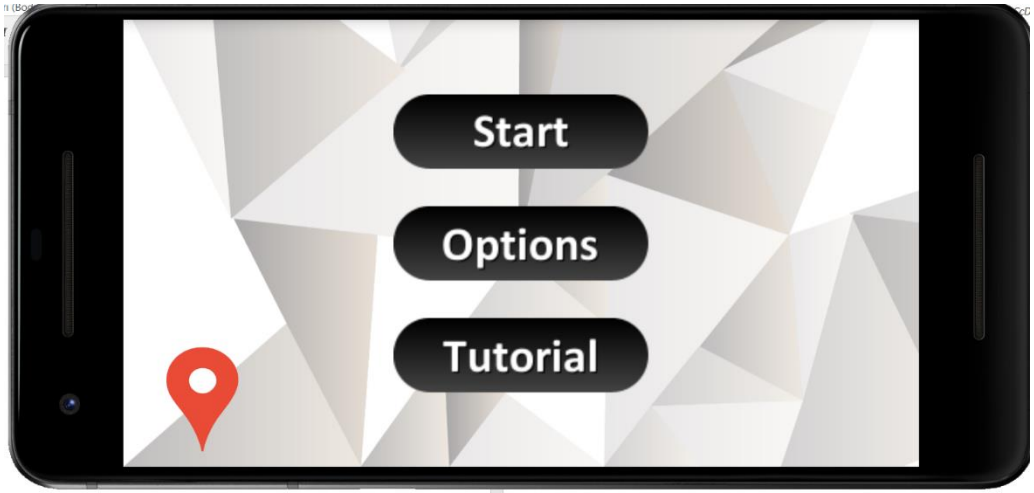


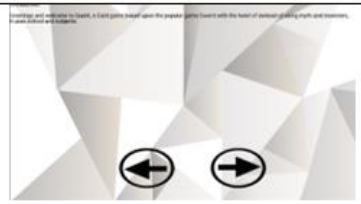

| ID | Use case ref | Description of Test | Execution Steps | Expected Results | Evidence | Passed? |
|---|---|---|---|---|---|---|
| 1 | splashScreen changeTo Screen() | Testing navigation to the menu screen | Click somewhere random on the screen | You should be directed to the menu screen |   | ✓ |
| 2 | splashScreen changeTo Screen() | Testing navigation to the menu screen | Repeat test 1 but click on a different location | You should be directed to the menu screen |   | ✓ |

## Menu Screen

This test plan is used to test if the navigation through the menu screen is correct. This is black box acceptance/unit testing. Unit tests for this class have also been written, as can be seen within the MenuScreenTest class in our project.

A screenshot of the menu screen is shown below, followed by the test plan for this screen.

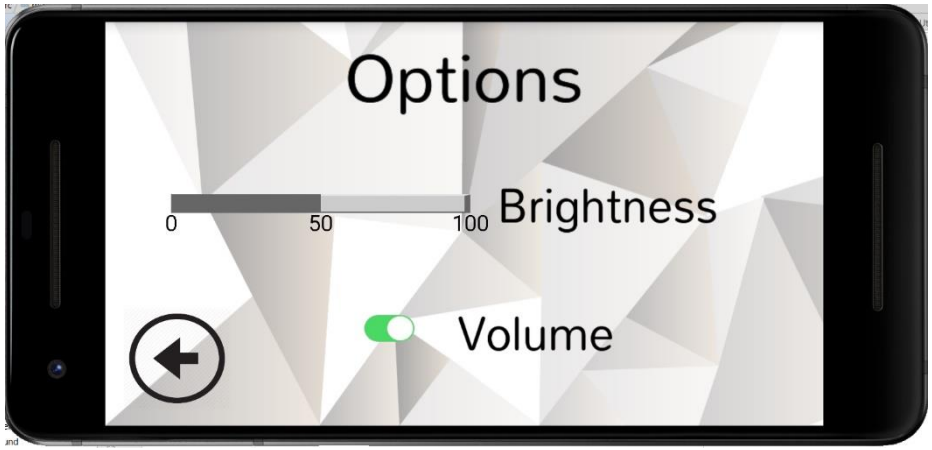


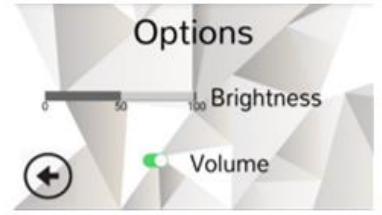
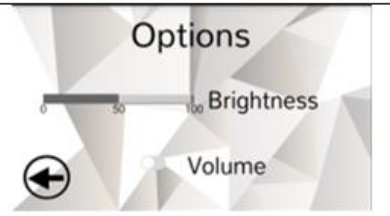

| ID | Use case ref | Description of Test | Execution Steps | Expected Results | Evidence | Passed? |
|---|---|---|---|---|---|---|
| 1 | MenuScreen changeToScreen() | Testing navigation to begin the game | Click the start button | You should be taken to the game screen |  | ✓ |
| 2 | MenuScreen changeToScreen() | Testing navigation to edit the game's options | Click the options button | You should be taken to the options screen |   | ✓ |
| 3 | MenuScreen changeToScreen() | Testing navigation to view the game's tutorial | Click the tutorial button | You should be taken to the tutorial screen |  | ✓ |
| 4 | MenuScreen changeToScreen() | Testing navigation does not occur anywhere you click | Click somewhere random on the screen | You should remain on the menu screen |   | ✓ |

## Options Screen Tests

This test plan is used to test if the navigation and features in the options screen function correctly. This is black box testing. Unit tests for this class have also been written, as can be seen within the OptionsScreenTest and SliderTest classes in our project.

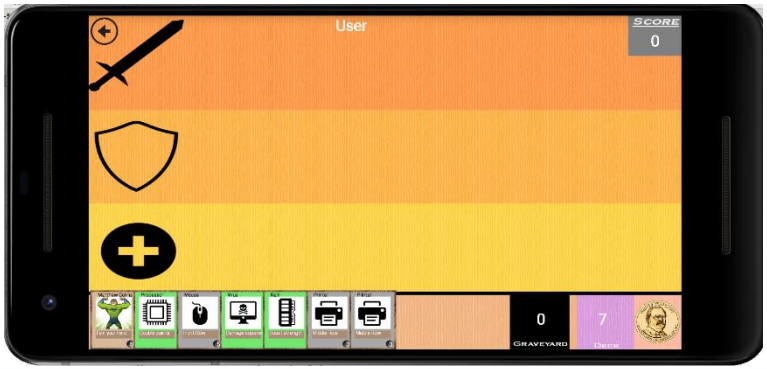
A screenshot of the options screen is shown below, followed by the test plan for this screen.

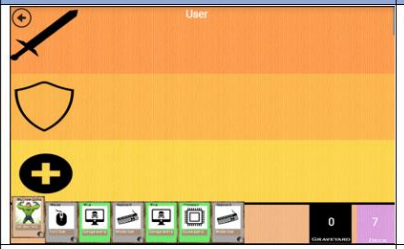
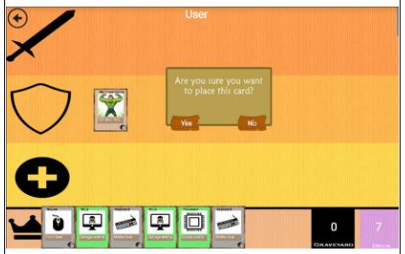
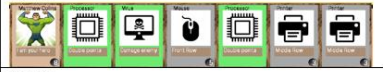
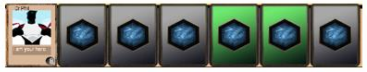
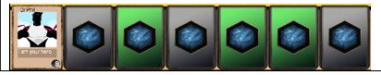


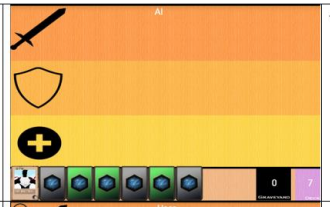
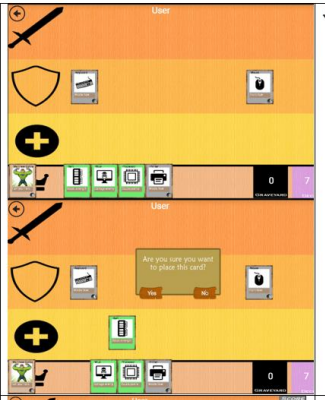

| ID | Use case ref | Description of Test | Execution Steps | Expected Results | Evidence | Passed? |
|---|---|---|---|---|---|---|
| 1 | OptionsScreen changeToScreen() | Test navigation occurs to go back to the main menu | Click the back button | You should be taken to the main menu |   | ✓ |
| 2 | OptionsScreen | Test brightness slider- when scrolled up and down the brightness of the screen should increase and decrease | Drag the slider up and down | Not yet implemented- expected to fail |   | X |
| 3 | OptionsScreen | Test volume toggle button- when turned/left on the game music should be on | Turn/leave the slider on | Not yet implemented- expected to fail |   | X |
| 4 | OptionsScreen | Test volume toggle button- when turned off the game music should be off | Turn the slider off | Not yet implemented- expected to fail |   | X |

## Card Demo Screen Tests

Tests based the users actions through the game screen. This test plan is used to test all the functionality the user using the Card Demo Screen. It a unit test solely testing the core functionality of this screen. Example of game screen below. I will take some extracts from my test plan and will evidence this in the adherence to process through my android tests.

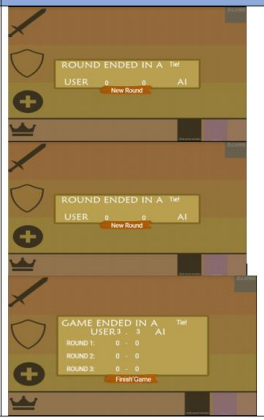


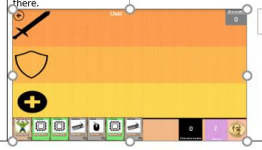

| CARD TESTS | | | | | | |
|---|---|---|---|---|---|---|
| ID | Use case ref | Description of Test | Execution Steps | Expected Results | Evidence | Passed ? |
| 1 | Select Card | Test card selection | Click on one of the available cards in Users hand | Card should 'pop' up to indicate selection has been made. |  | ✓ |
| 2 | Drag Card | Test card placement | Click on the board to place card | Card should move place on the board with a prompt asking the user if they wish to confirm their placement and the score should adjust to the cards strength number. |  | ✓ |
| 10. | Draw Cards | Test random user hand generation. | Run the game three different times and test to see if you get a different hand every time. The users hand is randomly generated every game to ensure both user and AI have a fair chance. | The users hand should have a different hand every time consisting of different spell cards and minion cards. The hero card should always remain the same and any hand should not have more than 3 spell cards. | Run 1:<br>  <br>Run 2:<br>  <br>Run 3:<br>  | ✓ |
| 11. | Draw Cards | Test random AI hand generation | Again similar to the test before. Run the game 3 times and test the AI has been given a randomly generated hand | The AI hand should again have 7 cards and a max of 3 spell cards and 1 consistent hero. The rest will be minion cards | Run 1:<br>  <br>Run 2:<br>  <br>Run 3:<br>  | ✓ |

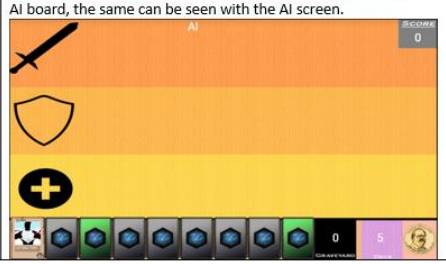
| | | | | | | |
|---|---|---|---|---|---|---|
| 4. | Place Card | Confirm selection | Select yes to placing the card on the board | Card should become and active card on the board of play and stick to its position, the game should then switch to the opponents view. |  | ✓ |
| 5. | Spell Animation | Play a double points spell card | Select and place a double points spell card on a board with a strength card already previously placed | The users score should double from 3 to 6. |  | ✓ |
| 7 | Spell Card | Play a “boost” strength spell card | Select and play the “boost strength” spell card on a board with two other strength cards. | The board already has two other strength cards of 2 and 5 creating a user score of 7. The “boost strength” spell card should double the strength of all spell cards therefore their strength attributes should double to 4 and 10. As well as the overall score doubling. |  | ✓ |
| 8 | Spell Card | Play “damage enemy” spell card | Select and play the “damage enemy” spell card from the user’s hand. | Once played the opponents score should minus by 3. |  | ✓ |

## Round & Game Base Tests

Tests based off how the game calculates if a round or game is won. The next set of tests are set out to test the round & game ending mechanism. I will take some extracts from my test plan and will evidence this in the adherence to process through my android tests.

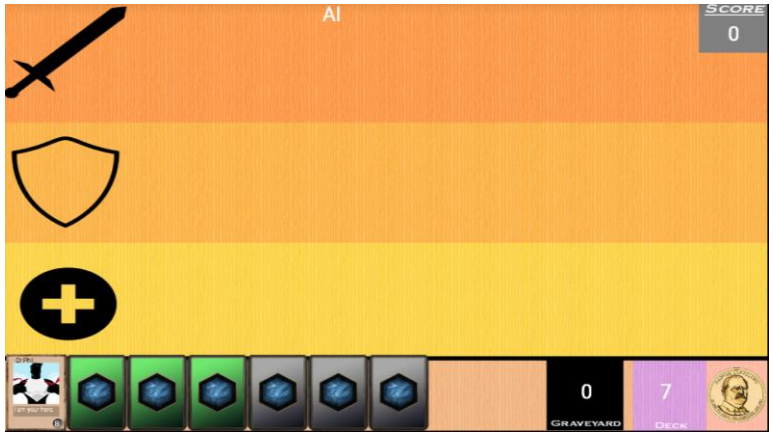
| Test Round & Game Ending/Turn Based Mechanism | | | | | | |
|---|---|---|---|---|---|---|
| ID | Use Case Ref | Test Description | Execution Steps | Expected Results | Evidence | Passed? |
| 1. | Win/ Lose Animation | User & AI pass their go every time | Ensure the AI & the User Pass every time | 3 rounds should pass with 2 round end screens shown before a final gam e ending screen displaying the result as a tie. |   | ✓ |

| | | | | | | |
|---|---|---|---|---|---|---|
| 5. | | Test "finish game" button | Continue to the end of the game until you reach the final screen. Press finish game. | User should be brought back to the original home screen and be able to start a new game with all the progress of the old game discarded. | Press finish game here   User gets brought to menu screen   Press start game to test if all progress of the old game is no longer there.  | ✓ |

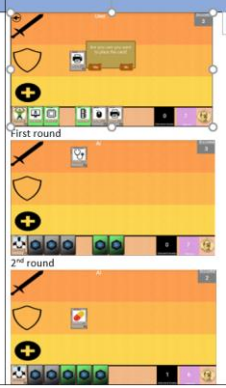
| | | | | | | |
|---|---|---|---|---|---|---|
| 2. | | Test new round is working correctly | Play 1 card for the user and then continue to pass for both ai and user until the round has finished and a new round is created. | Ensure User plays 1 card in the first round to test if on the new round both user & AI scores are set to 0 and each player gets a card drawn from their deck. Deck value should go from 7 to 6. |   User board, graveyard is now 1 and deck is now 6.  AI board, deck is now 6 and graveyard is 0 as no cards were placed  | ✓ |
| 3. | | Test round 3 draws cards from deck | Similarly to the test before, continue to pass until you reach round 3. | Both AI & User's deck should be 5 as they have had a card drawn every new round. Graveyards should be 0 as we have only passed and both scores should be 0. | Users board has 5 left in the deck and 9 cards now in their hand.  AI board, the same can be seen with the AI screen.  | ✓ |

## AI Tests

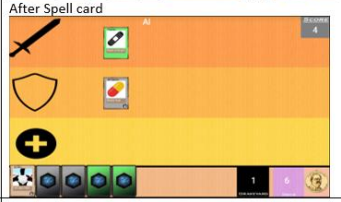
Tests based the AI's actions through the game screen. This test plan is used to test all the functionality the user using the AI Demo Screen. It a unit test solely testing the core functionality of this screen. Example of game screen below. I will take some extracts from my test plan and will evidence this in the adherence to process through my android tests.

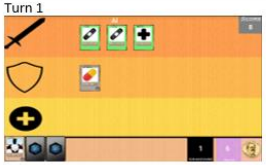
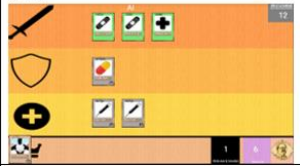


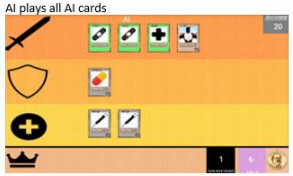
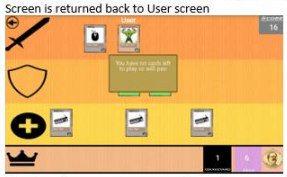
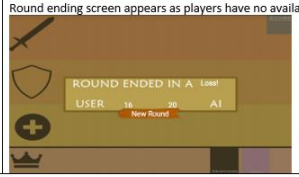

**AI Demo Screen Tests**

| ID | Use Case Ref | Test Description | Execution Steps | Expected Results | Evidence | Passed? |
|---|---|---|---|---|---|---|
| 1. | | AI Plays Minion Card first | Play a card and then the AI should play a minion card. I will also test this for a new round | The AI should always play a minion card unless it has no other cards left to play. I |  | ✓ |



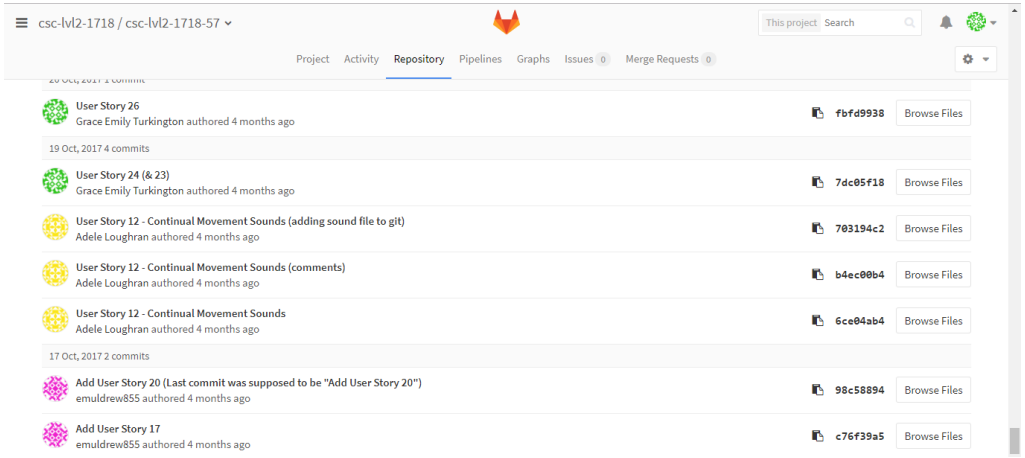
| | | | | | | |
|---|---|---|---|---|---|---|
| 3. | | Test to see if a boost strength spell card manipulates the score. | The score should manipulate based off the spell cards ability/ | The strength card of all placed strength cards should double and the score should change. | Before Spell Card  After Spell card  | ✓ |

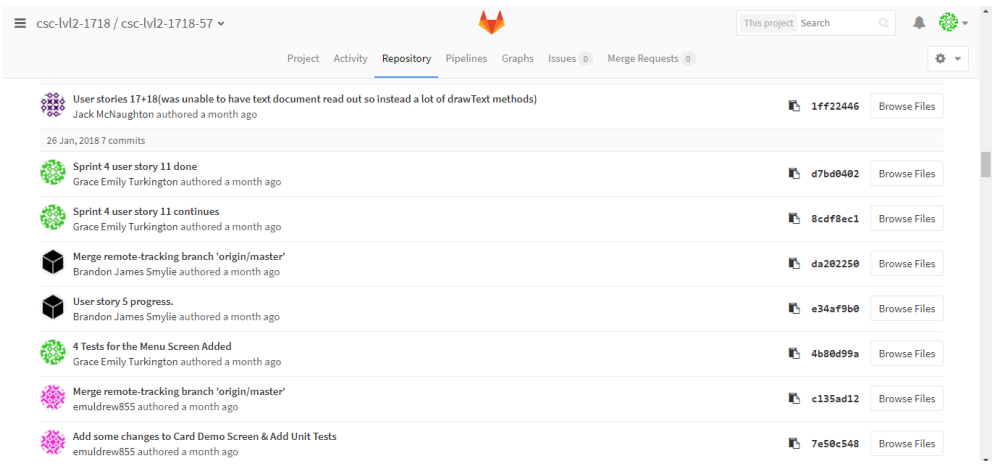
| | | | | | | |
|---|---|---|---|---|---|---|
| 10. | | Test to see if any cards collide | Simulate a few turns to see if any of the placed AI cards collide or on top of each other. | All the placed AI cards should have space between each of the placed cards. | Turn 1<br>  <br>Turn 2<br>  <br>Turn 3<br>  | ✓ |

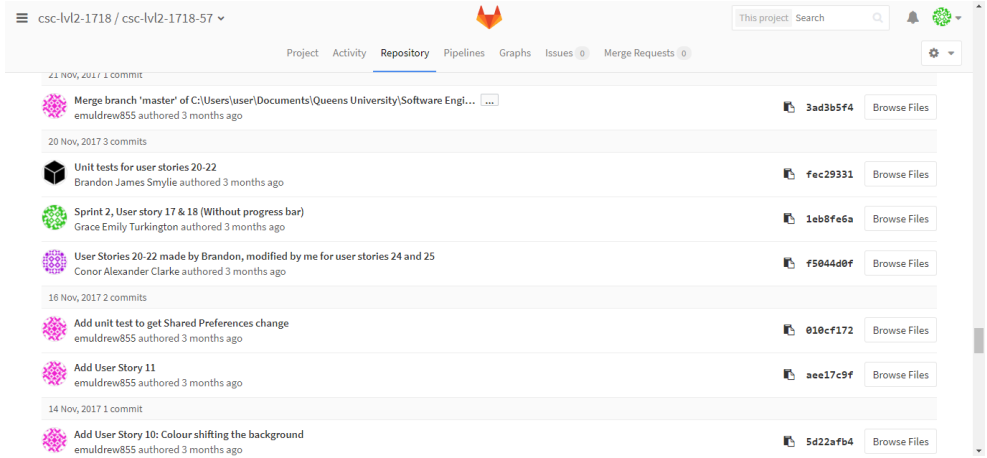
| | | | | | | |
|---|---|---|---|---|---|---|
| 15. | | Test to see what happens when the AI runs out of cards to play | Continue playing until the AI has no cards left to play | The AI should continue to play cards until they have to pass their turn which should return to the User and then eventually lead to the round ending. | AI plays all AI cards<br>  <br>Screen is returned back to User screen<br>  <br>Round ending screen appears as players have no available cards left<br>  | ✓ |

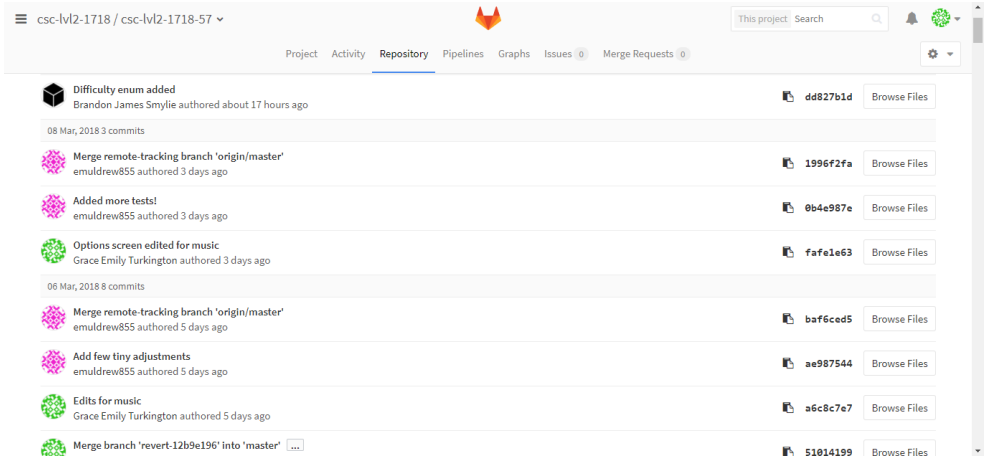
# Gitlab

As evidence of a populated Gitlab repository, I have included a couple of screenshots of our commits throughout the project. We have had more commits than shown, but this is to act as an example.

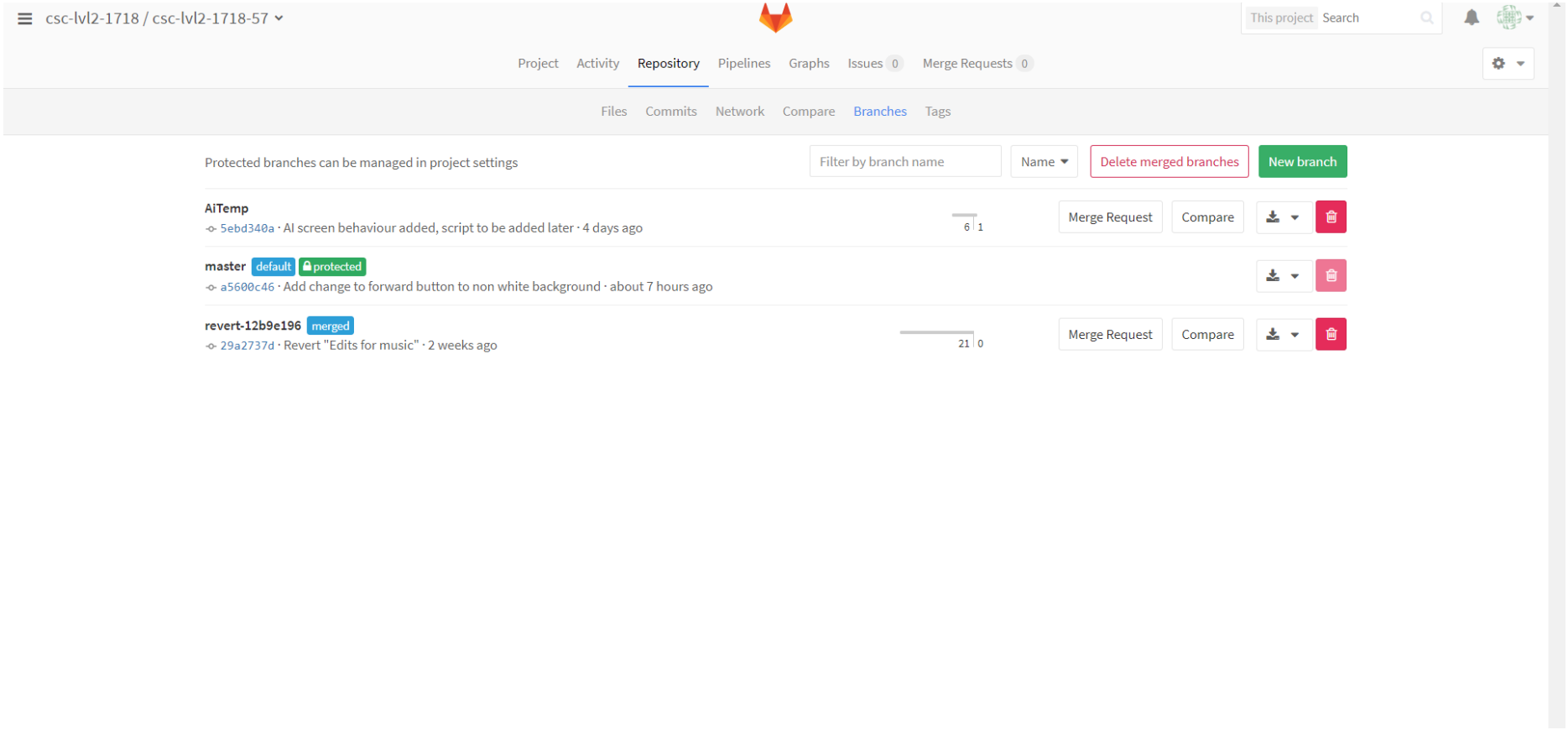


Here is Gitlab evidence of the various branches we used throughout our project.



# Appendix

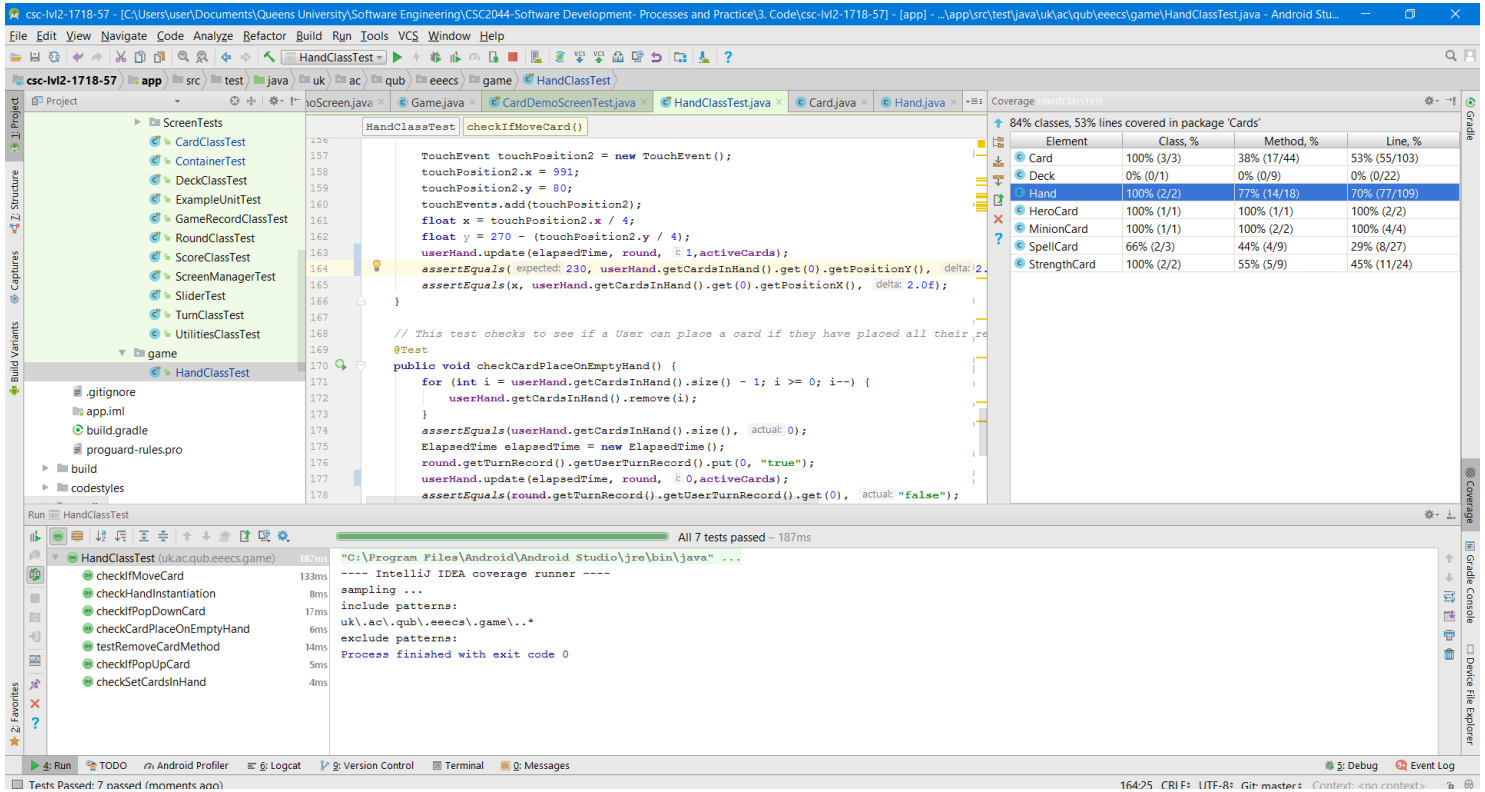
## Adherence to Process

## Proof of Testing

Here is some evidence of our testing that has gone on during our project which has been split up in to different test classes.

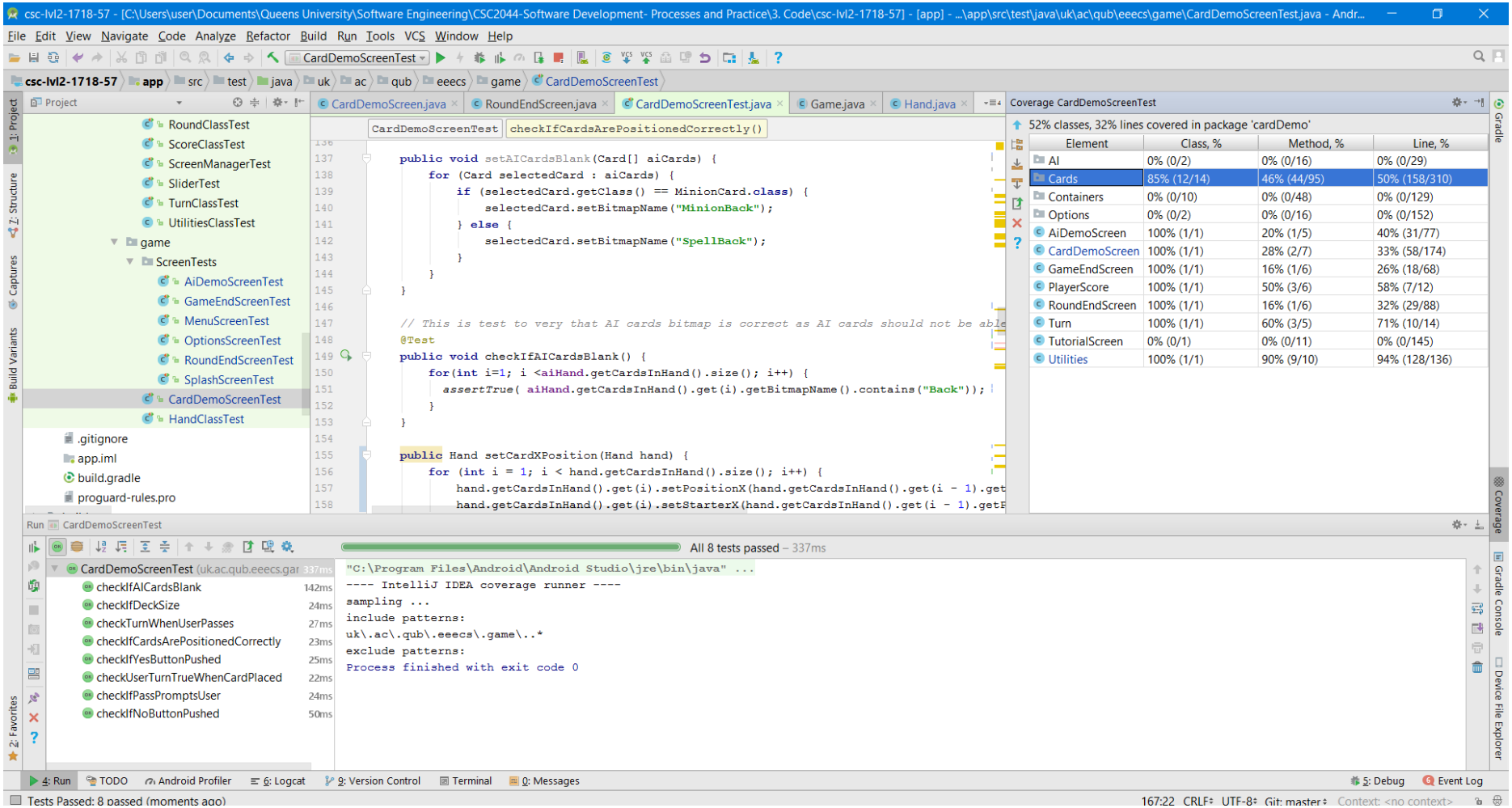
### Hand Class Test

For example here is the hand class test with 7 tests passing with code coverage. These tests test card selection, placement, de-selection, removing cards, placing card with cards available and instantiation.



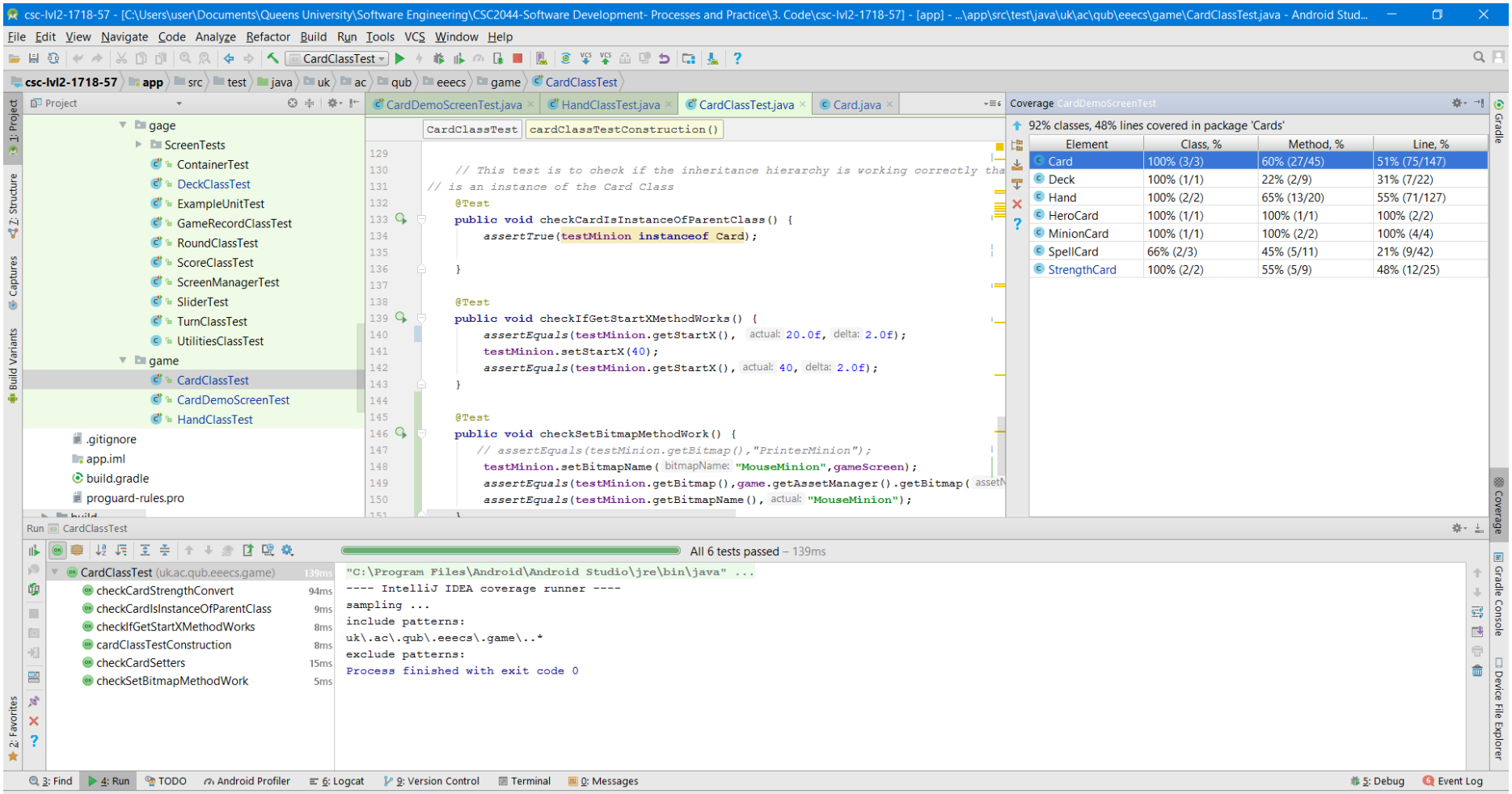
## Card Demo Screen Test

Here is another example of testing from the CardDemoScreenTest with 8 tests passing and code coverage shown. These tests check that all AI cards are blank at the start, check if cards are aligned in the user hand correctly, tests turns are managed correctly, checks if appropriate prompts appear when they should, etc.



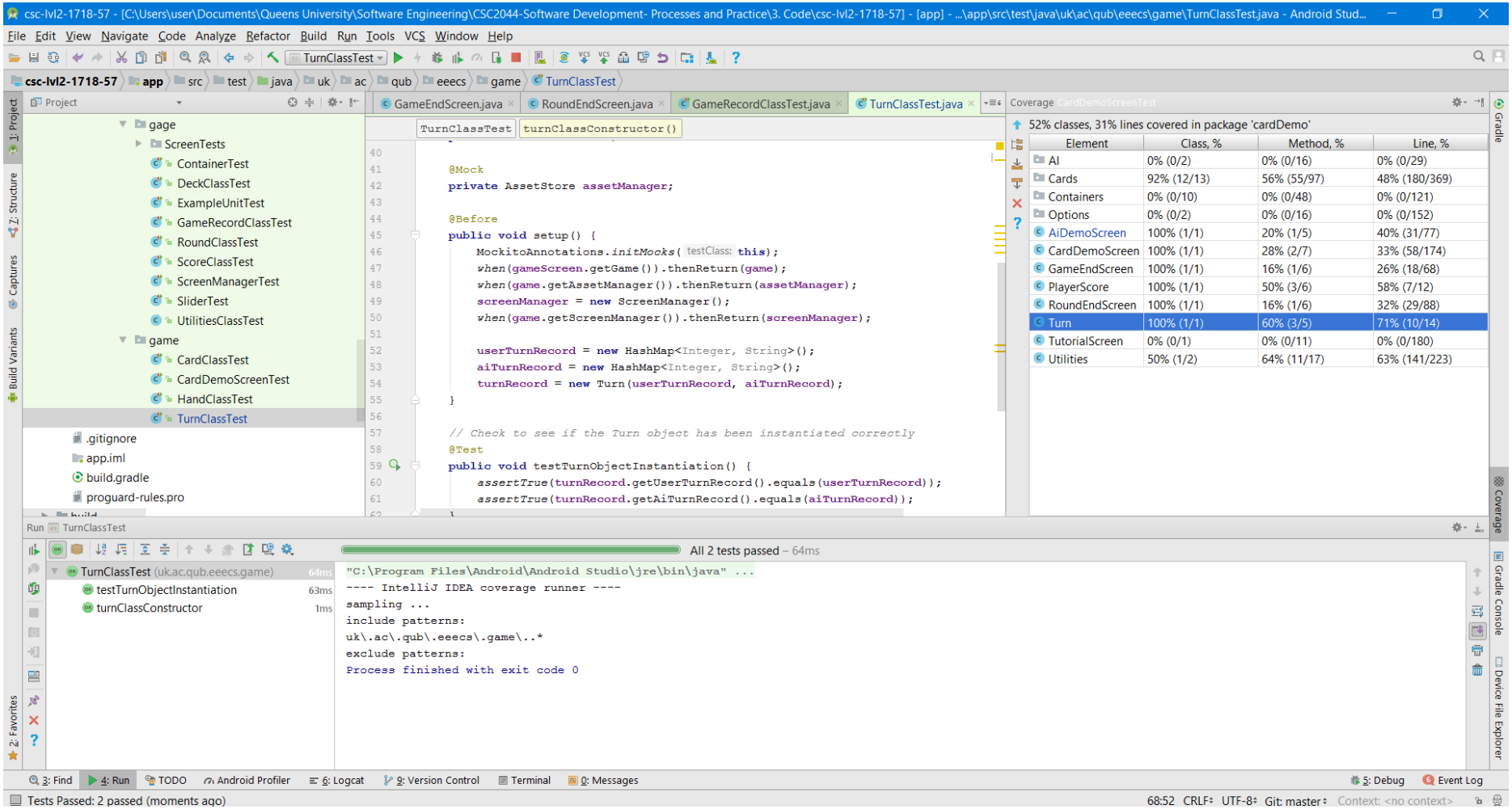
## Card Class Test

Here is another example of testing from the CardClassTest with 6 tests passing and code coverage shown. These tests test the inheritance hierarchy, all cards x position are equally aligned, the strength variable has been converted correctly, cards are constructed properly and bitmaps can be changed properly.



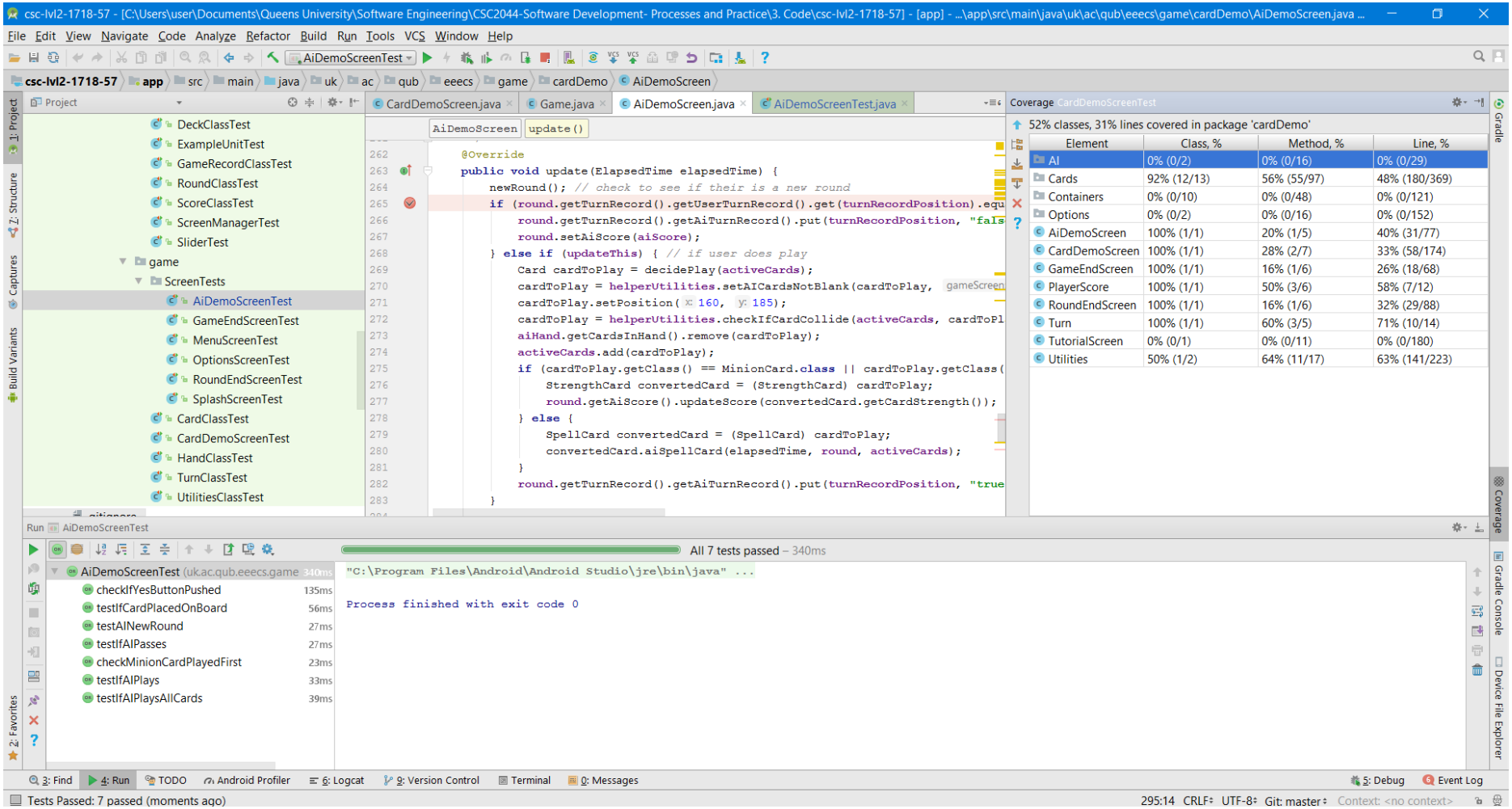
## Turn Class Tests

Here is another example of testing the turn class method with 2 tests passing and code coverage shown below. This test ensures the turn class is created and instantiated correctly.



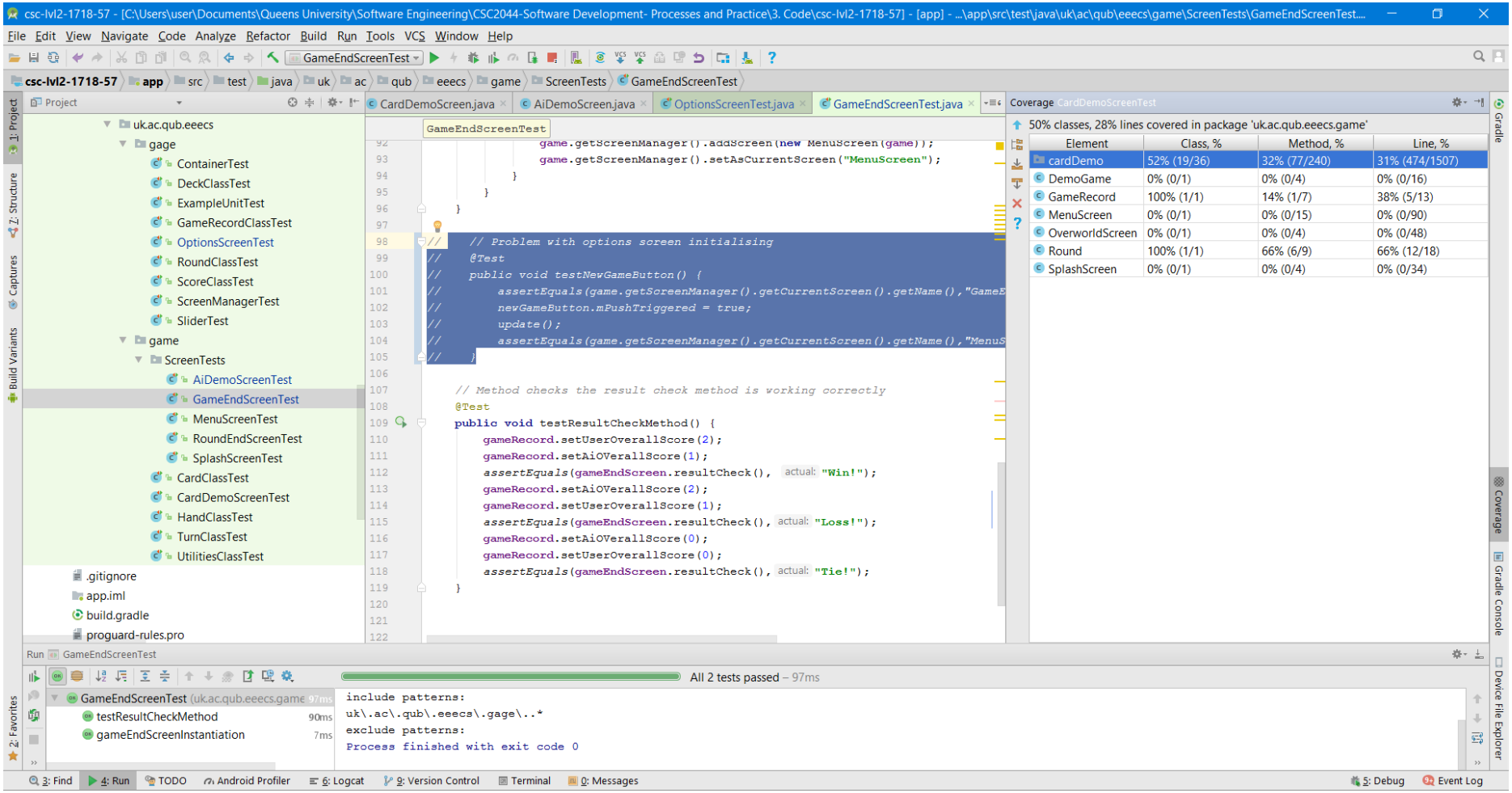
## Ai Card Demo Screen Tests

This class tests the AI and has 7 tests with code coverage. The tests test switching between appropriate screens, score change, new round method, if the user passes, if a card has successfully been placed on the board, etc.



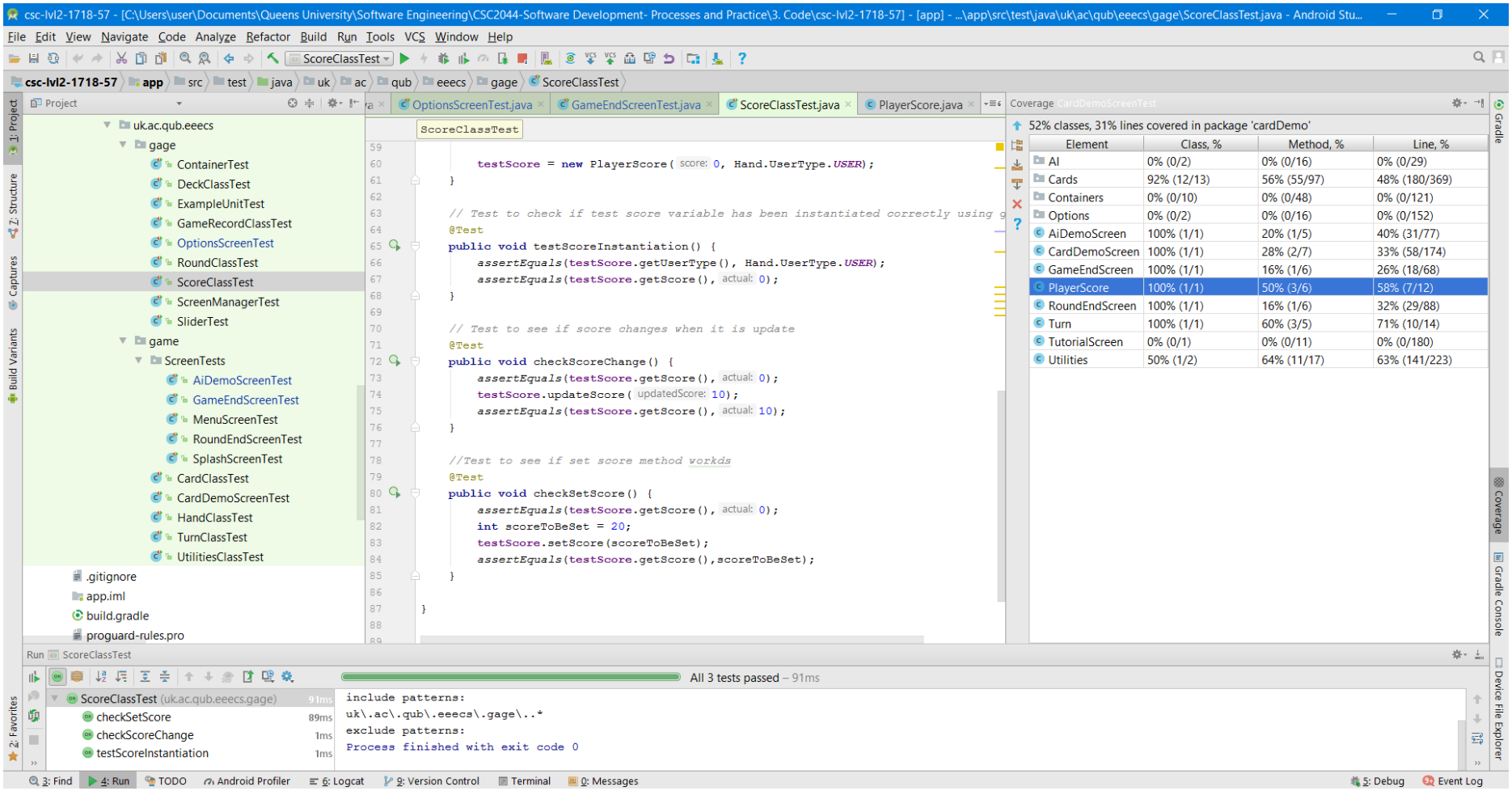
## Game End Screen Test

Here is the game end screen test which specifically tests the GameEndScreen class. The Unit test has two tests testing the game screen instantiation and testing the result check method.



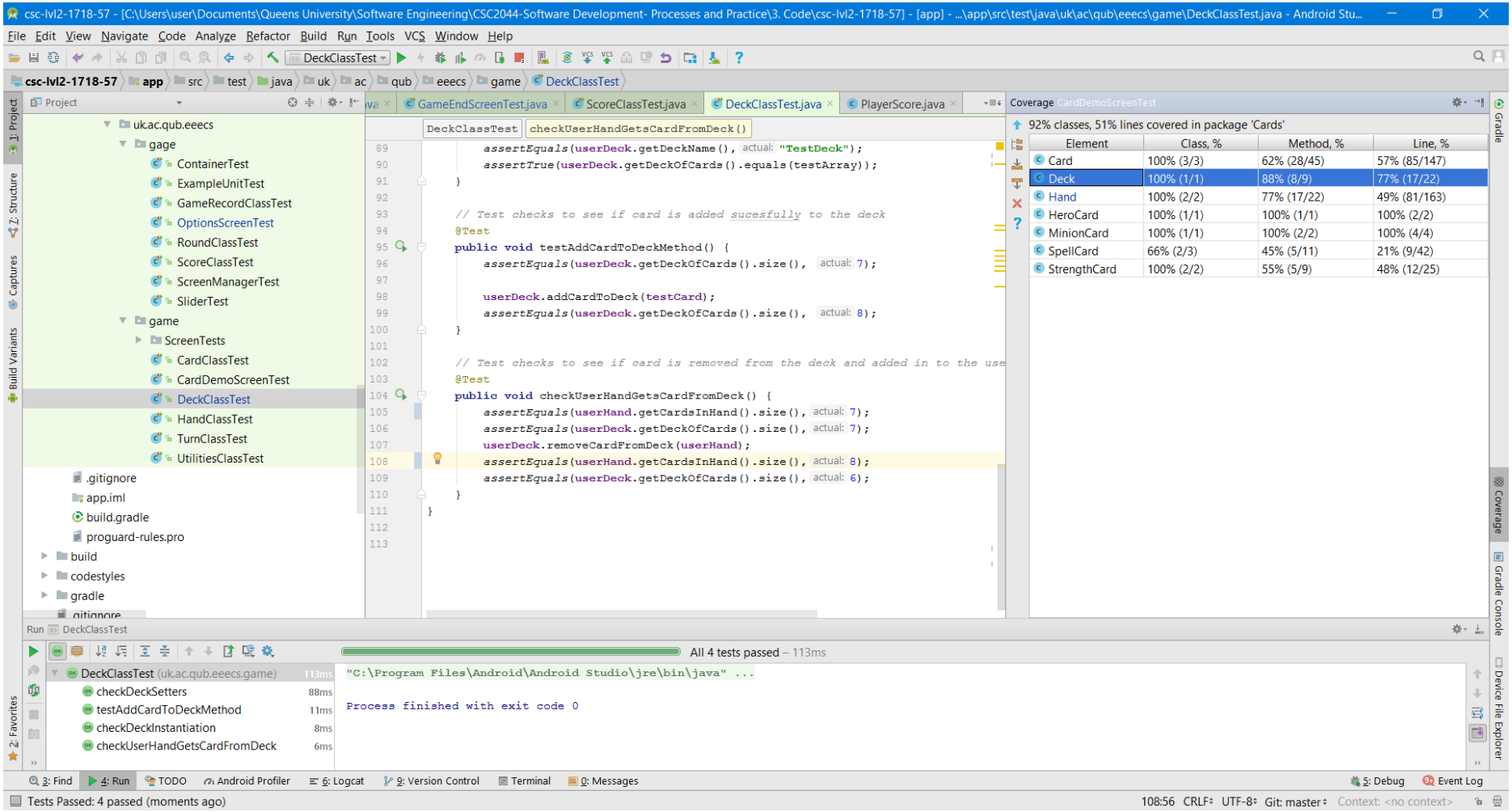
## Score Class Test

This is the score class test which is specifically testing only the score class. It has 3 tests testing the score objects instantiation, setters and score change method.

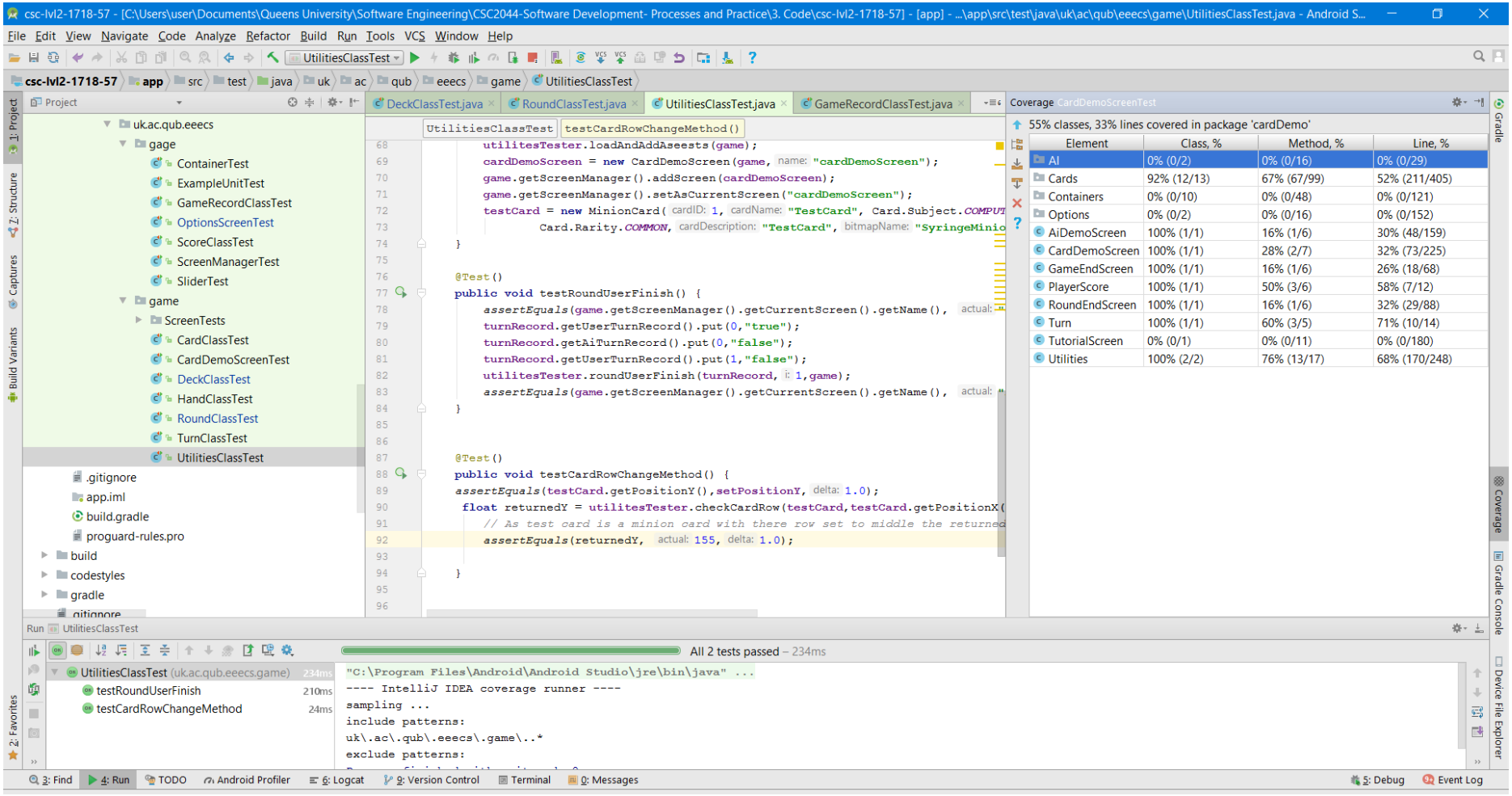


## Deck Class Tests

This is the deck class test which specifically tests the deck class test. This unit test tests adding and removing cards to and from the deck to the hand, deck instantiation, etc



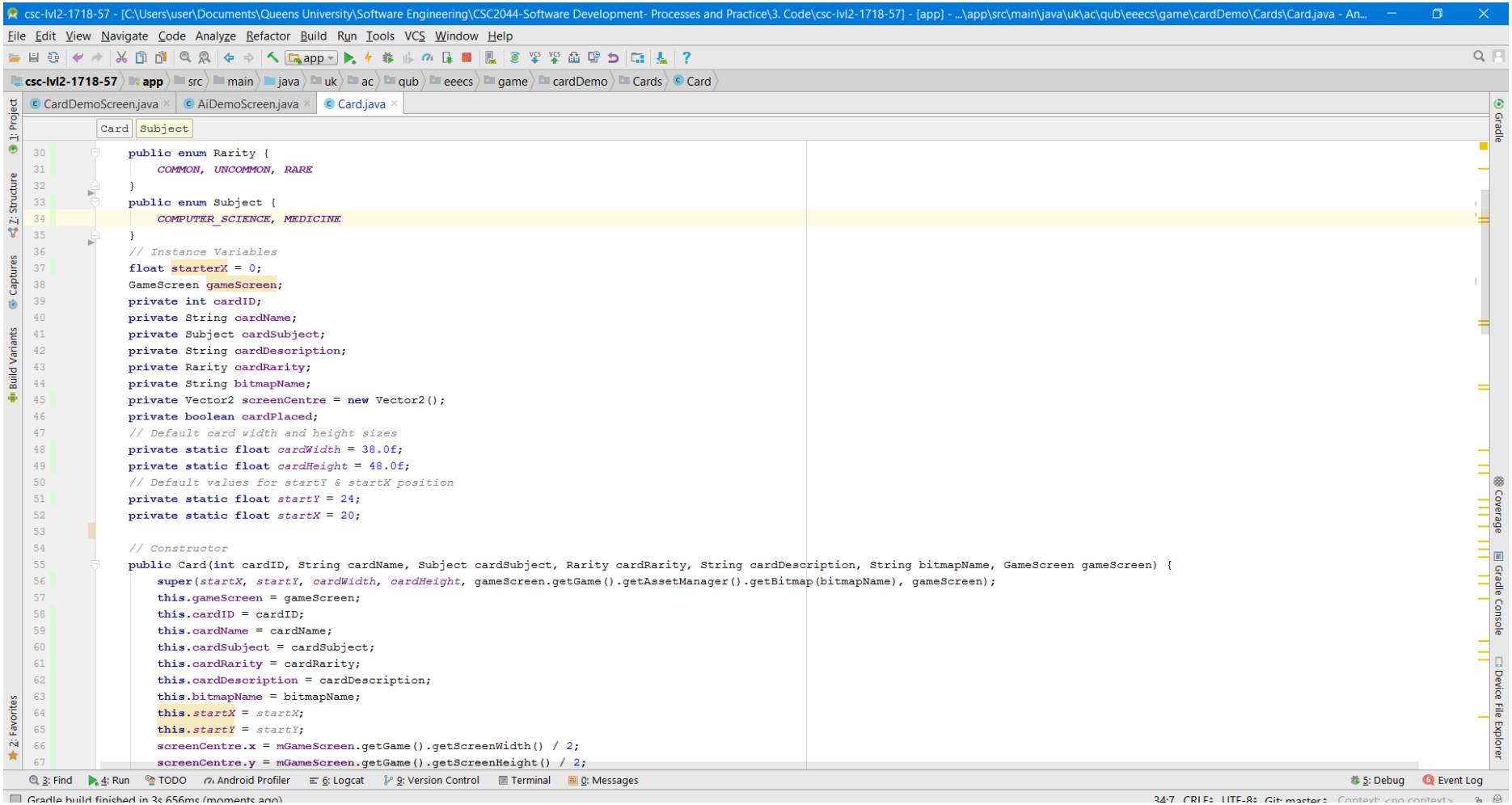
## Utilities Class Test



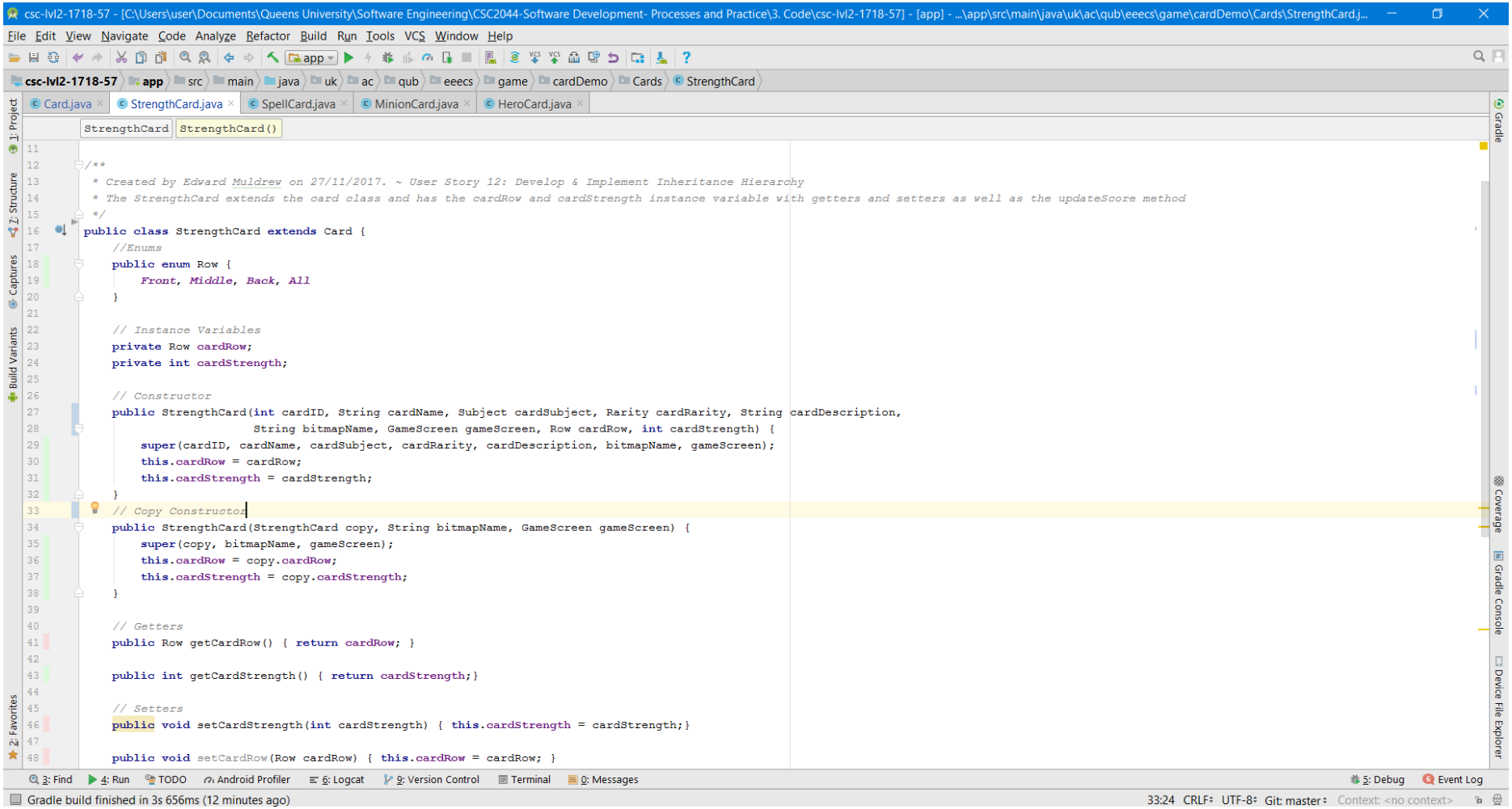
# Classes that match the documented design

Here is evidence of our class relationship model which represents the various classes involved in the Card superclass and its various subclasses

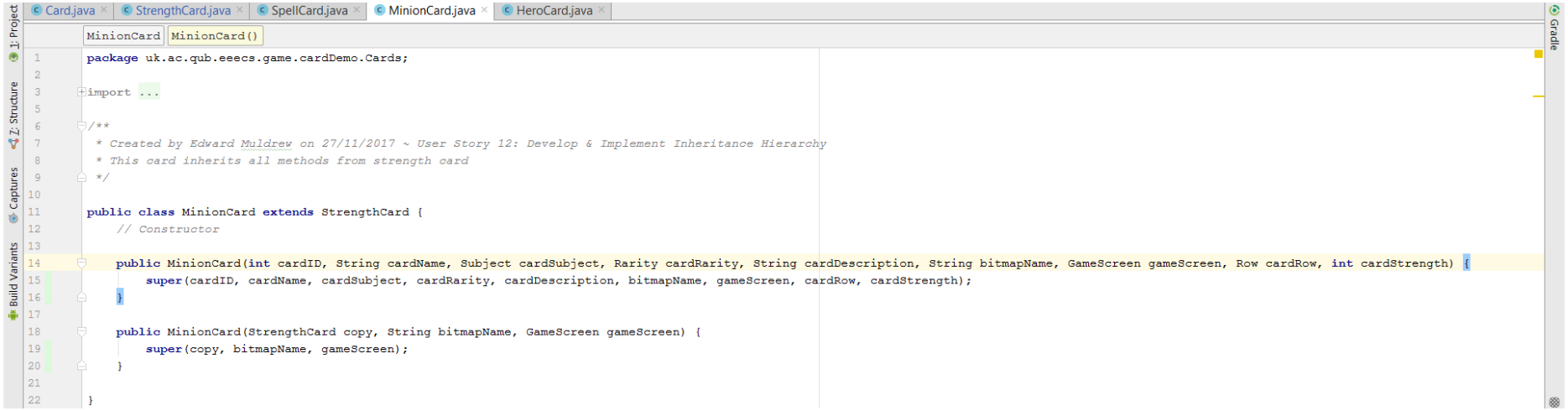
## Card Class



## Strength Card Class

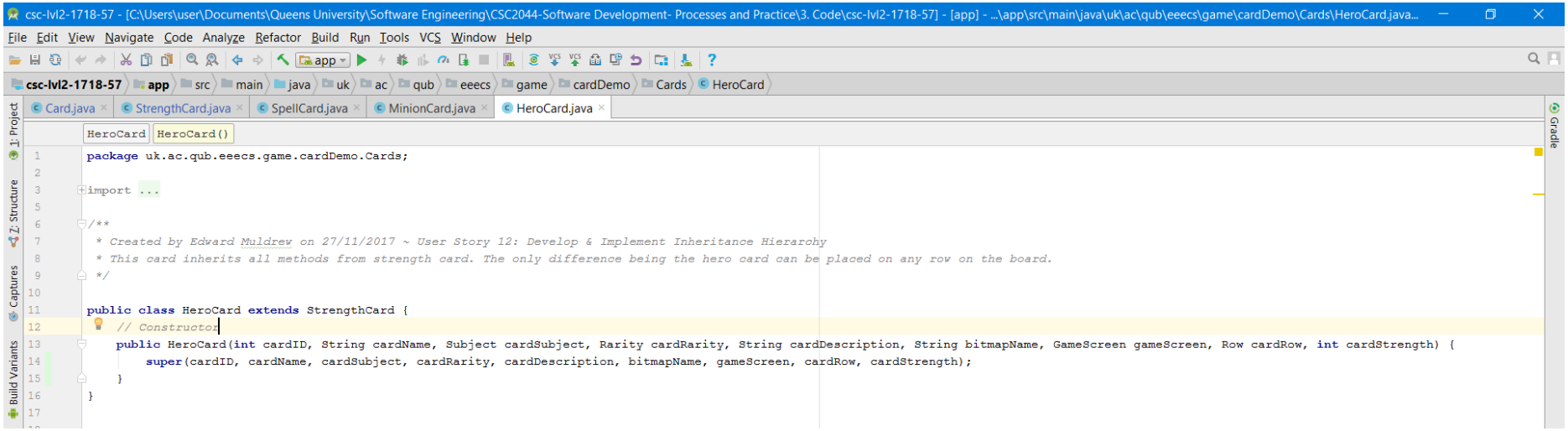


## Minion Card Class



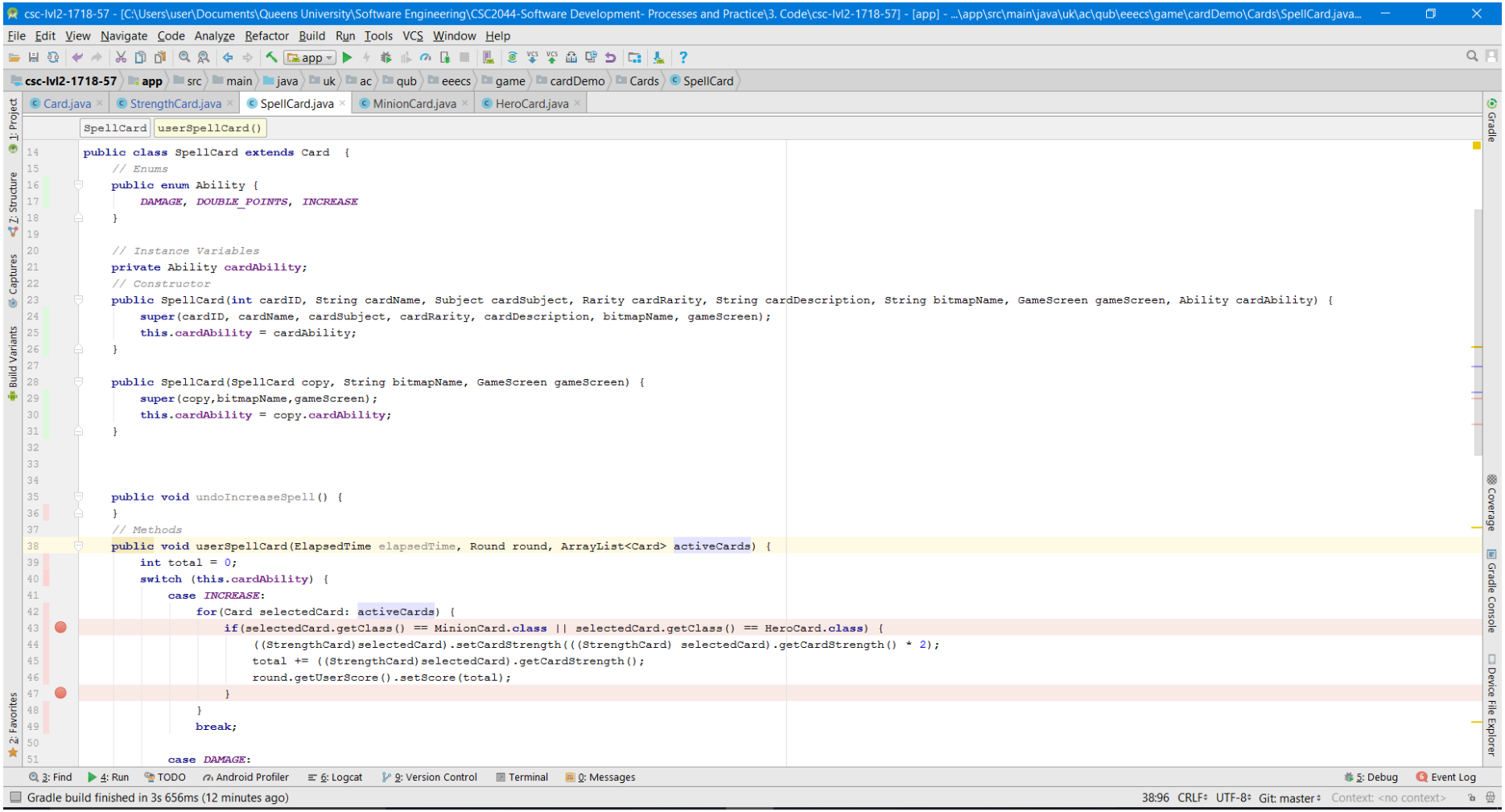
```java
package uk.ac.qub.eeecs.game.cardDemo.Cards;

import ...

/**
 * Created by Edward Muldrew on 27/11/2017 ~ User Story 12: Develop & Implement Inheritance Hierarchy
 * This card inherits all methods from strength card
 */

public class MinionCard extends StrengthCard {
    // Constructor

    public MinionCard(int cardID, String cardName, Subject cardSubject, Rarity cardRarity, String cardDescription, String bitmapName, GameScreen gameScreen, Row cardRow, int cardStrength) {
        super(cardID, cardName, cardSubject, cardRarity, cardDescription, bitmapName, gameScreen, cardRow, cardStrength);
    }

    public MinionCard(StrengthCard copy, String bitmapName, GameScreen gameScreen) {
        super(copy, bitmapName, gameScreen);
    }

}
```

## Hero Card Class



```java
package uk.ac.qub.eeecs.game.cardDemo.Cards;

import ...

/**
 * Created by Edward Muldrew on 27/11/2017 ~ User Story 12: Develop & Implement Inheritance Hierarchy
 * This card inherits all methods from strength card. The only difference being the hero card can be placed on any row on the board.
 */

public class HeroCard extends StrengthCard {
    // Constructor
    public HeroCard(int cardID, String cardName, Subject cardSubject, Rarity cardRarity, String cardDescription, String bitmapName, GameScreen gameScreen, Row cardRow, int cardStrength) {
        super(cardID, cardName, cardSubject, cardRarity, cardDescription, bitmapName, gameScreen, cardRow, cardStrength);
    }
}
```

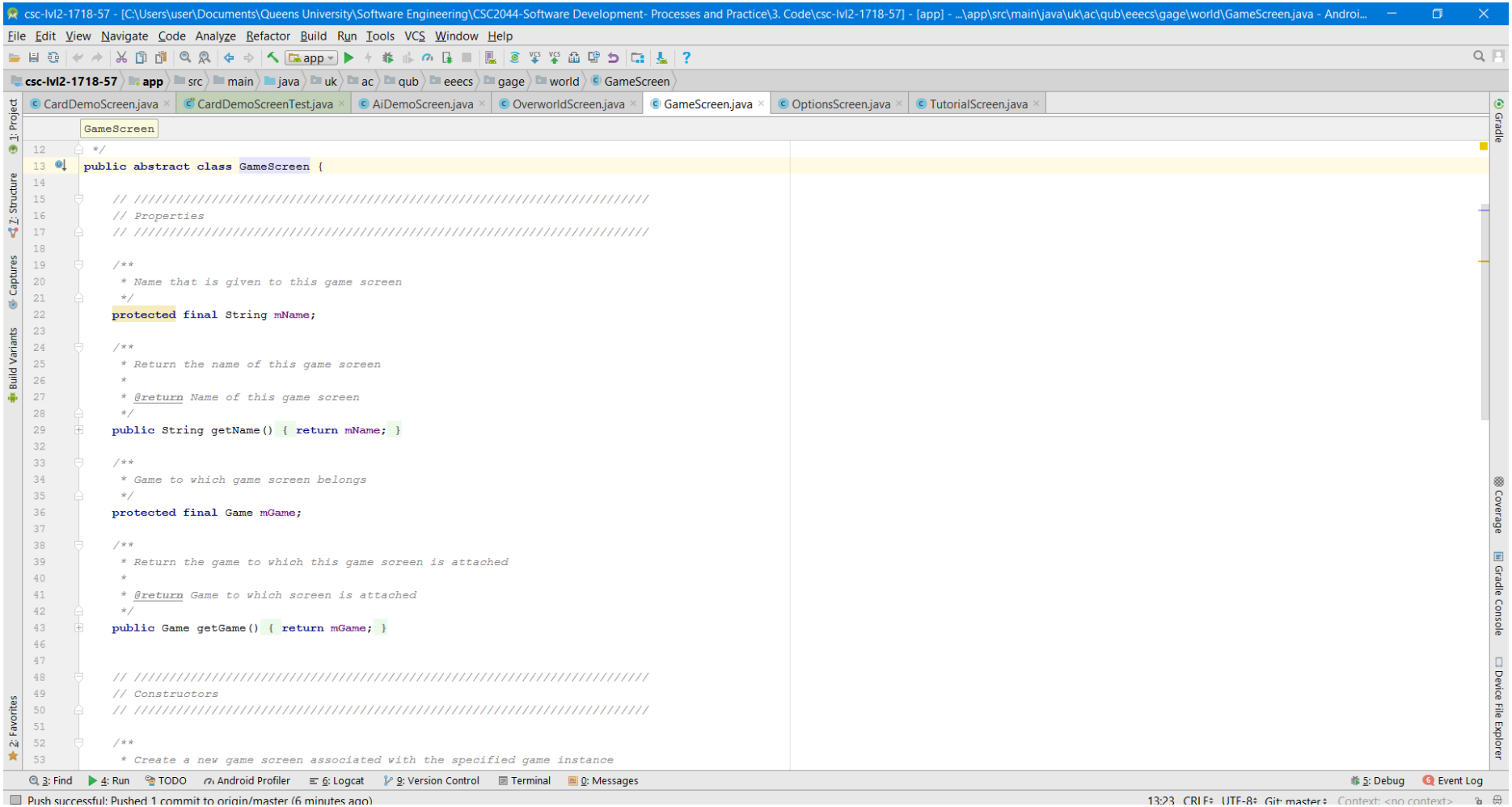
## Spell Card Class



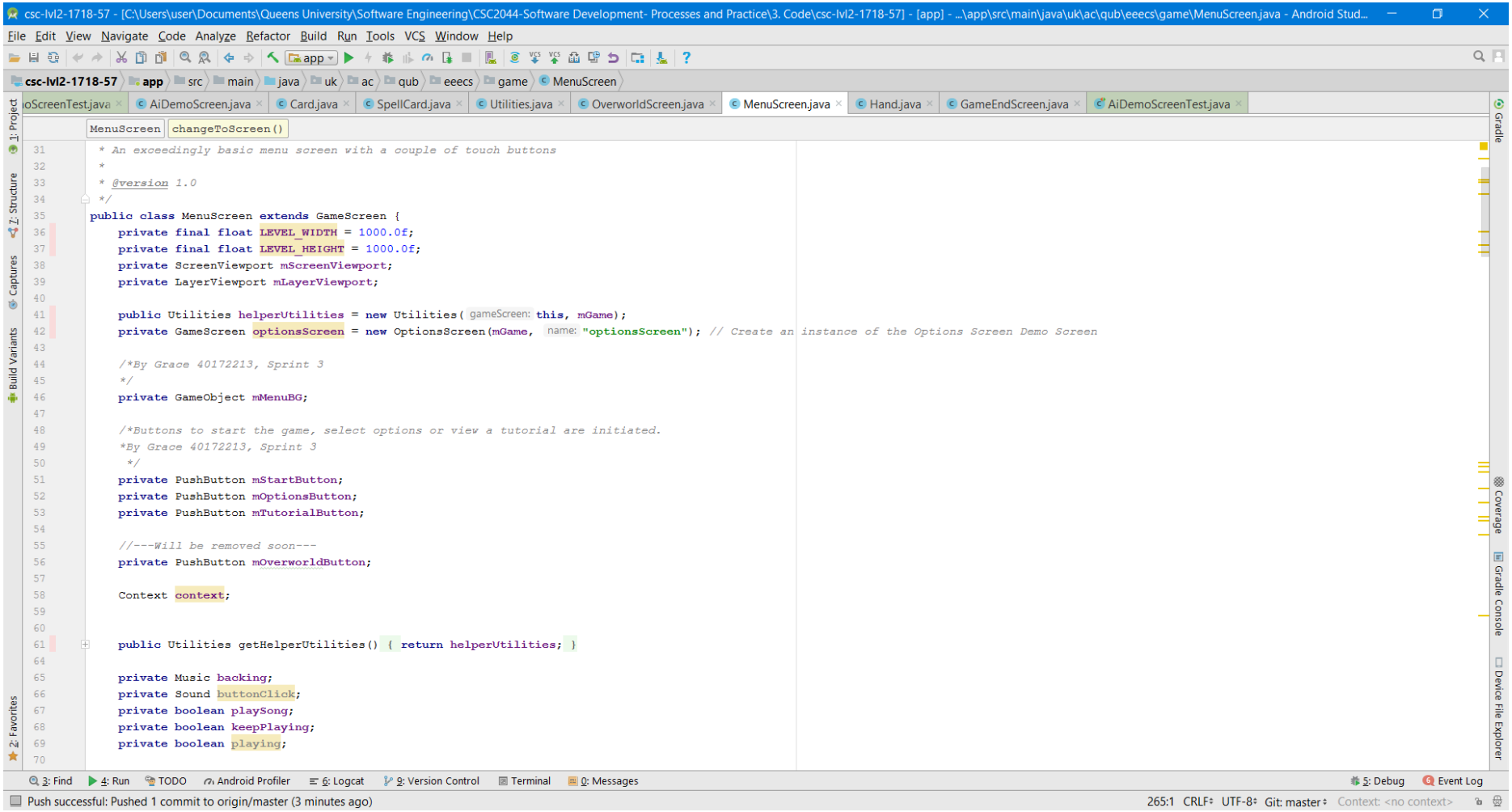

.

## Game Screen

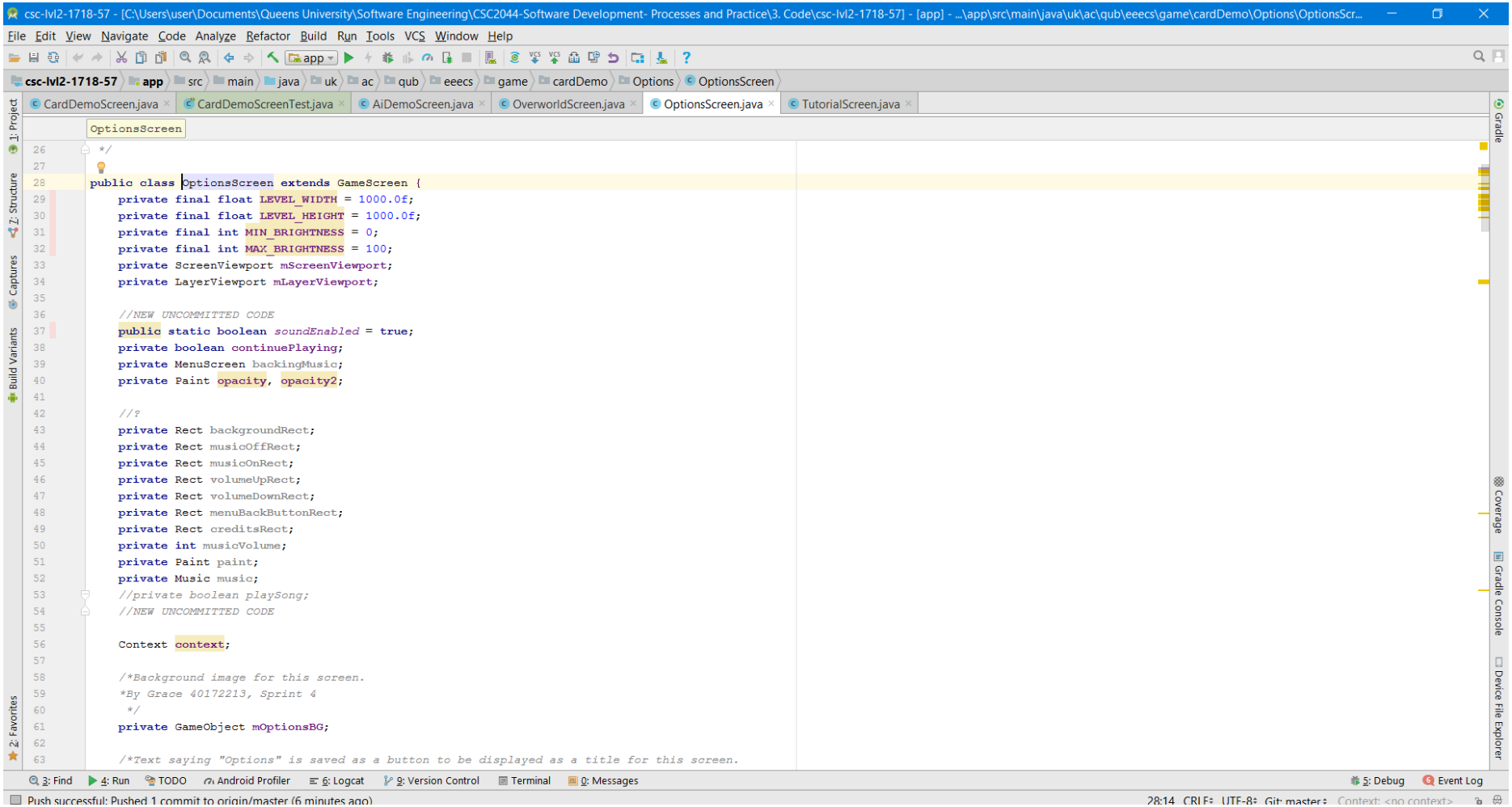
Here is evidence of the class relationship model which represents the various screens for our project which all inherit from the abstract class GameScreen



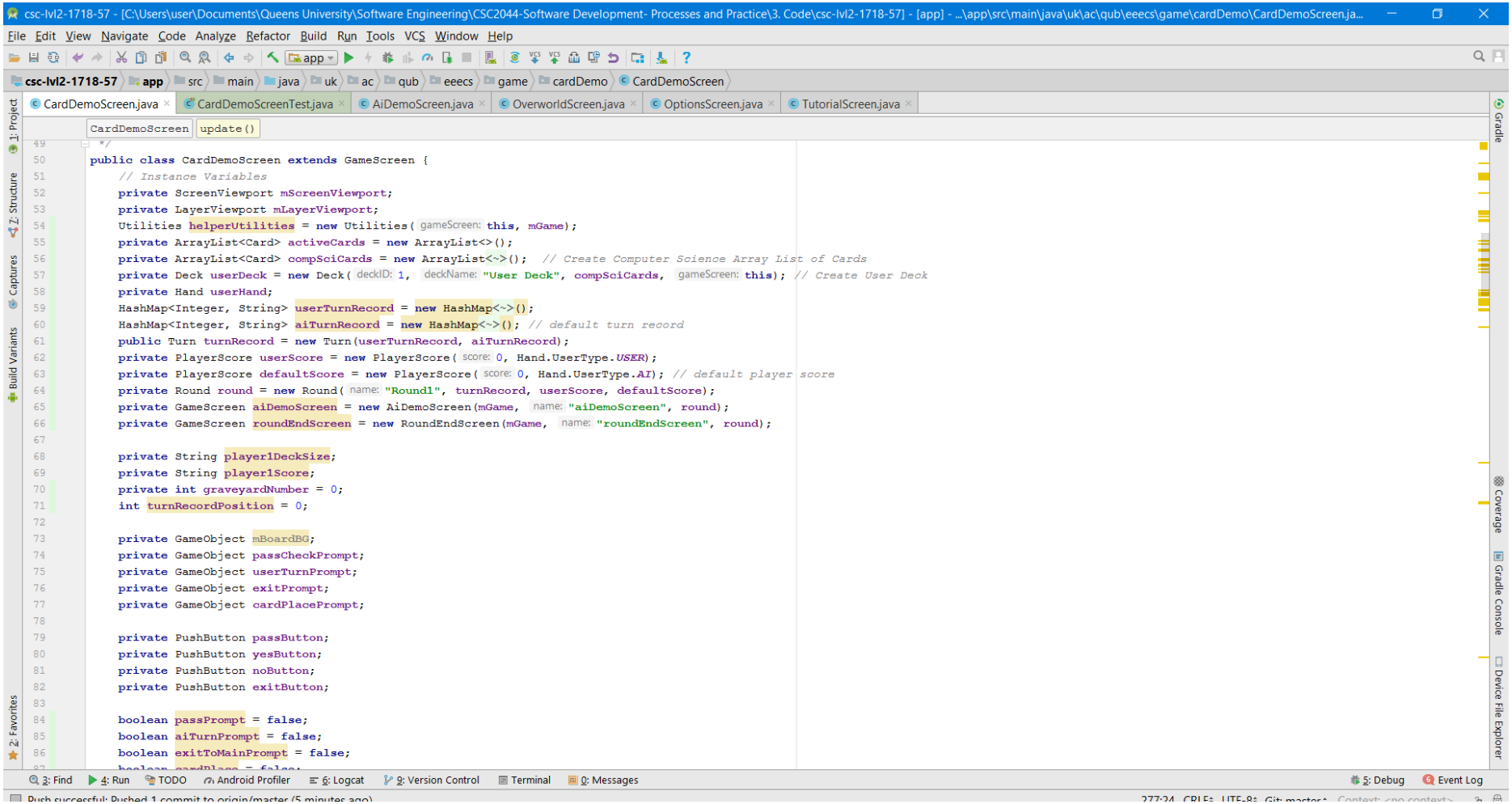
# Menu Screen


csc-lvl2-1718-57 - [C:\Users\user\Documents\Queens University\Software Engineering\CSC2044-Software Development- Processes and Practice\3. Code\csc-lvl2-1718-57] - [app] - ...\app\src\main\java\uk\ac\qub\eeecs\game\MenuScreen.java - Android Stud...
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
csc-lvl2-1718-57 app src main java uk ac qub eeecs game MenuScreen
AiDemoScreen.java Card.java SpellCard.java Utilities.java OverworldScreen.java MenuScreen.java Hand.java GameEndScreen.java AiDemoScreenTest.java
MenuScreen changeToScreen()
* An exceedingly basic menu screen with a couple of touch buttons
* @version 1.0
public class MenuScreen extends GameScreen {
private final float LEVEL_WIDTH = 1000.0f;
private final float LEVEL_HEIGHT = 1000.0f;
private ScreenViewport mScreenViewport;
private LayerViewport mLayerViewport;
public Utilities helperUtilities = new Utilities( gameScreen: this, mGame);
private GameScreen optionsScreen = new OptionsScreen(mGame, name: "optionsScreen"); // Create an instance of the Options Screen Demo Screen
/*By Grace 40172213, Sprint 3
*/
private GameObject mMenuBG;
/*Buttons to start the game, select options or view a tutorial are initiated.
*By Grace 40172213, Sprint 3
*/
private PushButton mStartButton;
private PushButton mOptionsButton;
private PushButton mTutorialButton;
//---Will be removed soon---
private PushButton mOverworldButton;
Context context;
public Utilities getHelperUtilities() { return helperUtilities; }
private Music backing;
private Sound buttonClick;
private boolean playSong;
private boolean keepPlaying;
private boolean playing;
3: Find 4: Run TODO Android Profiler 6: Logcat 9: Version Control Terminal 0: Messages 5: Debug Event Log
Push successful: Pushed 1 commit to origin/master (3 minutes ago)
265:1 CRLF UTF-8 Git: master Context: <no context>

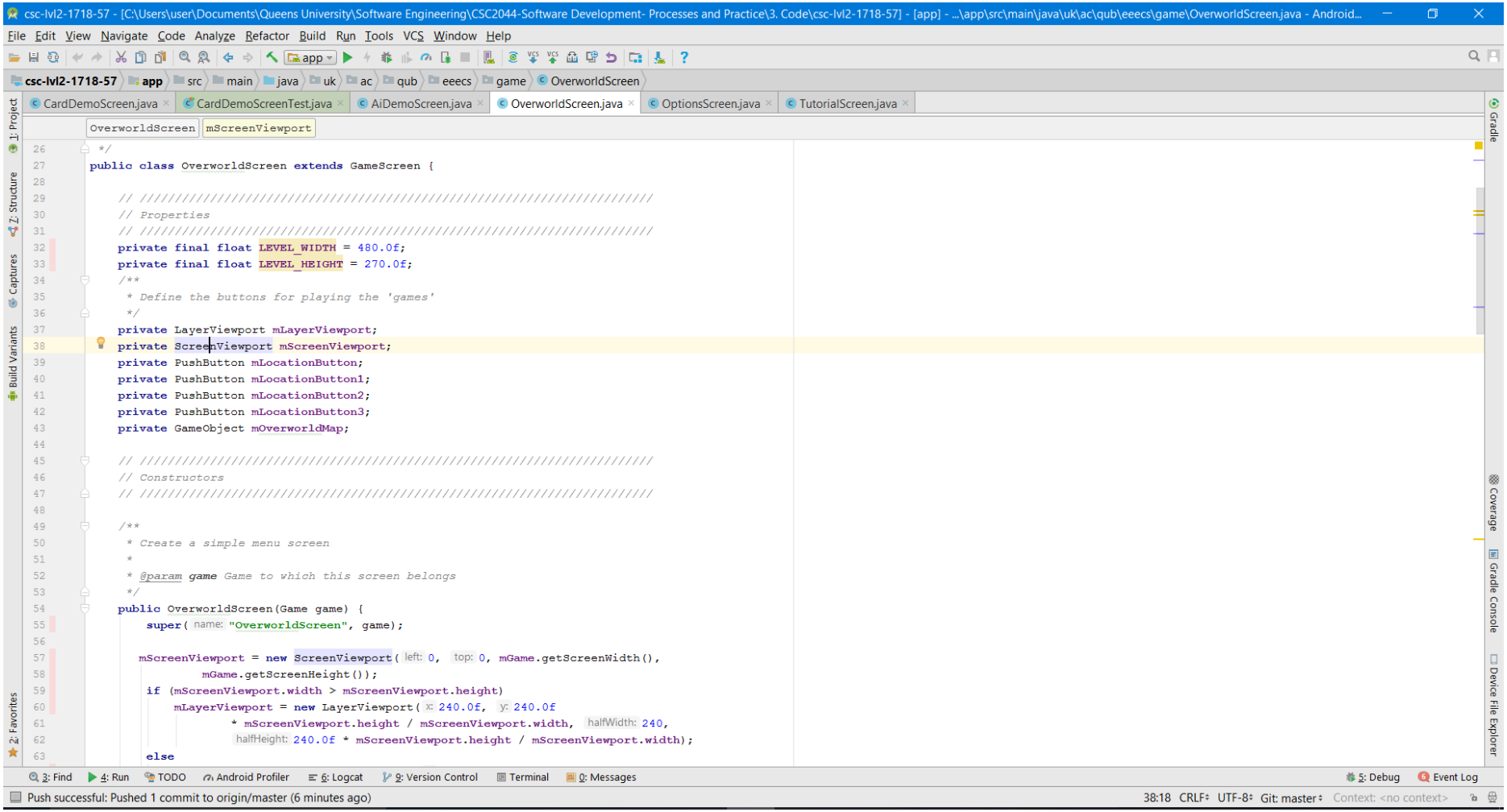
## Options Screen



# Card Demo Screen

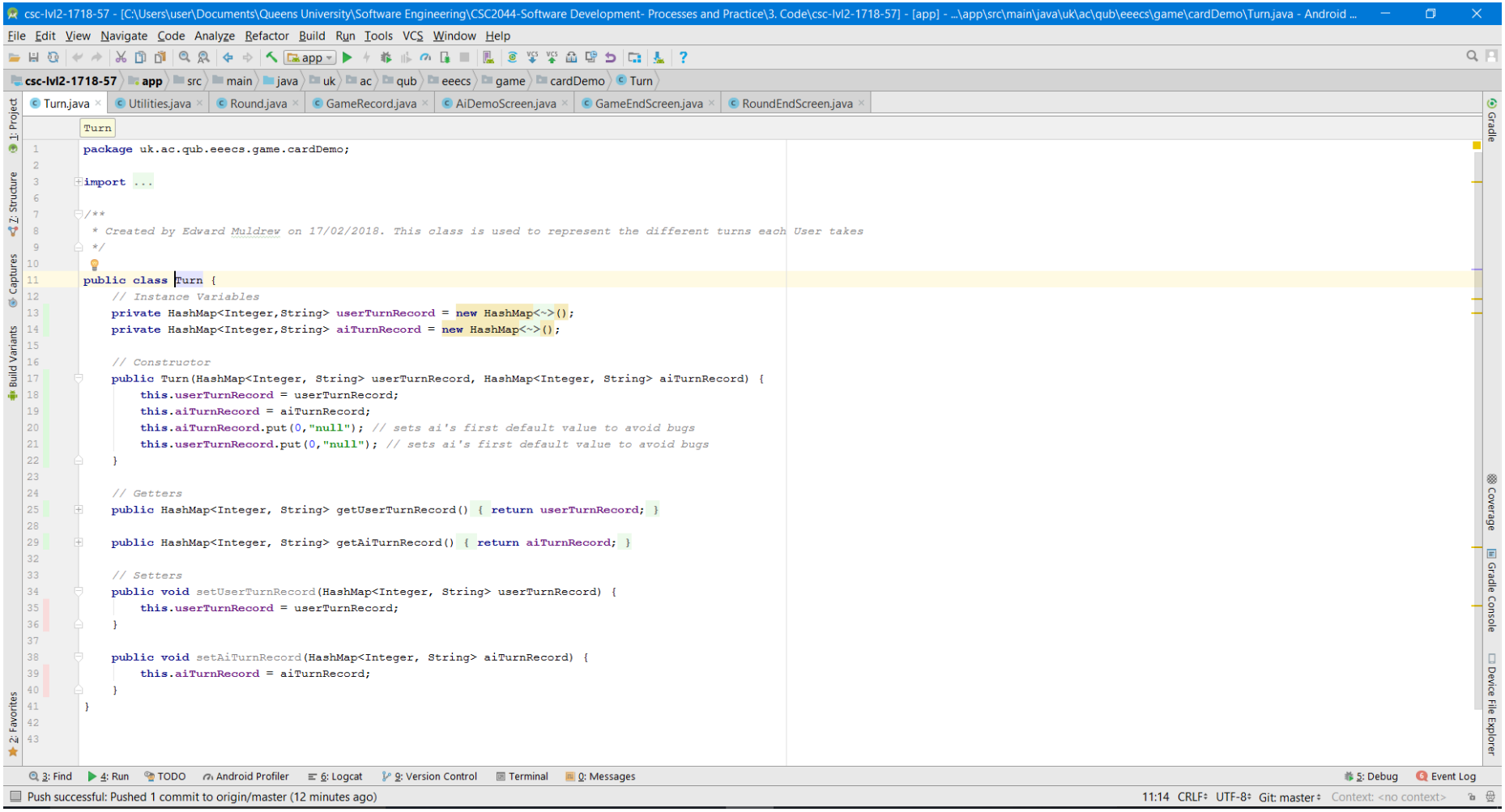


# Overworld Screen

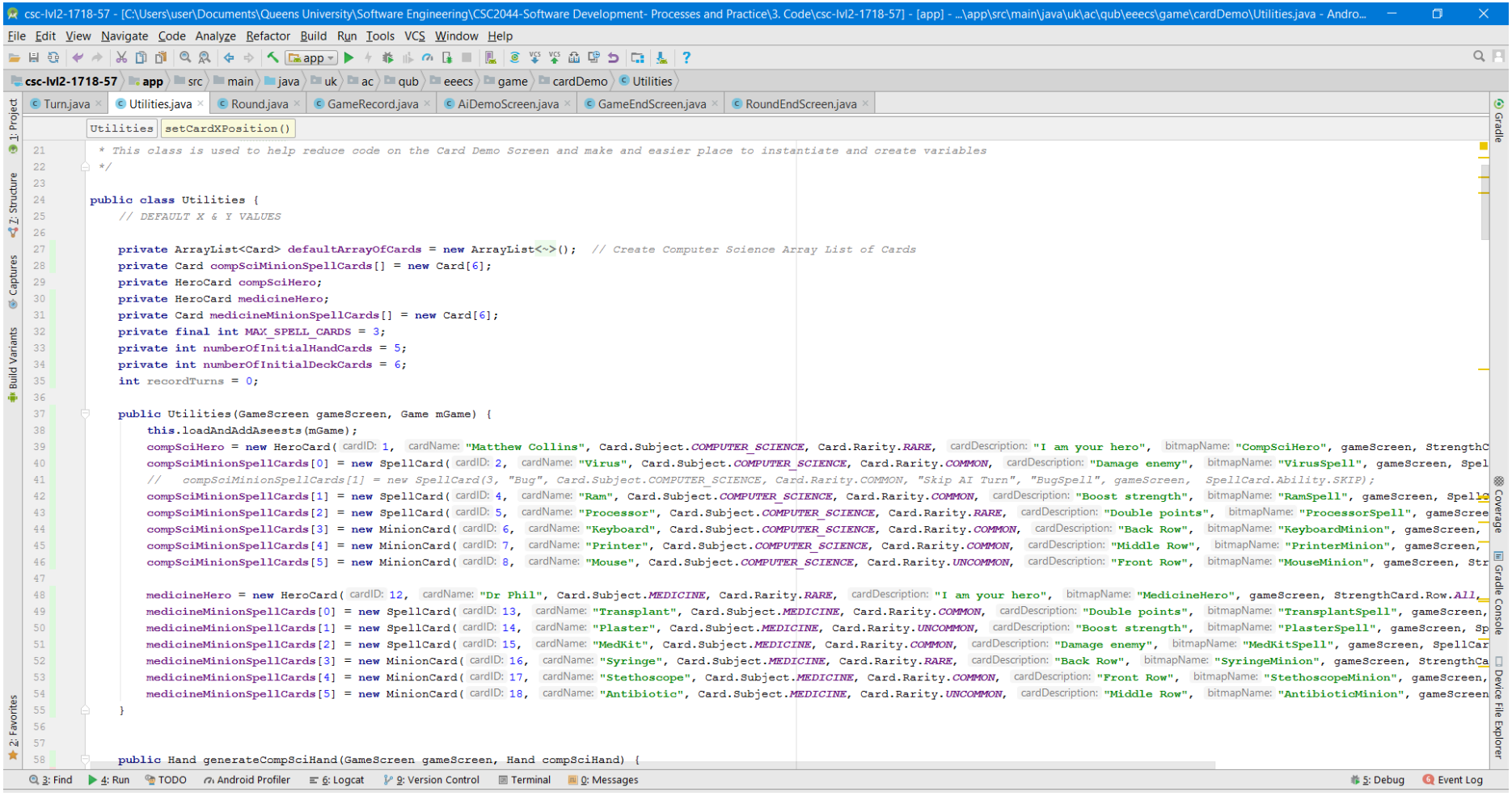


Here is evidence of the next class relationship model which represents the various screens for our actual game. This is a complex model showing the interactions between the game classes.

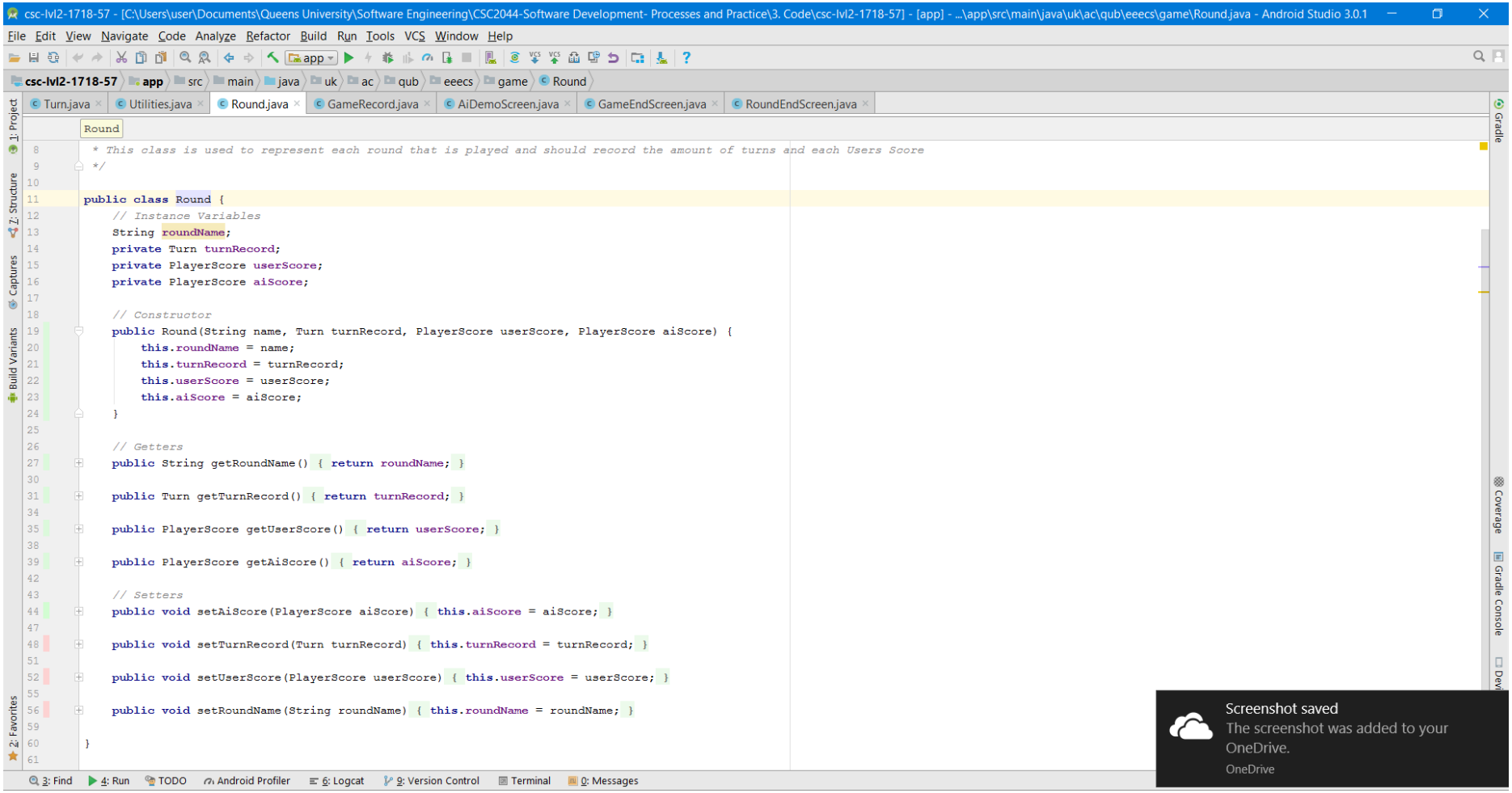
## Turn class



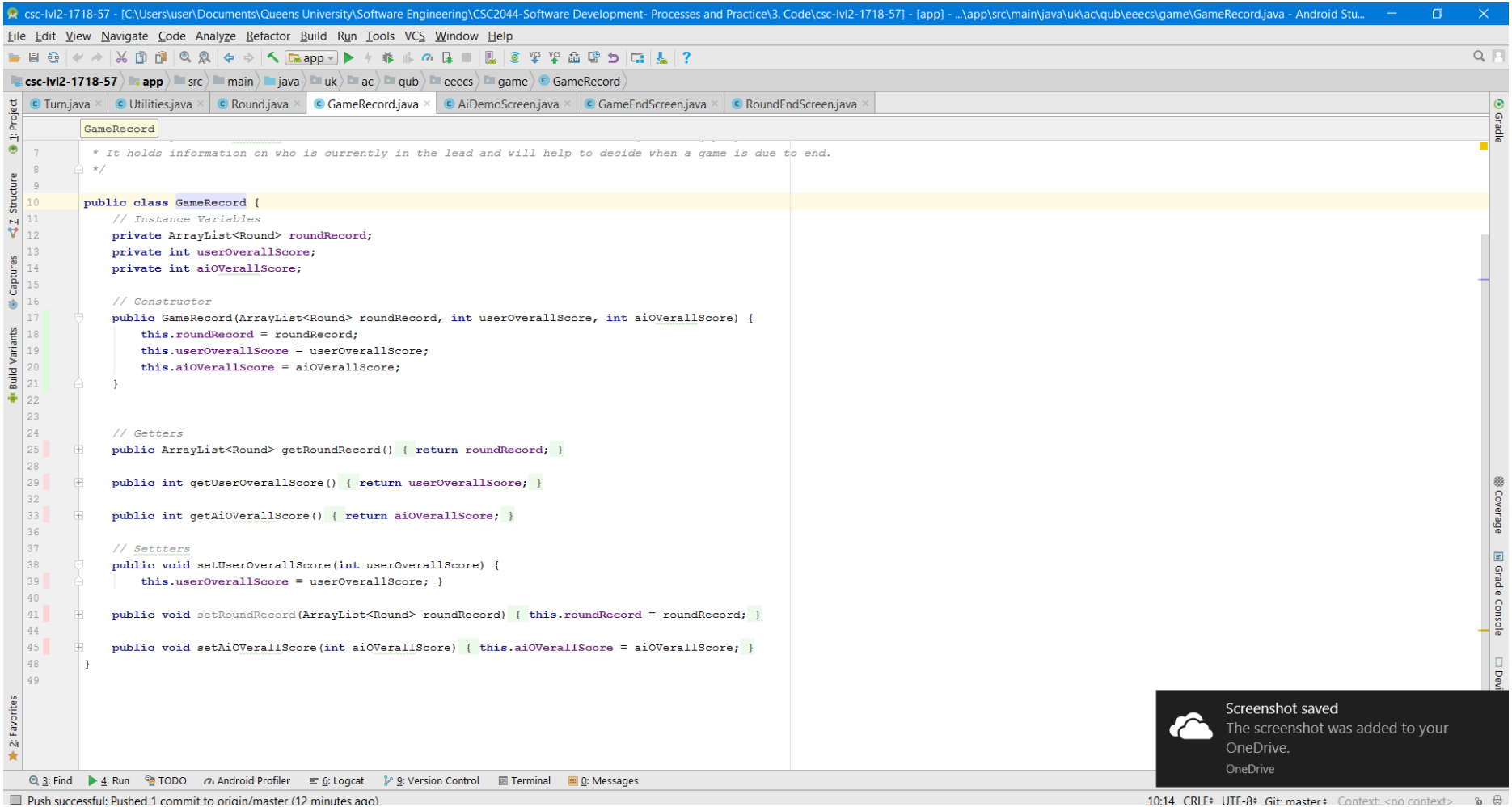
# Utilities Class



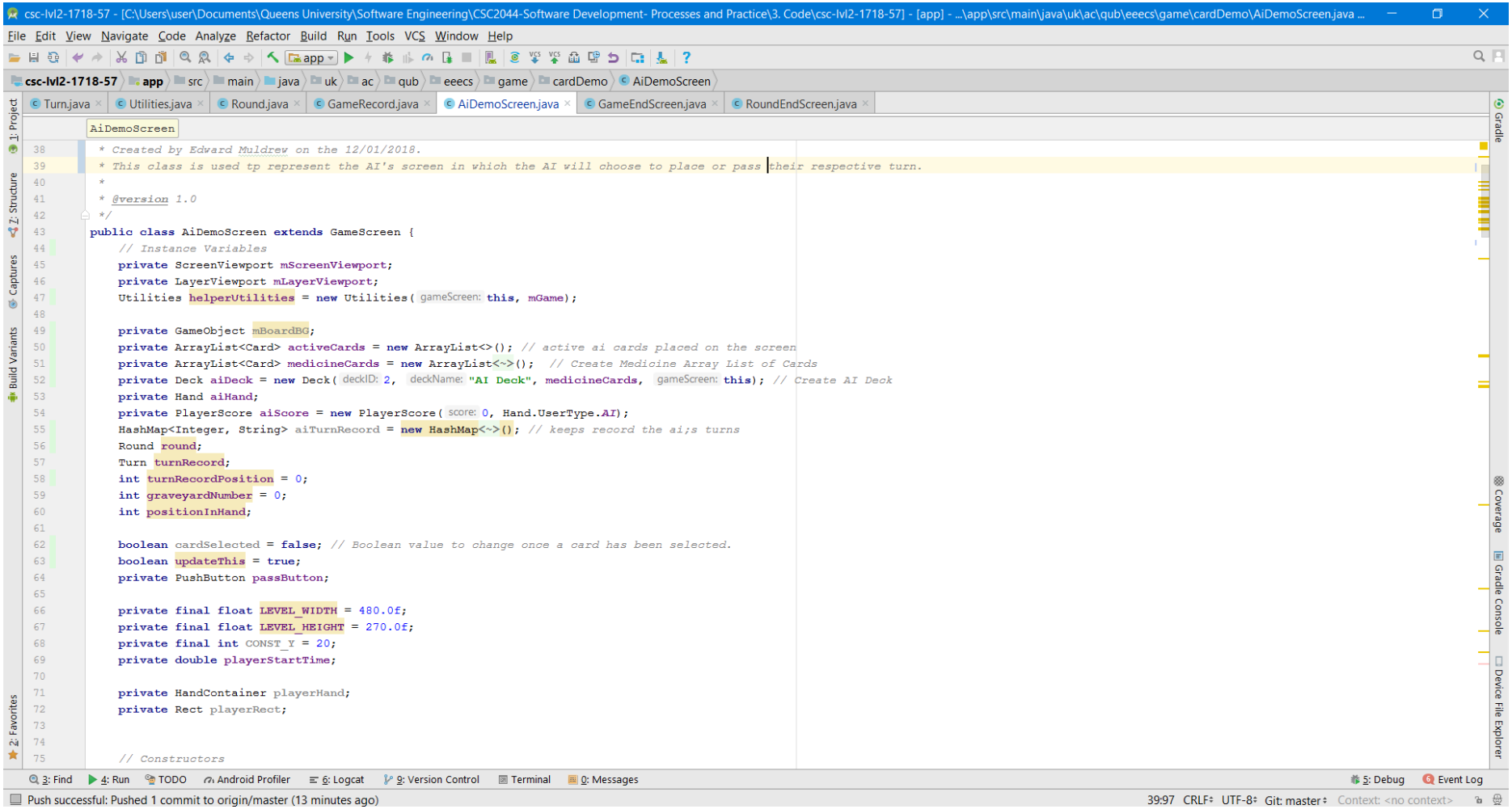
# Round Class


csc-lvl2-1718-57 - [C:\Users\user\Documents\Queens University\Software Engineering\CSC2044-Software Development- Processes and Practice\3. Code\csc-lvl2-1718-57] - [app] - ...\app\src\main\java\uk\ac\qub\eeecs\game\Round.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Turn.java
Utilities.java
Round.java
GameRecord.java
AiDemoScreen.java
GameEndScreen.java
RoundEndScreen.java
Round
* This class is used to represent each round that is played and should record the amount of turns and each Users Score
*/
public class Round {
// Instance Variables
String roundName;
private Turn turnRecord;
private PlayerScore userScore;
private PlayerScore aiScore;
// Constructor
public Round(String name, Turn turnRecord, PlayerScore userScore, PlayerScore aiScore) {
this.roundName = name;
this.turnRecord = turnRecord;
this.userScore = userScore;
this.aiScore = aiScore;
}
// Getters
public String getRoundName() { return roundName; }
public Turn getTurnRecord() { return turnRecord; }
public PlayerScore getUserScore() { return userScore; }
public PlayerScore getAiScore() { return aiScore; }
// Setters
public void setAiScore(PlayerScore aiScore) { this.aiScore = aiScore; }
public void setTurnRecord(Turn turnRecord) { this.turnRecord = turnRecord; }
public void setUserScore(PlayerScore userScore) { this.userScore = userScore; }
public void setRoundName(String roundName) { this.roundName = roundName; }
}
Screenshot saved
The screenshot was added to your OneDrive.
OneDrive

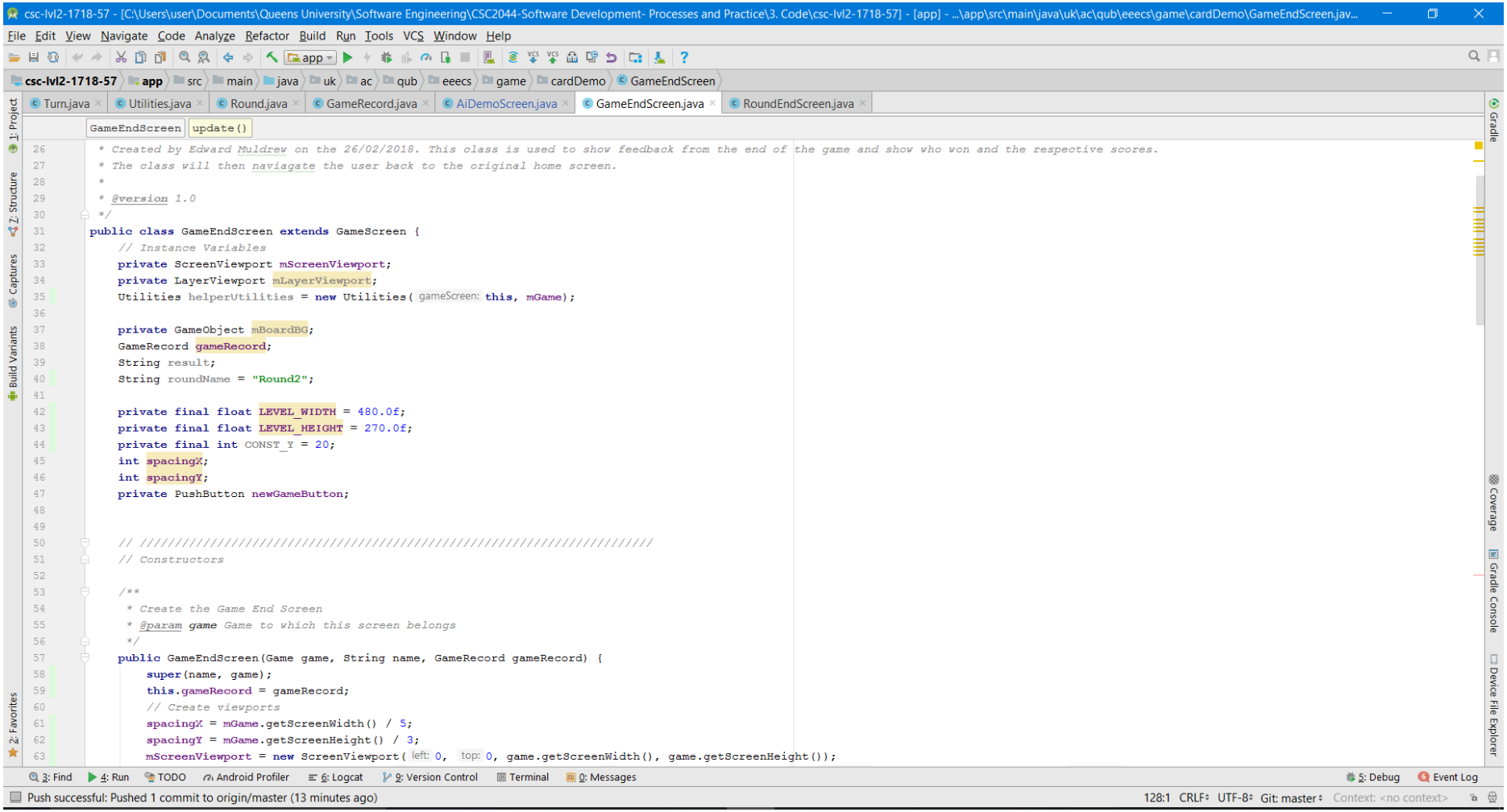
## Game Record Class



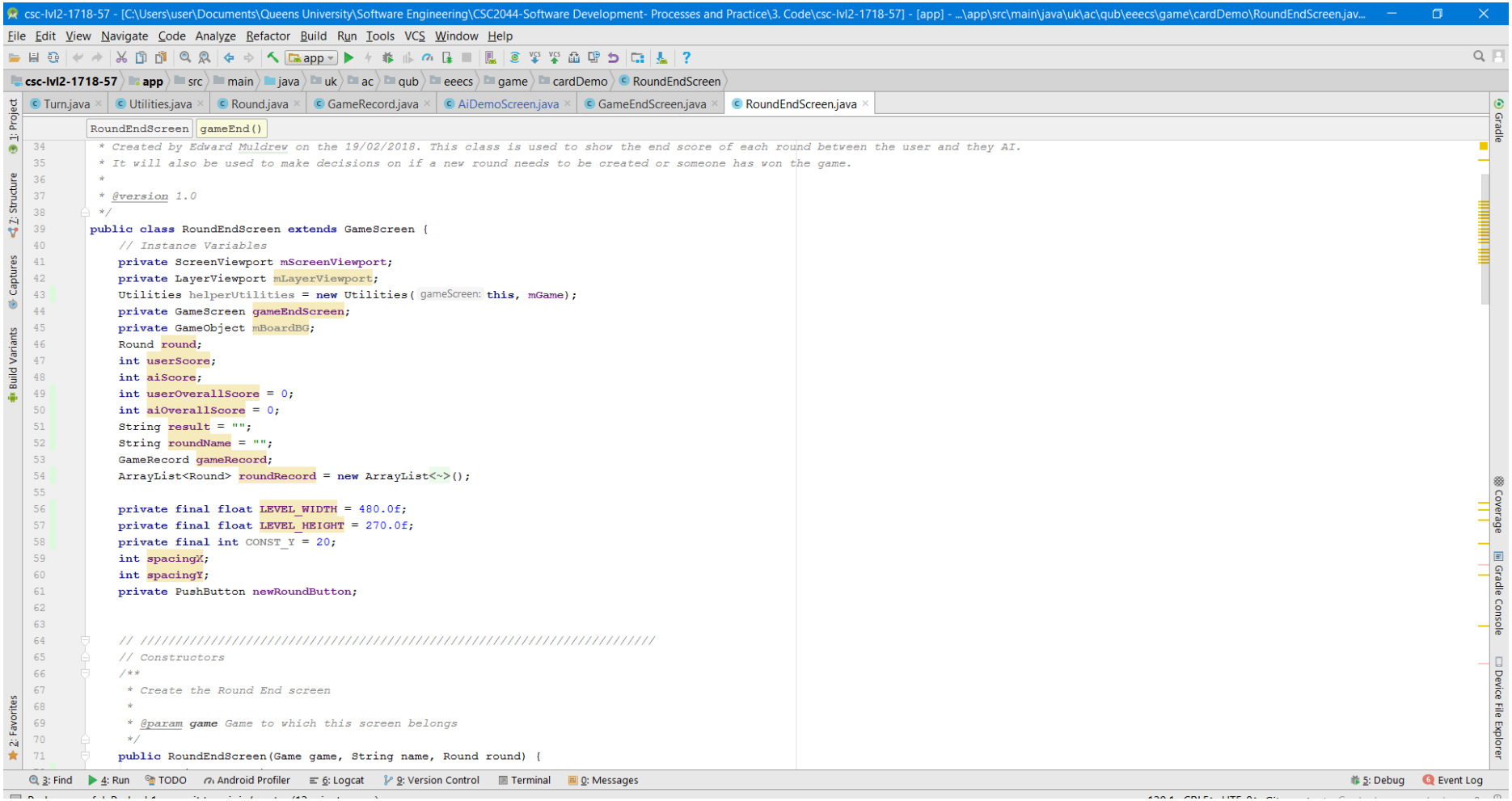
# AI Demo Screen Class



## Game End Screen Class



# Round End Screen



# Team Minutes

The following team members were present:

- Edward Muldrew
- Grace Turkington
- Brandon Smylie
- Conor Clarke
- Jack McNaughton

Name: Edward Muldrew Signature:

Name: Grace Turkington Signature:

Name: Brandon Smylie Signature:

Name: Conor Clarke Signature:

Name: Jack McNaughton Signature:

# Meeting 1: Sprint 4, Week 1

Week commencing: 08/01/2018

Date of this minute: 08/01/2018

## Task Reporting

First week of term- did not make progress in the last week

## Actions Planned

Edward Muldrew & Role: Programmer

- Complete User Stories for Card Class
- Submit User Stories document
- Continue to do some tests for card class
- Start on User Stories
- Meet up with group on Wednesday to discuss User Stories

Grace Turkington & Role (2):

- Meet with team on Wednesday to discuss feedback from sprint 3 and assign user stories for sprint 4
- Add a to-do list to the project page of my plan for an assigned user story
- Begin working on my assigned user stories
- Commit my finished changes to Git
- Watch this weeks youtube videos

Brandon Smylie & Role (3):

- Meet with team to discuss feedback and Sprint 4.
- Add my user stories to the document.
- Begin working on those user stories.
- Submit user story document to Phillip.

Conor Clarke & Role (4):

- Attend group meeting to discuss Sprint 3 feedback and Sprint 4 plans
- Add user stories to Sprint 4 document
- Work on user stories from older sprints

Jack McNaughton & Role (5):

- Work on any remaining user stories from previous sprint.
- Meet with team to discuss user stories for next sprint.
- Add new user stories to sprint document
- Start on new user stories
- Watch this YouTube videos

# Meeting 2: Sprint 4, Week 2

Week commencing: 15/01/2018

Date of this minute: 15/01/2018

## Task Reporting

Edward Muldrew & Role: Programmer

- Met with team on Wednesday to discuss the User Story document
- Started work on User story assigning random cards to each deck
- Initialized and declared medicine deck
- Removed old code from project file
- Added unit tests to test Card Class code

Grace Turkington & Role (2):

- Met with team to assign sprint 4 user stories
- Changed appearance of main menu screen, removing unused code and buttons
- Began user story 9
- Watched this weeks youtube videos

Brandon Smylie & Role (3):

- Met with team to discuss user stories.
- Re-evaluated user story 5 into a grand user story.
- Implemented a container superclass.
- Tested some superclass methods.

Conor Clarke & Role (4):

- Met with team to discuss user stories
- Work on Overworld user stories
- Edit sprite sheets

Jack McNaughton & Role (5):

- Met with team to discuss user stories.
- Wrote user stories to document.
- Attempted user story 17.
- Watched the weeks YouTube videos.

## Actions Planned

Edward Muldrew & Role: Programmer

- Continue work on my User Stories
- Add unit tests for the new User Stories created
- Meet up with the team on Tuesday to discuss user story progress and any problems they have thus far
- Commit all finished changes to Git
- Watch supplied YouTube videos from Philip

Grace Turkington & Role (2):

- Add a to-do list to the project page of my plan for an assigned user story
- Keep working on sprint 4 user stories and write some tests for these
- Commit my finished changes to Git
- Team meeting on Tuesday to discuss sprint 4 progress so far
- Watch this weeks youtube videos

Brandon Smylie & Role (3):

- Re-write my grand user story into 4 smaller, more manageable user stories.
- Commit my finished changes to Git.
- Meet with team on Tuesday to discuss progress.

Conor Clarke & Role (4):

- Meet with team to discuss progress and problems
- Watch helpful youtube videos

Jack McNaughton & Role (5):

- Change user story 17 so that the goal is to draw the text using draw methods instead of reading from a text document
- Start work on user story 18.
- Meet with team to discuss user story progression.
- Watch YouTube videos for the week.

# Meeting 3: Sprint 4, Week 3

Week commencing: 22/01/2018

Date of this minute: 22/01/2018

## Task Reporting

Edward Muldrew & Role: Programmer

- Team meeting to assess User Story progress with group. Discusses how our game is going to work and play out
- Added random deck generation for both decks
- Started work on Deck and Hand class
- Added copy constructors so duplicate objects of the same type can be created

Grace Turkington & Role (2):

- Met with team to discuss progress with sprint 4 user stories
- Added menu screen tests
- Created options screen (user story 10)
- Added options screen tests
- Began user story 11
- Watched this weeks youtube videos

Brandon Smylie & Role (3):

- Team meeting to assess user story progress.
- Reevaluated my user stories.
- Continued work on user story 5.

Conor Clarke & Role (4):

- Met with team to discuss progress and problems
- Continued to work on overworld, tried to string together a map from a tilesheet
- Listened to the youtube videos

Jack McNaughton & Role (5):

- Met with team to discuss user story progression.
- Watched YouTube videos
- Converted text document into draw text methods.
- Adjusted placement and sizes of text and images used so that they don’t block each other out.

## Actions Planned

Edward Muldrew & Role: Programmer

- Meet with team to discuss User Story progress
- Add Unit tests to test new code that has been added
- Continue work on Hand and Deck class
- Aim to try get a selected mechanism when User Clicks on card
- Add Amount of cards in deck class on Card Demo Screen

Grace Turkington & Role (2):

- Commit tests I've done to Git
- Keep working on user story 11
- Team meeting to complete sprint 4 submission document on Friday
- Watch this weeks youtube videos

Brandon Smylie & Role (3):

- Finish grand user story 5.
- Meet with group to submit the Sprint 4 report.
- Submit Sprint 4 report.

Conor Clarke & Role (4):

- Meet with group to add my contribution to sprint 4 report
- Continue to work on what I couldn’t finish in the sprint

Jack McNaughton & Role (5):

- Meet with team for user story progression and end of sprint document
- Research how to make an android screen scroll in android studio for user story 19.
- Upload completed code to git.
- Watch weeks YouTube video.

# Meeting 4: Sprint 5, Week 1

Week commencing: 29/01/2018

Date of this minute: 29/01/2018

## Task Reporting

Edward Muldrew & Role: Programmer

- Met with team to create Sprint 4 document and discussed progress thus far
- Add Unit tests to test recently added code
- Add Hand & Deck class
- Added AI cards set to be blank

Grace Turkington & Role (2):

- Met with team to discuss finalise sprint 4 document
- Completed and committed 3 user stories
- Watched this weeks youtube videos

Brandon Smylie & Role (3):

- Team meeting  to discuss, create and finalise sprint 4 document
- Completed and committed 1 grand user story
- Tested basic functionality of container superclass

Conor Clarke & Role (4):

- Met with team to discuss finalise sprint 4 document
- Did sprites and some of the map, no commit

Jack McNaughton & Role (5):

- Met with team to discuss Sprint 4 document and sprint progression.
- Uploaded stories 17 and 18 to git.
- Watched this week's YouTube videos.
- Checked for videos on how to scroll screen using touch.

## Actions Planned

Edward Muldrew & Role: Programmer

- Meet up with the team this week to discuss Sprint 4 progress & assign new user stories
- Start work on User Stories
- Commit changes to Git

Grace Turkington & Role (2):

- Meet with team to assign sprint 5 user stories
- Add a to-do list to the project page of my plan for an assigned user story
- Begin working on my assigned user stories
- Commit my finished changes to Git
- Watch this weeks youtube videos

Brandon Smylie & Role (3):

- Have team meeting to assign sprint 5 user stories
- Begin working on the user stories assigned to myself
- Commit any finished changes to Git
- Email user story document to Phillip

Conor Clarke & Role (4):

- Have team meeting to assign sprint 5 user stories
- Begin working on the user stories assigned to myself
- Put my user stories in the sprint 5 document

Jack McNaughton & Role (5):

- Meet with team to discuss Sprint 5 user stories
- Watch this week's YouTube videos
- Start on user stories.
- Input user stories into user stories document

**Meeting 5: Sprint 5, Week 2**

Week commencing: 05/02/2018

Date of this minute: 05/02/2018

## Task Reporting

Edward Muldrew & Role: Programmer

- Met with team to discuss User Stories
- Started work on Score class
- Completed User Story document

Grace Turkington & Role (2):

- Met with team to discuss create and assign sprint 5 user stories
- Added splash, menu and options screen tests
- Created to do lists for all my user stories (11-13)
- Completed and committed user story 12
- Watched this weeks youtube videos

Brandon Smylie & Role (3):

- Met with team to discuss, create and assign sprint 5 user stories
- Refactored some container code
- Added some more tests

Conor Clarke & Role (4):

- Met with team to discuss, create and assign sprint 5 user stories
- Made a start on the walk method for the player sprite

Jack McNaughton& Role (5):

- Watched this week's YouTube videos
- Met with team to discuss user stories.
- Determined that making the screen scroll in response to touch will take too long
- Broke large segment of text segments into smaller pages.

## Actions Planned

Edward Muldrew & Role: Programmer

- Meet with team to discuss User Story progress
- Finish work on Score class
- Add working show size of deck on screen
- Add User Stories to Test Score class

Grace Turkington & Role (2):

- Meet with team to discuss progress on sprint 5 user stories
- Continue working on user stories
- Commit my finished changes to Git
- Continue writing unit tests
- Watch this weeks youtube video

Brandon Smylie & Role (3):

- Meet with team to discuss progress on sprint 5 user stories
- Continue working on refactoring the Container hierarchy
- Commit my finished changes to Git
- Test any additional container methods

Conor Clarke & Role (4):

- Meet with team to discuss progress and problems
- Watch the youtube videos
- Continue work on overworld, try to finish walk method

Jack McNaughton & Role (5):

- Watch this weeks YouTube videos
- Meet with team to discuss user story progress
- Make the screen transition to a different draw method via a button click
- Upload code to git.

# Meeting 6: Sprint 5, Week 3

Week commencing: 12/02/2018

Date of this minute: 12/02/2018

## Task Reporting

Edward Muldrew & Role: Programmer

- Met with team on Friday to discuss sprint progress. Also discussed future of project and what needs completed in the upcoming sprints
- Added working select and place card
- Added working update score
- Added working show number of cards in deck on screen
- Added working switch between AI and User screens
- Added Utilities class so code can be maintained more easily and code is more readable

Grace Turkington & Role (2):

- Met with team to discuss sprint 5 progress
- Worked on user stories 11 & 13
- Watched this weeks youtube videos

Brandon Smylie & Role (3):

- Met with team on Friday to discuss sprint 5
- Finished refactoring container hierarchy

Conor Clarke & Role (4):

- Did not finish walk method
- Met with team on Friday to discuss sprint 5
- Listened to youtube videos

Jack McNaughton& Role (5):

- Watched this weeks YouTube videos
- Adjusted positioning of text and pictures for draw methods.
- Uploaded code to git
- Met with team to discuss sprint 5

## Actions Planned

Edward Muldrew & Role: Programmer

- Meet with team to complete User Story document
- Tidy up code which has been added
- Ensure code works the way it should and removing any bugs
- Add Unit tests to test the newly added classes and code
- Met with team to complete sprint 5 document

Grace Turkington & Role (2):

- Continue working on user stories
- Commit my finished changes to Git
- Continue writing unit tests
- Watch this weeks youtube videos
- Met with team to complete sprint 5 document

Brandon Smylie & Role (3):

- Finish off containers for good
- Commit any finished changes to Git
- Continue writing unit tests
- Meet with team to complete sprint 5 document
- Submit said group document to Phil

Conor Clarke & Role (4):

- Meet with team to complete sprint 5 document
- Continue work on overworld

Jack McNaughton & Role (5):

- Watch this week's YouTube videos
- Finish tutorial.
- Meet team for sprint 5 document
- Add new user stories

# Meeting 7: Sprint 6, Week 1

Week commencing: 19/02/2018

Date of this minute: 19/02/2018

## Task Reporting

Edward Muldrew & Role: Programmer

- Met up with team on Friday to finalize sprint 5 report and discuss User Story progress
- Add Unit tests to test the various new code that has been implemented
- Added in working pop up, pop down and place cards method
- Added User Prompt to check User is sure they wish to pass their go
- Added text to allow the User to understand the difference between the User and the AI screen

Grace Turkington & Role (2):

- Met up with team on Friday to finalize sprint 5 report and discuss User Story progress
- Worked on user stories 11 & 13 (Did not get these finished)
- Watched this weeks youtube videos
- Added & committed unit tests he various new code that has been implemented

Brandon Smylie & Role (3):

- Met with team to create Sprint 5 report
- Containers were finished
- Report was emailed to Phil

Conor Clarke & Role (4):

- Met with team to create Sprint 5 report
- Try to implement an interaction method between player and AI sprites

Jack McNaughton & Role (5):

- Met with team for sprint 5 document.
- Watched Youtube videos for the week
- Made the tutorial transition between draw methods using a button.

## Actions Planned

Edward Muldrew & Role: Programmer

- Meet with team to discuss User Stories and create User Story document
- Start work on User Stories
- Add turn based mechanism & Round ending screen

Grace Turkington & Role (2):

- Meet with team to assign sprint 6 user stories
- Add a to-do list to the project page of my plan for an assigned user story
- Begin working on my assigned user stories
- Commit my finished changes to Git
- Watch this weeks youtube videos

Brandon Smylie & Role (3):

- Meet with team to create and assign sprint 6 user stories
- Begin working on my assigned user stories
- Commit my finished changes to Git
- Submit user stories to Phil

Conor Clarke & Role (4):

- Meet with team to create and assign sprint 6 user stories
- Begin working on my assigned user stories

Jack McNaughton & Role (5):

- Meet with team to create user stories for sprint 6.
- Begin work on user stories.
- Watch YouTube videos of the week.
- Commit any significant changes to git.

# Meeting 8: Sprint 6, Week 2

Week commencing: 26/02/2018

Date of this minute: 26/02/2018

## Task Reporting

Edward Muldrew & Role: Programmer

- Met up with team on Friday to discuss User Story progress and completed User Story document
- Completed Turn Based mechanism
- Completed creating a round ending screen
- Started work on round based mechanism
- Cleaned up code
- Committed code to git

Grace Turkington & Role (2):

- Meet with team to assign sprint 6 user stories
- Add a to-do list to the project page of my plan for an assigned user story
- Begin working on my assigned user stories
- Commit my finished changes to Git
- Watch this weeks youtube videos

Brandon Smylie & Role (3):

- Met with team to assign sprint 6 user stories
- Began working on my assigned user stories
- Commit any finished changes to Git
- Submitted document to Phil

Conor Clarke & Role (4):

- Met with team to assign sprint 6 user stories
- Began working on my assigned user stories

Jack McNaughton & Role (5):

- Watched week's YouTube videos
- Met with team to discuss User stories
- Began working on assigned user stories.

## Actions Planned

Edward Muldrew & Role: Programmer

- Meet up with team during the week to discuss User Story progress
- Start working on some bugs in the code (i.e when user plays all their cards, only allow the ai to play)
- Start work on Spell cards
- Finish round-based mechanism and final screen to show who won the game.
- Commit code to GIT

Grace Turkington & Role (2):

- Meet with team to discuss progress on sprint 6 user stories
- Continue working on user stories
- Commit some changes to Git
- Watch this weeks youtube videos

Brandon Smylie & Role (3):

- Meet with team to discuss progress
- Continue working my assigned user stories
- Commit some changes to Git
- Start researching AI.

Conor Clarke & Role (4):

- Meet with team to discuss progress
- Continue working my assigned user stories
- Look into AI development

Jack McNaughton & Role (5):

- Meet with team to discuss progress on sprint 6 user stories
- Continue working on user stories
- Watch this week's YouTube videos
- Attempt to get screenshots of actual game to improve tutorial

# Meeting 9: Sprint 6, Week 3

Week commencing: 05/03/2018

Date of this minute: 05/03/2018

## Task Reporting

Edward Muldrew & Role: Programmer

- Completed round based and game based mechanism
- Game ending screen will appear once a winner has been achieved
- Committed code to git

Grace Turkington & Role (2):

- Met with team to discuss progress on sprint 6 user stories
- Continued working on user stories
- Committed some changes to Git
- Watched this weeks youtube videos

Brandon Smylie & Role (3):

- Met with team to discuss progress on sprint 6 user stories
- Watched multiple videos on AI development

Conor Clarke & Role (4):

- Met with team to discuss progress on sprint 6 user stories
- Went over lecture slides to get an idea of the right way to make an AI

Jack McNaughton & Role (5):

- Met with team to discuss progress on sprint 6 user stories
- Continued working on user stories
- Watched this week's YouTube videos
- Tested game to see if there were any bugs.

## Actions Planned

Edward Muldrew & Role: Programmer

- Meet up with the team during the week to complete the sprint submission document
- Work on testing the added code
- Ensure game ending and round ending code work efficently
- Start working on Spell Cards
- Commit code to GIT

Grace Turkington & Role (2):

- Meet with team to discuss complete sprint 6 submission document
- Continue working on user stories
- Commit some changes to Git
- Write unit tests
- Watch this weeks youtube videos

Brandon Smylie & Role (3):

- Meet with team to discuss and fill out sprint 6 report
- Refactor some code
- Commit some changes to Git
- Write unit tests for existing classes

Conor Clarke & Role (4):

- Meet with team to discuss and fill out sprint 6 report
- Put finishing touches on overworld screen

Jack McNaughton & Role (5):

- Meet with team to discuss complete sprint 6 submission document
- Continue working on user stories
- Watch this week's YouTube videos
- Attempted to get some screen shots from the game.

# Meeting 9: Sprint 7, Week 1

Week commencing: 12/03/2018

Date of this minute: 12/03/2018

## Task Reporting

Edward Muldrew & Role: Programmer

- Met up with team on Friday to discuss Sprint Progress and software engineering assignment
- Cleaned up some code
- Added tests and new test classes to test the newly added code
- Committed code to GIT

Grace Turkington & Role (2):

- Met with team to discuss progress on sprint 6 user stories
- Continued working on user stories
- Committed some changes to Git
- Watched this weeks youtube videos

Brandon Smylie & Role (3):

- Met with team to discuss progress on sprint 6 user stories and software engineering document
- Committed some changes to Git

Conor Clarke & Role (4):

- Due to the random nature of the hands and decks it is very difficult to make smart AI
- Met with team to discuss progress on sprint 6 user stories and software engineering document

Jack McNaughton& Role (5):

- Met with team to discuss complete sprint 6 submission document
- Continued working on user stories
- Watched this week's YouTube videos
- Rewrite tutorial to account for changes in game.

## Actions Planned

Edward Muldrew & Role: Programmer

- Meet up with team on Tuesday to discuss the final core pieces of functionality that need uploaded
- Work on spell cards
- Tidy up code and refactor
- Improve on existing code & methods
- Continue work on Software Engineering assignment

Grace Turkington & Role (2):

- Meet with team to discuss complete sprint 6 submission document
- Continue working on user stories
- Commit some changes to Git
- Write unit tests
- Watch this weeks youtube videos

Brandon Smylie & Role (3):

- Meet with team to discuss final functionality
- Work on AI script wrapper
- Commit some changes to Git

Conor Clarke & Role (4):

- Create simple AI that plays cards
- Meet with team to discuss final functionality

Jack McNaughton& Role (5):

- Meet with team to discuss complete sprint 6 submission document
- Continue working on user stories
- Watch this week's YouTube videos
- Committed changes to git

# Meeting 10: Sprint 7, Week 2

Week commencing: 12/03/2018

Date of this minute: 12/03/2018

## Task Reporting

Edward Muldrew & Role: Programmer

- Added working spell cards
- Added exit button
- Added tweaks to placement of cards and collision detection
- Added bug fixes
- Met up with the team on Tuesday to discuss progress

Grace Turkington & Role (2):

- Met with team to discuss progress and work on software engineering assignment
- Refactored code
- Edited unit tests
- Watch this weeks youtube videos

Brandon Smylie & Role (3):

Conor Clarke & Role (4):

- Met with team to discuss progress and work on software engineering assignment
- Assign roles for the assignment

Jack McNaughton& Role (5):

## Actions Planned

Edward Muldrew & Role: Programmer

- Complete any last bug fixes/tweaks to code
- Meet with team to complete project submission document
- Continue work on Software Engineering assignment
- Add relevant tests to newly added code
- Clean up code

Grace Turkington & Role (2):

- Meet with team to finalise software engineering assignment
- Keep refactoring code & testing for bugs
- Write more unit tests

Brandon Smylie & Role (3):

Conor Clarke & Role (4):

- Meet with team to finalise software engineering assignment
- Add collision detection to the player sprites

Jack McNaughton& Role (5):

# Peer Assessment

| **Evaluation** | Group Number: 57 | Group Name: Black Box Studios | | |
|---|---|---|---|---|
| Name | Contribution of time and effort[1] | Contribution to team-working and motivation[1] | Contributions to this deliverable [1,2] | Peer Score (Range 85 – 115) |
| Conor Clarke | 3 | 3 | 3 | 100 |
| Jack McNaughton | 2 | 3 | 2 | 95 |
| Edward Muldrew | 5 | 5 | 5 | 115 |
| Brandon Smylie | 5 | 4 | 4 | 105 |
| Grace Turkington | 4 | 5 | 4 | 105 |

[1]Values: 1 = Less than average; 2 = Slightly less than average; 3 = Average; 4 = Slightly more than average; 5 = More than average

[2]This value should consider contributions in the round – direct contributions to required deliverables, and contributions that have made the deliverables possible.

**Declaration**

"I declare that I have read the Queen's University regulations on plagiarism, and that any contribution I have made to the attached submission is my own original work, except for any elements that I have clearly attributed to third parties. I understand that this submission will be subject to an electronic test for plagiarism and will also be subject to the University's regulations concerning late submission if it is received after the deadline."

| Name | Date | Confirmation *(use the words shown in the example below!)* |
|---|---|---|
| Conor Clarke | 23/03/18 | I agree to the terms of the declaration |
| Jack McNaughton | 23/03/18 | I agree to the terms of the declaration |
| Edward Muldrew | 23/03/18 | I agree to the terms of the declaration |
| Brandon Smylie | 23/03/18 | I agree to the terms of the declaration |
| Grace Turkington | 23/03/18 | I agree to the terms of the declaration |