*Florida International University*

*School of Computing and Information Sciences*

Software Engineering Focus

Final Deliverable

SkillCourt v10.0

**Team Members:** Carlos Alva, Sandra Hurtado, Sergio Rosales

**Product Owner(s)**: Guðmundur Traustason

**Mentor(s)**: Joshua Mclendon

**Instructor**: Masoud Sadjadi

The MIT License (MIT)

Copyright (c) *2016 Florida International University*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

***Abstract***

Skillcourt is an application that allows soccer fanatics around the world to develop their skills and become as good as the soccer players they idolize whenever and wherever. SkillCourt is the application trainers would love to have access to. It allows the user to personalize soccer games and simulate them as how playing out in the field would be like. The application has a user-friendly design that allows users to easily get setup with a game of their liking, there are a few game mode which they are allowed to choose from and at the end of each game, a summary of the total points accumulated over the game is given, giving the user an insight of how their skills out in the field are.

Table of Contents

[Introduction 5](#_Toc511737025)

[Current System 5](#_Toc511737026)

[Purpose of New System 5](#_Toc511737027)

[User Stories 6](#_Toc511737028)

[Implemented User Stories 6](#_Toc511737029)

[Pending User Stories 14](#_Toc511737030)

[Project Plan 15](#_Toc511737031)

[Hardware and Software Resources 15](#_Toc511737032)

[Sprints Plan 16](#_Toc511737033)

[System Design 19](#_Toc511737034)

[Architectural Patterns 19](#_Toc511737035)

[System and Subsystem Decomposition 19](#_Toc511737036)

[System Validation 21](#_Toc511737037)

[Glossary 24](#_Toc511737038)

[Appendix 25](#_Toc511737039)

[Appendix A - UML Diagrams 25](#_Toc511737040)

[25](#_Toc511737041)

[31](#_Toc511737042)

[32](#_Toc511737043)

[36](#_Toc511737044)

[38](#_Toc511737045)

[39](#_Toc511737046)

[44](#_Toc511737047)

[44](#_Toc511737048)

[Appendix B - User Interface Design 45](#_Toc511737049)

[Appendix C - Sprint Review Reports 46](#_Toc511737050)

[**Sprint 2** 46](#_Toc511737051)

[**Sprint 3** 46](#_Toc511737052)

[**Sprint 4** 47](#_Toc511737053)

[**Sprint 5** 47](#_Toc511737054)

[**Sprint 6** 48](#_Toc511737055)

[Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents 49](#_Toc511737056)

[References 49](#_Toc511737057)

# **Introduction**

Soccer is a worldwide well recognized sport that all ages love to play. Not only is it fun but it also tests your agility in running, reflection, and aim. SkillCourt is an application that allows anyone to train like a professional at the comfort of their own home.

## 

## **Current System**

Currently, the system contained a single player game that allowed the user to play a certain amount of time in a random sequence and calculated hits at the end of the game. User interface has very few selections for the user to choose from.

## **Purpose of New System**

The new system shall allow the users to have more control over the application and their personal training. User will be allowed to have a multiplayer mode that lets them compete with their peers. Instead of only allowing the user to have a random sequence, a new mode will be implemented called manual mode. Manual mode allows user to preconfigure the sequence and decide which pads will light up at a given order allowing them to challenge the person playing the game. Also the interface will be given a more appealing and user friendly look to it.

# **User Stories**

The following section provides the detailed user stories that were implemented in this iteration of the SkillCourt 10.0 project. These user stories served as the basis for the implementation of the project’s features. This section also shows the user stories that are to be considered for future development.

## **Implemented User Stories**

#### User Story Name: Change UI from activities to fragments

* Description: As a Developer I would like to use fragment layouts **so that** the application costs less memory space and appears smoother during layout transitions.

Acceptance Criteria:

* Access to all tabs and game pages.
* Services bound to MainActivity.
* Behavior of program should remain the same.

Use Case

* Name: Refactor code from Activities to Fragments
* Actor: User
* Preconditions: Application launches and starts MainActivity (Navigation Page) which contains an empty FrameLayout.
* Description: User taps on any button/tab while in Navigation Page, then application will switch to that button/tab’s respective content. However, only the contents of the Navigation Page should be replaced, instead of starting a new application window (Activity).

#### **User Story Name:** Design architecture to identify Pads

* Description: As a Developer I would like design the software architecture so that I can identify a pad that can later be used on the game.

Acceptance Criteria

* Creation of class “Pads”
* Skeleton class creation

Use Case

* Name: Configure Pad order
* Actor: User
* Preconditions: User just opened the app
* Description:   
  1. User clicks on pad and sees it light up  
  2. Long Click Hold on the screen to enter Edit mode  
  3. Drag Pads to the order desired  
  4. Long Click Hold to exit Edit mode

#### User Story Name: Design Architecture for Sequence

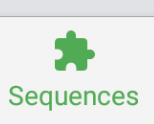
* Description: As a developer, I must create skeleton class for a Player so it is easier to create different sequence mode.

Acceptance Criteria:

* Contains information of Pads that correspond to the player
* Pads should light up in the sequence assigned by user.

Use Case

* Name: Design Sequence
* Actor: User
* Preconditions: Must have pads identified and application open
* Description:

1. Select button
2. Different options of sequences will display
3. Select the sequence of your preference
4. Show the newly created sequence to the player

#### User Story Name: Implement Player Class

* Description: As a Developer I would like to implement Player class so that future game modes can be implemented.

Acceptance Criteria:

* Create Player class
* Integrate Player class to current functionality.
* Behavior of program should remain the same.

Use Case

* Name: Implement Player Class
* Actor: User
* Preconditions: Must have pads identified and user must be in Create Game page.
* Description: User selects the number of players and assigns each player a set of Pads. User then starts game. After game ends, the application will show the stats of each Player in separate views via dropdown menu.

#### **User Story Name:** Implement and Refactor Pad Class

* Description: As a Developer I should refactor and implement a Pad class so that I can identify a pad that can later be used on the game.

Acceptance Criteria

* Creation of class “Pads”
* Refactoring of Pad class
* Implementation of new methods

Use Case

* Name: Start a game
* Actor: User
* Preconditions: User is on the Home screen
* Description:   
    
  1. User Clicks “Play Game”

2. Selects Time to play

3. Play Game

4. When time runs out then game is finished and shows Score

#### User Story Name: Implement and Refactor Sequence Class

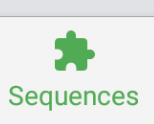
* Description: As a developer, I should implement Sequence class so that the user can have more variety of sequence modes to choose from.

Acceptance Criteria:

* Sequence class should have random sequence mode implemented.
* Refactor current use of sequence in GameService.
* Allow future sequence methods to be added through methods.

Use Case

* Name: Implement and Refactor Sequence Class
* Actor: User
* Preconditions: Must have pads identified and application open
* Description:

1. Select button
2. Different options of sequences will display
3. Select random sequence mode
4. Pads light up in a random sequence

#### **User Story Name:** Implement sending Hex Html color codes

* As a Developer I should send HTML hex color codes to the pads so that Users can differentiate different states of the pads using color feedback

Acceptance Criteria

* Pad should send message using HTML hex code
* Should keep other commands as reserved strings
* Should receive html hex codes
* Pad should light up with the Hex color

Use Case

* Name: Play game
* Actor: User
* Preconditions: User is on the Play game screen and has selected game mode and time
* Description:   
    
  1. User Clicks “Play Game”

2. Starts playing game

3. When time runs out then game is finished and shows Score

#### User Story Name: Refactor UI Design for user to play game

* Description: As a developer, I need to update UI to allow user to see the new features.

Acceptance Criteria:

* Should add a multiplayer option
* Should prompt user to pick number of players in game
* Add frame showing different sequence options

Use Case

* Name: Refactor UI Design for User to Play Game
* Actor: User
* Preconditions: Must have pads identified and application open, and have number of player predetermined.
* Description:

1. Select  button
2. Select MultiPlayer
3. Choose number of player (max = 4)

#### User Story Name: Implement Multiplayer Game Template

* Description: As a User I would like to play the game with multiple players so that the pads light up in different colors or multiple games run simultaneously.

Acceptance Criteria:

* Create method skeletons for creating multiplayer game
* Should keep stats inside Player
* Should maintain functionality

Use Case

* Name: Implement Multiplayer Game Template
* Actor: User
* Preconditions: User must be in the Create Game Page
* Description: User sets the number of players to be more than one. The application will then prompt user to choose between two multiplayer modes. The user’s choice will notify the application on how the sequences should be read and define the set(s) of pads for the players.

#### **User Story Name:** Refactor PadConnection as a Service Class

* As a USER, I want my pads to connect faster when starting the app and not disconnect when screen locks or App is minimized, so that I can have a better user experience

Acceptance Criteria

* Maintain current functionality
* Pads should not disconnect when Screen is locked or App is minimized
* Remove redundant and duplicate code
* Create a new file for Enum and Pad commands
* Test refactored code with current Hardware version

Use Case

* Name: Start App and connect Pads
* Actor: User
* Preconditions: Physical Pads are turned on User has a hotspot in his phone with the name: skillcourt and password: skillcourt
* Description:

1. User opens the app
2. Pads should flash when connected to app
3. Symbol on the top right should display the number of pads currently connected

#### User Story Name: Fix "Play" floating button bug

* Description: As a User I would like to see the play button only on the home page so that the button is not in my way when viewing stats or setting sequences.

Acceptance Criteria:

* Maintain functionality
* Remove floating button when not in Home page

Use Case

* Name: Fix "Play" Floating Button Bug
* Actor: User
* Preconditions: User must be in the Home Page
* Description: User switches tabs or selects a page link. Upon leaving the Home Page the “Play’ floating button’s visibility is set to false. When directed back to Home Page, the button’s visibility is set to true.

#### User Story Name: Customize Pad light up time

* Description: As a developer, I want to give user (soccer coach) full control of the pads .

Acceptance Criteria:

* Make pads light up to the time user decides to
* Allow user to choose the time between pad light up
* Add frame showing different sequence options

Use Case

* Name:Customize Pad light up time
* Actor: User
* Preconditions: Must have pads identified and application open, and have number of player predetermined.
* Description:

1. Select  button
2. Choose Sequence mode
3. Select manual mode
4. Navigate to pad configuration
5. Choose the amount of time to light up and transition time.

#### User Story Name: Modify UI for Play Game

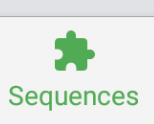
* Description: As a developer, I want to change Game View to easily show me game modes and options, so that choosing a game be intuitive and simple.

Acceptance Criteria:

* Play Game Screen should display time to play
* Play Game Screen should let you play Random Mode or Sequence Mode
* Should work with current Hardware version

Use Case

* Name:Customize Modify UI for Play Game
* Actor: User (coach)
* Preconditions: Must have pads identified and application open, and have number of player predetermined.
* Description:

1. Select  button
2. Choose Sequence mode
3. Select mode wanted

#### User Story Name: Implement Multiplayer Game Over Template

* Description: As a User I would like Multiplayer mode to show players’ respective scores on the Game Over screen so thatI can distinguish each player’s scores.

Acceptance Criteria:

* Should display different scores for each player
* Should work with Emulator
* Should work with current Hardware Version

Use Case

* Name: Implement Multiplayer Game Over Template
* Actor: User
* Preconditions: User must have finished a multiplayer game and be in the Game Over Page
* Description: User starts a multiplayer game. After the game ends, the Game Over Page should display all players’ respective scores.

#### **User Story Name:** Modify Pad Configuration UI to match Models

* As a USER, I want the Pad Configuration screen to show current connected Pads , so that the USER can give pads an individual order.

Acceptance Criteria

* Back End needs to reflect current UI changes
* Pad order should be stored on the Pad class
* Should play random/recorded sequence using these order.
* Should work with current Hardware version

Use Case

* Name: Pad Configuration Screen
* Actor: User
* Preconditions: At least 1 Pad is connected
* Description:

1. User opens the app
2. User click “Pad connected” square at top right corner

## **Pending User Stories**

# **Project Plan**

This section describes the planning that went into the realization of this project. This project incorporated the agile development techniques and as such required the sprints to be planned. These sprint plannings are detailed in the section. This section also describes the components, both software and hardware, chosen for this project.

## **Hardware and Software Resources**

The following is a list of all hardware and software resources that were used in this project:

* + Android Studio 3.0.1
  + Raspberry Pi 3 Model B
  + MEAS (Piezo vibration sensor)
  + Samsung Galaxy Note 4
  + Samsung Galaxy Tab S3
  + WS2812B LED strips

## 

## **Sprints Plan**

#### Sprint 2

Attendees: Carlos Alva, Sandra Hurtado, Sergio Rosales, Gummi, Michael

Date: 01/29/2018

Start time: 7:30 PM

End time: 7:50 PM

After discussion, the velocity of the team were estimated to be an average of 2 weeks

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

* User Story ID: SC-1137 - Change UI from Activity to Fragments
* User Story ID: SC-1138 - Modify the code to fit the new architectural design for identifying pads
* User Story ID: SC-1139 - Modify the code to fit the new architectural design to play a sequence of instructions

The team members indicated their willingness to work on the following user stories.

* Carlos Alva
  + User Story ID: SC-1138
* Sandra Hurtado
  + User Story ID: SC-1139
* Sergio Rosales
  + User Story ID: SC-1137

#### **Sprint 3**

Attendees: Carlos Alva, Sandra Hurtado, Sergio Rosales, Gummi, Michael

Date: 02/12/2018

Start time: 7:30 PM

End time: 7:50 PM

After discussion, the velocity of the team were estimated to be an average of 2 weeks

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

* User Story ID: SC-1145 - Implement Player class
* User Story ID: SC-1146 - Implement and Refactor Pad class
* User Story ID: SC-1147 - Implement and Refactor Sequence class

The team members indicated their willingness to work on the following user stories.

* Carlos Alva
  + User Story ID: SC-1146
* Sandra Hurtado
  + User Story ID: SC-1147
* Sergio Rosales
  + User Story ID: SC-1145

#### **Sprint 4**

Attendees: Carlos Alva, Sandra Hurtado, Sergio Rosales, Gummi, Michael

Date: 02/26/2018

Start time: 6:00 PM

End time: 7:30 PM

After discussion, the velocity of the team were estimated to be an average of 2 weeks

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

* User Story ID: SC-1156 - Implement sending Hex Html color codes
* User Story ID: SC-1157 - Refactor UI design for User to Play Game
* User Story ID: SC-1158 - Implement Multiplayer Game Service

The team members indicated their willingness to work on the following user stories.

* Carlos Alva
  + User Story ID: SC-1156
* Sandra Hurtado
  + User Story ID: SC-1157
* Sergio Rosales
  + User Story ID: SC-1158

#### **Sprint 5**

Attendees: Carlos Alva, Sandra Hurtado, Sergio Rosales, Gummi, Michael

Date: 03/09/2018

Start time: 6:00 PM

End time: 7:30 PM

After discussion, the velocity of the team were estimated to be an average of 2 weeks

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

* User Story ID: SC-1159 - Refactor PadConnectionService Class
* User Story ID: SC-1160 - Fix "Play" floating button bug
* User Story ID: SC-1161 - Customize Pad light up time

The team members indicated their willingness to work on the following user stories.

* Carlos Alva
  + User Story ID: SC-1159
* Sandra Hurtado
  + User Story ID: SC-1161
* Sergio Rosales
  + User Story ID: SC-1160

#### **Sprint 6**

Attendees: Carlos Alva, Sandra Hurtado, Sergio Rosales, Gummi, Michael

Date: 03/30/2018

Start time: 6:00 PM

End time: 7:30 PM

After discussion, the velocity of the team were estimated to be an average of 2 weeks

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

* User Story ID: SC-1168 - Change Pad Configuration UI to match Models
* User Story ID: SC-1169 - Modify UI for Play Game
* User Story ID: SC-1170 - Modify Multiplayer UI to display current Players Pads

The team members indicated their willingness to work on the following user stories.

* Carlos Alva
  + User Story ID: SC-1168
* Sandra Hurtado
  + User Story ID: SC-1169
* Sergio Rosales
  + User Story ID: SC-1170

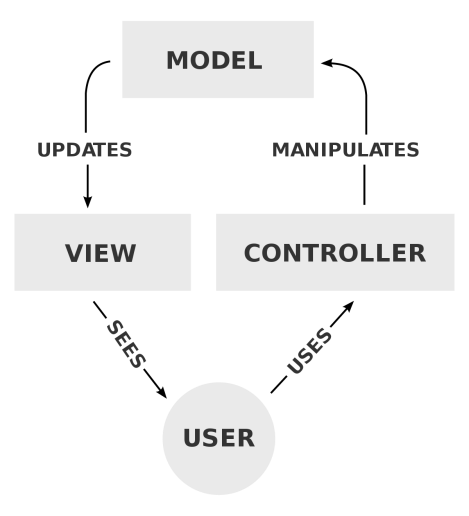
### 

# **System Design**

This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

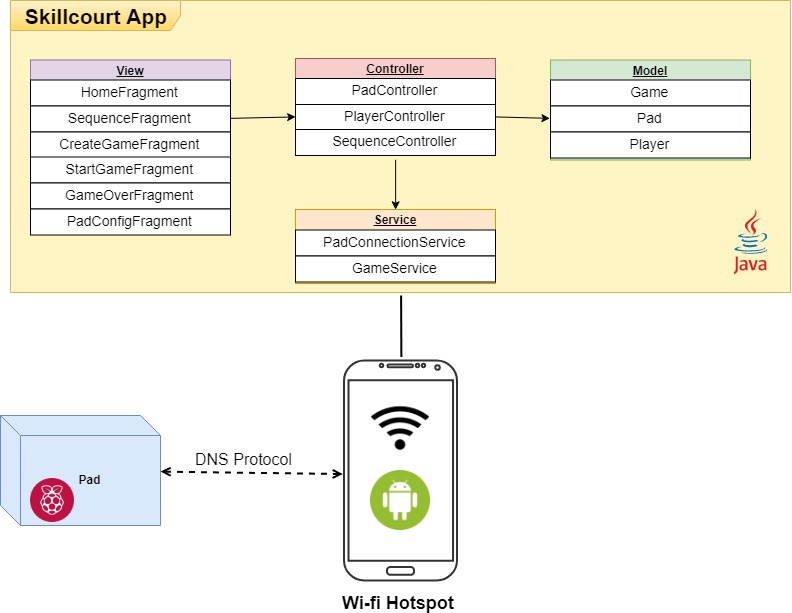
## **Architectural Patterns**

* MVC Model

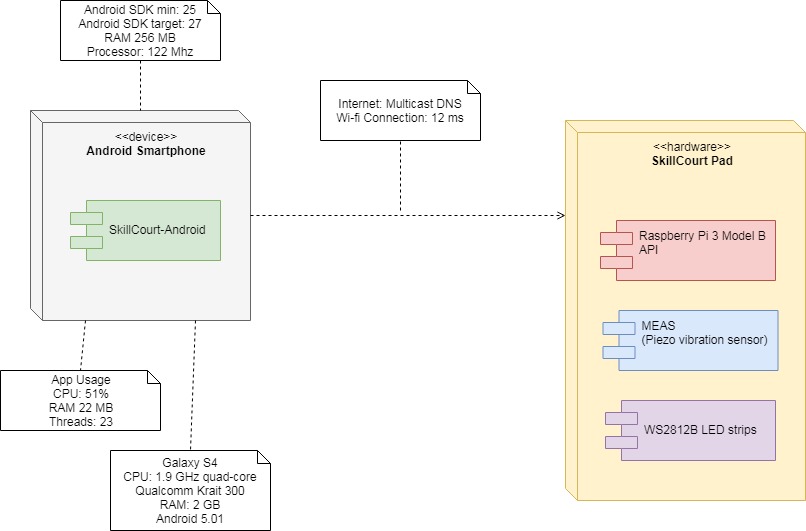


## 

## **System and Subsystem Decomposition**



**Deployment Diagram**



**Design Patterns**

* Singleton: Main Activity, Game and Pad connection service
* Iterators: Pad and Player classes
* Event-based asynchronous: Hardware send Async messages to Connection service
* Thread Pool: Connection service creates different Socket threads to handle multiple communication concurrently
* API: Pad Class provides an interface to communicate with Hardware Pads

# **System Validation**

**Test case ID: UT-1145\_getHitPercentage**

* Description/Summary of Test: Player class must return correct Hit Percentage
* Pre-condition: User must have finished a game with 4 hits out of 10 shots
* Expected Results: Game Over screen should display 40% on the progress bar
* Actual Result: Game Over screen displayed 40% on the progress bar
* Status (Fail/Pass): Pass

**Test case ID: SC-1146\_SendMessageTestString**

* Description/Summary of Test: As a tester I should use sendMessage(“TEST”) method to verify if a test string is received on the socket connection.
* Pre-condition: A socket is created and connected
* Expected Results: string received at socket endpoint equals “TEST”
* Actual Result: string “TEST” was received at endpoint
* Status (Fail/Pass): Pass

**Test case ID: SC-1147 \_SequenceContoller**

* Test Sequence Connection
* Description/Summary of Test: As a tester, I am suppose to make sure the sequence class is getting correct input from controller.
* Pre-condition: Must be signed on to application, need to have at least one pad connected and selected random sequence mode.
* Expected Results: random sequence mode
* Actual Result: random sequence mode
* Status (Fail/Pass): Pass

**Test case ID: SC-1152\_SendHtmlHexColorCode**

* Description/Summary of Test: As a tester I should use turnOn(FF0000) method to verify if a test string is received on the socket connection.
* Pre-condition: A socket is created and connected
* Expected Results: string received at socket endpoint equals “FF0000”
* Actual Result: string “FF0000” was received at endpoint
* Status (Fail/Pass): Pass

**Test case ID: SC-1157\_SequenceConnection**

* Test Sequence Connection
* Description/Summary of Test: As a tester, I am supposed to make sure user interaction with multiplayer is correct.
* Pre-condition: Must be signed on to application, need to have at least 4 pads connected and select multiplayer with 2 players
* Expected Results: shows pads corresponding for 2 players
* Actual Result: 2 players shown with 2 pads each
* Status (Fail/Pass): Pass

**Test case ID: UT-1158\_startGame**

* Description/Summary of Test: startGame() should consider multiplayer mode when creating game.
* Pre-condition: User must be in Create Game page
* Expected Results: startGame() should run the contents of if mGameMode == “Multiplayer”
* Actual Result: startGame() ran the contents of if mGameMode == “Multiplayer”
* Status (Fail/Pass): Pass

**Test case ID: IT-1158\_startGameIntegrationTest**

* Description/Summary of Test: Game Service creates Game threads and all of its components
* Pre-condition: User must be in Create Game page
* Expected Results: A list of Game threads must be created
* Actual Result: A list of Game threads is created
* Status (Fail/Pass): Pass

**Test case ID: SC-1006\_ShouldUseTurnOnEnumWhenSendingMessage**

* Description/Summary of Test: As a tester I should use turnOn() method in Pad class to verify if a call to sendMessage(“LIGHT\_UP”) is done.
* Pre-condition: A socket is created and connected.
* Expected Results: A call to sendMessage() should be made with parameter “LIGHT\_UP”
* Actual Result: A call to sendMessage() with parameter value “LIGHT\_UP” was made.
* Status (Fail/Pass): Pass

**Test case ID: SC-1007\_ShouldNotSendEnumIfSocketIsNotConnected**

* Description/Summary of Test: The method sendMessage() should not write on the OutputStreamWriter if socket is not connected.
* Pre-condition: A socket is created but is not connected.
* Expected Results: No call to OutputStreamWriter.write() should be made.
* Actual Result: A call to sendMessage() with parameter value “LIGHT\_UP” was made but no call to OutputStreamWriter.write() was made.
* Status (Fail/Pass): Pass

**Test case ID: SC1010\_fragmentsSunny**

* Description/Summary of Test: Fragment manager should change the current fragment when a tab (Stats) is clicked.
* Pre-condition: User must be Home Page
* Expected Results: currentFragment is equal to StatsFragment.class
* Actual Result: currentFragment is equal to StatsFragment.class
* Status (Fail/Pass): Pass

**Test case ID: SC1011\_fragmentsRainy**

* Description/Summary of Test: Fragment manager should not change the current fragment when the current tab is clicked.
* Pre-condition: User must be Home Page
* Expected Results: currentFragment is equal to HomeFragment.class
* Actual Result: currentFragment is equal to HomeFragment.class
* Status (Fail/Pass): Pass

**Test case ID: UT-1160\_removePlayButton**

* Description/Summary of Test: “Play” floating button must only be visible on Home Page
* Pre-condition: User must be in Home Page
* Expected Results: “Play ” floating button visibility should be false when NOT in Home Page
* Actual Result: “Play ” floating button visibility is false when NOT in Home Page
* Status (Fail/Pass): Pass

**Test case ID: SC-1161-Sequence**

* Test Sequence Connection
* Description/Summary of Test: As a tester, I am supposed to make sure the pads are lighting on the amount of time assigned
* Pre-condition: Must be signed on to application, need to have at least 1 pads connected, set the time to hold the pad light.
* Expected Results: pad lights up for the given amount time (3 secs)
* Actual Result: pad lit up for 3 secs
* Status (Fail/Pass): Pass

**Test case ID: SC-1169 \_AcceptanceSequence**

* Test Sequence Connection
* Description/Summary of Test: As a tester, I am suppose to make sure the sequence modes are set correctly
* Pre-condition: Must be signed on to application, need to have at least 2 pads connected, select random sequence mode
* Expected Results: pad to run in random sequence mode
* Actual Result: pads lit in random sequence
* Status (Fail/Pass): Pass

# 

# 

# 

# 

# **Glossary**

* SkillCourt: Android application that allows soccer fans to practice their skills at the comfort of their own home.
* Raspberry Pi: Tiny and affordable computer.
* Android studio: Integrated environment for Google’s android system.
* Pad: Engineered pad that user kicks ball to when it lights up.
* Java: object-oriented computer programing language
* Python: high level programming language

# **Appendix**

## **Appendix A - UML Diagrams**

## 

## 

## 

## 

## 

## 

## 

## 

## 

Fig. SC-1137 Use Case Diagram

Change UI from Activity to Fragments

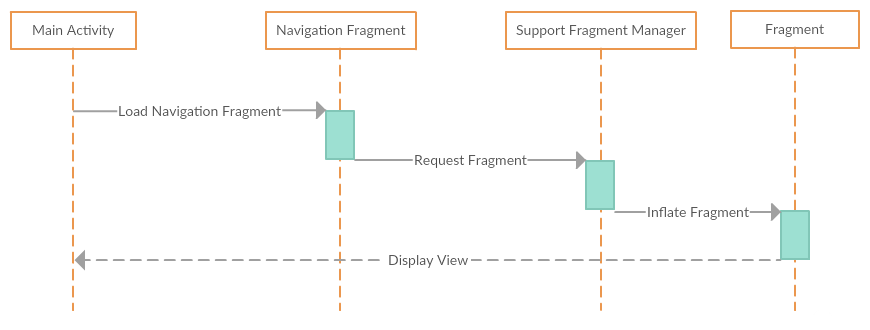


Fig. SC-1137 Sequence Diagram

Change UI from Activity to Fragments

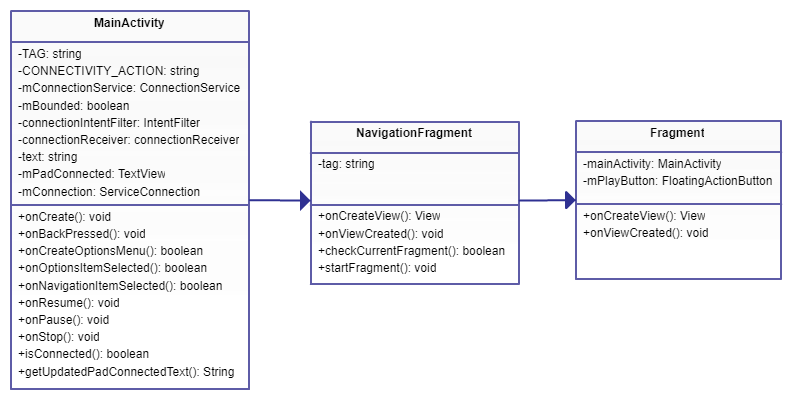


Fig. SC-1137 Class Diagram

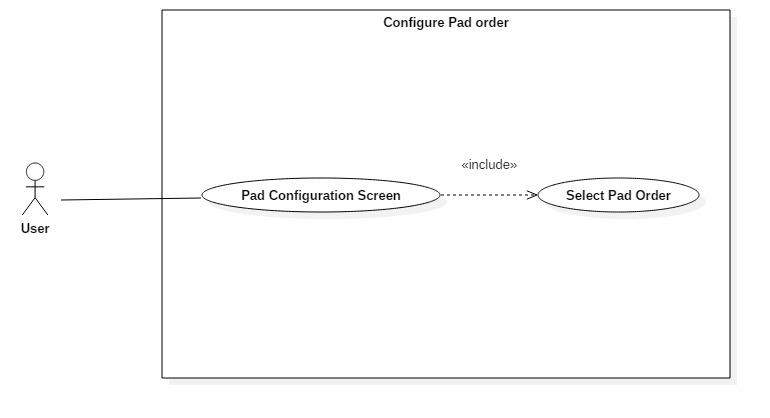


Fig. SC-1138 Use Case Diagram

Design architecture to identify Pads

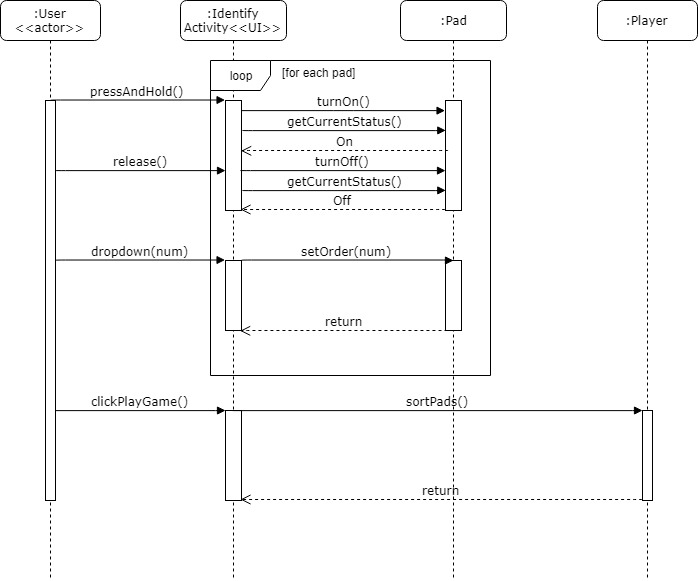


Fig. SC-1138 Sequence Diagram

Design architecture to identify Pads

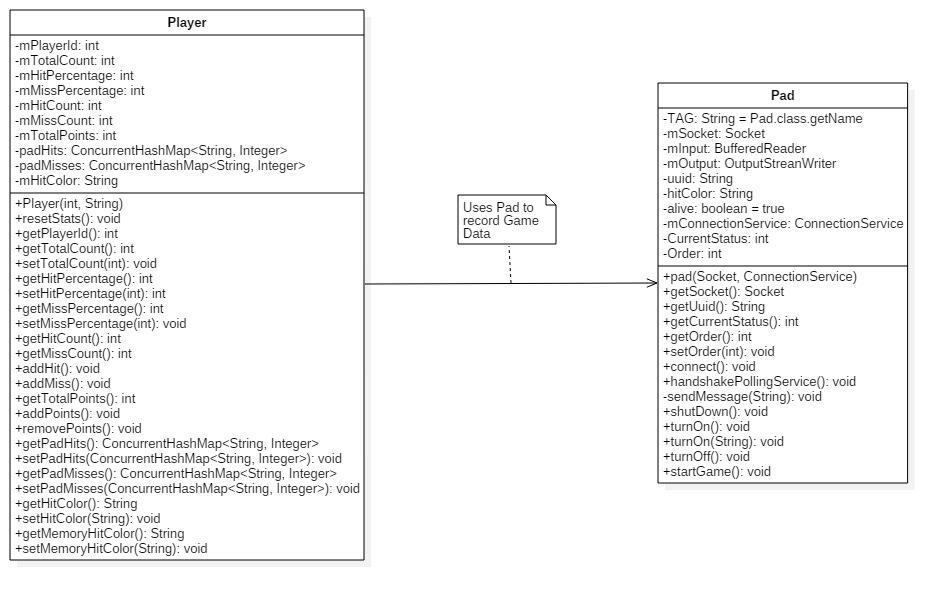
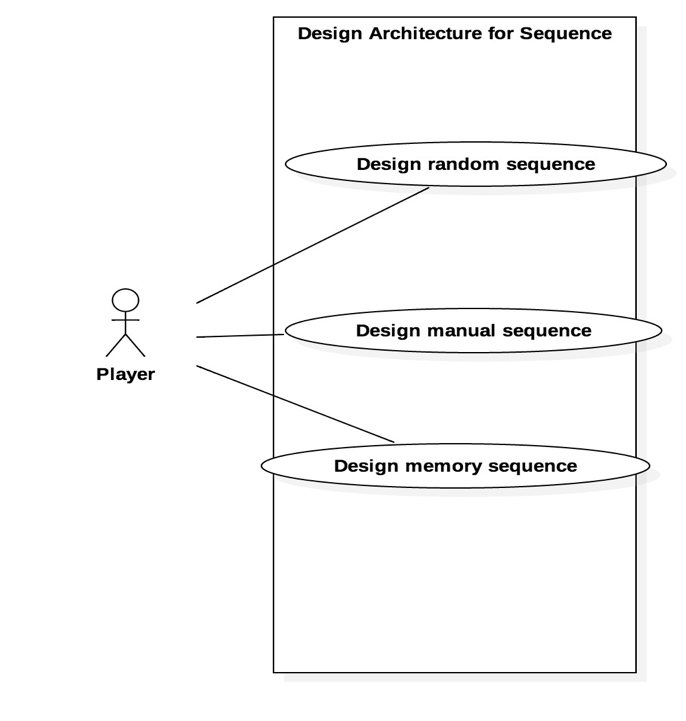


Fig. SC-1138 Class Diagram

Design architecture to identify Pads



## 

Fig. SC-1139 Use Case Diagram

Design Architecture For Sequence

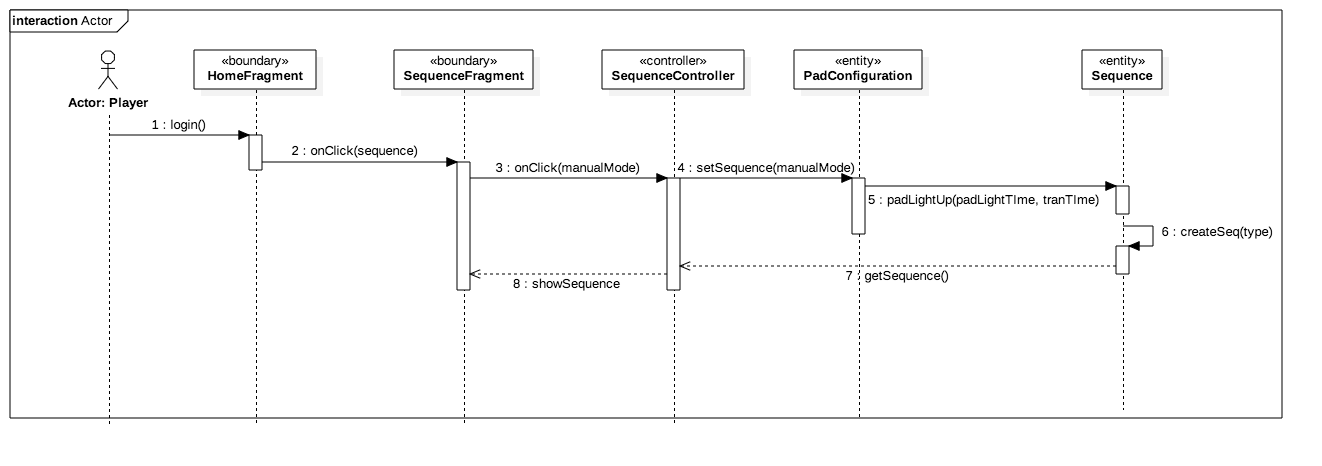


Fig. SC-1139 Sequence Diagram

Design Architecture For Sequence

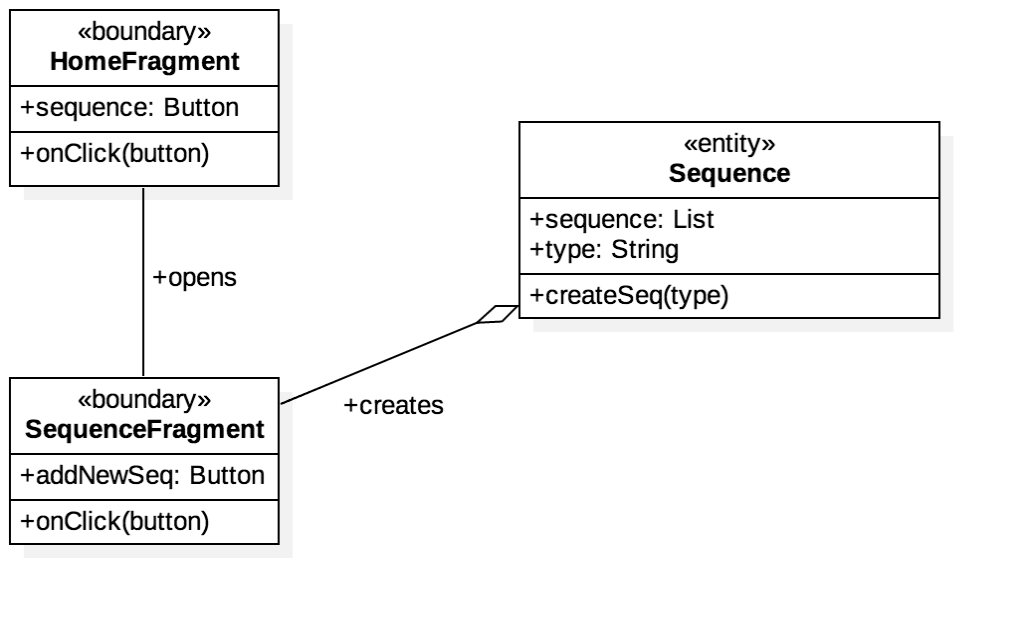


Fig. SC-1139 Class Diagram

Design Architecture For Sequence

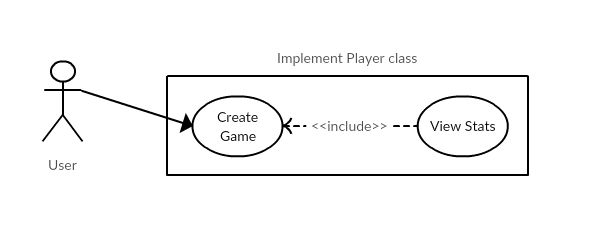
****

Fig. SC-1145 Use Case Diagram

Implement Player Class

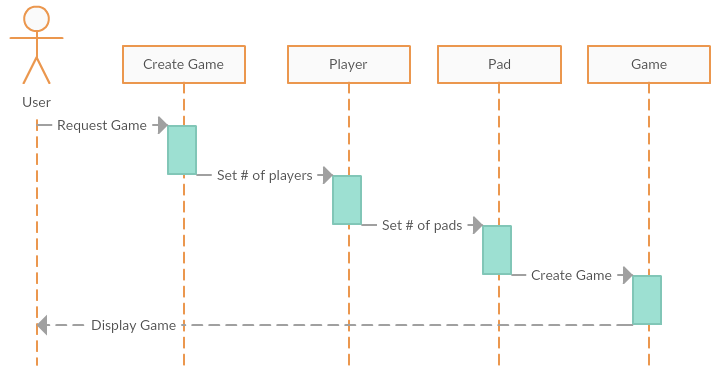
****

Fig. SC-1145 Sequence Diagram

Implement Player Class

****

Fig. SC-1145 Class Diagram

Implement Player Class

## https://lh6.googleusercontent.com/fkFH1KFQR_GojDUmvPFbmDR4irCP1K4i3L_H0Og_2A78bgCnp6qq8RL3f7z0v3pepVVZS-6VT0LRUf0v-LaFh0tvQt3w9kxAdUs1y0OtyJG6wOWAJo0eeoaUHCYubvvr1mFn23FR

Fig. SC-1146 Use Case Diagram

Implement and Refactor Pad Class

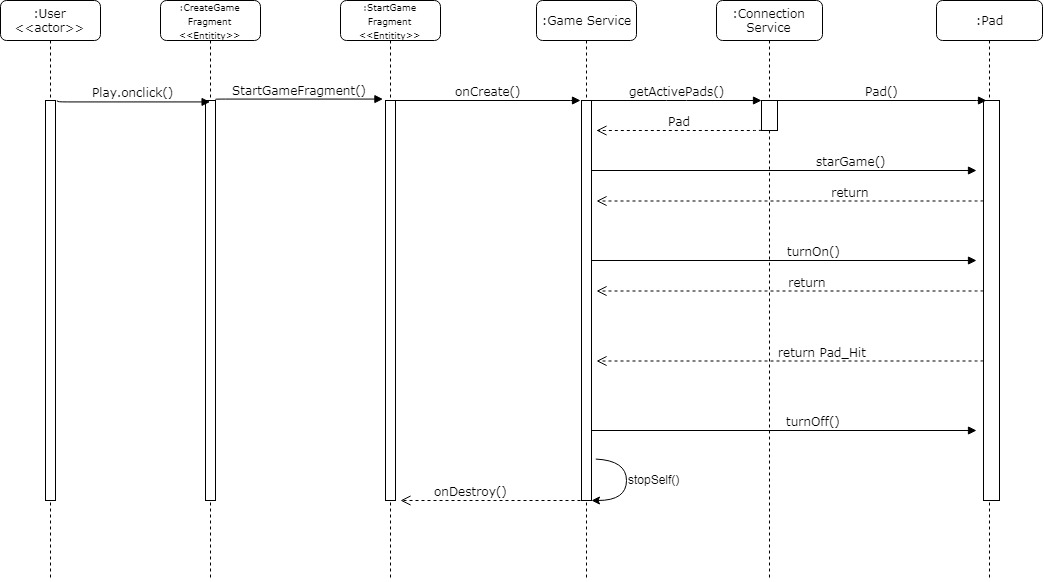


Fig. SC-1146 Sequence Diagram

Implement and Refactor Pad Class

Fig. SC-1146 Class Diagram

Implement and Refactor Pad Class

## https://lh4.googleusercontent.com/qb7NMxxaf7rSHpepFAt0jbACWGhuU8_cj0cdWYl5b8I3yPoo-LxFDSRh3P9mPGjJkIOKIPuo965-u5O1vsooIY26SUEXgdDxof_7uy0ClkE58uUSUcnG3Nq0qonCW6uPIKy87tLI

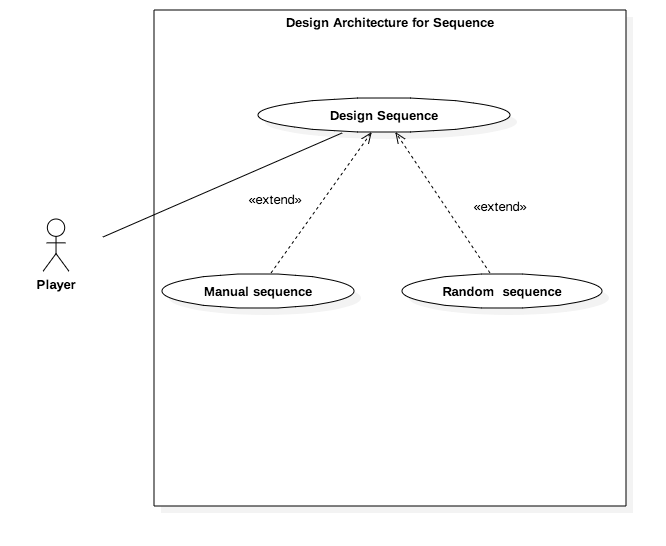


Fig. SC-1147 Use Case Diagram

Implement and Refactor Sequence Class

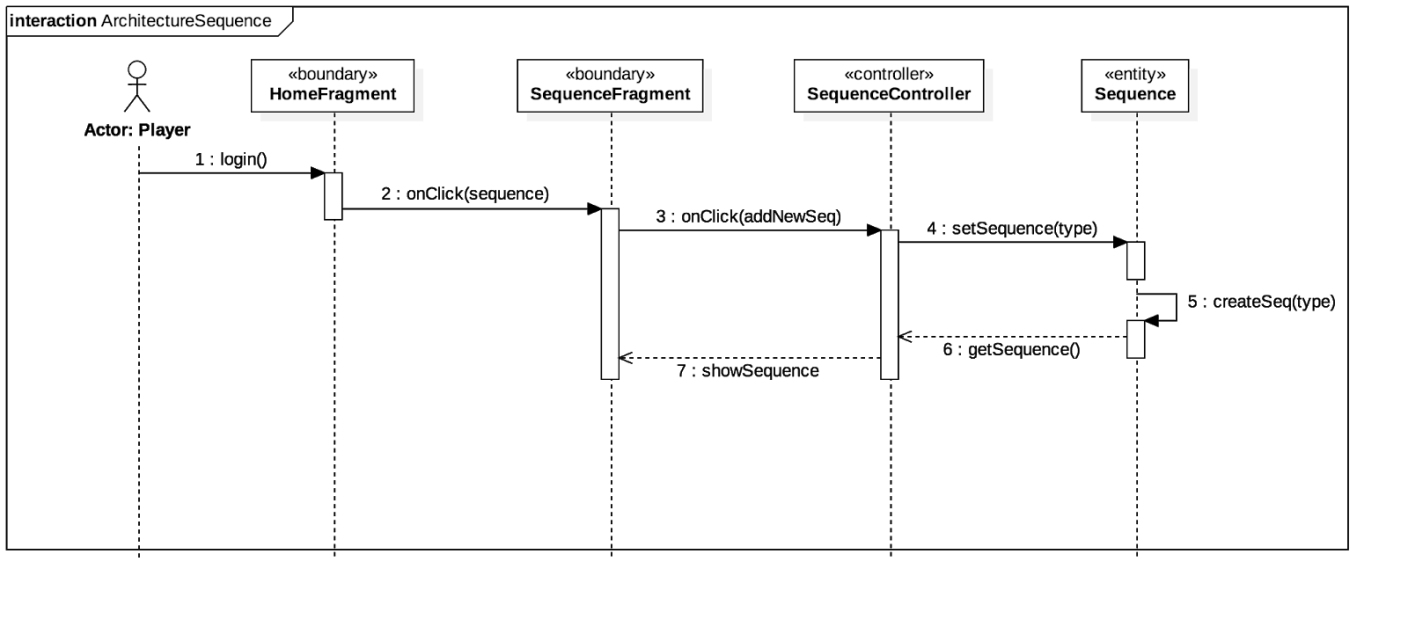


Fig. SC-1147 Sequence Diagram

Implement and Refactor Sequence Class

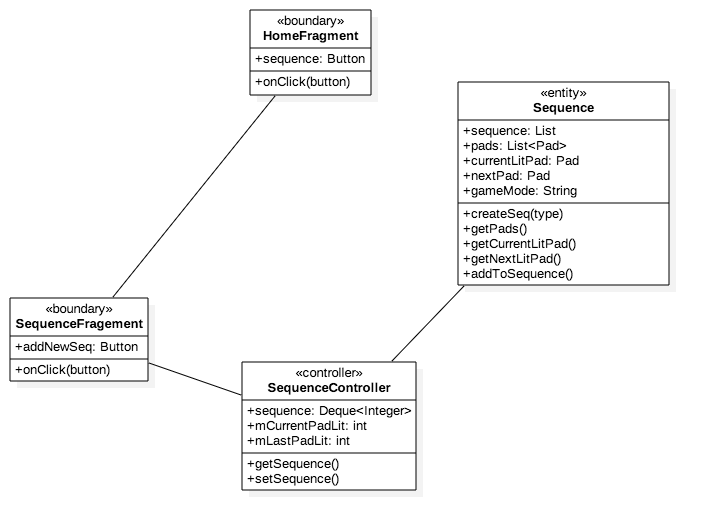


Fig. SC-1147 Class Diagram

Implement and Refactor Sequence Class

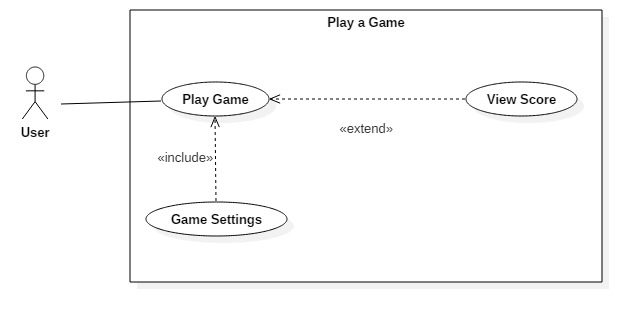


Fig. SC-1156 Use Case Diagram

Implement sending Hex html color codes

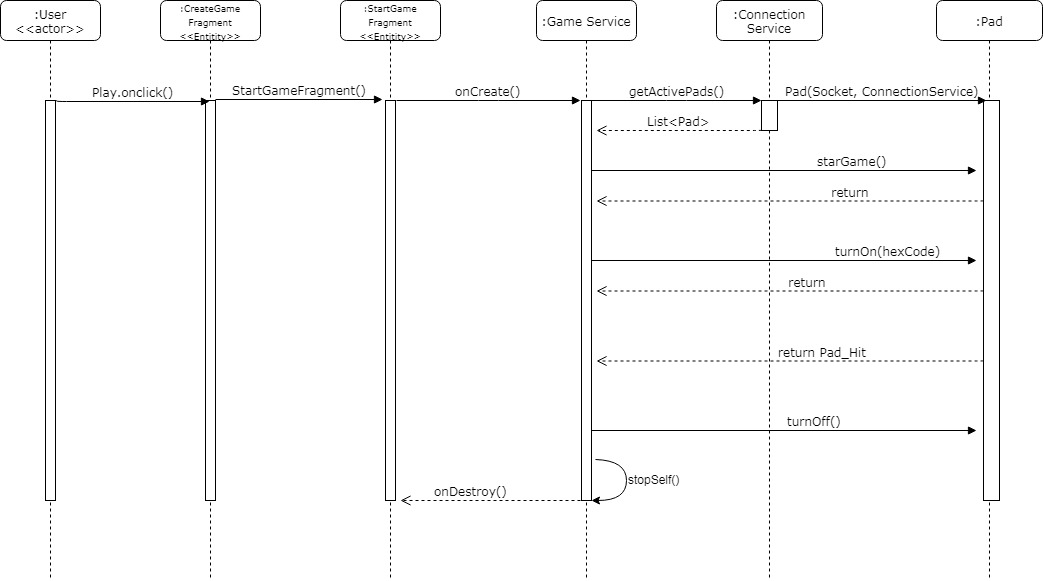


Fig. SC-1156 Sequence Diagram

Implement sending Hex html color codes

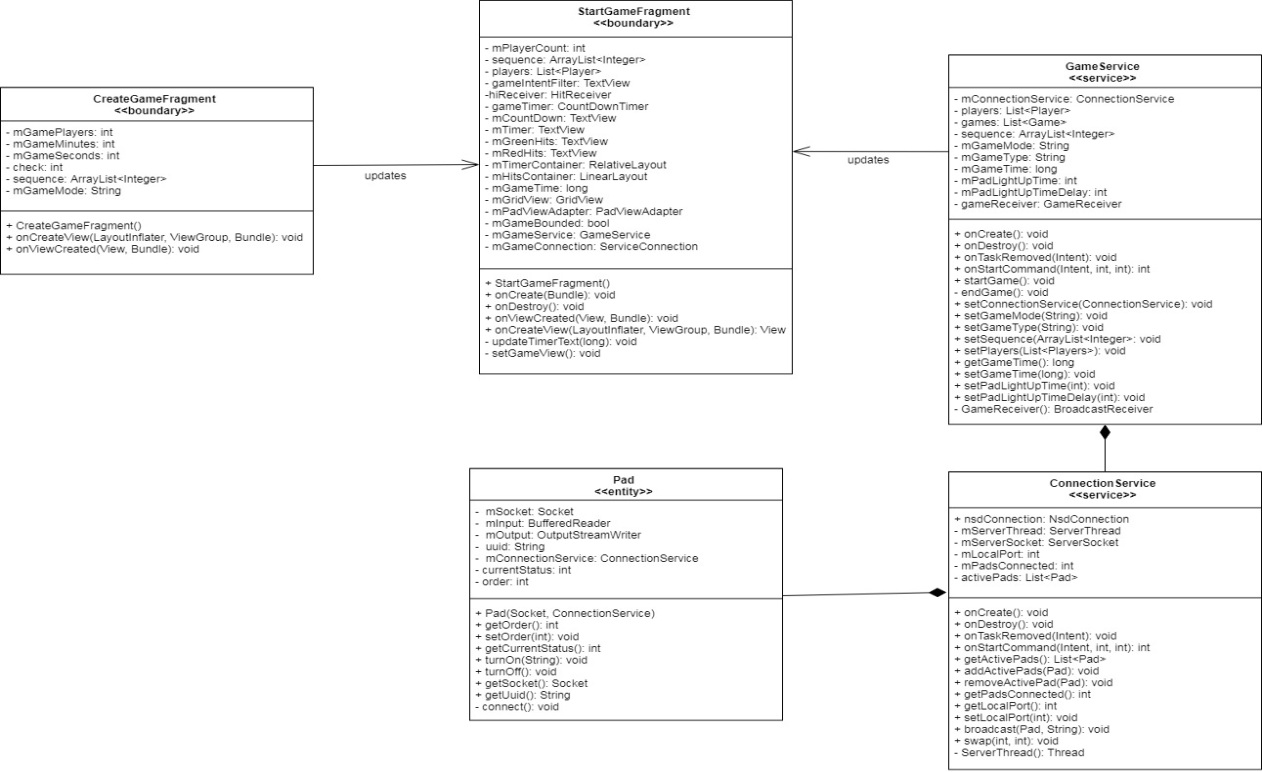


Fig. SC-1156 Class Diagram

Implement sending Hex html color codes

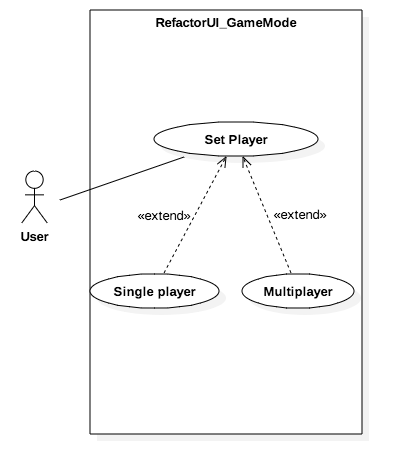
****

Fig. SC-1157 Use Case Diagram

Refactor Design for User To Play Game

## https://lh3.googleusercontent.com/6tXMS5NcvYFql1K9LWoSo6DxstrDUY3VRw0V05a3DX_sPDkVZDw5rCyW1AfumoBe72twfhaao8JNH8PGR35vpWHNuzq5CJC_hng9PYcfVsnocAazCGxELiEAQRFHYb7xdz5T8Eej

Fig. SC-1157 Sequence Diagram

Refactor Design for User To Play Game

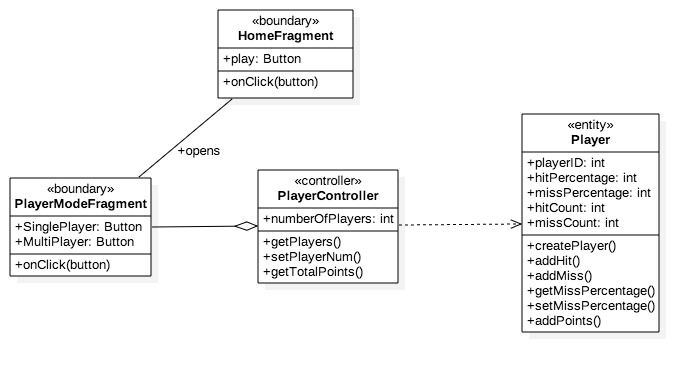


Fig. SC-1157 Class Diagram

Refactor Design for User To Play Game

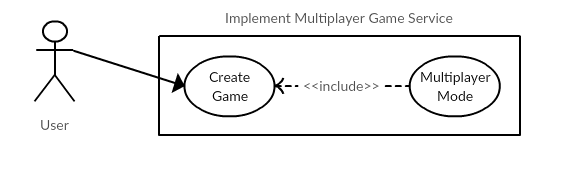
****

Fig. SC-1158 Use Case Diagram

Implement Multiplayer Game Template

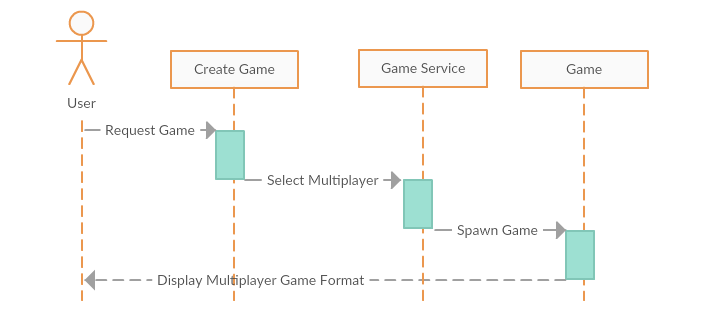
****

Fig. SC-1158 Sequence Diagram

Implement Multiplayer Game Template

****

Fig. SC-1158 Class Diagram

Implement Multiplayer Game Template

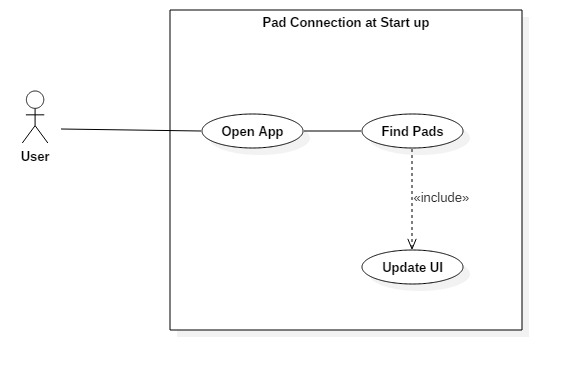
****

Fig. SC-1159 Use Case Diagram

Refactor PadConnectionService Class

## 

Fig. SC-1159 Sequence Diagram

Refactor PadConnectionService Class

## https://lh4.googleusercontent.com/ow2oqopmqLA-X21EeHMYIDDuL6GYkch7vOuAjztMbmBa_ybIqRZHlGM9OsHlAbG4zZ2dyn38fKadbOJYtRQNhmMV1v1K8x8WSBro-fvpi4wZUOCvjwdxPOjXeTVbvylvj5fgnDzG

Fig. SC-1159 Class Diagram

Refactor PadConnectionService Class

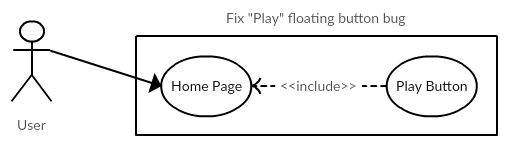
****

Fig. SC-1160 Use Case Diagram

Fix “Play” Floating Button Bug

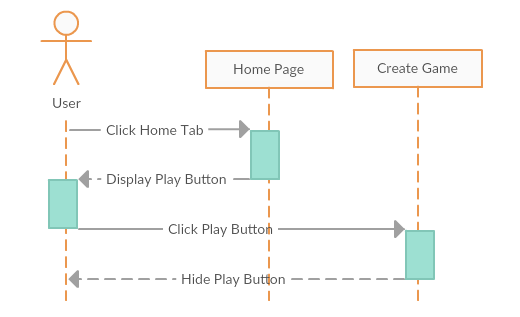
****

Fig.SC-1160 Sequence Diagram

Fix “Play” Floating Button Bug

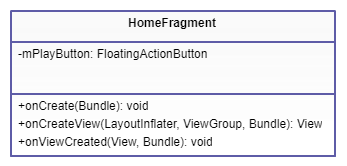
****

Fig.SC-1160 Class Diagram

Fix “Play” Floating Button Bug

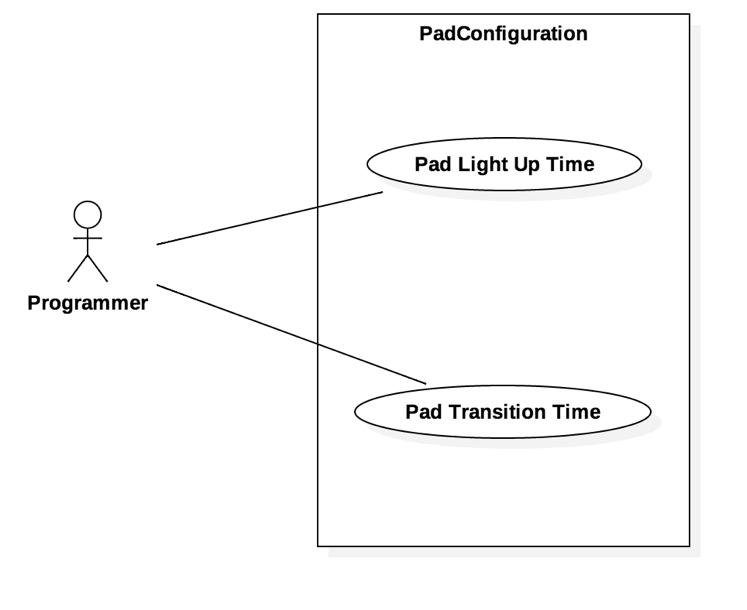


Fig. SC-1161 Use Case Diagram

Customize Pad Light Up Time

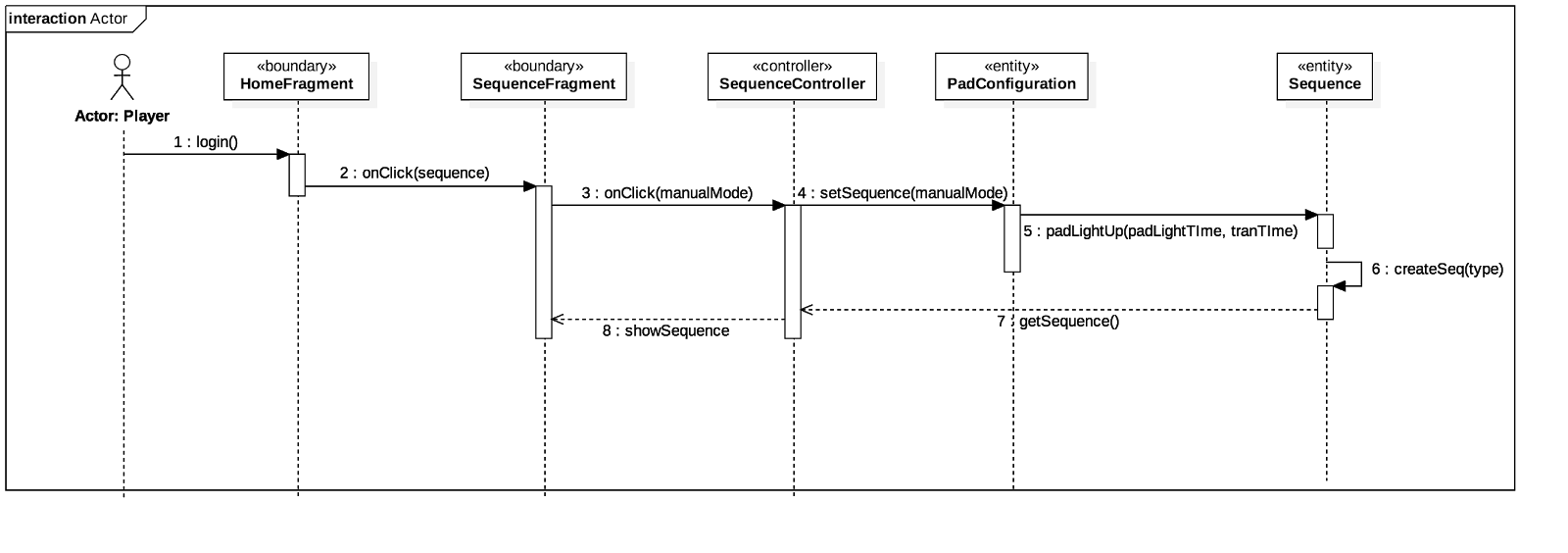


Fig. SC-1161 Sequence Diagram

Customize Pad Light Up Time

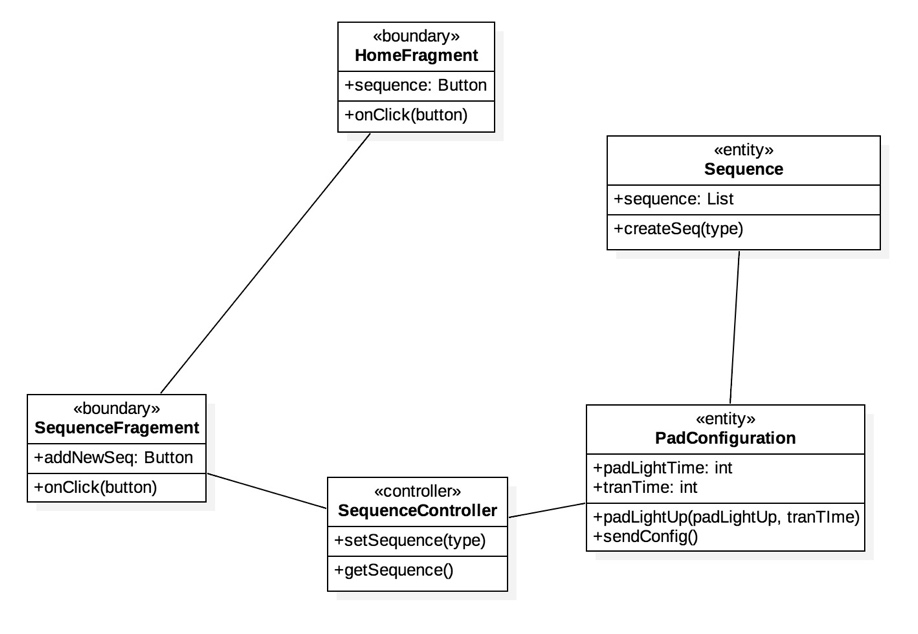


Fig. SC-1161 Class Diagram

Customize Pad Light Up Time

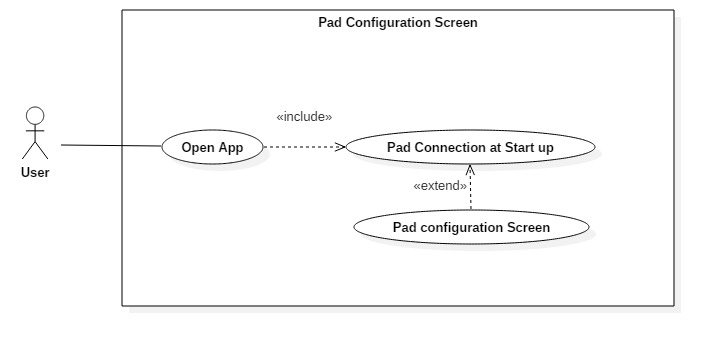


Fig. SC-1168 Use Case Diagram

Modify Pad Configuration UI to match Models

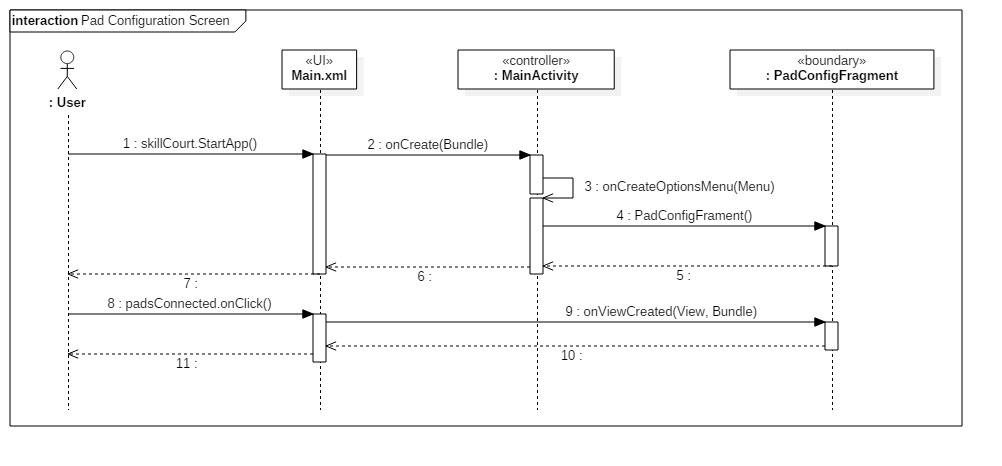


Fig. SC-1168 Sequence Diagram

Modify Pad Configuration UI to match Models

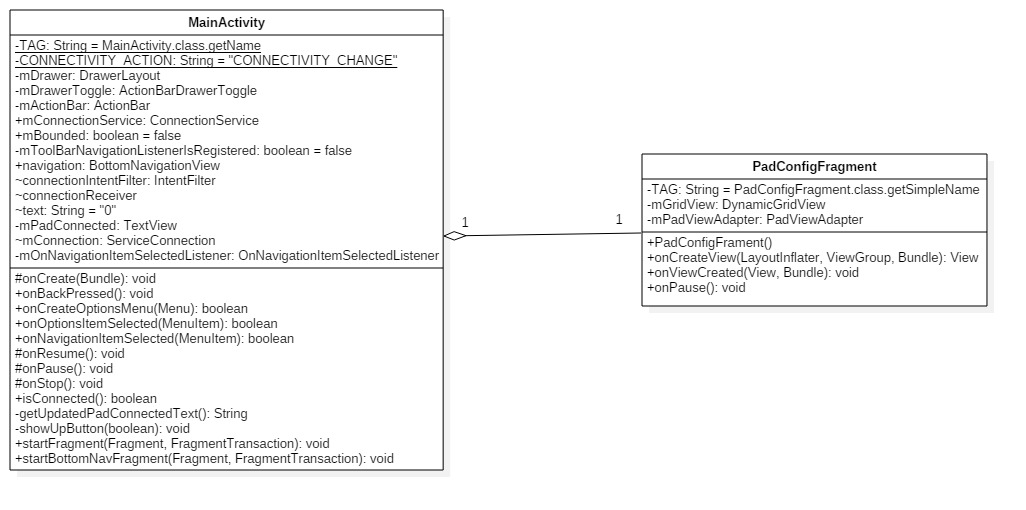


Fig. SC-1168 Class Diagram

Modify Pad Configuration UI to match Models

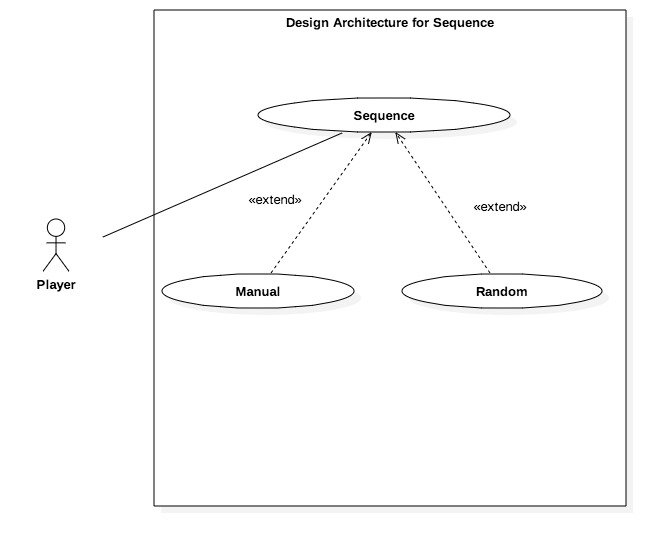


Fig. SC-1169 Use Case Diagram

Modify UI for Play Game

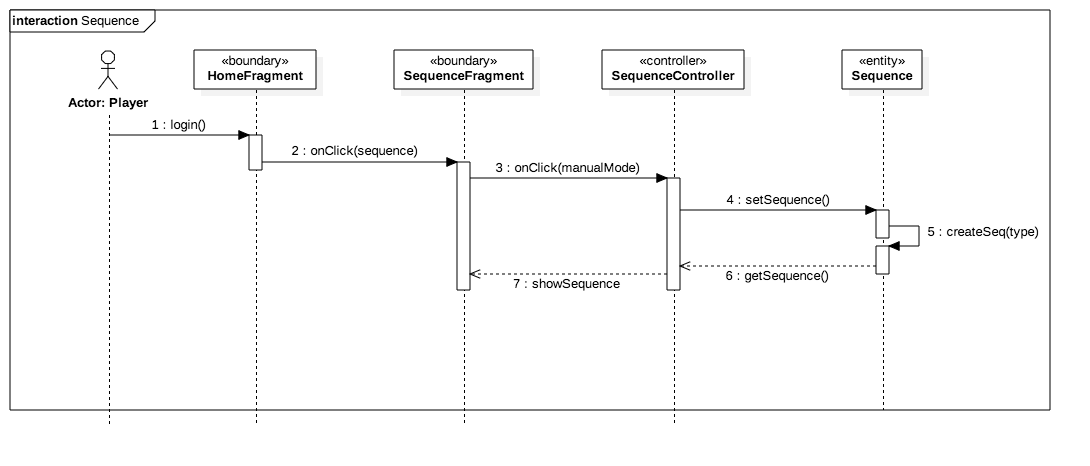


Fig. SC-1169 Sequence Diagram

Modify UI for Play Game

## https://lh6.googleusercontent.com/VbdX3Fza0DrGOwa5QYcXwpQUBtDNDQi6F3nZxN3rMhlPDOuvvxDTNgASf0CWE_rj1Yd63Id7IusEZJcxjmTwSD9M21LJ7oMN0yhoyzRE_Utn6WFNypanb8sRNdqmKzKG9ve-iwPe

Fig. SC-1169 Class Diagram

Modify UI for Play Game

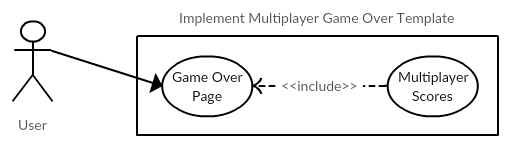
****

Fig. SC-1170 Use Case Diagram

Implement Multiplayer Game Over Template

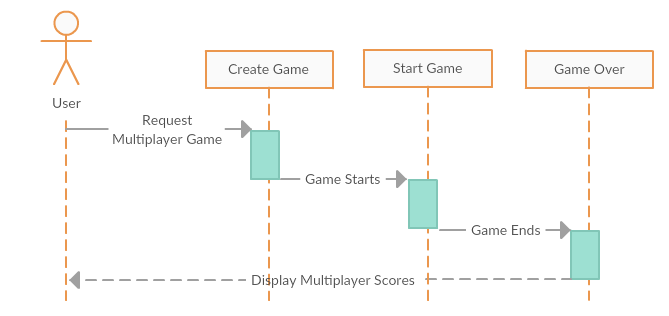
****

Fig. SC-1170 Sequence Diagram

Implement Multiplayer Game Over Template

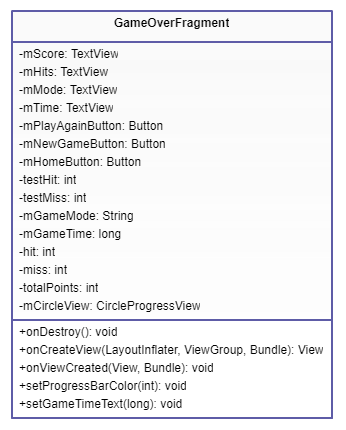
****

Fig. SC-1170 Class Diagram

Implement Multiplayer Game Over Template

## **Appendix B - User** **Interface Design**

## 

## 

## **Appendix C - Sprint Review Reports**

### **Sprint 2**

Attendees: Carlos Alva, Sandra Hurtado, Sergio Rosales, Gummi, Michael

Date: 01/29/2018

Start time: 7:30 PM

End time: 7:50 PM

After discussion, the velocity of the team were estimated to be an average of 2 weeks

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

* User Story ID: SC-1137 - Change UI from Activity to Fragments
* User Story ID: SC-1138 - Modify the code to fit the new architectural design for identifying pads
* User Story ID: SC-1139 - Modify the code to fit the new architectural design to play a sequence of instructions

The team members indicated their willingness to work on the following user stories.

* Carlos Alva
  + User Story ID: SC-1138
* Sandra Hurtado
  + User Story ID: SC-1139
* Sergio Rosales
  + User Story ID: SC-1137

### **Sprint 3**

Attendees: Carlos Alva, Sandra Hurtado, Sergio Rosales, Gummi, Michael

Date: 02/12/2018

Start time: 7:30 PM

End time: 7:50 PM

After discussion, the velocity of the team were estimated to be an average of 2 weeks

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

* User Story ID: SC-1145 - Implement Player class
* User Story ID: SC-1146 - Implement and Refactor Pad class
* User Story ID: SC-1147 - Implement and Refactor Sequence class

The team members indicated their willingness to work on the following user stories.

* Carlos Alva
  + User Story ID: SC-1146
* Sandra Hurtado
  + User Story ID: SC-1147
* Sergio Rosales
  + User Story ID: SC-1145

### **Sprint 4**

Attendees: Carlos Alva, Sandra Hurtado, Sergio Rosales, Gummi, Michael

Date: 02/26/2018

Start time: 6:00 PM

End time: 7:30 PM

After discussion, the velocity of the team were estimated to be an average of 2 weeks

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

* User Story ID: SC-1156 - Implement sending Hex Html color codes
* User Story ID: SC-1157 - Refactor UI design for User to Play Game
* User Story ID: SC-1158 - Implement Multiplayer Game Service

The team members indicated their willingness to work on the following user stories.

* Carlos Alva
  + User Story ID: SC-1156
* Sandra Hurtado
  + User Story ID: SC-1157
* Sergio Rosales
  + User Story ID: SC-1158

### **Sprint 5**

Attendees: Carlos Alva, Sandra Hurtado, Sergio Rosales, Gummi, Michael

Date: 03/09/2018

Start time: 6:00 PM

End time: 7:30 PM

After discussion, the velocity of the team were estimated to be an average of 2 weeks

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

* User Story ID: SC-1159 - Refactor PadConnectionService Class
* User Story ID: SC-1160 - Fix "Play" floating button bug
* User Story ID: SC-1161 - Customize Pad light up time

The team members indicated their willingness to work on the following user stories.

* Carlos Alva
  + User Story ID: SC-1159
* Sandra Hurtado
  + User Story ID: SC-1161
* Sergio Rosales
  + User Story ID: SC-1160

### **Sprint 6**

Attendees: Carlos Alva, Sandra Hurtado, Sergio Rosales, Gummi, Michael

Date: 03/30/2018

Start time: 6:00 PM

End time: 7:30 PM

After discussion, the velocity of the team were estimated to be an average of 2 weeks

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

* User Story ID: SC-1168 - Change Pad Configuration UI to match Models
* User Story ID: SC-1169 - Modify UI for Play Game
* User Story ID: SC-1170 - Modify Multiplayer UI to display current Players Pads

The team members indicated their willingness to work on the following user stories.

* Carlos Alva
  + User Story ID: SC-1168
* Sandra Hurtado
  + User Story ID: SC-1169
* Sergio Rosales
  + User Story ID: SC-1170

## **Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents**

**USER MANUAL**

Hardware: Pads

Software: Android SkillCourt App v-10.0

* 1. Connect Pads
     + 1. Turn on Hardware Pads
       2. Wait until Pad turn on red
       3. Open App
       4. Once the Pads are connected then they should flash white.
       5. Verify that the number of connected pads on the top right square displays the number of pads turned on.
       6. If the number of Pads turned on are not the same as the ones on the app then restart process again.
       7. If Pads match the number of pads turned on then you can continue with the next section
  2. Configure Pad Order

At the home screen click on the top right square containing the pads connected

The screen should show the pads connected visually represented by rectangles

Click on a Pad and see the Hardware Pad light up.

Press and Hold for 2 second until entering Edit mode (Pads start shaking)

Drag and Drop the Pads on the order that matches your current setup.

Try to match your pads to physically correspond to their position in space relative to you.

To exit edit mode just click the Pads again

* 1. Play Game

Click the Play button at the bottom right corner

Choose between Random or Manual

If game is Random then choose play time and light up time and delay time and hit play

If game is Manual then record the sequence by clicking on the appropriate pads and then select time to play, delay time and light up time, then hit Play to start game

**Play Time:** Time the Game runs for.

**Delay Time:** Time it takes for the next pad to turn on.

**Light Up Time:** Time the Pads stay on before moving onto the next.

User Guide Video: <https://youtu.be/E4j0bEOXH5E>

**INSTALLATION/MAINTENANCE DOCUMENT**

Video: <https://youtu.be/oe3svICPq4I>

**SHORTCOMINGS/WISHLIST DOCUMENT**

Video: <https://youtu.be/YHHwwBzfCDU>

# **References**

Android Developer Guide:

<https://developer.android.com/guide/>

Java Documentation:

<https://docs.oracle.com/javase/7/docs/api/>

JUnit Documentation:

<https://junit.org/junit4/javadoc/latest/>

Mockito Documentation:

<http://static.javadoc.io/org.mockito/mockito-core/2.18.3/org/mockito/Mockito.html>