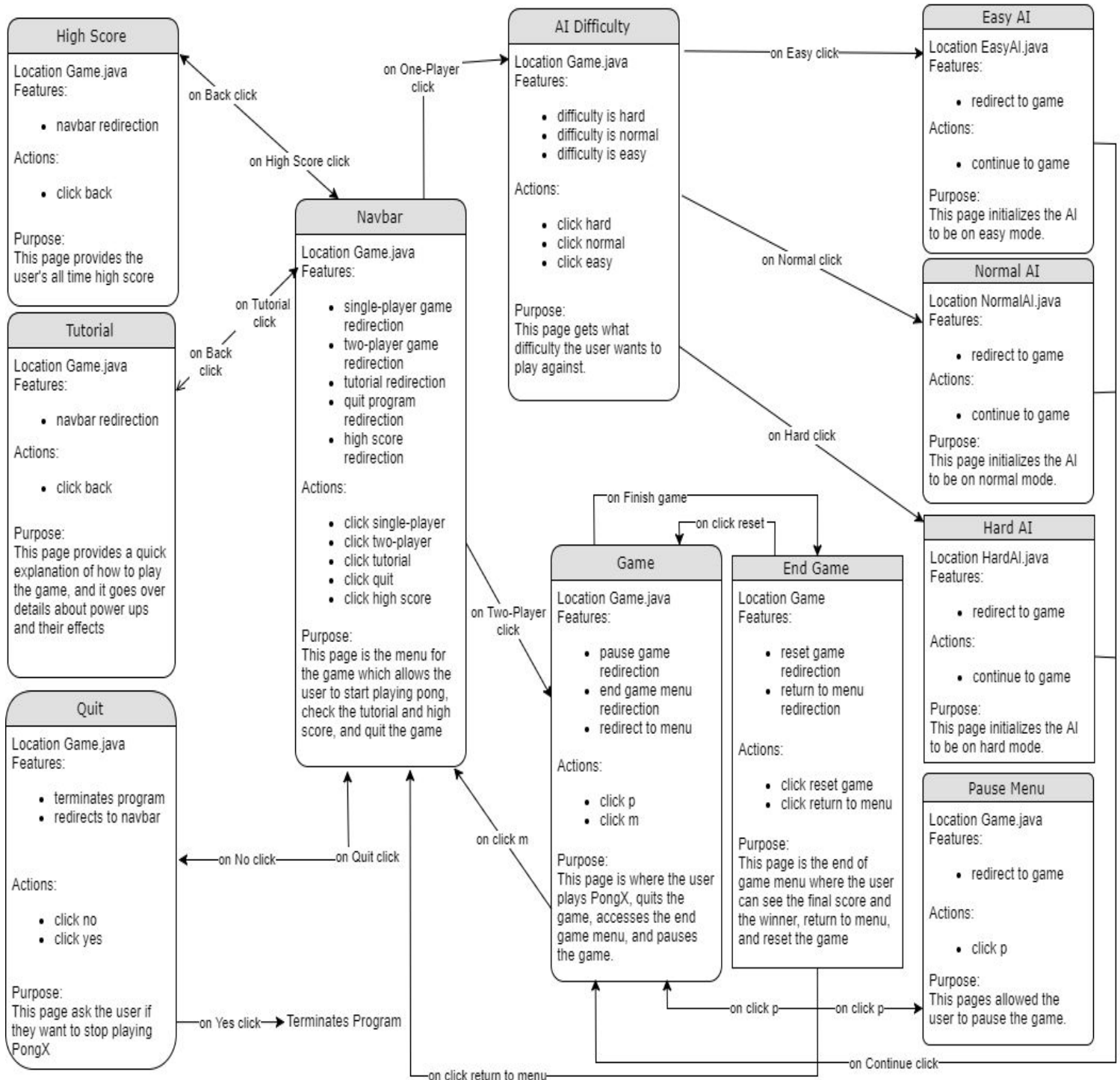# PongX

## Team 12 -  Sprint 2 Incremental and Regression Testing

**Tomasz Parzadka, Michael Quade, Ashwin Prasad, Kyle Bentain, Nathanael Monaghan**

# Components

## High Score

Location Game.java
Features:

- navbar redirection

Actions:

- click back

Purpose:
This page provides the user's all time high score

## Tutorial

Location Game.java
Features:

- navbar redirection

Actions:

- click back

Purpose:
This page provides a quick explanation of how to play the game, and it goes over details about power ups and their effects

## Quit

Location Game.java
Features:

- terminates program
- redirects to navbar

Actions:

- click no
- click yes

Purpose:
This page ask the user if they want to stop playing PongX

*on Back click*

*on High Score click*

*on Tutorial click*

*on Back click*

## Navbar

Location Game.java
Features:

- single-player game redirection
- two-player game redirection
- tutorial redirection
- quit program redirection
- high score redirection

Actions:

- click single-player
- click two-player
- click tutorial
- click quit
- click high score

Purpose:
This page is the menu for the game which allows the user to start playing pong, check the tutorial and high score, and quit the game

*on One-Player click*

*on Two-Player click*

*on No click* — *on Quit click*

*on Yes click* → Terminates Program

*on click return to menu*

## AI Difficulty

Location Game.java
Features:

- difficulty is hard
- difficulty is normal
- difficulty is easy

Actions:

- click hard
- click normal
- click easy

Purpose:
This page gets what difficulty the user wants to play against.

*on Easy click*

*on Normal click*

*on Hard click*

## Game

Location Game.java
Features:

- pause game redirection
- end game menu redirection
- redirect to menu

Actions:

- click p
- click m

Purpose:
This page is where the user plays PongX, quits the game, accesses the end game menu, and pauses the game.

*on Finish game*

*on click reset*

*on click m*

## End Game

Location Game
Features:

- reset game redirection
- return to menu redirection

Actions:

- click reset game
- click return to menu

Purpose:
This page is the end of game menu where the user can see the final score and the winner, return to menu, and reset the game

*on click p* — *on click p*

## Easy AI

Location EasyAI.java
Features:

- redirect to game

Actions:

- continue to game

Purpose:
This page initializes the AI to be on easy mode.

## Normal AI

Location NormalAI.java
Features:

- redirect to game

Actions:

- continue to game

Purpose:
This page initializes the AI to be on normal mode.

## Hard AI

Location HardAI.java
Features:

- redirect to game

Actions:

- continue to game

Purpose:
This page initializes the AI to be on hard mode.

## Pause Menu

Location Game.java
Features:

- redirect to game

Actions:

- click p

Purpose:
This pages allowed the user to pause the game.

*on Continue click*

# Incremental Testing

- **Defined Components: See design above for defined components.**

- **Type of Incremental Testing Used: We implemented top-down incremental testing since it would be very difficult to create drivers to test less critical modules such as power-ups. It is far more efficient to test the main modules first using stubs and then building from there. Thus, we tested Game and AI first, followed by the less critical modules such as power-ups and Menu.**

**Menu:**

| Defect No. | Description | Severity | How To Correct |
|:---:|---|:---:|---|
| 1 | One-player button would start two-player game. | 1 | Adjusted the coordinates of the button to properly execute correct game. |
| 2 | Two Player button did not have any functionality | 1 | Added a MouseInput event for the area of the two player button |

**Paddles:**

| Defect No. | Description | Severity | How To Correct |
|:---:|---|:---:|---|
| 1 | Paddles were moving slowly | 1 | See below. |
| 2 | In Two player, you could not move both paddles at the same time. | 2 | Started using flags for when a key was pressed, and checked them during the main actionPerformed update loop. |

**Ball:**

| Defect No. | Description | Severity | How To Correct |
|:---:|---|:---:|---|
| 1 | When the ball would first appear, the user could not reach it | 3 | Increase the paddle's speed and reduce the angle of the ball |

**Game:**

| Defect No. | Description | Severity | How To Correct |
|:---:|---|:---:|---|
| 1 | Game would not be able to start again | 1 | Removed everything from being painted and added a new END |

| Defect No. | Description | Severity | How To Correct |
|---|---|---|---|
| | after ending. | | state that would restart game. |
| 2 | Game would not go to menu after ending. | 1 | Removed everything from being painted and game state would change to Menu if selected. |
| 3 | Single player game would launch after a user would like to play two-player game again | 1 | Recorded the previous state to launch correct game, being single-player or two-player. |

**Powerups:**

| Defect No. | Description | Severity | How To Correct |
|---|---|---|---|
| 1 | Paddle stops when gets Decrease Paddle Speed at initial speed | 2 | Added check for when the paddle speed is 1 and if it is, the powerup doesn't apply to the paddle |
| 2 | Ball doesn't reset to initial size after inc/dec ball size | 3 | Reset the balls based off size and not multiples |
| 3 | Paddle doesn't reset to initial size after inc/dec ball size | 3 | Reset the paddles based off size and not multiples |
| 4 | Add ball powerup adds 2 extra balls instead of just 1 | 3 | Removed upgrades from upgrade arraylist right after hitting paddle |
| 5 | When a person has more than one paddle, they don't stay apart; they collide and merge into one paddle. | 2 | If one paddles can't move, then all of the paddles now can't move |
| 6 | Upgrades gave an outofboundsexceptio n | 3 | Added checks to make sure indexes referenced through going through upgrades arraylist were valid |
| 7 | Sticky Paddle powerup doesn't release ball when powerup times out | 2 | |

| 8 | Both paddles didn't stop when add paddle is added and one of the paddles hits a wall | 2 | Added a check for add paddle to not move any paddles if one of the paddles couldn't move within the window |
| --- | --- | --- | --- |

**AI:**

| Defect No. | Description | Severity | How To Correct |
| --- | --- | --- | --- |
| 1 | Would search for a ball even when none available and crash game. | 1 | Made a check to see if ball was available before starting AI. |
| 2 | AI difficulty selection would not execute correct AI. | 2 | Created a getter and setter that would launch the correct AI when user selects difficulty. |
| 3 | If there was more than 1 ball, the AI would only follow the starting ball | 2 | Updated the AI that it would trace the closest ball from the paddle |
| 4 | The AI wouldn't search for the closet power up | 3 | Changed the algorithm for finding the closest power up |

**Scores:**

| Defect No. | Description | Severity | How To Correct |
| --- | --- | --- | --- |
| 1 | Would access null scores | 2 | Created a check to make sure null scores would not print |
| 2 | Score would reset after getting repainted | 2 | Created an additional variable to store score which would be printed instead |
| 3 | Going back to the menu wouldn't reset the score | 3 | Reset the score after invoking change to menu state |
| 4 | Scores would be printed off screen | 2 | Set a limit to 20 previous scores being viewed. |

**Pause:**

| Defect No. | Description | Severity | How To Correct |
|---|---|---|---|
| 1 | Text wouldn't be drawn when paused | 3 | Repainted whenever program was not in GAME state |
| 2 | Pausing didn't stop movement of balls | 1 | Set condition for balls to move only during GAME state |
| 3 | Pausing resetted the ball's positions | 1 | Froze the ball's positions instead of resetting to initial position |
| 4 | Unpausing during two player game turned it into a one player game | 2 | Made unpause revert to previous game state |

## Regression Testing

| Defect No. | Description | Severity | How To Correct |
|---|---|---|---|
| 1 | Frame state would only repaint once. | 1 | Changed the check for repaint() to allow all buttons to function correctly. |
| 2 | Score button functionality is not working. | 2 | Stored the score in an arraylist before game world reset and listed the previous scores in order. Created coordinates that allowed for the display of scores. |
| 3 | Paddle KeyListeners kept losing focus, and would sometimes not move. | 2 | Switched from using KeyListeners to setting KeyBindings. |
| 4 | Some balls were not removed when the user would reset game | 1 | Call on resetPower when the user would click reset game |
| 5 | Powerups didn't reset properly | 2 | added timer to each type of powerup and once they hit the end, |
| 6 | Powerups for Top Paddles didn't work | 2 | Added the same implementation as for the bottom paddles but for the top paddles while changing items and calculations |

| 7 | Would overwrite the previous score. | 2 | Multiplied the y-axis by the number of scores inside array to prevent overlap. |
|---|---|---|---|