

BFC Playground Folio



CONTENTS

<u>Design Brief</u>	1	<u>Program Testing</u>	36
- Aims		- Testing Plan	
- Objectives			
<u>Detailed Design</u>	3	<u>External Files / Documents</u>	39
- Sketches of the screens		- Unity Project	
- Program Specifications		- Executable Program	
- Structure Charts		- User Manual	
		- Technical Manual	
		- Logbook	
<u>Algorithm Design</u>	8		
- Memory Game Pseudocode			
- Higher or Lower Pseudocode			
- Pfp Guesser Pseudocode			
- Username Guesser Pseudocode			

Design Brief

Aim

To create a program that helps people in the *Best Friends Club* (BFC) discover others in the BFC and learn tidbits about aspects of their experience in the community, such as how long they've been watching *Brodie*, the founder, on [Twitch](#).

Objectives

Memory Game

Incorporates Easy (2x3), Medium (3x4) and Hard (4x5) difficulties, with the difficulty increasing in relation to the amount of tiles that you have to match. The profile picture and name of the member will be shown and the player has to match them according to what they think goes together. There are also sudden death versions that will not allow any mistakes to be made.

Higher or Lower Follower Age

The player is given the username and profile picture of a BFC member and the player has to guess how long the given user has been following the streamer *Brodie*. The objective for the player is to guess the follower age of 3 twitch users in as least amount of attempts as possible. This game mode will also have the options of Easy (10 lives), Medium (7 lives) and Hard (5 life) difficulties alongside sudden death versions of Easy (1min), Medium (45secs) and Hard (30secs).

Twitch Username Guesser

The player is given the profile picture of a BFC member and they have to guess what their username is. This is a quick-fire game, where the objective is to see how many matches the player can get in the allocated time slots. This game mode will also have the options of Easy (1min), Medium (45sec) and Hard (30sec) difficulties alongside sudden death versions.

Profile Picture Guesser

Similar to the Twitch Username Guesser, but instead, the player is shown the username of a BFC member and you have to choose which profile picture (pfp) is theirs. This is a quick-fire game, where the objective is to see how many matches the player can get in the allocated time slots. This game mode will also have the options of Easy (1min), Medium (45sec) and Hard (30sec) difficulties alongside sudden death versions.

Statistics

Have a section of the game that holds statistics for each game mode.

Memory Game	Higher or Lower
Total Matches	Total Matches
Total Mistakes	Total Mistakes
Total Regular Easy Wins	Total Regular Easy Wins
Total Regular Medium Wins	Total Regular Medium Wins
Total Regular Hard Wins	Total Regular Hard Wins
Total Sudden Death Easy Wins	Total Regular Easy Losses
Total Sudden Death Medium Wins	Total Regular Medium Losses
Total Sudden Death Hard Wins	Total Regular Hard Losses
Total Sudden Death Easy Losses	Total Sudden Death Easy Wins
Total Sudden Death Medium Losses	Total Sudden Death Medium Wins
Total Sudden Death Hard Losses	Total Sudden Death Hard Wins
	Total Sudden Death Easy Losses
	Total Sudden Death Medium Losses
	Total Sudden Death Hard Losses

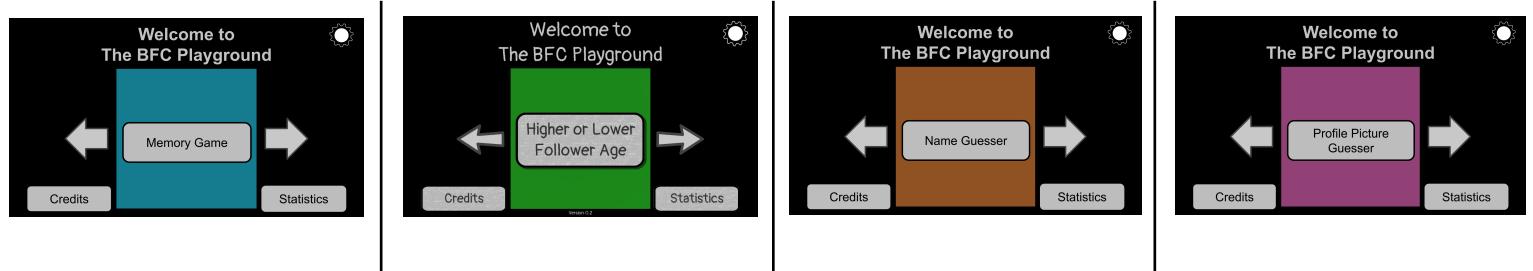
Name Guesser	Profile Picture Guesser
Total Matches	Total Matches
Total Mistakes	Total Mistakes
Total Regular Easy Games	Total Regular Easy Games
High Score: Matches - ???, Mistakes - ???	High Score: Matches - ???, Mistakes - ???
Total Regular Medium Games	Total Regular Medium Games
High Score: Matches - ???, Mistakes - ???	High Score: Matches - ???, Mistakes - ???
Total Regular Hard Games	Total Regular Hard Games
High Score: Matches - ???, Mistakes - ???	High Score: Matches - ???, Mistakes - ???
Total Sudden Death Easy Games	Total Sudden Death Easy Games
High Score: Matches - ???, Mistakes - ???	High Score: Matches - ???, Mistakes - ???
Total Sudden Death Medium Games	Total Sudden Death Medium Games
High Score: Matches - ???, Mistakes - ???	High Score: Matches - ???, Mistakes - ???
Total Sudden Death Hard Games	Total Sudden Death Hard Games
High Score: Matches - ???, Mistakes - ???	High Score: Matches - ???, Mistakes - ???

Detailed Design

Sketches

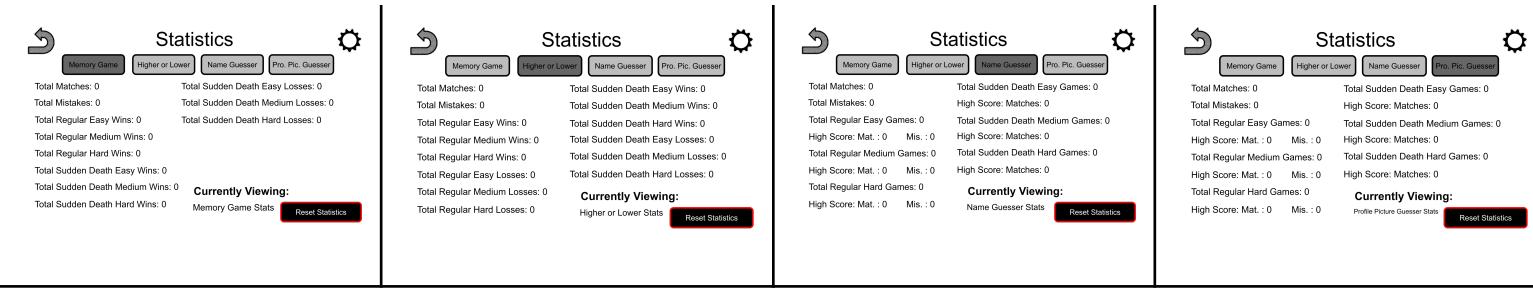
https://drive.google.com/drive/folders/1p2U8y_NsZlwKzdrfWbj5h0p-6BFM8VhF?usp=sharing

Start Screen



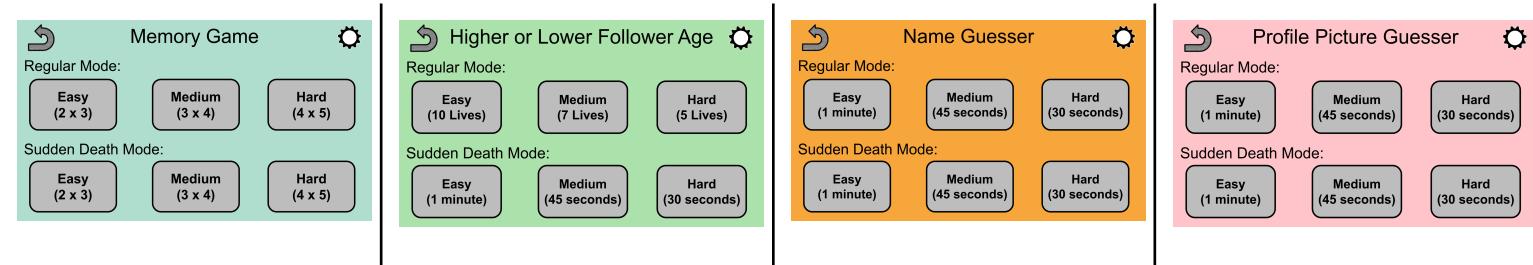
The cube that holds the game modes in the center of the screen rotates using the arrows. Each face of the cube on the horizontal axis has a game mode - Memory Game, Higher or Lower, Name Guesser and Profile Picture Guesser.

Statistics Screen



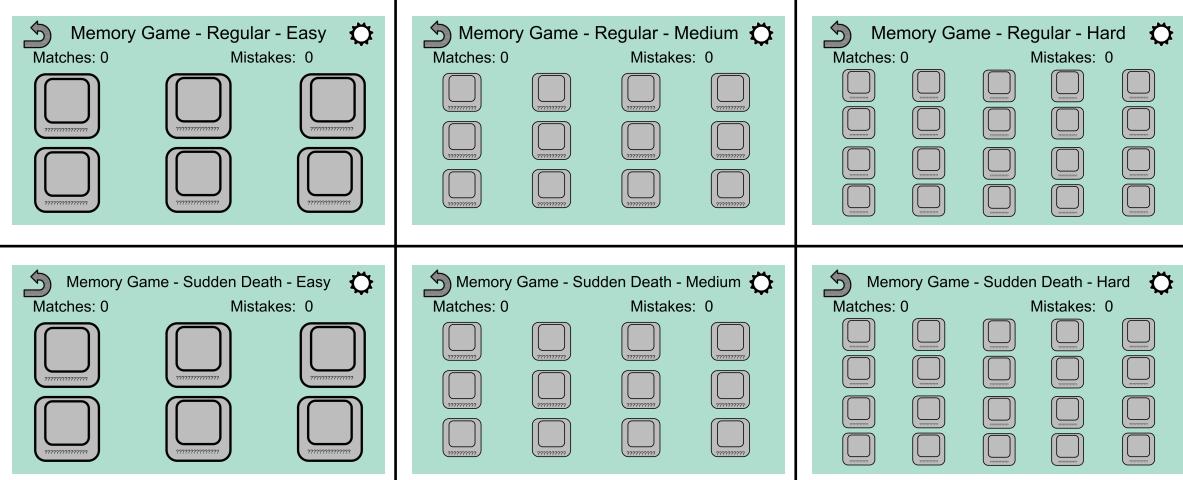
The player can choose which statistics they would like to view. For example, if *Name Guesser* was chosen for instance, the *Memory Game* button would be re-enabled and the *Name Guesser* button would be disabled, then all of the text GameObjects belonging to the *Memory Game* will disappear (their SetActive boolean set to false) and all those belonging to the *Name Guesser* will have appear (their SetActive boolean set to true).

Selection Screens



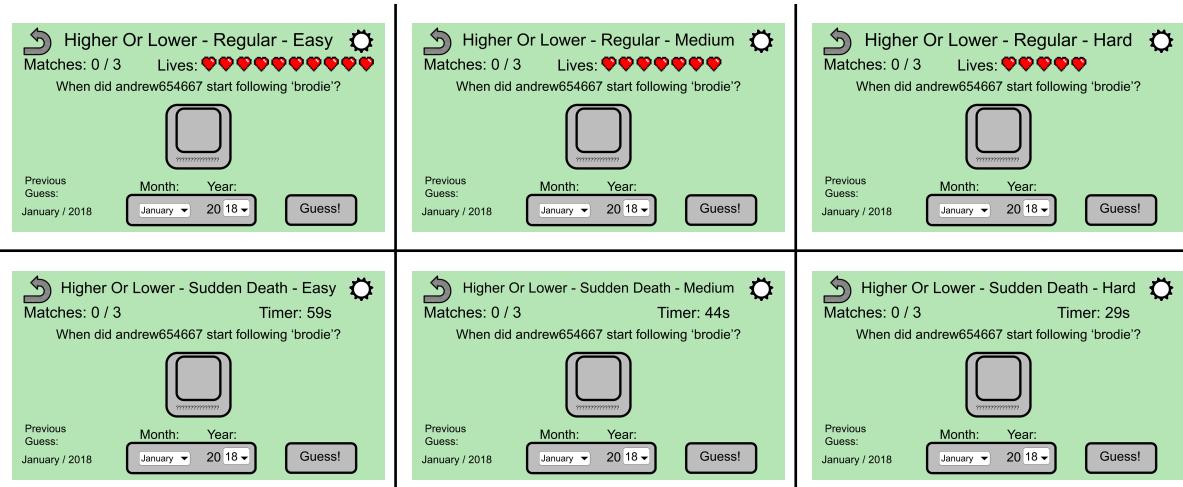
Each side of the cube on the start screen will take the player to one of these screens. From here, the player can choose which difficulty they want to play in each of the given game modes.

Memory Game



In the *Regular* game mode, the player is shown the profile pictures and names of members of the BFC. The objective is to match all the tiles in the least amount of attempts. In the *Sudden Death* game mode, the layout is the same as *Regular* mode, but if the player matches the wrong tiles, they lose immediately.

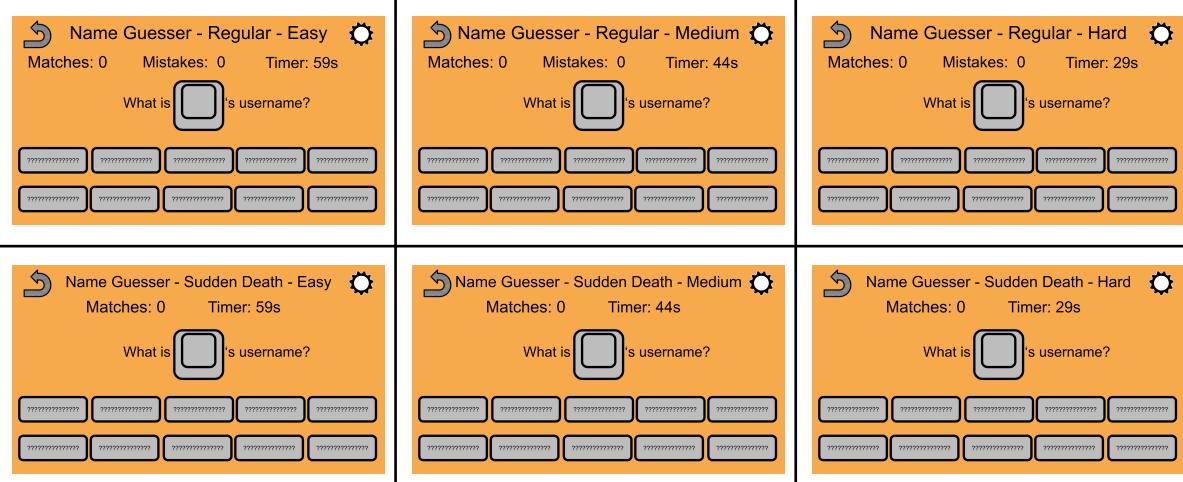
Higher or Lower Follower Age



In the *Regular* game mode, the player is shown the profile picture and name of 3 random members of the BFC. The objective is to correctly guess which year and month did the given members start following 'brodie' on twitch without running out of lives. In the *Sudden Death* game mode, if the player guesses the wrong month, guesses the wrong year or runs out of time, they lose immediately. Whenever the player guesses incorrectly in the *Regular* game mode, the heart shakes and turns grey.

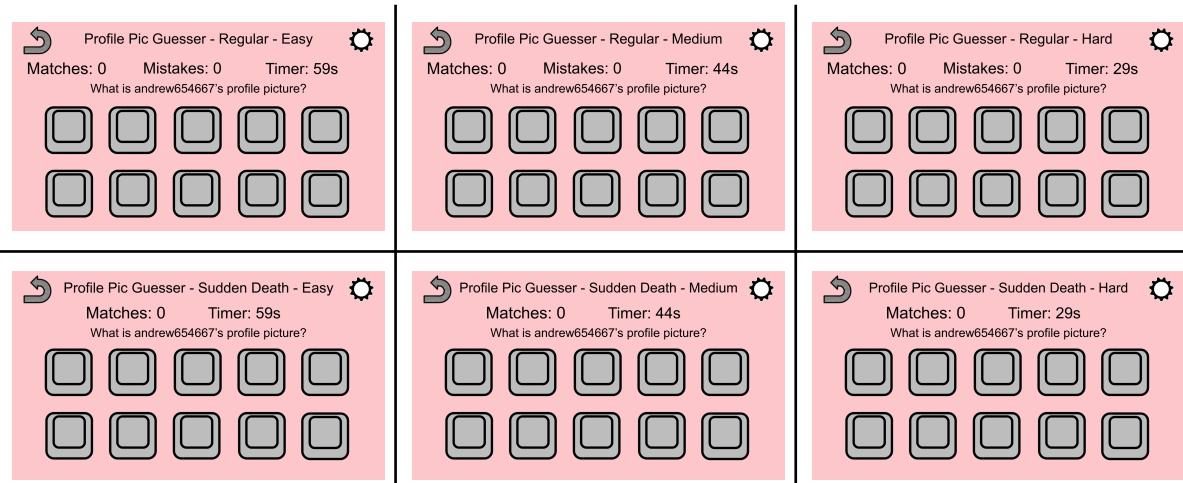


Twitch Username Guesser



In the *Regular* game mode, the player is given 1 profile picture and shown 10 usernames of random BFC members per round. The objective is to correctly guess which username belongs to the profile picture that the player has been given in each round. This is a quickfire game, so the player is racing against the clock to see how many matches they can get in the dedicated time slot. In the *Sudden Death* game mode, if the player guesses the wrong username, they lose immediately. In the actual game the tile in the center of the screen would have the profile picture of the member in the inner box and instead of the question marks within the button tiles along the bottom of the screen, it will hold the correct answer of who's profile picture is shown and 9 other random members. I wanted to have the game end with the condition of the player running out of lives and if the timer ran out, but I was restricted by the time I was given to make the project. Therefore for the *Regular* game modes, I went with the timer and a mistakes counter.

Profile Picture Guesser



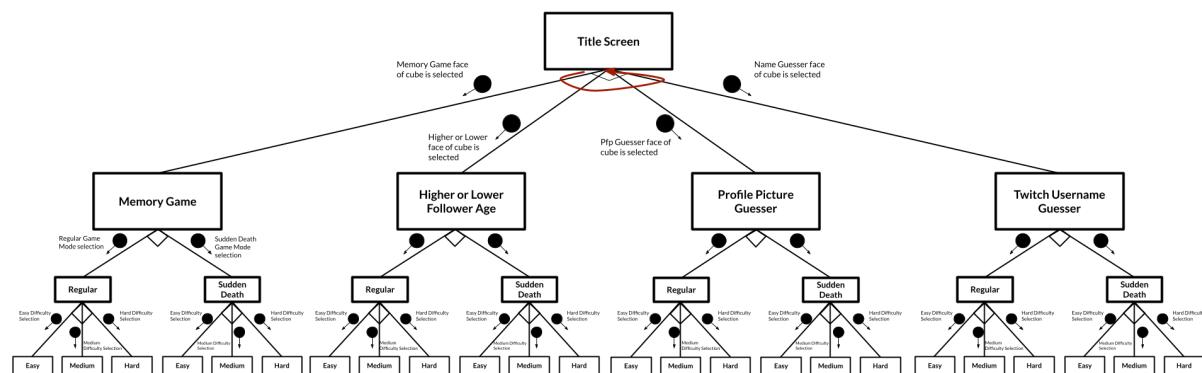
In the *Regular* game mode, the player is given 1 username and shown 10 profile pictures of random BFC members per round. The objective is to correctly guess which profile picture belongs to the username that the player has been given in each round. This is a quickfire game, so the player is racing against the clock to see how

many matches they can get in the dedicated time slot. In the *Sudden Death* game mode, if the player guesses the wrong profile picture, they lose immediately. In the actual game username in the center of the screen would have 1 of the 3 randomly chosen BFC members and along the bottom of the screen, the inner boxes of the button tiles will hold the correct answer of who's username is shown and 9 other random members. I wanted to have the game end with the condition of the player running out of lives and if the timer ran out, but I was restricted by the time I was given to make the project. Therefore for the *Regular* game modes, similar to the name guesser, I went with the timer and a mistakes counter.

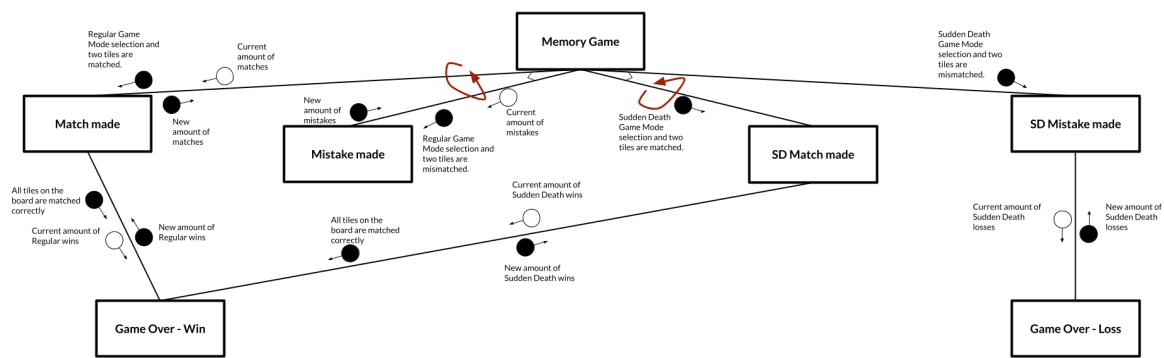
Structure Charts

https://drive.google.com/drive/folders/1wPZfJAR4pyfH80t69rAC60G_BpXEubNK?usp=sharing

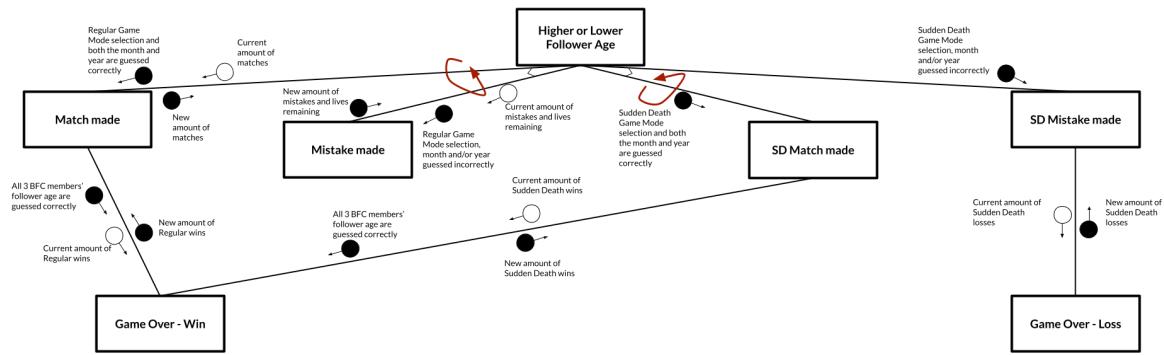
Start Screen



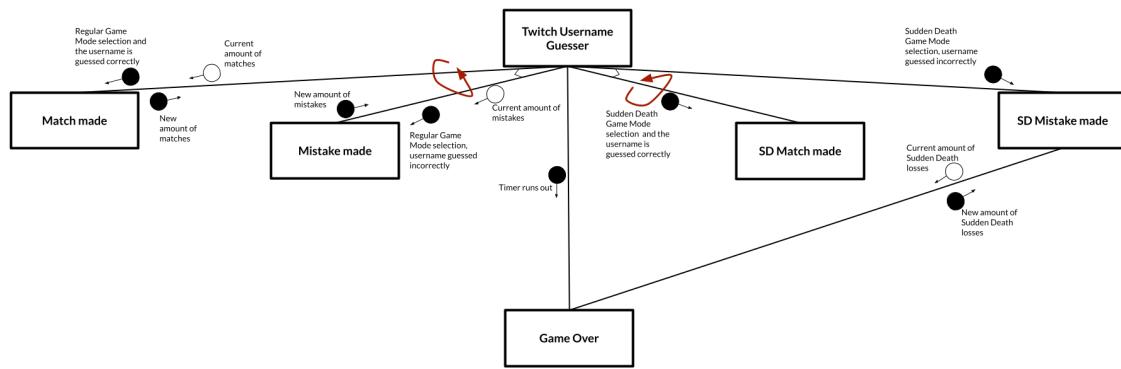
Memory Game



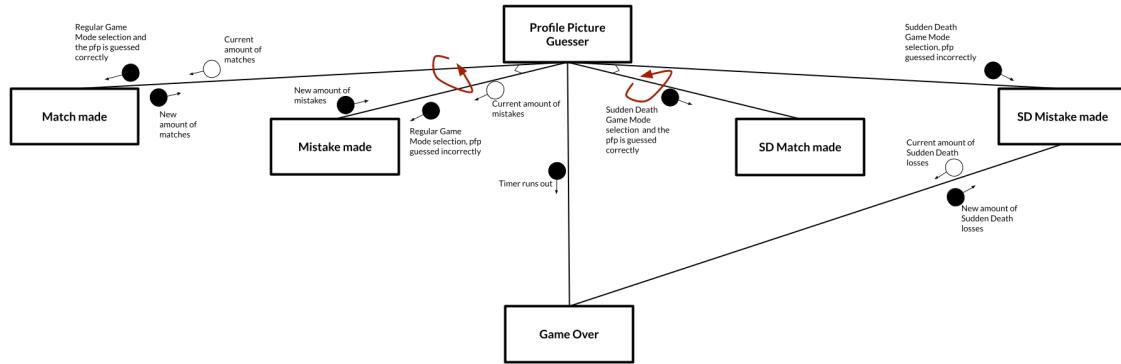
Higher or Lower Follower Age



Twitch Username Guesser



Profile Picture Guesser



Algorithm Design

Note:

- “Invoke” is calling a subprogram after a certain amount of time, which is indicated by the float next to the name of the subprogram, stating how long to wait, in terms of seconds, before loading the code written in said subprogram.
- When Invoking a subprogram with a similar name to “Button1to1Mistaken”, it will be triggering an animation. Which is done through the Unity Editor and not through a C# script, therefore there is no way to show this in pseudocode.
- Whenever something is “PRINT(ED)” it is coming up in the console of the program, which cannot be seen by the player, I only put these there to help me find bugs within the program and check that everything is working as intended in the background.
- When saying something is going to be “DISPLAY(ED)”, I am saying that it will be shown to the player.
- When there is “.SetActive = false” at the end of an object’s name, it means that said object will disappear.
- When there is “.interactable = false” at the end of an object, it means that the player will not be able to press said button component. Whereas, when there is “.interactable= true” at the end of an object, it means that the player will be able to press said button component.

Start Screen Pseudocode

Start Function (called before the first frame is loaded)

```
BEGIN
    currentFace = 1
END

BEGIN LeftArrowPressed
CASEWHERE currentFace =
    1:
        memoryButton.SetActive = false
        leftArrow.interactable = false
        rightArrow.interactable = false
        Animator.SetTrigger("Memory turn to H or L")
        Invoke("ShowHOrLButton", 0.5f)
        currentFace = 4
    2:
        nameGButton.SetActive = false
        leftArrow.interactable = false
        rightArrow.interactable = false
        Animator.SetTrigger("Name G turn to Memory")
        Invoke("ShowMemoryButton", 0.5f)
        currentFace = 1
    3:
```

```
proPicGButton.SetActive = false
leftArrow.interactable = false
rightArrow.interactable = false
Animator.SetTrigger("Pro Pic G turn to Name G")
Invoke("ShowNameGButton", 0.5f)
currentFace = 2
4:
hOrLButton.SetActive = false
leftArrow.interactable = false
rightArrow.interactable = false
Animator.SetTrigger("H or L turn to Pro Pic G")
Invoke("ShowProPicGButton", 0.5f)
currentFace = 3
ENDCASE
END LeftArrowPressed

BEGIN RightArrowPressed
CASEWHERE currentFace =
1:
memoryButton.SetActive = false
leftArrow.interactable = false
rightArrow.interactable = false
Animator.SetTrigger("Memory turn to Name G")
Invoke("ShowNameGButton", 0.5f)
currentFace = 2
2:
nameGButton.SetActive = false
leftArrow.interactable = false
rightArrow.interactable = false
Animator.SetTrigger("Name G turn to Pro Pic G")
Invoke("ShowProPicGButton", 0.5f)
currentFace = 3
3:
proPicGuesser.SetActive = false
leftArrow.interactable = false
rightArrow.interactable = false
Animator.SetTrigger("Pro Pic G turn to H or L")
Invoke("ShowHOrLButton", 0.5f)
currentFace = 4
4:
hOrLButton.SetActive = false
leftArrow.interactable = false
rightArrow.interactable = false
Animator.SetTrigger("H or L turn to Memory")
Invoke("ShowMemoryButton", 0.5f)
currentFace = 1
ENDCASE
END RightArrowPressed

BEGIN ShowMemoryButton
```

```
DISPLAY memoryButton
leftArrow.interactable = true
rightArrow.interactable = true
END ShowMemoryButton

BEGIN ShowNameGButton
    DISPLAY nameGButton
    leftArrow.interactable = true
    rightArrow.interactable = true
END ShowNameGButton

BEGIN ShowProPicGButton
    DISPLAY proPicGButton
    leftArrow.interactable = true
    rightArrow.interactable = true
END ShowProPicGButton

BEGIN ShowHOrLButton
    DISPLAY hOrLButton
    leftArrow.interactable = true
    rightArrow.interactable = true
END ShowHOrLButton
```

Memory Game Pseudocode

Start Function (called before the first frame is loaded)

```
BEGIN
    EasyXCoords(0) = -400
    EasyYCoords(0) = 55
    ...
    EasyXCoords(5) = 400
    EasyYCoords(5) = -150
    MediumXCoords(0) = -420
    MediumYCoords(0) = 80
    ...
    MediumXCoords(11) = 420
    MediumYCoords(11) = -200
    HardXCoords(0) = -400
    HardYCoords(0) = 100
    ...
    HardXCoords(19) = 400
    HardYCoords(19) = -230
    gameModeNum(0) = 5
    gameModeNum(1) = 11
    gameModeNum(2) = 19
    FOR i = 0 to 81
        randomPicBool1(i) = false
        PRINT "The random bool (1) of " + i + " has been made false."
```

```
NEXT i
CASEWHERE current =
0:
FOR i = 0 to 2
    REPEAT
        pictureNum(i) = Random.Range(0, 82);
        UNTIL randomPicBool1[pictureNum(i)] <> true
        randomPicBool1[pictureNum(i)] = true;
        gameImages(i).sprite = pictures[pictureNum(i)]
        imageNames(i).text = pictures[pictureNum(i)].name
        gameImages[(i + 3)].sprite = pictures[pictureNum(i)]
        imageNames[(i + 3)].text = pictures[pictureNum(i)].name
        PRINT "Picture Number (" + i + ") = " + pictureNum(i)
        PRINT "Picture Name (" + i + ") = " + pictures[pictureNum(i)].name
NEXT i
FOR i = 0 to gameModeNum(current)
    REPEAT
        tileLocation(i) = Random.Range(0, gameModeNum(current))
        UNTIL randomPicBool2[tileLocation(i)] <> true

        randomPicBool2[tileLocation(i)] = true
        Tiles[tileLocation(i)].anchoredPosition = new Vector3(xCoords(i), yCoords(i))
        PRINT "Tile location (" + tileLocation(i) + ") = " + tileLocation(i)
        PRINT "Position Taken (" + tileLocation(i) + ") = " +
        randomPicBool2[tileLocation(i)]
NEXT i
1:
FOR i = 0 to 5
    REPEAT
        pictureNum(i) = Random.Range(0, 82);
        UNTIL randomPicBool1[pictureNum(i)] <> true
        randomPicBool1[pictureNum(i)] = true;
        gameImages(i).sprite = pictures[pictureNum(i)]
        imageNames(i).text = pictures[pictureNum(i)].name
        gameImages[(i + 6)].sprite = pictures[pictureNum(i)]
        imageNames[(i + 6)].text = pictures[pictureNum(i)].name
        PRINT "Picture Number (" + i + ") = " + pictureNum(i)
        PRINT "Picture Name (" + i + ") = " + pictures[pictureNum(i)].name
NEXT i
FOR i = 0 to gameModeNum(current)
    REPEAT
        tileLocation(i) = Random.Range(0, gameModeNum(current))
        UNTIL randomPicBool2[tileLocation(i)] <> true
        randomPicBool2[tileLocation(i)] = true
        Tiles[tileLocation(i)].anchoredPosition = new Vector3(xCoords(i), yCoords(i))
        PRINT "Tile location (" + tileLocation(i) + ") = " + tileLocation(i)
        PRINT "Position Taken (" + tileLocation(i) + ") = " +
        randomPicBool2[tileLocation(i)]
NEXT i
2:
```

```
FOR i = 0 to 9
    REPEAT
        pictureNum(i) = Random.Range(0, 82);
    UNTIL randomPicBool1[pictureNum(i)] <> true
    randomPicBool1[pictureNum(i)] = true;
    gameImages(i).sprite = pictures[pictureNum(i)]
    imageNames(i).text = pictures[pictureNum(i)].name
    gameImages[(i + 10)].sprite = pictures[pictureNum(i)]
    imageNames[(i + 10)].text = pictures[pictureNum(i)].name
    PRINT "Picture Number (" + i + ") = " + pictureNum(i)
    PRINT "Picture Name (" + i + ") = " + pictures[pictureNum(i)].name
NEXT i
FOR i = 0 to gameModeNum(current)
    REPEAT
        tileLocation(i) = Random.Range(0, gameModeNum(current))
    UNTIL randomPicBool2[tileLocation(i)] <> true
    randomPicBool2[tileLocation(i)] = true
    Tiles[tileLocation(i)].anchoredPosition = new Vector3(xCoords(i), yCoords(i))
    PRINT "Tile location (" + tileLocation(i) + ") = " + tileLocation(i)
    PRINT "Position Taken (" + tileLocation(i) + ") = " +
        randomPicBool2[tileLocation(i)]
NEXT i
ENDCASE
FOR i = 0 to gameModeNum(current)
    randomPicBool2(i) = false
    PRINT "The random bool (2) of " + i + " has been made false."
NEXT i
IF Settings.suddenDeath = true THEN
    FOR i = 0 to gameModeNum(current)
        gameButtons(i).enabled = false
    NEXT i
    Animator.SetTrigger("SDShowAll")
    Invoke("AllEnabled", gameModeNum(current)f)
    Settings.suddenDeath = false
END IF
END
```

Update Function (called every frame)

```
BEGIN Update
    matchesText.text = "Matches: " + matches
    CASEWHERE current =
        0:
        IF matchBool(0) = true AND matchBool(1) = true AND matchBool(2) = true THEN
            FOR i = 0 to 2
                matchBool(i) = false
            Next i
            DISPLAY winScreen
        END IF
        1:
        IF matchBool(0) = true AND matchBool(1) = true AND matchBool(2) = true AND
```

```
matchBool(3) = true AND matchBool(4) = true AND matchBool(5) = true THEN
    FOR i = 0 to 5
        matchBool(i) = false
    Next i
    DISPLAY winScreen
END IF
2:
IF matchBool(0) = true AND matchBool(1) = true AND matchBool(2) = true AND
matchBool(3) = true AND matchBool(4) = true AND matchBool(5) = true AND
matchBool(6) = true AND matchBool(7) = true AND matchBool(8) = true AND
matchBool(9) = true THEN
    FOR i = 0 to 9
        matchBool(i) = false
    Next i
    DISPLAY winScreen
END IF
ENDCASE
END Update

BEGIN AllEnabled
    FOR i = 0 to gameModeNum(current)
        gameButtons(i).enabled = true
    NEXT i
END AllEnabled

BEGIN DebugTest
    PRINT "Currently Clicked = " + currentlyClicked
    PRINT "Matches = " + matches
    PRINT "Mistakes = " + mistakes
    CASEWHERE current =
        1:
        PRINT "Match 1 = " + matchBool(0) + ". Match 2 = " + matchBool(1) + ". Match 3
= " + matchBool(2) + ":"+
        2:
        PRINT "Match 1 = " + matchBool(0) + ". Match 2 = " + matchBool(1) + ". Match 3
= " + matchBool(2) + ". Match 4 = " + matchBool(3) + ". Match 5 = " + matchBool(4) + ".
Match 6 = " + matchBool(5) + ":"+
        3:
        PRINT "Match 1 = " + matchBool(0) + ". Match 2 = " + matchBool(1) + ". Match 3
= " + matchBool(2) + ". Match 4 = " + matchBool(3) + ". Match 5 = " + matchBool(4) + ".
Match 6 = " + matchBool(5) + ". Match 7 = " + matchBool(6) + ". Match 8 = "
+ matchBool(7) + ". Match 9 = " + matchBool(8) + ". Match 10 = " + matchBool(9) + "."
    END CASE
    PRINT "-----"
END DebugTest
```

The following function is called when the player clicks on a tile while playing the Regular difficulties

```
BEGIN Button1To1
    IF currentlyClicked = false THEN
```

```
currentlyClicked = true
gameButtons(0).enabled = false
curentlyPressedNum = 0
Animator.SetTrigger("Tile1To1Appeared")
ELSE IF currentlyClicked = true AND gameButtons(1).enabled = false THEN
    matchBool(0) = true
    AddMatches
        Animator.SetTrigger("Tile1Disappeared")
ELSE IF currentlyClicked = true AND gameButtons(1).enabled = true THEN
    AddMistakes
        Invoke("Button1to1Mistaken", 0.4f)
END IF
DebugTest
END Button1To1

BEGIN AddMatches
currentlyClicked = false
Increment matches
Increment Settings.totalMatchesInMemory
CASEWHERE current =
    0:
        IF matchBool(0) = true AND matchBool(1) = true AND matchBool(2) = true THEN
            Increment Settings.totalMemoryRegularEasyWins
        END IF
    1:
        IF matchBool(0) = true AND matchBool(1) = true AND matchBool(2) = true AND
        matchBool(3) = true AND matchBool(4) = true AND matchBool(5) = true THEN
            Increment Settings.totalMemoryRegularMediumWins
        END IF
    2:
        IF matchBool(0) = true AND matchBool(1) = true AND matchBool(2) = true AND
        matchBool(3) = true AND matchBool(4) = true AND matchBool(5) = true AND
        matchBool(6) = true AND matchBool(7) = true AND matchBool(8) = true AND
        matchBool(9) = true THEN
            Increment Settings.totalMemoryRegularHardWins
        END IF
    END CASE
END AddMatches

BEGIN AddMistakes
currentlyClicked = false
Increment mistakes
Increment Settings.totalMistakesInMemory
gameButtons(currentlyPressedNum).enabled = true
CASEWHERE currentlyPressedNum =
    0: Animator.SetTrigger("Tile1To1HalfMistaken")
    1: Animator.SetTrigger("Tile1To2HalfMistaken")
    2: Animator.SetTrigger("Tile2To1HalfMistaken")
    3: Animator.SetTrigger("Tile2To2HalfMistaken")
    4: Animator.SetTrigger("Tile3To1HalfMistaken")
```

```
    5: Animator.SetTrigger("Tile3To2HalfMistaken")
ENDCASE
END AddMistakes
```

The following function is called when the player clicks on a tile while playing the Sudden Death difficulties

```
BEGIN SDButton1To1
    IF currentlyClicked = false THEN
        currentlyClicked = true
        gameButtons(0).enabled = false
        currentlyPressedNum = 0
        Animator.SetTrigger("Tile1To1Appeared")
    ELSE IF currentlyClicked = true AND gameButtons(1).enabled = false THEN
        matchBool(0) = true
        SDAddMatches
        Animator.SetTrigger("Tile1Disappeared")
    ELSE IF currentlyClicked = true AND gameButtons(1).enabled = true THEN
        SDAddMistakes
    END IF
    DebugTest
END SDButton1To1
```

```
BEGIN SDAddMatches
    currentlyClicked = false
    Increment matches
    Increment Settings.totalMatchesInMemory
    CASEWHERE current =
        0:
        IF matchBool(0) = true AND matchBool(1) = true AND matchBool(2) = true THEN
            Increment Settings.totalMemorySuddenDeathEasyWins
        END IF
        1:
        IF matchBool(0) = true AND matchBool(1) = true AND matchBool(2) = true AND
        matchBool(3) = true AND matchBool(4) = true AND matchBool(5) = true THEN
            Increment Settings.totalMemorySuddenDeathMediumWins
        END IF
        2:
        IF matchBool(0) = true AND matchBool(1) = true AND matchBool(2) = true AND
        matchBool(3) = true AND matchBool(4) = true AND matchBool(5) = true AND
        matchBool(6) = true AND matchBool(7) = true AND matchBool(8) = true AND
        matchBool(9) = true THEN
            Increment Settings.totalMemorySuddenDeathHardWins
        END IF
    ENDCASE
END SDAddMatches
```

```
BEGIN SDAddMistakes
    Increment Settings.totalMistakesInMemory
    CASEWHERE current =
        0:
```

```
Increment Settings.totalMemorySuddenDeathEasyLosses
1:
Increment Settings.totalMemorySuddenDeathMediumLosses
2:
Increment Settings.totalMemorySuddenDeathHardLosses
ENDCASE
FOR i = 0 to gameModeNum(current)
    gameButtons(i).enabled = false
NEXT i
Animator.SetTrigger("LostSDGame")
randomEndScreen = Random.Range(0,4)
DISPLAY EndScreens(randomEndScreen)
END SDAddMistakes
```

Higher or Lower Follower Age Pseudocode

Start Function (called before the first frame is loaded)

```
BEGIN
    userMonth(1) = "May"
    userYear(1) = "2020"
    ...
    userMonth(100) = "September"
    userYear(100) = "2020"
    REPEAT
        FOR i = 0 to 2
            REPEAT
                pictureNum[i] = Random.Range(0, 100)
                UNTIL pictureNum[i] <> 0
                PRINT "Picture Number (" + i + ") = " + pictureNum(i)
                PRINT "Picture Name (" + i + ") = " + names[pictureNum(i)]
            NEXT i
        UNTIL pictureNum(0) <> pictureNum(1) OR pictureNum(0) <> pictureNum(2) OR
        pictureNum(1) = pictureNum(2)
        CASEWHERE current =
            0: livesRemaining = 10
            1: livesRemaining = 7
            2: livesRemaining = 5
        ENDCASE
        IF Settings.suddenDeath = true THEN
            CASEWHERE current =
                0: timeRemaining = 60
                1: timeRemaining = 45
                2: timeRemaining = 30
            ENDCASE
            timerIsRunning = true
            Settings.suddenDeath = false
        END IF
        guessingPlayerNum = 0
        LogCurrentlyGuessing
```

```
PRINT "This person started following in " + currentMonth + "," + currentYear + "."
END

BEGIN LogCurrentlyGuessing
    tileImage.sprite = pictures[pictureNum(guessingPlayerNum)]
    tileName.text = names[pictureNum(guessingPlayerNum)]
    WhenDid.text = "When did " + names[pictureNum(guessingPlayerNum)] + " start following
    'brodie'?"
    CASEWHERE userMonth[pictureNum(guessingPlayerNum)] =
        "January": currentMonth = 0
        "February": currentMonth = 1
        "March": currentMonth = 2
        "April": currentMonth = 3
        "May": currentMonth = 4
        "June": currentMonth = 5
        "July": currentMonth = 6
        "August": currentMonth = 7
        "September": currentMonth = 8
        "October": currentMonth = 9
        "November": currentMonth = 10
        "December": currentMonth = 11
    ENDCASE
    CASEWHERE userYear[pictureNum[guessingPlayerNum]] =
        "2018": currentYear = 0
        "2019": currentYear = 1
        "2020": currentYear = 2
        "2021": currentYear = 3
    ENDCASE
END LogCurrentlyGuessing
```

Update Function (called every frame)

```
BEGIN Update
    float seconds = Mathf.FloorToInt(timeRemaining % 60)
    timerText.text = "Timer: " + seconds + "s"
    IF timerIsRunning = true THEN
        IF timeRemaining > 0 THEN
            timeRemaining = timeRemaining - Time.deltaTime
        ELSE
            SDAddMistakes
            timeRemaining = 0
            timerIsRunning = false
        END IF
    END IF
    matchesText.text = "Matches: " + matches.ToString() + " / 3"
    IF matchBool(0) = true AND matchBool(1) = true AND matchBool(2) = true THEN
        matchBool(0) = false
        matchBool(1) = false
        matchBool(2) = false
        FOR i = 0 to 21
            disappearThings(i).SetActive = false
    END IF
```

```
NEXT i
Animator.SetTrigger("Disappear Tile")
DISPLAY winScreen
END IF
END Update
```

The following function is called when the player clicks on the 'Guess' button

```
BEGIN GuessButton
    PRINT "Chosen month is " + monthDropdown.value
    PRINT "Chosen year is " + yearDropdown.value
    chosenMonth = monthDropdown.value
    chosenYear = yearDropdown.value
    DISPLAY monthTitle
    DISPLAY yearTitle
    CASEWHERE chosenMonth
        = currentMonth:
            monthDropdown.interactable = false
            monthHigher.SetActive = false
            monthLower.SetActive = false
            DISPLAY monthLocked
        < currentMonth:
            DISPLAY monthHigher
            monthLower.SetActive = false
            singleMistake = true
            AddMistakes
        > currentMonth:
            monthHigher.SetActive = false
            DISPLAY monthLower
            singleMistakes = true
            AddMistakes

    ENDCASE
    IF chosenYear = currentYear THEN
        yearDropdown.interactable = false
        yearHigher.SetActive = false
        yearLower.SetActive = false
        DISPLAY yearLocked
    END IF
    IF gameEnded = false
        IF chosenYear < currentYear
            IF singleMistake = true THEN
                doubleMistake = true
                singleMistake = false
            END IF
            AddMistakes
        ELSE IF chosenYear > currentYear
            yearHigher.SetActive = false
            DISPLAY yearLower
            IF singleMistake = true THEN
                doubleMistake = true
            END IF
    END IF
END Update
```

```
    singleMistake = false
END IF
AddMistakes
END IF
END IF
IF monthDropdown.interactable = false AND yearDropdown.interactable = false THEN
    PRINT "You guessed the month and year correctly!"
    monthHigher.SetActive = false
    monthLower.SetActive = false
    yearHigher.SetActive = false
    yearLower.SetActive = false
    monthLocked.SetActive = false
    yearLocked.SetActive = false
    monthTitle.SetActive = false
    yearTitle.SetActive = false
    monthDropdown.interactable = true
    yearDropdown.interactable = true
    IF matchBool(0) = false THEN
        matchBool(0) = true
        guessingPlayerNum = 1
        Animator.SetTrigger("Remove Pic and Name")
        Invoke("DoTheLog", 0.3f)
    ELSE IF matchBool(0) = true AND matchBool(1) = false THEN
        matchBool(1) = true
        guessingPlayerNum = 2
        Animator.SetTrigger("Remove Pic and Name")
        Invoke("DoTheLog", 0.3f)
    ELSE IF matchBool(1) = true AND matchBool(2) = false THEN
        matchBool(2) = true
    END IF
AddMatches
CASEWHERE chosenMonth =
    0: previousMonth.text = "January"
    1: previousMonth.text = "February"
    2: previousMonth.text = "March"
    3: previousMonth.text = "April"
    4: previousMonth.text = "May"
    5: previousMonth.text = "June"
    6: previousMonth.text = "July"
    7: previousMonth.text = "August"
    8: previousMonth.text = "September"
    9: previousMonth.text = "October"
    10: previousMonth.text = "November"
    11: previousMonth.text = "December"
ENDCASE
CASEWHERE chosenYear =
    0: previousYear.text = "2018"
    1: previousYear.text = "2019"
    2: previousYear.text = "2020"
    3: previousYear.text = "2021"
```

```
        ENDCASE
    END IF
END GuessButton

BEGIN AddMatches
    Increment matches
    Increment Settings.totalMatchesInHigherOrLower
    IF matchBool(0) = true AND matchBool(1) = true AND matchBool(2) = true THEN
        CASEWHERE current =
            0: Increment Settings.totalHigherOrLowerRegularEasyWins
            1: Increment Settings.totalHigherOrLowerRegularMediumWins
            2: Increment Settings.totalHigherOrLowerRegularHardWins
    ENDCASE
END IF
END AddMatches

BEGIN AddMistakes
    Increment mistakes
    Increment Settings.totalMistakesInHigherOrLower
    Decrement livesRemaining
    CASEWHERE livesRemaining =
        9: Animator.SetTrigger("Killed Tenth Heart")
        8:
            IF doubleMistake = true THEN
                Invoke("NinthHeartDeath", 1f)
                doubleMistake = false
            ELSE
                Animator.SetTrigger("Killed Ninth Heart")
            END IF
        7:
            IF doubleMistake = true THEN
                Invoke("EighthHeartDeath", 1f)
                doubleMistake = false
            ELSE
                Animator.SetTrigger("Killed Eighth Heart")
            END IF
        6:
            IF doubleMistake = true THEN
                Invoke("SeventhHeartDeath", 1f)
                doubleMistake = false
            ELSE
                Animator.SetTrigger("Killed Seventh Heart")
            END IF
        5:
            IF doubleMistake = true THEN
                Invoke("SixthHeartDeath", 1f)
                doubleMistake = false
            ELSE
                Animator.SetTrigger("Killed Sixth Heart")
            END IF
    END IF
END AddMistakes
```

```
4:  
IF doubleMistake = true THEN  
    Invoke("FifthHeartDeath", 1f)  
    doubleMistake = false  
ELSE  
    Animator.SetTrigger("Killed Fifth Heart")  
END IF  
3:  
IF doubleMistake = true THEN  
    Invoke("FourthHeartDeath", 1f)  
    doubleMistake = false  
ELSE  
    Animator.SetTrigger("Killed Fourth Heart")  
END IF  
2:  
IF doubleMistake = true THEN  
    Invoke("ThirdHeartDeath", 1f)  
    doubleMistake = false  
ELSE  
    Animator.SetTrigger("Killed Third Heart")  
END IF  
1:  
IF doubleMistake = true THEN  
    Invoke("SecondHeartDeath", 1f)  
    doubleMistake = false  
ELSE  
    Animator.SetTrigger("Killed Second Heart")  
END IF  
<= 0:  
IF doubleMistake = true THEN  
    Invoke("FirstHeartDeath", 1f)  
    doubleMistake = false  
ELSE  
    Animator.SetTrigger("Killed First Heart")  
END IF  
PRINT "You Lose :"  
gameEnded = true  
FOR i = 0 to 21  
    disappearThings(i).SetActive = false  
NEXT i  
Animator.SetTrigger("Disappear Tile")  
Increment Settings.totalHigherOrLowerRegularEasyLosses  
randomEndScreen = Random.Range (0,3)  
DISPLAY EndScreens(randomEndScreen)  
ENDCASE  
END AddMistakes  
  
BEGIN DoTheLog  
    LogCurrentlyGuessing  
END DoTheLog
```

```
BEGIN SDGuessButton
    PRINT "Chosen month is " + monthDropdown.value
    PRINT "Chosen year is " + yearDropdown.value
    chosenMonth = monthDropdown.value
    chosenYear = yearDropdown.value
    DISPLAY monthTitle
    DISPLAY yearTitle
    CASEWHERE chosenMonth
        = currentMonth:
            monthDropdown.interactable = false
            monthHigher.SetActive = false
            monthLower.SetActive = false
            DISPLAY monthLocked
            < currentMonth:
                DISPLAY monthHigher
                monthLower.SetActive = false
                singleMistake = true
                SDAddMistakes
            > currentMonth:
                monthHigher.SetActive = false
                DISPLAY monthLower
                singleMistake = true
                SDAddMistakes
    ENDCASE
    CASEWHERE chosenYear
        = currentYear:
            yearDropdown.interactable = false
            yearHigher.SetActive = false
            yearLower.SetActive = false
            DISPLAY yearlocked
            < currentYear:
                DISPLAY yearHigher
                yearLower.SetActive = false
                SDAddMistakes
            > currentYear:
                yearHigher.SetActive = false
                DISPLAY yearLower
                SDAddMistakes
    ENDCASE
    IF monthDropdown.interactable = false AND yearDropdown.interactable = false THEN
        PRINT "You guessed the month and year correctly!"
        monthHigher.SetActive = false
        monthLower.SetActive = false
        yearHigher.SetActive = false
        yearLower.SetActive = false
        monthLocked.SetActive = false
        yearLocked.SetActive = false
        monthTitle.SetActive = false
        yearTitle.SetActive = false
```

```
monthDropdown.interactable = true
yearDropdown.interactable = true
IF matchBool(0) = false THEN
    matchBool(0) = true
    guessingPlayerNum = 1
    Animator.SetTrigger("Remove Pic and Name")
    Invoke("DoTheLog", 0.3f)
ELSE IF matchBool(0) = true AND matchBool(1) = false THEN
    matchBool(1) = true
    guessingPlayerNum = 2
    Animator.SetTrigger("Remove Pic and Name")
    Invoke("DoTheLog", 0.3f)
ELSE IF matchBool(1) = true AND matchBool(2) = false THEN
    matchBool(2) = true
    timerIsRunning = false
ENDIF
SDAddMatches
CASEWHERE chosenMonth =
    0: previousMonth.text = "January"
    1: previousMonth.text = "February"
    2: previousMonth.text = "March"
    3: previousMonth.text = "April"
    4: previousMonth.text = "May"
    5: previousMonth.text = "June"
    6: previousMonth.text = "July"
    7: previousMonth.text = "August"
    8: previousMonth.text = "September"
    9: previousMonth.text = "October"
    10: previousMonth.text = "November"
    11: previousMonth.text = "December"
ENDCASE
CASEWHERE chosenYear =
    0: previousYear.text = "2018"
    1: previousYear.text = "2019"
    2: previousYear.text = "2020"
    3: previousYear.text = "2021"
ENDCASE
ENDIF
END SDGuessButton

BEGIN SDAddMatches
    Increment matches
    Settings.totalMatchesInHigherOrLower
    IF matchBool(0) = true AND matchBool(1) = true AND matchBool(2) = true THEN
        CASEWHERE current =
            0: Increment Settings.totalHigherOrLowerSuddenDeathEasyWins
            1: Increment Settings.totalHigherOrLowerSuddenDeathMediumWins
            2: Increment Settings.totalHigherOrLowerSuddenDeathHardWins
        ENDCASE
        timerIsRunning = false
    ENDIF
END
```

```
    PRINT "You have won!"  
END IF  
END SDAddMatches  
  
BEGIN SDAddMistakes  
    Increment Settings.totalMistakesInHigherOrLower  
    CASEWHERE current =  
        0: Increment Settings.totalHigherOrLowerSuddenDeathEasyLosses  
        1: Increment Settings.totalHigherOrLowerSuddenDeathMediumLosses  
        2: Increment Settings.totalHigherOrLowerSuddenDeathHardLosses  
    ENDCASE  
    PRINT "You Lose :("  
    gameEnded = true  
    timerIsRunning = false  
    FOR i = 0 to 21  
        disappearThings(i).SetActive = false  
    NEXT i  
    Animator.SetTrigger("Disappear Tile")  
    randomEndScreen = Random.Range(0,3)  
    DISPLAY EndScreens(randomEndScreen)  
END SDAddMistakes
```

Twitch Username Guesser Pseudocode

Start Function (called before the first frame is loaded)

```
BEGIN  
    LoadPlayer  
    gameIsSuddenDeath = false  
    xCoords(0) = -740  
    yCoords(0) = -330  
    ...  
    xCoords(9) = 740  
    yCoords(9) = -530  
    FOR i = 0 to 79  
        REPEAT  
            throwaway2(i) = Random.Range(0, 79)  
        UNTIL usernamesToDisplay[throwaway2(i)] <> true  
        usernamesToDisplay[throwaway2(i)] = true  
        PRINT "The username of place " + i + " was taken by " + throwaway2(i)  
    NEXT i  
    currentPlaceInOrder = 0  
    gameImage.sprite = pictures[throwaway2(currentPlaceInOrder)]  
    gameTexts(0).text = pictures[throwaway2(currentPlaceInOrder)].name  
    usernamesToDisplay1[throwaway2(currentPlaceInOrder)] = true  
    FOR i = 1 to 9  
        REPEAT  
            throwaway3(i) = Random.Range(0, 79)
```

```
UNTIL usernamesToDisplay1[throwaway3(i)] <> true
usernamesToDisplay1[throwaway3(i)] = true
gameTexts(i).text = pictures[throwaway3(i)].name
PRINT "The username (" + throwaway3(i) + ") is in the tile (" + i + ")."

NEXT i
FOR i = 0 to 9
    REPEAT
        throwaway4(i) = Random.Range(0, 9)
    UNTIL usernamesToDisplay2[throwaway4(i)] <> true

    usernamesToDisplay2[throwaway4(i)] = true
    Tiles[throwaway4(i)].anchoredPosition = new Vector3(xCoords(i), yCoords(i))
    PRINT "The tile (" + throwaway4(i) + ") is in the location of tile (" + i + ")."

NEXT i
CASEWHERE current =
    0: timeRemaining = 60
    1: timeRemaining = 45
    2: timeRemaining = 30
ENDCASE
timerIsRunning = true
IF Settings.suddenDeath = true THEN
    highScoreMistakesObject.SetActive = false
    gameIsSuddenDeath = true
    Settings.suddenDeath = false
ENDIF
END
```

Update Function (called every frame)

```
BEGIN Update
    matchesText.text = "Matches: " + matches
    IF gameIsSuddenDeath = true THEN
        CASEWHERE current =
            0: highScoreMatches.text = "Matches: " +
                Settings.highestNameGuesserSuddenDeathEasyMatchesScore
            1: highScoreMatches.text = "Matches: " +
                Settings.highestNameGuesserSuddenDeathMediumMatchesScore
            2: highScoreMatches.text = "Matches: " +
                Settings.highestNameGuesserSuddenDeathHardMatchesScore
    END CASE
    ELSE
        mistakesText.text = "Mistakes: " + mistakes
        CASEWHERE current =
            0: highScoreMatches.text = "Matches: " +
                Settings.highestNameGuesserRegularEasyMatchesScore
            1: highScoreMatches.text = "Matches: " +
                Settings.highestNameGuesserRegularEasyMistakesScore
            2: highScoreMatches.text = "Matches: " +
                Settings.highestNameGuesserRegularMediumMatchesScore
            3: highScoreMatches.text = "Matches: " +
                Settings.highestNameGuesserRegularMediumMistakesScore
```

```
2: highScoreMatches.text = "Matches: " +
Settings.highestNameGuesserRegularHardMatchesScore
highScoreMistakes.text = "Mistakes: " +
Settings.highestNameGuesserRegularHardMistakesScore
ENDCASE
END IF
float seconds = Mathf.FloorToInt(timeRemaining MOD 60)
timerText.text = "Timer: " + seconds + "s"
IF timerIsRunning = true THEN
    IF timeRemaining > 0 THEN
        timeRemaining = timeRemaining - Time.deltaTime
    ELSE
        IF gameIsSuddenDeath = true THEN
            SDGameOver
        ELSE
            GameOver
        END IF
        timeRemaining = 0
        timerIsRunning = false
    END IF
END IF
END Update

BEGIN AddMistakes
    Increment mistakes
    Increment Settings.totalMistakesInNameGuesser
    CASEWHERE currentlyPressedNum =
        1: gameButtons(currentlyPressedNum).enabled = false
        Animator.SetTrigger("Tile2Mistaken")
        2: gameButtons(currentlyPressedNum).enabled = false
        Animator.SetTrigger("Tile3Mistaken")
        3: gameButtons(currentlyPressedNum).enabled = false
        Animator.SetTrigger("Tile4Mistaken")
        4: gameButtons(currentlyPressedNum).enabled = false
        Animator.SetTrigger("Tile5Mistaken")
        5: gameButtons(currentlyPressedNum).enabled = false
        Animator.SetTrigger("Tile6Mistaken")
        6: gameButtons(currentlyPressedNum).enabled = false
        Animator.SetTrigger("Tile7Mistaken")
        7: gameButtons(currentlyPressedNum).enabled = false
        Animator.SetTrigger("Tile8Mistaken")
        8: gameButtons(currentlyPressedNum).enabled = false
        Animator.SetTrigger("Tile9Mistaken")
        9: gameButtons(currentlyPressedNum).enabled = false
        Animator.SetTrigger("Tile10Mistaken")
    ENDCASE
END AddMistakes

BEGIN AddMatches
    Increment matches
```

```
Increment Settings.totalMatchesInNameGuesser
FOR i = 0 to 9
    gameButtons(i).enabled = false
    PRINT "The tile (" + i + ") is disabled"
NEXT i
END AddMatches

BEGIN GameOver
    timerIsRunning = false
    FOR i = 0 to 9
        gameButtons(i).enabled = false
    NEXT i
    Animator.SetTrigger("GameDone")
    CASEWHERE current =
        0:
        IF matches > Settings.highestNameGuesserRegularEasyMatchesScore THEN
            DISPLAY newHighScore
            Settings.highestNameGuesserRegularEasyMatchesScore = matches
            Settings.highestNameGuesserRegularEasyMistakesScore = mistakes
        ELSE IF matches = Settings.highestNameGuesserRegularEasyScore AND mistakes <
            Settings.highestNameGuesserRegularEasyMistakesScore THEN
            DISPLAY newHighScore
            Settings.highestNameGuesserRegularEasyMistakesScore = mistakes
        END IF
        Increment Settings.totalNameGuesserRegularEasyGames
        1:
        IF matches > Settings.highestNameGuesserRegularMediumMatchesScore THEN
            DISPLAY newHighScore
            Settings.highestNameGuesserRegularMediumMatchesScore = matches
            Settings.highestNameGuesserRegularMediumMistakesScore = mistakes
        ELSE IF matches = Settings.highestNameGuesserRegularMediumScore AND mistakes <
            Settings.highestNameGuesserRegularMediumMistakesScore THEN
            DISPLAY newHighScore
            Settings.highestNameGuesserRegularMediumMistakesScore = mistakes
        END IF
        Increment Settings.totalNameGuesserRegularMediumGames
        2:
        IF matches > Settings.highestNameGuesserRegularHardMatchesScore THEN
            DISPLAY newHighScore
            Settings.highestNameGuesserRegularHardMatchesScore = matches
            Settings.highestNameGuesserRegularHardMistakesScore = mistakes
        ELSE IF matches = Settings.highestNameGuesserRegularHardScore AND mistakes <
            Settings.highestNameGuesserRegularHardMistakesScore THEN
            DISPLAY newHighScore
            Settings.highestNameGuesserRegularHardMistakesScore = mistakes
        END IF
        Increment Settings.totalNameGuesserRegularHardGames
    ENDCASE
    Invoke("ShowGameOverPanel", 0.6f)
END GameOver
```

```
BEGIN ShowGameOverPanel
    DISPLAY gameOverPanel
    whatIsObject.SetActive = false
    guessTile.SetActive = false
END ShowGameOverPanel
```

```
BEGIN DebugTest
    PRINT "Matches = " + matches
    PRINT "Mistakes = " + mistakes
    PRINT "-----"
END DebugTest
```

The following function is called when the player clicks on the correct button

```
BEGIN CorrectButton
    Animator.SetTrigger("GameOver")
    AddMatches
    Invoke("ResetTheBoard", 0.6f)
END CorrectButton
```

```
BEGIN ResetTheBoard
    Increment currentPlaceInOrder
    gameTexts(0).text = pictures[throwaway2(currentPlaceInOrder)].name
    gameImage.sprite = pictures[throwaway2(currentPlaceInOrder)]?
    FOR i = 0 to 79
        usernamesToDisplay1(i) = false
    NEXT i
    FOR i = 0 to 9
        usernamesToDisplay(i) = false
    NEXT i
    usernamesToDisplay1[throwaway2(currentPlaceInOrder)] = true
    FOR i = 1 to 9
        REPEAT
            throwaway3(i) = Random.Range(0, 79)
            UNTIL usernamesToDisplay1[throwaway3(i)] <> true
            usernamesToDisplay1[throwaway3(i)] = true
            gameTexts(i).text = pictures[throwaway3(i)].name
            PRINT "The username (" + throwaway3[i] + ") is in the tile (" + i + ")."
    NEXT i
    FOR i = 0 to 9
        REPEAT
            throwaway4(i) = Random.Range(0, 9)
            UNTIL usernamesToDisplay2[throwaway4(i)] <> true
            usernamesToDisplay2[throwaway4(i)] = true
            Tiles[throwaway4(i)].anchoredPosition = new Vector3(xCoords(i), yCoords(i))
            PRINT "The tile (" + throwaway4[i] + ") is in the location of tile (" + i + ")."
    NEXT i
    FOR i = 0 to 9
        gameButtons(i).enabled = true
        PRINT "The tile (" + i + ") is enabled"
```

```
NEXT i
    Animator.SetTrigger("ResetBoard")
END ResetTheBoard
```

The following function is called when the player clicks on the wrong button while playing the Regular difficulties

```
BEGIN WrongButton2
    currentlyPressedNum = 1
    AddMistakes
END WrongButton2
...
BEGIN WrongButton10
    currentlyPressedNum = 9
    AddMistakes
END WrongButton10
```

The following function is called when the player clicks on the wrong button while playing the Sudden Death difficulties

```
BEGIN SDGameOver
    timerIsRunning = false
    FOR i = 0 to 9
        gameButtons(i).enabled = false
    NEXT i
    Animator.SetTrigger("GameDone")
    CASEWHERE current =
        0:
            IF matches > Settings.highestNameGuesserSuddenDeathEasyMatchesScore THEN
                DISPLAY newHighScore
                Settings.highestNameGuesserSuddenDeathEasyMatchesScore = matches
            END IF
            Increment Settings.totalNameGuesserSuddenDeathEasyGames
        1:
            IF matches > Settings.highestNameGuesserSuddenDeathMediumMatchesScore THEN
                DISPLAY newHighScore
                Settings.highestNameGuesserSuddenDeathMediumMatchesScore = matches
            END IF
            Increment Settings.totalNameGuesserSuddenDeathMediumGames
        2:
            IF matches > Settings.highestNameGuesserSuddenDeathHardMatchesScore THEN
                DISPLAY newHighScore
                Settings.highestNameGuesserSuddenDeathHardMatchesScore = matches
            END IF
            Increment Settings.totalNameGuesserSuddenDeathHardGames
    ENDCASE
    Invoke ("Show Game Over Panel", 0.6f)
END SDGameOver
```

Profile Picture Guesser Pseudocode

Start Function (called before the first frame is loaded)

```
BEGIN
    LoadPlayer
    gameIsSuddenDeath = false
    xCoords(0) = -660
    yCoords(0) = -145
    ...
    xCoords(9) = 660
    yCoords(9) = -475
    FOR i = 0 to 79
        REPEAT
            throwaway2(i) = Random.Range(0,79)
            UNTIL usernamesToDisplay[throwaway2(i)] <> true
            usernamesToDisplay[throwaway2(i)] = true
            PRINT "The username of place " + i + " was taken by " + throwaway2[i]
    NEXT i
    currentPlaceInOrder = 0
    gameImages(0).sprite = pictures[throwaway2(currentPlaceInOrder)]
    whatIs.text = "What is " + pictures[throwaway2[currentPlaceInOrder]].name + "'s profile
picture?"
    usernamesToDisplay1[throwaway2(currentPlaceInOrder)] = true
    FOR i = 1 to 9
        REPEAT
            throwaway3(i) = Random.Range(0,79)
            UNTIL usernamesToDisplay1[throwaway3(i)] <> true
            usernamesToDisplay1[throwaway3(i)] = true
            gameImages(i).sprite = pictures[throwaway3(i)]
            PRINT "The picture (" + throwaway3[i] + ") is in the tile (" + i + ")."
    NEXT i
    FOR i = 0 to 9
        REPEAT
            throwaway4(i) = Random.Range(0,9)
            UNTIL usernamesToDisplay2[throwaway4(i)] <> true

            usernamesToDisplay[throwaway4(i)] = true
            Tiles[throwaway4(i)].anchoredPosition = new Vector3(xCoords(i), yCoords(i))
            PRINT "The tile (" + throwaway4[i] + ") is in the location of tile (" + i + ")."
    NEXT i
    CASEWHERE current =
        0: timeRemaining = 60
        1: timeRemaining = 45
        2: timeRemaining = 30
    ENDCASE
    timerIsRunning = true
    IF Settings.suddenDeath = true THEN
        highScoreMistakesObject.SetActive = false
        gameIsSuddenDeath = true
```

```
    Settings.suddenDeath = false
END IF
END
```

Update Function (called every frame)

```
BEGIN Update
    matchesText.text = "Matches: " + matches
    IF gameIsSuddenDeath = true
        CASEWHERE current =
            0: highScoreMatches.text = "Matches: " +
                Settings.highestProPicGuesserSuddenDeathEasyMatchesScore
            1: highScoreMatches.text = "Matches: " +
                Settings.highestProPicGuesserSuddenDeathMediumMatchesScore
            2: highScoreMatches.text = "Matches: " +
                Settings.highestProPicGuesserSuddenDeathHardMatchesScore
        ENDCASE
    ELSE
        mistakesText.text = "Mistakes: " + mistakes
        CASEWHERE current =
            0: highScoreMatches.text = "Matches: " +
                Settings.highestProPicGuesserRegularEasyMatchesScore
            highScoreMistakes.text = "Mistakes: " +
                Settings.highestProPicGuesserRegularEasyMistakesScore
            1: highScoreMatches.text = "Matches: " +
                Settings.highestProPicGuesserRegularMediumMatchesScore
            highScoreMistakes.text = "Mistakes: " +
                Settings.highestProPicGuesserRegularMediumMistakesScore
            2: highScoreMatches.text = "Matches: " +
                Settings.highestProPicGuesserRegularHardMatchesScore
            highScoreMistakes.text = "Mistakes: " +
                Settings.highestProPicGuesserRegularHardMistakesScore
        ENDCASE
    END IF
    float seconds = Mathf.FloorToInt(timeRemaining MOD 60)
    timerText.text = "Timer: " + seconds + "s"
    IF timerIsRunning = true THEN
        IF timeRemaining > 0 THEN
            timeRemaining = timeRemaining - Time.deltaTime
        ELSE
            IF gameIsSuddenDeath = true THEN
                SDGameOver
            ELSE
                GameOver
            END IF
            timeRemaining = 0
            timerIsRunning = false
        END IF
    END IF
END Update
```

```
BEGIN AddMistakes
    Increment mistakes
    Increment Settings.totalMistakesInProPicGuesser
    CASEWHERE currentlyPressedNum =
        1: gameButtons(currentlyPressedNum).enabled = false
        Animator.SetTrigger("Tile2Mistaken")
        2: gameButtons(currentlyPressedNum).enabled = false
        Animator.SetTrigger("Tile3Mistaken")
        3: gameButtons(currentlyPressedNum).enabled = false
        Animator.SetTrigger("Tile4Mistaken")
        4: gameButtons(currentlyPressedNum).enabled = false
        Animator.SetTrigger("Tile5Mistaken")
        5: gameButtons(currentlyPressedNum).enabled = false
        Animator.SetTrigger("Tile6Mistaken")
        6: gameButtons(currentlyPressedNum).enabled = false
        Animator.SetTrigger("Tile7Mistaken")
        7: gameButtons(currentlyPressedNum).enabled = false
        Animator.SetTrigger("Tile8Mistaken")
        8: gameButtons(currentlyPressedNum).enabled = false
        Animator.SetTrigger("Tile9Mistaken")
        9: gameButtons(currentlyPressedNum).enabled = false
        Animator.SetTrigger("Tile10Mistaken")
    ENDCASE
END AddMistakes
```

```
BEGIN AddMatches
    Increment matches
    Increment Settings.totalMatchesInProPicGuesser
    FOR i = 0 to 9
        gameButtons(i).enabled = false
        PRINT "The tile (" + i + ") is disabled"
    NEXT i
END AddMatches
```

```
BEGIN GameOver
    timerIsRunning = false
    FOR i = 0 to 9
        gameButtons(i).enabled = false
    NEXT i
    Animator.SetTrigger("GameDone")
    CASEWHERE current =
        0:
        IF matches > Settings.highestProPicGuesserRegularEasyMatchesScore THEN
            newHighScore.SetActive = true
            Settings.highestProPicGuesserRegularEasyMatchesScore = matches
            Settings.highestProPicGuesserRegularEasyMistakesScore = mistakes
        ELSE IF matches = highestProPicGuesserRegularEasyMatchesScore AND mistakes <
            Settings.highestProPicGuesserRegularEasyMistakesScore THEN
                newHighScore.SetActive = true
                Settings.highestProPicGuesserRegularEasyMistakesScore = mistakes
```

```
END IF
Increment Settings.totalProPicGuesserRegularEasyGames
1:
IF matches > Settings.highestProPicGuesserRegularMediumMatchesScore THEN
    newHighScore.SetActive = true
    Settings.highestProPicGuesserRegularMediumMatchesScore = matches
    Settings.highestProPicGuesserRegularMediumMistakesScore = mistakes
ELSE IF matches = highestProPicGuesserRegularMediumMatchesScore AND mistakes
< Settings.highestProPicGuesserRegularMediumMistakesScore THEN
    newHighScore.SetActive = true
    Settings.highestProPicGuesserRegularMediumMistakesScore = mistakes
END IF
Increment Settings.totalProPicGuesserRegularMediumGames
2:
IF matches > Settings.highestProPicGuesserRegularHardMatchesScore THEN
    newHighScore.SetActive = true
    Settings.highestProPicGuesserRegularHardMatchesScore = matches
    Settings.highestProPicGuesserRegularHardMistakesScore = mistakes
ELSE IF matches = highestProPicGuesserRegularHardMatchesScore AND mistakes
< Settings.highestProPicGuesserRegularHardMistakesScore THEN
    newHighScore.SetActive = true
    Settings.highestProPicGuesserRegularHardMistakesScore = mistakes
END IF
Increment Settings.totalProPicGuesserRegularHardGames
ENDCASE
Invoke("ShowGameOverPanel", 0.6f)
END GameOver
```

```
BEGIN ShowGameOverPanel
    DISPLAY gameOverPanel
    whatIsObject.SetActive = false
END ShowGameOverPanel
```

```
BEGIN DebugTest
    PRINT "Matches = " + matches
    PRINT "Mistakes = " + mistakes
    PRINT "-----"
END DebugTest
```

The following function is called when the player clicks on the correct button

```
BEGIN CorrectButton
    Animator.SetTrigger("GameOver")
    AddMatches
    _____ Invoke("ResetTheBoard", 0.6f)
END CorrectButton
```

```
BEGIN ResetTheBoard
    Increment currentPlaceInOrder
    gameImages(0).sprite = pictures[throwaway2(currentPlaceInOrder)]
    whatIs.text = "What is " + pictures[throwaway2(currentPlaceInOrder)].name + "'s profile
```

```
picture?"  
FOR i = 0 to 79  
    usernamesToDisplay1(i) = false  
NEXT i  
FOR i = 0 to 9  
    usernamesToDisplay2(i) = false  
NEXT i  
usernamesToDisplay1[throwaway2(currentPlaceInOrder)] = true  
FOR i = 1 to 9  
    REPEAT  
        throwaway3(i) = Random.Range(0, 79)  
        UNTIL usernamesToDisplay1[throwaway3(i)] <> true  
        usernamesToDisplay1[throwaway3(i)] = true  
        gameImages(i).sprite = pictures[throwaway3(i)]  
        PRINT "The picture (" + throwaway3[i] + ") is in the tile (" + i + ")."  
    NEXT i  
    FOR i = 0 to 9  
        REPEAT  
            throwaway4(i) = Random.Range(0, 9)  
            UNTIL usernamesToDisplay2[throwaway2(throwaway4[i])] <> true  
            usernamesToDisplay2[throwaway4(i)] = true  
            Tiles[throwaway4(i)].anchoredPosition = new Vector3(xCoords(i), yCoords(i))  
            PRINT "The tile (" + throwaway4(i) + ") is in the location of tile (" + i + ")."  
    NEXT i  
    FOR i = 0 to 9  
        gameButtons(i).enabled = true  
        PRINT "The tile (" + i + ") is enabled"  
    NEXT i  
    Animator.SetTrigger("ResetBoard")  
END ResetTheBoard
```

The following function is called when the player clicks on the wrong button while playing the Sudden Death difficulties

```
BEGIN SDGameOver  
    timerIsRunning = false  
    FOR i = 0 to 9  
        gameButtons(i).enabled = false  
    NEXT i  
    Animator.SetTrigger("GameDone")  
    CASEWHERE current =  
        0:  
            IF matches > Settings.highestProPicGuesserSuddenDeathEasyMatchesScore THEN  
                DISPLAY newHighScore  
                Settings.highestProPicGuesserSuddenDeathEasyMatchesScore = matches  
            END IF  
            Increment Settings.totalProPicGuesserSuddenDeathEasyGames  
        1:  
            IF matches > Settings.highestProPicGuesserSuddenDeathMediumMatchesScore  
            THEN  
                DISPLAY newHighScore
```

```
    Settings.highestProPicGuesserSuddenDeathMediumMatchesScore = matches
END IF
Increment Settings.totalProPicGuesserSuddenDeathMediumGames
2:
IF matches > Settings.highestProPicGuesserSuddenDeathHardMatchesScore THEN
    DISPLAY newHighScore
    Settings.highestProPicGuesserSuddenDeathHardMatchesScore = matches
END IF
Increment Settings.totalProPicGuesserSuddenDeathHardGames
ENDCASE
Invoke("ShowGameOverPanel", 0.6f);
END SDGameOver
```

The following function is called when the player clicks on the wrong button while playing the Regular difficulties

```
BEGIN WrongButton2
    currentlyPressedNum = 1
    AddMistakes
END WrongButton2
...
BEGIN WrongButton10
    currentlyPressedNum = 9
    AddMistakes
END WrongButton10
```

Program Testing

You are to prepare a suitable plan for how your program will be tested for logic and runtime errors. This plan should include (where appropriate) suitable test data that will be used to test the program.

Should you have a program where traditional test data is not suitable you will need to detail how your program can be tested to ensure that “all paths” through the program operate correctly.

Testing Plan

Overview

Throughout the BFC Playground there were multiple types of test data that I needed to emulate when experimenting with the game modes and their respective difficulties. Unity has a window called the *Console*, that displays readable text that is written in C# while the program is running for the purpose of assisting developers debug their applications. This was primarily the source of my testing as I was creating this game because it allowed me to know, in the instance of running into a runtime error or overflow error (such as making an endless loop or referencing a variable that does not exist) or logic error (such as referencing a variable that has not been given a value). This is essentially done to ensure that all paths of the program operate correctly.

Memory Game

In the *Memory Game* a lot of the testing took place before the first frame of the game is presented to the player. This would be in the “*Start*” function/method of the algorithm and entails the need to check:

- That all of the boolean that belong to the random values regarding the members of the BFC have been set to false.
- Which of the 82 members of the BFC have been randomly chosen through displaying their name and picture index in the console.
- That all of the boolean that belong to the random values regarding the tile locations have been set to false.
- Where the chosen BFC members have been placed on the board.

In the “*DebugTest*” function/method of the algorithm, the following segments are displayed in the console whenever the player clicks a tile on the board:

- Which button has been clicked
- What is the current amount of matches
 - Which of the tiles have been matched
- What is the current amount of mistakes

Higher or Lower Follower Age

Similarly, in the *Higher or Lower Follower Age* game mode, there is testing that also takes place before the first frame of the game is presented to the player. In the “*Start*” function/method of the algorithm the:

- 3 members of the BFC that have been randomly selected will have their name and picture index displayed in the console.
- Chosen member’s month and year of following will be displayed in the console.

In the “*GuessButton*” function/method of the *Regular* game modes’ algorithm, the following segments were displayed in the console whenever the player clicked the guess button after choosing a month and year through the dropdowns:

- The month that the user guessed the given BFC member started following
- The year that the user guessed the given BFC members started following
- If the user chose correctly, then a message saying so will be displayed in the console.

Likewise, in the “*SDGuessButton*” function/method of the *Sudden Death* game modes’ algorithm, the month and year messages are displayed in the console whenever the player clicks the guess button as well as notifying the programmer if a correct guess was made in the console - which triggers the “*SDAddMatches*” function/method. But if the user makes a mistake, the “*SDAddMistakes*” function/method will be called that states in the console that the game has ended with the player losing.

Profile Picture Guesser

In the *Profile Picture Guesser* a lot of the testing took place before the first frame of the game is presented to the player. This would be in the “*Start*” function/method of the algorithm and entails the need to check:

- The random order of the BFC members
- In which tiles are each of the wrong pictures placed
- Where each of the tiles have been placed on the board

In the “*DebugTest*” function/method of the algorithm, the following messages are displayed in the console whenever the player clicks a tile on the board:

- What is the current amount of matches
- What is the current amount of mistakes

In the “*AddMatches*” function/method, a message is shown that states that the button component of each of the tiles has been disabled so the player can not click on a button before the “*ResetTheBoard*” function/method has been called. Within “*ResetTheBoard*”, similar to the “*Start*” function/method, a message states in which of the tiles are the wrong pictures placed and where each of the tiles are on the board, but dissimilarly, it also shows that all of the button components have been reenabled.

Twitch Username Guesser

In the *Twitch Username Guesser* a lot of the testing took place before the first frame of the game is presented to the player. This would be in the “*Start*” function/method of the algorithm and entails the need to check:

- The random order of the BFC members
- In which tiles are each of the wrong names placed
- Where each of the tiles have been placed on the board

In the “*DebugTest*” function/method of the algorithm, the following messages are displayed in the console whenever the player clicks a tile on the board:

- What is the current amount of matches
- What is the current amount of mistakes

In the “*AddMatches*” function/method, a message is shown that states that the button component of each of the tiles has been disabled so the player can not click on a button before the “*ResetTheBoard*” function/method has been called. Within “*ResetTheBoard*”, similar to the “*Start*” function/method, a message states in which of the tiles are the wrong names placed and where each of the tiles are on the board, but dissimilarly, it also shows that all of the button components have been reenabled.

External Files / Documents

Coded Program (refer to User Manual for installation guide):

<https://andrewyoussef.itch.io/bfc-playground>

Sketches:

https://drive.google.com/drive/folders/1p2U8y_NsZlwKzdrfWbj5h0p-6BFM8VhF?usp=sharing

Structure Charts:

https://drive.google.com/drive/folders/1wPZfJAR4pyfH80t69rAC60G_BpXEubNK?usp=sharing

User Manual:

https://docs.google.com/document/d/1BGwrp_GqyKWpQJF7wYdhiSsqxICULnitatgdCFPuxOg/edit?usp=sharing

Technical Manual:

https://docs.google.com/document/d/1d1_jznRsE6q_FFh7KhVOtFRlcHpcd8G7A3DiggPDz7A/edit?usp=sharing