

3 in 1 Game Project Backlog

Chris Coraggio, Muhammad Hamza Farrukh, Brad Howell, Rizwanulla
Mohammed, Syed Ahad Sagheer

Problem Statement:

Many classic games have been ported over to mobile apps for users to pass the time. For a user to fully enjoy the nostalgia and entertainment offered by these games, they must commit a significant portion of device memory to these applications. We aim to solve this issue by creating one centralized application that bundles together three classic games: Connect Four, Tic-Tac-Toe, and Hangman, in order to provide users a fun experience while saving space on their phones.

Furthermore, the majority of these applications are only available in single player platforms. In the current digital age we live in, users are craving connection with their peers. Our application aims to meet this need by providing an online multiplayer experience that will allow our users to share their favorite classic games with their friends.

2 Background Information (5 points)

A short paragraph about the history behind the problem at hand. Explain why this is an issue and why your team is attempting to solve this issue.

Background:

Mobile gaming is an important and significant part of our lives that isn't as convenient or as simple as it should be. In order to play some of our favorite games, such as connect 4, hangman, or tic-tac-toe, we need to devote a significant portion of device memory to download several apps. Many of these mobile games are also limited to either singleplayer or multiplayer. Our product is a centralized Android application that bundles

three classic games into a single app. We will also allow users to play both singleplayer and multiplayer.

3 Environment

A short paragraph pertaining detail about your development environment. This may include the language you will be coding in, external resources and libraries, databases, etc.

We will focus on creating an android application using Android Studio and will be coding in Java. Our server will be written in Javascript using nodeJS and will be deployed on Heroku.

4 Functional Requirements (35 points)

A list of functional requirements of your software product. Your product should be approximately 2500 lines of code or more – not counting comments or blank lines, that is 2500 actual lines of code or more.

Backlog ID	Functional Requirements	Hours	Initial Sprint Number	Sprint 1 Status
1	As a user, I would like to be able to set a unique nickname	5	Sprint 1	Moved to sprint 2
2	As a user, I would like to be able to change my nickname	2	Sprint 1	Moved to sprint 2
3	As a user, I would like to be able to play a game of connect 4 with an AI	7	Sprint 1	Completed in sprint 1
4	As a user, I would like to be able to play Tic-Tac-Toe with an AI	6	Sprint 1	Completed in sprint 1
5	As a user, I would like to be able to play Hangman with an AI	7	Sprint 1	In progress, moved for sprint 2
6	As a user, I would like to play Connect 4 with a friend online	6	Sprint 2	Planned for sprint 2

7	As a user, I would like to play Tic-Tac-Toe with a friend online	4	Sprint 2	Planned for sprint 2
8	As a user, I would like to play Hangman with a friend online	5	Sprint 2	Planned for sprint 2
9	As a user, I want to be able to see my score after completing and game	3	Sprint 2	Planned for sprint 2
10	As a user, I want to be able to see a menu of all available games	2	Sprint 1	Completed in sprint 1
11	As a user, I would like to be able to select a game from the menu	3	Sprint 1	Completed in sprint 1
12	As a client user, I would like to be able to enter my friend's unique nickname and connect to them	4	Sprint 2	Planned for sprint 2
13	As a host user, I would like to be able to wait for my friend to connect to the server	5	Sprint 2	Planned for sprint 2
14	As a user, I would like to see the connect 4 board once I select connect 4	6	Sprint 1	Completed in sprint 1
15	As a user, I would like to see the tic-tac-toe board once I select tic-tac-toe	7	Sprint 1	Completed sprint 1
16	As a user, I would like to see the hangman screen once I select hangman	8	Sprint 1	In progress, moved for sprint 2
17	As a user, I would like the connect 4 game to end when I select the forfeit option	2	Sprint 2	Planned for sprint 2
18	As a user, I would like the tic-tac-toe game to end when I select the forfeit option	2	Sprint 2	Planned for sprint 2
19	As a user, I would like the hangman game to end when I select the forfeit option	2	Sprint 2	Planned for sprint 2

20	As a user, when I am playing connect 4 and I close the app, I would like the connect 4 game to end	2	Sprint 2	Planned for sprint 2
21	As a user, when I am playing tic-tac-toe and I close the app, I would like the tic-tac-toe game to end	2	Sprint 2	Planned for sprint 2
22	As a user, when I am playing hangman and I close the app, I would like the hangman game to end	2	Sprint 2	Planned for sprint 2
	Total	92		

5 Non-Functional Requirements (15 points)

Appropriate non-functional requirements that are related to your project. These may include security, scalability, reliability, easy recovery, etc.

Backlog ID	Non-functional requirements	Hours	Initial Sprint Number	Sprint 1 Status
23	As a user, I would like to be able to connect to a user quickly and easily	10	Sprint 2	Planned for sprint 2
24	As a developer, I would like to learn how to use nodeJS	15	Sprint 2	Planned for sprint 2
25	As a user, I would like an easy to navigate interface	30	Sprint 1 and 2	In progress, moved for sprint 2
26	As a developer, I would like to learn Android development	30	Sprint 1	In progress, moved for

				sprint 2
27	As a developer, I would like he codebase to have a solid, consistent architecture. Code should be clean and easy to read/navigate.	80	Sprint 1 and 2	In progress, moved for sprint 2
28	As a developer, I would like to have approval from another developer before I commit my code	10	Sprint 1 and 2	In progress, moved for sprint 2
	Total	175		

6 Use Cases (35 points)

Describe how a user will use your system. Make sure that every functional requirement is covered with a use case. A use case should include the following two details:

Name: Give a short, descriptive name to the use case.

Steps: Describe each step using a 2-column format (Actor actions and System responses).

Case 1: Setup a nickname

Action	System Response
1. Type in a nickname into an input box	
2. Confirm Nickname	3. Check to make sure nickname is unique
	4. Save the name locally on the system

Case 2: Change nickname

Action	System Response
1. Select change nickname option	2. Open a dialog box with an input box
3. Type in a new nickname into an input box	4. Check to make sure nickname is unique
	5. Save the new name as a replacement to the already there nickname

Case 3: Play Connect 4 with an AI

Action	System Response
1. Choose "Connect 4" on home page	2. Open a dialog box asking if you want to play single player mode or multiplayer mode
3. Select single player mode	4. Open a clear board of connect 4
	5. Randomly select a player to start first
6. Select a location to place a coin	7. Place an appropriately colored coin in the appropriate location

Case 4: Play Tic-Tac-Toe with an AI

Action	System Response
1. Choose "Tic-Tac-Toe" on home page	2. Open a dialog box asking if you want to play single player mode or multiplayer mode
3. Select single player mode	4. Open a clear board of Tic-Tac-Toe
	5. Randomly select a player to start first
6. Select a location to place character	7. Place an appropriately character in the appropriate location

Case 5: Play Hangman with an AI

Action	System Response
1. Choose "Hangman " on home page	2. Open a dialog box asking if you want to play single player mode or multiplayer mode
3. Select single player mode	4. Open a clean board of Hangman with the AI having already selected a word
5. Select a letter from a selection box	6. Determine whether that letter is in the word or not.

Case 6: Play Connect 4 with another player online

Action	System Response
1. Choose "Connect 4 " on home page	2. Open a dialog box asking if you want to play single player mode or multiplayer mode
3. Select multiplayer mode	4. Open a clean board of Connect 4
	5. Random select either player 1 or player 2 as the first to go
6. First player to go selects a location to place a coin	7. Determine whether that letter is in the word or not.

Case 7: Play Tic-Tac-Toe with another player online

Action	System Response
1. Choose "Tic-Tac-Toe " on home page	2. Open a dialog box asking if you want to play single player mode or multiplayer mode
3. Select multiplayer mode	4. Open a clean board of Hangman with the AI having already selected a word

5. Select a letter from a selection box	6. Determine whether that letter is in the word or not.
---	---

Case 8: Play Hangman with another player online

Action	System Response
1. Choose "Hangman " on home page	2. Open a dialog box asking if you want to play single player mode or multiplayer mode
3. Select single player mode	4. Open a clean board of Hangman with the AI having already selected a word
5. Select a letter from a selection box	6. Determine whether that letter is in the word or not.

Case 9: As a user, I want to be able to see my score after completing and game

Action	System Response
1. Finish a game	2. Calculate score
	3. Update score count
	4. Display updated score

Case 10: As a user, I want to be able to see a menu of all available games

Action	System Response
1. Open the app	2. App opens and menu page displays
	3. Buttons to open games load and are displayed

Case 11: As a user, I would like to be able to select a game from the menu

Action	System Response
1. Click the game button	2. Launch selected game activity
	3. Display game

Case 12: As a client user, I would like to be able to connect with my friend to play a game

Action	System Response
1. Click “connect to friend” button	2. Textbox opens which allows user to enter friends name
	3. Server connects to ensure game is valid
	4. Server connects both players to the same game

Case 13: As a host user, I would like to be able to wait for my friend to connect to the server

Action	System Response
1. Click “host game” button	2. Server makes game
	3. Server waits for client player to join

Case 14: As a user, I would like to see the connect 4 board once I select connect 4

Action	System Response
1. User clicks on “connect 4” button	2. Open a dialog box asking if you want to play single player mode or multiplayer mode

3. User selects single player	4. Load and display an empty connect 4 board
-------------------------------	--

Case 15: As a user, I would like to see the tic-tac-toe board once I select tic-tac-toe

Action	System Response
1. User clicks on "Tic-Tac-Toe" button	2. Open a dialog box asking if you want to play single player mode or multiplayer mode
3. User selects single player	4. Load and display an empty Tic-Tac-Toe board

Case 16: As a user, I would like to see the hangman screen once I select hangman

Action	System Response
1. User clicks on the hangman button	2. Open a dialog box asking if you want to play single player mode or multiplayer mode
3. User selects single player	4. Load and display a new hangman game

Case 17: As a user, I would like the connect 4 game to end when I select the forfeit option

Action	System Response
1. User clicks "forfeit" button	2. Game ends and score count updates accordingly

Case 18: As a user, I would like the tic-tac-toe game to end when I select the forfeit option

Action	System Response
1. User clicks “forfeit” button	2. Game ends and score count updates accordingly

Case 19: As a user, I would like the hangman game to end when I select the forfeit option

Action	System Response
1. User clicks “forfeit” button	2. Game ends and score count updates accordingly

Case 20: As a user, when I am playing connect 4 and I close the app, I would like the connect 4 game to end

Action	System Response
1. User kills app manually	2. App closes gracefully

Case 21: As a user, when I am playing tic-tac-toe and I close the app, I would like the tic-tac-toe game to end

Action	System Response
1. User kills app manually	2. App closes gracefully

Case 22: As a user, when I am playing hangman and I close the app, I would like the hangman game to end

Action	System Response
1. User kills app manually	2. App closes gracefully

