ECE 493 Capstone Test Plan and Suite for PolyPong

Arun Woosaree, Michael Antifaoff, Joshua Chang

April 15, 2021

1 Test Plan

Overall, the tests for the backend code are automated, and will have over 80% line coverage, as outlined in the software requirements specification document. Because the backend is using the Deno runtime, we are using a built-in testing tool. In conjunction with this, we are also using a library called 'superoak', which is designed to help with testing oak applications. Oak is a framework we use for routing requests in the server.

The frontend code will be subject to manual testing, due to the difficulty with automating tests for a multiplayer game. We do have an automated test which clicks buttons and navigates the site to ensure that our navigation works, and certain things are displayed correctly. However, we found that manual tests were necessary to ensure that our functional requirements are satisfied and the application works as expected. The automated test which click buttons and navigates around is written using the puppeteer framework. Puppeteer is a tool written by Google, which allows one to automate actions in the Chrome browser. With the combination of this tool plus manual testing, we are confident that the lines covered far exceed 80%, however, since a coverage report is not automatically generated like in automated tests, we do not know the exact percentage of lines covered.

1.1 User Login

1.1.1 FR1

Setting up a repeatable test that covers everything for this feature would be difficult to say the least, because it would require burning email accounts. That is, once an email address is used to sign up, it cannot be re-used to

test this functional requirement. As a result, a new email would need to be created each time we would want to run the test. A compromise would be to delete the email in our database, so that the user is still new to us, even though the OAuth provider (In this case, Auth0) would still remember that the user had previously authorized our application.

An automated test will test the server side code, while a manual test will ensure that a user can in fact, log in.

1.2 User Registration

1.2.1 FR2

Setting up a test for login will also be slightly difficult. For security reasons, both GitHub and Google (the social login providers we chose) occasionally require a code to be entered which is sent by e-mail. For this reason, an manual test is required to test the frontend part of this feature.

An automated test will test the server side code, while a manual test will ensure that a user can in fact, log in.

1.3 Play Game

1.3.1 FR3 Create Game

To test this functionality, at least three users are required to test the requirement that the game can start when three or more players are present as outlined in the Software Requirements Specification. This needs to happen in three separate browser windows either on the same computer, or on different computers. This makes an automated test difficult, so a manual test should be designed instead.

1.3.2 FR4 Share Link

To test this functionality, a link needs to be copied, and then pasted into a new browser window. Even if you ignore the security implications of using a script to copy clipboard contents, this requires opening a new browser window and pasting a link. A manual test will be designed for this feature.

1.3.3 FR5 Join Game

To test this functionality, a lobby id would have to already exist, meaning that this test would depend on the create game functionality, which will be a manual test. Therefore, a manual test will also be designed for this feature.

1.3.4 FR6 Play Game

To test this functionality, one would have to actually play the game. There are a lot of things that can happen in a game, and many scenarios to think about that can happen in the game. There are virtually infinite possibilities of combinations of game states, which depend on things like:

- the number of players
- the positions of paddles
- the powerups used
- network latency

We found from preliminary playtesting that the game gets really chaotic past about 12 players. Beyond that point, the distance you can move a paddle becomes really small. So, the test will start with 12 players, each with powerups such that all of the power ups will be used multiple times throughout the game, and will run until there is one player left. Also, it will be a manual test because of the amount of players required.

1.4 Leaderboard and Statistics

1.4.1 FR7 Earn XP

Parts of this feature can be tested with automated tools. In the backend, an automated test can be made to ensure that the function which adds XP to a user updates the database correctly.

For a player to actually earn XP however, they must play the game. So, the other part of this test would be to ensure that when a game ends, the correct amount of XP is awarded. This will be a manual test case.

1.4.2 FR8 View Local Leaderboard

Parts of this feature can be tested with automated tools. In the backend, an automated test can be made to ensure that the function which returns the data that the frontend requests is in the correct format.

A manual test will be used to ensure that the correct data is shown in the frontend. Another acceptable solution would be to have a unit test in the frontend which mocks the response from the server and ensures the data is displayed correctly.

1.4.3 FR9 View Global Leaderboard

Parts of this feature can be tested with automated tools. In the backend, an automated test can be made to ensure that the function which returns the data that the frontend requests is in the correct format.

A manual test will be used to ensure that the correct data is shown in the frontend. Another acceptable solution would be to have a unit test in the frontend which mocks the response from the server and ensures the data is displayed correctly.

1.4.4 FR10 User statistics

Parts of this feature can be tested with automated tools. In the backend, an automated test can be made to ensure that the function which returns the user statistics data that the frontend requests is in the correct format.

A manual test will be used to ensure that the correct data is shown in the frontend. Another acceptable solution would be to have a unit test in the frontend which mocks the response from the server and ensures the data is displayed correctly.

1.5 Power Ups

1.5.1 FR11 Power Ups

There are 11 Choose 3 = 165 possible combinations of power up choices that users can make. Given more time, we could make that many test cases, however, because a game has multiple players, we could have a game with say, 11 players (because there are 11 power ups in total), and get each player to pick one power up. This is a much more reasonable solution, and we still ensure that all of the power ups can be picked. Each power up is still has their respective tests for ensuring that the power ups itself works as expected.

1.5.2 FR12 Expanded Paddle

This functional requirement is a power up, and a power up is activated during a game. The actions that happen in a game cannot be automated, so testing is to be done manually.

1.5.3 FR13 Shrink Paddle

This functional requirement is a power up, and a power up is activated during a game. The actions that happen in a game cannot be automated, so testing

is to be done manually.

1.5.4 FR14 Self Invisible Paddle

This functional requirement is a power up, and a power up is activated during a game. The actions that happen in a game cannot be automated, so testing is to be done manually.

1.5.5 FR15 Others Invisible Paddle

This functional requirement is a power up, and a power up is activated during a game. The actions that happen in a game cannot be automated, so testing is to be done manually.

1.5.6 FR16 Invisible Ball

This functional requirement is a power up, and a power up is activated during a game. The actions that happen in a game cannot be automated, so testing is to be done manually.

1.5.7 FR17 Self Curved Outwards Paddle

This functional requirement is a power up, and a power up is activated during a game. The actions that happen in a game cannot be automated, so testing is to be done manually.

1.5.8 FR18 Self Curved Inwards Paddle

This functional requirement is a power up, and a power up is activated during a game. The actions that happen in a game cannot be automated, so testing is to be done manually.

1.5.9 FR19 Self Bumpy Paddle

This functional requirement is a power up, and a power up is activated during a game. The actions that happen in a game cannot be automated, so testing is to be done manually.

1.5.10 FR20 Distracting Background

This functional requirement is a power up, and a power up is activated during a game. The actions that happen in a game cannot be automated, so testing is to be done manually.

1.5.11 FR21

This optional functional requirement was scrapped. Therefore, the feature does not exist, so no tests are required for a feature that does not exist.

1.5.12 FR22

This optional functional requirement was scrapped. Therefore, the feature does not exist, so no tests are required for a feature that does not exist.

1.5.13 FR23 Add Ball

This functional requirement is a power up, and a power up is activated during a game. The actions that happen in a game cannot be automated, so testing is to be done manually.

1.5.14 FR24

This optional functional requirement was scrapped. Therefore, the feature does not exist, so no tests are required for a feature that does not exist.

1.5.15 FR25

This optional functional requirement was scrapped. Therefore, the feature does not exist, so no tests are required for a feature that does not exist.

1.5.16 FR26 Path Trace

This functional requirement is a power up, and a power up is activated during a game. The actions that happen in a game cannot be automated, so testing is to be done manually.

1.6 Skins

1.6.1 FR27 Earn Skin

Parts of this feature can be tested with automated tools. In the backend, an automated test can be made to ensure the following:

- the function which returns the available skins (based on current XP) is correct
- the user can select a new skin when the XP threshold for that skin is earned

  A manual test would be needed to ensure that the user can actually select a skin when the skin has been earned

1.6.2 FR28 Select skin

Parts of this feature can be tested with automated tools. In the backend, an automated test can be made to ensure the following:

- the function which returns the available skins (based on current XP) is correct
- the function that sets the user's skin works, and does not allow a user to set a skin they have not unlocked yet

A manual test would be needed to ensure that the user's currently selected skin is visible when the game starts, since testing the game is also done manually and cannot be automated.

2 Test Suite

2.1 User Login

2.1.1 FR1

This is a manual test

1. Test steps
   (a) Go to `https://polypong.ca`
   (b) Click on the `Sign Up/Log In` buttons
   - you will be redirected to log in to either your github or google account

2. Expectations

   (a) When the login process is finished, you should be redirected to the home page. If it was successful, you will see two new buttons on the home page: `My Stats and Leaderboard`, and `Settings`. Also, the `Sign Up/Log In` button should no longer be visible.

   (b) In the developer tools for your browser, you should see a cookie set by auth0:

   - Name: auth0.isauthenticated
   - Value: true

2.2 User Registration

2.2.1 FR2

On the server side, there is an automated test. Use the provided test.sh script in the server/ folder to run it, along with other tests

A manual test is used to test the functionality from the user's perspective, and also to test portions of code that are not covered by automated tests.

1. Test steps

   (a) Pick a Google or GitHub account with an email that does not already exist in the database

   - if the user already exists, either delete the entry in the database, or create a new GitHub/Google account.

   (b) Go to `https://polypong.ca`

   (c) Click on the `Sign Up/Log In` buttons

   - you will be redirected to log in to either your github or google account

   (d) When login in to Google or GitHub is done, you should be redirected to a sign up page where you are asked to choose a username for your new account

   (e) Type a username that already exists in the database. (If one is not known already, you can look at the leaderboard to see some already existing usernames)

   (f) Hit `Create Account`

   (g) Change the username requested to something unique

   (h) Hit `Create Account`

2. Expectations

   (a) When the sign up process is finished, you should be redirected to the home page. If it was successful, you will see two new buttons on the home page: `My Stats and Leaderboard`, and `Settings`. Also, the `Sign Up/Log In` button should no longer be visible.

   (b) In the developer tools for your browser, you should see a cookie set by auth0:

   - Name: auth0.isauthenticated
   - Value: true

   (c) On step 6, when a user attempts to sign up with an already existing username, an error should notify the user that the requested username is taken.

   (d) Also on step 6, on the network tab of your browser's developer tools, a request to `https://polyserver.polypong.ca/signup` should be made, with the response status code being `409 Conflict` indicating that the username is already taken

   (e) After step 8, you should be redirected to the home page for polypong.ca. If it was successful, you will see two new buttons on the home page: `My Stats and Leaderboard`, and `Settings`. Also, the `Sign Up/Log In` button should no longer be visible.

   (f) In the developer tools for your browser, you should see a cookie set by auth0:

   - Name: auth0.isauthenticated
   - Value: true

2.3 Play Game

2.3.1 FR3 Create Game

This is a manual test

1. Test steps

   (a) Go to `https://polypong.ca`

   (b) Click on `Create Private Game`

2. Expectations

(a) You should see that you are put into a new lobby, and that you are waiting for users to join the lobby.

(b) You should also see a lobby id

2.3.2 FR4 Share Link

This is a manual test

1. Test steps
   (a) Go to `https://polypong.ca`
   (b) Click on `Create Private Game`
   (c) Click on `Copy Link to Clipboard to Invite Friends`
2. Expectations
   (a) You should see that you are put into a new lobby, and that you are waiting for users to join the lobby.
   (b) You should also see a lobby id
   (c) When step 3 above is completed, a link should be copied to your system clipboard

2.3.3 FR5 Join Game

This is a manual test

1. Test steps
   (a) Go to `https://polypong.ca`
   (b) Click on `Create Private Game`
   (c) Click on `Copy Link to Clipboard to Invite Friends`
   (d) Open a new browser window, paste the link, and hit enter
2. Expectations
   (a) You should see that you are put into a new lobby, and that you are waiting for users to join the lobby.
   (b) You should also see a lobby id
   (c) When step 3 above is completed, a link should be copied to your system clipboard
   (d) When step 4 is completed, there should now be two players in the lobby

2.3.4 FR6 Play Game

This is a manual test

1. Test steps
   (a) Go to `https://polypong.ca`
   (b) Click on `Create Private Game`
   (c) Click on `Copy Link to Clipboard to Invite Friends`
   (d) Open a new browser window, paste the link, and hit enter
   (e) Repeat step 4, so that a third person joins the lobby
   (f) Have all users in the lobby click on the `Let's Play` button at the bottom of the page
2. Expectations
   (a) You should see that you are put into a new lobby, and that you are waiting for users to join the lobby.
   (b) You should also see a lobby id
   (c) When step 3 above is completed, a link should be copied to your system clipboard
   (d) When step 4 is completed, there should now be two players in the lobby
   (e) When step 5 is completed, there should now be three players in the lobby
   (f) When step 6 is completed, the game should start and users can start playing.

2.4 Leaderboard and Statistics

2.4.1 FR7 Earn XP

This is a manual test

1. Test steps
   (a) Go to `https://polypong.ca`
   (b) Click on `Create Private Game`
   (c) Click on `Copy Link to Clipboard to Invite Friends`

(d) Open a new browser window, and log in as a different user, or get a friend to log in with their account and paste the link and hit enter

(e) Repeat step 4, so that a third person joins the lobby

(f) Have all users in the lobby click on the `Let's Play` button at the bottom of the page

(g) Play the game until each player either wins or gets eliminated.

2. Expectations

(a) You should see that you are put into a new lobby, and that you are waiting for users to join the lobby.

(b) You should also see a lobby id

(c) When step 3 above is completed, a link should be copied to your system clipboard

(d) When step 4 is completed, there should now be two players in the lobby

(e) When step 5 is completed, there should now be three players in the lobby

(f) When step 6 is completed, the game should start and users can start playing.

(g) When step 7 is completed, there should be a notification at the end of the game indicating the amount of XP earned for playing the game.

i. The first player to get eliminated should have earned 1 XP

ii. The second player should have earned 2 XP

iii. The player that won the game should have earned 3 XP

2.4.2 FR8 View Local Leaderboard

On the server side, there is an automated test. Use the provided test.sh script in the server/ folder to run it, along with other tests

A manual test is used to test the functionality from the user's perspective, and also to test portions of code that are not covered by automated tests.

1. Test Steps

(a) Go to `https://polypong.ca`

(b) Log in if not already logged in yet.

(c) If you have 0 XP, play the game to earn some XP

(d) On the home page, click on the `My Stats and Leaderboard` button

2. Expectations

(a) You should see a list of usernames and their corresponding XP. The users that appear directly next to your name should be the ones that have the closest amount of XP to you

2.4.3 FR9 View Global Leaderboard

On the server side, there is an automated test. Use the provided test.sh script in the server/ folder to run it, along with other tests

A manual test is used to test the functionality from the user's perspective, and also to test portions of code that are not covered by automated tests.

1. Test Steps

(a) Go to `https://polypong.ca`

(b) Log in if not already logged in yet.

(c) If you have 0 XP, play the game to earn some XP

(d) On the home page, click on the `My Stats and Leaderboard` button

(e) Click on the `Top in the World` button

2. Expectations

(a) You should see a top 10 list of usernames and their corresponding XP. These users are the 10 players in descending order with the highest XP.

2.4.4 FR10 User statistics

On the server side, there is an automated test. Use the provided test.sh script in the server/ folder to run it, along with other tests

A manual test is used to test the functionality from the user's perspective, and also to test portions of code that are not covered by automated tests.

1. Test Steps

(a) Go to `https://polypong.ca`

(b) Log in if not already logged in yet.

(c) If you have 0 XP, play the game to earn some XP

(d) On the home page, click on the `My Stats and Leaderboard` button

2. Expectations

   (a) You should see three statistics which should match with your past play history

      i. Games Played

      ii. Games Won

      iii. XP Level

2.5 Power Ups

2.5.1 FR11 Power Ups

2.5.2 FR12 Expanded Paddle

2.5.3 FR13 Shrink Paddle

2.5.4 FR14 Self Invisible Paddle

2.5.5 FR15 Others Invisible Paddle

2.5.6 FR16 Invisible Ball

2.5.7 FR17 Self Curved Outwards Paddle

2.5.8 FR18 Self Curved Inwards Paddle

2.5.9 FR19 Self Bumpy Paddle

2.5.10 FR20 Distracting Background

2.5.11 FR21

This optional functional requirement was scrapped. Therefore, the feature does not exist, so no tests are required for a feature that does not exist.

2.5.12 FR22

This optional functional requirement was scrapped. Therefore, the feature does not exist, so no tests are required for a feature that does not exist.

2.5.13 FR23 Add Ball

2.5.14 FR24

This optional functional requirement was scrapped. Therefore, the feature does not exist, so no tests are required for a feature that does not exist.

2.5.15 FR25

This optional functional requirement was scrapped. Therefore, the feature does not exist, so no tests are required for a feature that does not exist.

2.5.16 FR26 Path Trace

2.6 Skins

2.6.1 FR27 Earn Skin

On the server side, there is an automated test. Use the provided test.sh script in the server/ folder to run it, along with other tests

A manual test is used to test the functionality from the user's perspective, and also to test portions of code that are not covered by automated tests.

1. Test Steps
   (a) Go to `https://polypong.ca`
   (b) Log in if not already logged in yet.
   (c) Make note of your current XP level. This can be checked on the leaderboard page by clicking on the `My Stats and Leaderboard` button on the home page
   (d) Make note of the current skins you have unlocked. This can be checked on the settings page by clicking on the `Settings` button on the home page.
   (e) Play enough games to earn enough XP to unlock a new skin: Skins are unlocked at the following thresholds

```
{
  [Color.White]: 0,
  [Color.BlueGrey]: 7,
  [Color.Grey]: 11,
  [Color.Brown]: 18,
  [Color.DeepOrange]: 29,
```

```
  [Color.Orange]: 47,
  [Color.Amber]: 76,
  [Color.Yellow]: 123,
  [Color.Lime]: 199,
  [Color.LightGreen]: 322,
  [Color.Green]: 521,
  [Color.Teal]: 843,
  [Color.Cyan]: 1364,
  [Color.LightBlue]: 2207,
  [Color.Blue]: 3571,
  [Color.Indigo]: 5778,
  [Color.DeepPurple]: 9349,
  [Color.Purple]: 15127,
  [Color.Pink]: 24476,
  [Color.Red]: 39603,
  [Color.Black]: 64079,
  [Color.BackgroundColor]: 103682
}
```

(f) When enough experience points are earned to unlock a new skin, go to the home page and click on the `Settings` button

2. Expectations

   (a) You should now see a new skin available to choose from

2.6.2 FR28 Select skin

On the server side, there is an automated test. Use the provided test.sh script in the server/ folder to run it, along with other tests

A manual test is used to test the functionality from the user's perspective, and also to test portions of code that are not covered by automated tests.

1. Test Steps

   (a) Go to `https://polypong.ca`

   (b) Log in if not already logged in yet.

   (c) Go to the settings page by clicking on the `Settings` button

   (d) Select a different skin from the one you currently have equipped

   (e) Play a game to see the new skin in action

2. Expectations

   (a) You should see a selection of skins that you can choose

   (b) After step 3, in the network tab of the developer console of your browser, a network request should be seen made to `https://polyserver.polypong.ca/getavailableskins` The response code should be 200

   (c) After step 4, in the network tab of the developer console of your browser, a network request should be seen made to `https://polyserver.polypong.ca/setskin` The response code should be 204

   (d) During step 5, you should see the new skin that was selected active in the game. Your opponents should also be able to see the skin colour you chose.