

H446

A Level Computer Science



2023-2024

Fulford School

Centre Number: 48317 Candidate Number: 7768 Name: Ben Nesom

A Level Computer Science

Practical programming project

Contents

[Analysis 1](#_Toc151021386)

[Summary 1](#_Toc151021387)

[Computational methods 1](#_Toc151021388)

[Stakeholders 2](#_Toc151021389)

[Similar Solutions 2](#_Toc151021390)

[Essential Features 2](#_Toc151021391)

[Limitations 3](#_Toc151021392)

[Hardware and Software Requirements 3](#_Toc151021393)

[Success Criteria 3](#_Toc151021394)

[Design 6](#_Toc151021395)

[Decomposition 6](#_Toc151021396)

[For Host Player 6](#_Toc151021397)

[For Client Players 7](#_Toc151021398)

[Algorithms 8](#_Toc151021399)

[Procedures 8](#_Toc151021400)

[Functions 8](#_Toc151021401)

[Usability Features 9](#_Toc151021402)

[Testing Data 9](#_Toc151021403)

[Development and Testing 12](#_Toc151021404)

[Evaluation 12](#_Toc151021405)

# Analysis

## Summary

I would like to create a multiplayer game based on the board game “*The Chameleon*” made by *Big Potato Games.* The game would need to connect 8 players (including a host) to one another and be able to interact and send information between each of the players.

## Computational methods

**Abstraction** – I will use abstraction to ensure that my software isnt bloated and only contains essential features. This will allow for lower system requirements. I will also use abstraction so that connecting to other players is easy. This will mean that my stakeholders will not need to have an advanced knowledge of networking. The game will be based around a board game, so I will also use abstraction so that I can display all of the information that the game would have on cards, dice, and boards in a way that is easy to understand and likely more efficient than dealing out cards.

**Decomposition** – Decompostion is the process of breaking down a large project into smaller, more managable ones. In the context of this game some of the smaller problems could be connecting to another player and allowing information to be sent between them; displaying the same cards on each player’s screen; choosing players to be the chameleon; and having an interactive UI.

**Sequence and Selection** – Without these I would not be able to implement any game logic into this game. Logic is essential in a lot of parts of the game such as displaying different information to the chameleon as to the other players. Loops allow me to repeat a section of code mutilple turns – this will be used to display the grid of words, as well as to display the list of players in the lobby.

## Stakeholders

The users of my app will be similar to the users of a traditional board game. This means that my app will need to be usable for people of a wide range of ages. All of my users must have access to a computer of some sort and must have some basic knowledge in how to use them. I will make my game as easy as possible to play, so that people will less techonological ability are still just as able to play with others. One advantage of a simulated board game is that the hand-eye coordination that is typically needed for video games is not as necessary, the majority of my inputs will my mouse clicks on buttons and keyboard inputs in text input fields. Anyone that has used a computer should be familiar with these concepts.

The stakeholders for my game will aim to represent the diversity of ages of the users of my app, so will be Josef Lambley and Jennie Nesom. Josef Lambley is very familiar with video-gaming, and the general use of a computer. Jennie Nesom is not familiar with games outside of simple mobile games, but does enjoy board games.

The stakeholders want the game to have the general feel of a board game, with a simple UI. It should be easy to join other players and host games. They want the host to have settings to change the game, such as the amount of points it requires to win.

## Similar Solutions

<https://bigpotato.co.uk/products/the-chameleon> - this is the physical version of the game I will be making.

<https://the-chameleon.onrender.com/> - this is a website used to play the game. I aim to have an easier to use UX, as well as being based within an app rather than a website.

## Essential Features

The essential features that my stakeholder’s have outlined say the game must:

* Must have a similar feel to playing a physical copy of the board game
* The game must be playable, having scoring and a winner
* The UI must be simple and easy to use
* Games should be easy to join and host
* The host should be able to decide on some settings, such as the points need to win
* The players must be able to send emoticons from a select list to other players
* The host of the game must be able to kick other players, as well as pause the game

## Limitations

The game will not have public lobbies, so that moderation will be the host of the game’s responsibility.

The game will not have a built-in voice chat, so if players wish to communicate outside of giving clues, they must either be in the same room, or have another voice chat application (e.g. discord, or facetime)

The game will have no accounts, so no social features such as friends and private messages.

## Hardware and Software Requirements

To develop my game, I will need Windows 7 or above, a CPU with X64 architecture, and a graphics API of DX10 or above.

The game will require a screen, keyboard, and either mouse or trackpad to play.

## Success Criteria

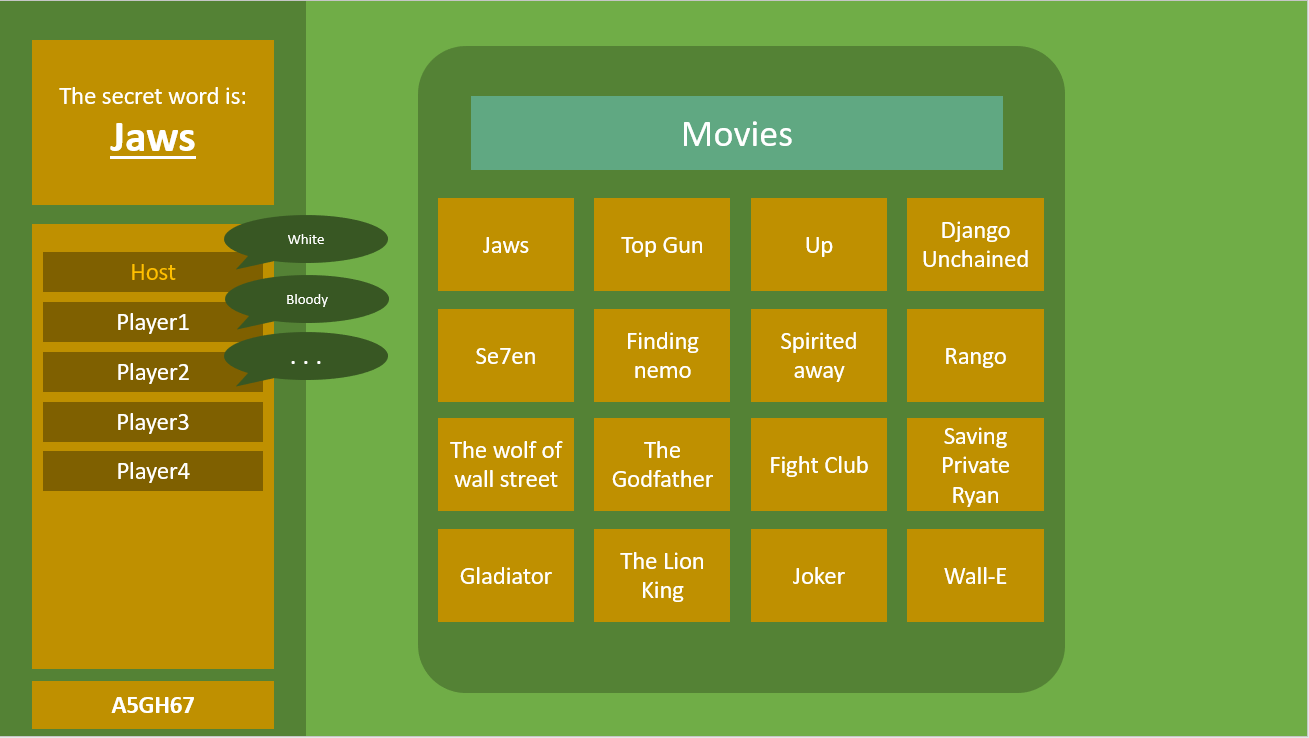
|  |  |  |
| --- | --- | --- |
| **ID** | **Success Criteria** | **Successful?** |
| **0** | **Game Launch** |  |
| 0.1 | Running the game must launch a game window |  |
| 0.2 | Closing the game must close the window |  |
| **1** | **Skeleton of Game** |  |
| 1.1 | Title Screen |  |
| 1.1a | Must have buttons for joining and hosting a game |  |
| 1.1b | Must have a text input area for inputting the username |  |
| 1.1c | Must have a text input area for inputting the join code |  |
| 1.1d | Username input box must be limited to 16 characters |  |
| 1.2 | Game Screen (must start hidden) |  |
| 1.2a | Must have a grid of 16 text boxes to display the chosen word grid |  |
| 1.2b | Must have a panel to display the players in the lobby |  |
| 1.2c | Must have a text input field to input your clue into (must start hidden) |  |
| 1.2d | Must have a button to send your clue to the other clients (must start hidden) |  |
| 1.2e | Must have a panel to display every player and the clue that they have given. |  |
| 1.2f | Must be a text box next to a player’s username to display their current score. |  |
| 1.2g | Host must have a start game button |  |
| 1.3 | Voting Screen |  |
| 1.3a | Must display all the player’s usernames onto buttons that can be clicked to vote for that player |  |
| 1.3b | Player’s clues must be displayed beneath their buttons |  |
| 1.3c | Player buttons and clues must be displayed in an ellipse |  |
| 1.3d | Must have a confirmation button (must start hidden) |  |
| 1.4 | Scoreboard Screen |  |
| 1.4a | Must show every player and their score |  |
| 1.4b | After a certain amount of time must show the change in score after a round |  |
| 1.4c | Screen must run automatically |  |
| **2** | **Networking** |  |
| 2.1 | Any player must be able to host a game session |  |
| 2.2 | Any player must be able to join another a currently hosted game |  |
| 2.3 | Clients must be able to send information to the server |  |
| 2.4 | The server must be able to send information to the clients |  |
| 2.5 | Clients should see a list of connected clients (and their usernames) |  |
| 2.6 | Host players must be able to kick other clients |  |
| 2.7 | A client must be able to leave the game |  |
| 2.8 | A host must be able to close the server |  |
| **3** | **Joining Game** |  |
| 3.1 | A player pressing join game must send their username to the server |  |
| 3.2 | A player pressing host game must send their username to the server |  |
| 3.3 | Once a player has connected to the server the game screen must be displayed |  |
| 3.3a | The current player list must be sent to the player |  |
| 3.3b | The current word grid must be sent to the player |  |
| 3.4 | Host must have a button to start the game |  |
| 4 | **Starting Game** |  |
| 4.1 | Host pressing “start game” must choose a new word grid, a random player to be the chameleon, and a random secret word |  |
| 4.2 | Player that is chosen to be chameleon must be told they’re the chameleon |  |
| 4.3 | Players not chosen to be chameleon must be given the secret word |  |
| **5** | **Playing the game** |  |
| 5.1 | Starting player must be given a text input panel that they can input their clue |  |
| 5.2 | Clue must be between 3-24 characters |  |
| 5.3 | Clue must be displayed on every client’s screen |  |
| 5.4 | Clue input panel must be disabled on the first players screen |  |
| 5.5 | Must repeat 5.1-5.4 for all players |  |
| 5.6 | After everyone has had their go, all players must be moved onto the voting screen |  |
| **6** | **Voting** |  |
| 6.1 | All players must see a button for each player arranged along a circle |  |
| 6.2 | These buttons must display a player’s name and the clue they have given each |  |
| 6.3 | Clicking a button send that you’ve cast a vote towards the player that you clicked, to the server |  |
| 6.4 | After every player has cast a vote, the votes are tallied, and every player’s secret identity is revealed |  |
| 6.5 | If the chameleon has the most votes, everyone gains 2 points |  |
| 6.6 | If someone other than the chameleon has the most votes, the chameleon gets 2 points and everyone that successfully voted for the chameleon gets 1 |  |
| 6.7 | Must loop the rounds, with a new word grid, new secret word, and new chameleon |  |
| 6.8 | Host must be able to stop the game at any time |  |

# Design

## Layout

**Title Screen**



**Game Screen**

**Vote Screen**

## Decomposition

### For Host Player

### For Client Players

I am going to start by developing the title screen, this is because it will be necessary to join other players, so I will need it to test the connections between players. The online part of my game is a key feature, so I will need this to be implemented early on, so that I can test and develop the other parts of the game that depend on it. I think that the next most important part of the game is the gameplay loop, especially everything contained on the game screen. This is because this contains all of the actual game logic, and ability to play the game. This means I will do this after the title screen, and online multiplayer logic. After the game screen I will start on the vote screen, as the game would be playable without it, just using a traditional voting system (e.g. everyone saying who they vote for). After I have developed everything else, I will develop the end screen, as this is not at all essential to the play of the game, and is purely for quality of life.

## Algorithms

### Procedures

Joining a lobby (Title Screen)

|  |  |
| --- | --- |
| Input  string lobbyCode | Process |

Hosting a lobby (Title Screen)

Arranging the player’s buttons in a circle (Vote Screen)

|  |  |
| --- | --- |
| Input  int numberOfPlayers  button[] buttons | Process  angle = 360/numberOfPlayers  xCoord = screen.width \* sin(angle)  yCoord = screen.height \* cos(angle)  for i in range (buttons.length())  if i > numberOfPlayers  buttons[i].disable()  else  buttons[i].pos = (xCoord,yCoord) |

### Functions

Choosing random word grid (Game Screen)

|  |  |  |
| --- | --- | --- |
| Input  WordGrid[] wordGrids | Process  WordGrid chosenWordGrid = wordGrids.Random()  return chosenWordGrid | Output  WordGrid chosenWordGrid |

Choosing random player to be the chameleon (Game Screen)

|  |  |  |
| --- | --- | --- |
| Input  ulong[] playerIds | Process  ulong chameleonID = playerIds.Random() | Output  ulong chameleonID |

## Usability Features

I will my UI scale with different screen sizes, so that my game can be played on different screens, as well as being able to be played without the game in fullscreen. This will also allow for use across different devices, if I ever implement a mobile version of the game. My game’s text will be large and have a high contrast against the background, so that it is easy to read. This will make it more accessible for people of all generations and eyesights, as my game will have a diverse player base, and be good for families. I will need to UI in my game to be easy to understand, as I am not assuming that all players are particularly experienced with video game UI’s

## Testing Data

**Prototype 1 – Iteration 1: Basic Skeleton**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Description | Success Criteria | Test Data | Expected Result |
| 1.1.1 | Tests that the program will run | 0.1 | User opens the game file | Window launches |
| 1.1.2 | Tests that the program will close | 0.2 | User closes the window | Window closes |
| 1.1.3 | The title screen must be displayed to the screen | 1.1 | User opens the game file | Window launches, displaying: join button, host button, username input box, join code input box. Nothing else should be displayed |
| 1.1.4 | The other screens aren’t yet displayed to the screen | 1 |
| 1.1.5 | All screens must be able to be hidden | 1.1 | User presses right arrow | Title screen gets hidden, Game screen is displayed, showing a: 4x4 grid of text boxes; a panel to input clues; a button to enter your clue; panel of text boxes; a start game button |
| 1.1.6 | The game screen must be loaded | 1.2 |
| 1.1.7 | The vote screen must be loaded | 1.3 | User presses up arrow | Game screen gets hidden, Vote screen is displayed, showing 8 buttons with a text box for each. |

**Prototype 1 – Iteration 2: Button Interactions**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Description | Success Criteria | Test Data | Expected Result |
| 1.2.1 | This tests that the host button is able to cause an action | 1.1a | User enters “Username” into the username input box and presses the host button | User is shown the game screen, with their inputted username on the playerlist |
| 1.2.2 | This tests that the join button is able to cause an action | 1.1a | User presses the join button | A prompt is shown to input the join code |
| 1.2.3 | Testing that the user can input a username | 1.1b | User inputs “PlayerName” into username input field | The username input box displays “PlayerName” |
| 1.2.4 | Testing that the username box is limited to 16 characters | 1.1d | User inputs “MoreThanSixteenChars” | The username input box displays “MoreThanSixteenC” |
| 1.2.5 | Testing that the user can input a join code | 1.1c | User inputs “F5AGH1” into join code input field | Input field displays “F5AGH1” |
| 1.2.6 | Validating the join code input field | 1.1c | User inputs nothing to the join code input field and presses join game | Text above input field displays “Please enter a valid join code” |
| 1.2.7 | User inputs “GH018” to join code input field and presses join game | Text above input field displays “Please enter a valid join code” |
| 1.2.8 | User inputs “GH018A7” to join code input field and presses join game | Text above input field displays “Please enter a valid join code” |
| 1.2.9 | User inputs “H01?A7” to join code input field and presses join game | Text above input field displays “Please enter a valid join code” |
| 1.2.10 | User inputs “GH018A” | The code error box says nothing |
| 1.2.11 | Validating the username input field | 1.1b | User doesn’t input anything to the username field and presses join game | Text above the username input box displays “Please input a username” |
| 1.2.12 | User inputs “A” to the username field and presses join game | The username error box says nothing |

**Prototype 1 – Iteration 3: Online Interactions**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Description | Success Criteria | Test Data | Expected Result |
| 1.3.1 | This tests that the host button is able to host a game session | 2.1 | User presses the host button | Should request a join code from the allocation service, as well as connecting as the host of a relay server |
| 1.3.2 | This tests that the join button can find a game session | 2.2, 3.1 & 3.3 | User inputs a valid code to a game session that exists. E.g. “A5P3JK” | Game screen should be displayed to the user. Adding their username to the other clients’ player list. And the other clients’ username to their player list. |
| 1.3.3 | This tests that the join button can tell whether a game session exists | 2.2 | User inputs a valid code to a game session that does not exists. E.g. “HG78AW” | Text above join code displays “Game not found” And game screen is not displayed |
| 1.3.4 | This tests that the host can cause something to happen on the client’s screen | 2.4 | Host types “TestHost” into the clue input box and presses the send clue button | Clients should get a debug saying “TestHost” |
| 1.3.5 | This tests that the clients can cause something to happen on the server | 2.3 | Client types “TestClient” into the clue input box and presses the send clue button | Host should get a debug saying “TestClient” |
| 1.3.6 | Tests that a client is able to leave the game | 2.7 | Client closes the window | Other clients see the client that left’s username disappear off of the playerList |
| 1.3.7 | Tests that only the host has the option to kick people from the game session | 2.6 | Host clicks on a player other than themselves username | Buttons displaying “Kick Player” should be shown |
| 1.3.8 | Host clicks on a player other than themselves username | Nothing happens |
| 1.3.9 | Tests that the Kick Button actually disconnects the selected client | 2.6 | Host clicks on a player other than themselves username, then presses the “Kick Player” button | Client is disconnected from the game session, and is removed from every player’s player list. That player is sent back to the title screen |
| 1.3.10 | Tests that the host can close the server | 2.8 | Host clicks on their own name in the player list | Button displaying “Close Server” should be displayed |
| 1.3.11 | Host clicks on their own name in the player list then clicks “Close Server” | Every client, and the host, should be disconnected from the server, and sent back to the title screen. The server connection should be closed |
| 1.3.12 | Tests that only the host has a start game button | 1.2g | Host presses host game | Host has a start game button |
| 1.3.13 | Client joins currently open game session using a valid join code | Client does not have a start game button |

**Prototype 1 – Iteration 4 - Game Logic**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Description | Success Criteria | Test Data | Expected Result |
| 1.4.1 | Tests that the host is able to start the game, and generate all the game information | 4.1, 4.2, 4.3 | Host presses the “Start game” button | The same grid of 16 related words should be shown to every client. One random player gets shown a message “You are the chameleon”. Every other player gets the same random word from the grid. |
| 1.4.2 | Tests that the players are able to enter a clue | 5.1 | Host presses “Start game” button | One player gets an area to input their clue |
| 1.4.3 | Validates the clue | 5.2 | Player inputs “A” and presses “Enter Clue” | Error message above the clue input field reads “Clue too short" |
| 1.4.4 | Player inputs “This Clue Is Way Too Long” and presses the enter clue button | Error message above the clue input field reads “Clue too long” |
| 1.4.5 | Player inputs “Acceptable” and presses the enter clue button | Clue is accepted, and displayed on every clients screen next to the player that inputted it’s name. The next player is shown the clue input field, and this player’s is hidden. |
| 1.4.6 | Player inputs “Car” and presses the enter clue button |
| 1.4.7 | Player inputs “ThisClueIsAccepted123123” and presses the enter clue button |
| 1.4.8 | Tests that the clue is displayed on everyone’s screen | 5.3 | Player inputs “ValidClue” and presses the enter clue button | “ValidClue” is displayed next to the player that inputted it’s name |
| 1.4.9 | Tests that the clue panel will be hidden if its not your turn | 5.4 | Clue input field and enter button are no longer visible |
| 1.4.10 | Tests that another player’s clue panel is made visible | 5.5 | Another player inputs “ValidClue” and pressed the enter clue button | Clue input field and enter button are made visible for one client |
| 1.4.11 | Tests that every player will get a chance to input a clue | 5.5 | All players take their turn inputting “ValidClue” and pressing the enter clue button | Every client should eventually have the clue panel made visible, then when they’ve entered their clue it should be hidden. No clients should have the panel displayed at the same time. No client should give 2 clues. |
| 1.4.12 | Tests that after every player has had their turn, they will be moved onto the voting screen | 5.6 | The last player enters “ValidClue” and presses the enter clue button | Every client should be moved onto the voting screen. |

# Development and Testing

## Iteration 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Description | Success Criteria | Test Data | Expected Result |
| 1.1.1 | Tests that the program will run | 0.1 | User opens the game file | Window launches |
| Actual Result:  Window launches as expected | | | | |
| Changes to be made:  N/A | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Description | Success Criteria | Test Data | Expected Result |
| 1.1.2 | Tests that the program will close | 0.2 | User closes the window | Window closes |
| Actual Result:  Window closes as expected | | | | |
| Changes to be made:  N/A | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Description | Success Criteria | Test Data | Expected Result |
| 1.1.3 | The title screen must be displayed to the screen | 1.1 | User opens the game file | Window launches, displaying: join button, host button, username input box, join code input box. Nothing else should be displayed |
| Actual Result: | | | | |
| Changes to be made:  N/A | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Description | Success Criteria | Test Data | Expected Result |
| 1.1.4 | The other screens aren’t yet displayed to the screen | 1 | User opens the game file | Window launches, displaying: join button, host button, username input box, join code input box. Nothing else should be displayed |
| Actual Result:  See above | | | | |
| Changes to be made:  N/A | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Description | Success Criteria | Test Data | Expected Result |
| 1.1.5 | All screens must be able to be hidden | 1.1 | User presses right arrow | Title screen gets hidden, Game screen is displayed, showing a: 4x4 grid of text boxes; a panel to input clues; a button to enter your clue; panel of text boxes; a start game button |
| Actual Result: | | | | |
| Changes to be made:  N/A | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Description | Success Criteria | Test Data | Expected Result |
| 1.1.7 | The vote screen must be loaded | 1.3 | User presses up arrow | Game screen gets hidden, Vote screen is displayed, showing 8 buttons with a text box for each. |
| Actual Result: | | | | |
| Changes to be made:  N/A | | | | |

### Comments

All my tests passed, meaning that no changes need to be made as of yet

## Iteration 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Description | Success Criteria | Test Data | Expected Result |
| 1.2.1 | This tests that the host button is able to cause an action | 1.1a | User enters “Username” into the username input box and presses the host button | User is shown the game screen, with their inputted username on the playerlist |
| Actual Result: | | | | |
| Changes to be made:  N/A | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Description | Success Criteria | Test Data | Expected Result |
| 1.2.2 | This tests that the join button is able to cause an action | 1.1a | User presses the join button | A prompt is shown to input the join code |
| Actual Result: | | | | |
| Changes to be made:  N/A | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Description | Success Criteria | Test Data | Expected Result |
| 1.2.3 | Testing that the user can input a username | 1.1b | User inputs “PlayerName” into username input field | The username input box displays “PlayerName” |
| Actual Result: | | | | |
| Changes to be made:  N/A | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Description | Success Criteria | Test Data | Expected Result |
| 1.2.4 | Testing that the username box is limited to 16 characters | 1.1d | User inputs “MoreThanSixteenChars” | The username input box displays “MoreThanSixteenC” |
| Actual Result: | | | | |
| Changes to be made:  N/A | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Description | Success Criteria | Test Data | Expected Result |
| 1.2.5 | Testing that the user can input a join code | 1.1c | User inputs “F5AGH1” into join code input field | Input field displays “F5AGH1” |
| Actual Result: | | | | |
| Changes to be made:  N/A | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Description | Success Criteria | Test Data | Expected Result |
| 1.2.6 | Validating the join code input field | 1.1c | User inputs nothing to the join code input field and presses join game | Text above input field displays “Please enter a valid join code” |
| Actual Result: | | | | |
| Changes to be made:  N/A | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Description | Success Criteria | Test Data | Expected Result |
| 1.2.7 | Validating the join code input field | 1.1c | User inputs “GH018” to join code input field and presses join game | Text above input field displays “Please enter a valid join code” |
| Actual Result: | | | | |
| Changes to be made:  N/A | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Description | Success Criteria | Test Data | Expected Result |
| 1.2.8 | Validating the join code input field | 1.1c | User inputs “GH018A7” to join code input field and presses join game | Text above input field displays “Please enter a valid join code” |
| Actual Result: | | | | |
| Changes to be made:  N/A | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Description | Success Criteria | Test Data | Expected Result |
| 1.2.9 | Validating the join code input field | 1.1c | User inputs “H01?A7” to join code input field and presses join game | Text above input field displays “Please enter a valid join code” |
| Actual Result: | | | | |
| Changes to be made:  N/A | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Description | Success Criteria | Test Data | Expected Result |
| 1.2.10 | Validating the join code input field | 1.1c | User inputs “GH018A” | The code error box says nothing |
| Actual Result: | | | | |
| Changes to be made:  N/A | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Description | Success Criteria | Test Data | Expected Result |
| 1.2.11 | Validating the username input field | 1.1b | User doesn’t input anything to the username field and presses join game | Text above the username input box displays “Please input a username” |
| Actual Result: | | | | |
| Changes to be made:  Make it say input instead of enter | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Number | Description | Success Criteria | Test Data | Expected Result |
| 1.2.12 | Validating the username input field | 1.1b | User inputs “A” to the username field and presses join game | The username error box says nothing |
| Actual Result: | | | | |
| Changes to be made:  N/A | | | | |

### Comments

The change from “enter” to input” has been made

# Evaluation