**Team Project Sprint Report #3**

Report

**Team Name:** Delta

**Team Members:** Bayard Rucker, Muhammad Usman, Zeal Patel, Ergin Bostanci, Sabrina Djeddi

1. **Project Micro-Charter**

**Project name**: American Checkers Board Game by Team Delta

This project is to build a web app using an OOP language that allows users to play American checkers against each other. Our development team vision is to deliver a secure and scalable checkers game while using the agile development mythology and multiple modern technologies such as mySQL and the Django web framework.

The purpose of this project is to deliver to our product owners a well designed program and deploy a web app for users to play American checkers. Our aim is a stable application built on a modern framework allowing the dev team to learn new technologies and hone necessary soft skills that are imperative for working on dev teams and in large scale projects.

The business value for our product owners is in allowing users to play checkers as well as the flexible design of the game that will allow for new functionality and additional games to be added. The primary product owner for the project is our professor Dianxiang Xu however all other students in class are encouraged to try the app as well.

The dev team's goal is to use agile for our development mythology. There are many different ways to evaluate a team's value such as tracking K locks or the number of stories completed. The team focus is on delivery of high quality working software, getting the local Django framework running and then integrating with our mysql backend as well as getting the logic for the game engine finished. The team's biggest hurdles heading into the next sprint are a lack of testing. Like many projects, certain decisions were made with the understanding that some amount of refactoring would be needed so the team needs to be ready to integrate and refactor as well as robustly test our code. The second major hurdle is in connecting our frontend Django uI and back end game engine; however, given the robust documentation from Django, this should be a more straightforward challenge to overcome.

1. **Updated User Stories**

**NOTE: stories are pointed in terms of complexity. They are meant to be a general marker for the average developers skills of the team. While they are not a one to one mapping of time spent developing a general rule of thumb is that a 3 pts story is about one full day of development work. Given that most students don’t work full 8 hour shifts like industry. It's safe to assume a 3 pts story might take a little longer in terms of development time.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **User Story Name** | **User Story Description** | **Priority** | **Estimated effort (pts)** | **Actual effort (if completed)** | **Status (Done, To Do, in Progress)** | **Developer names** |
| 1 | Research spike | As a use I want to play checkers on a web app | Research | 3 pts | 3 pts | Done | Bayard |
| 2 | Save games | As a user I want to be able to save my games | High | 3 pts | 3 pts | Done | Sabrina |
| 3 | Login | As a user I want to be able to login to the game | High | 5 pts | 5 pts | Done | Sabrina |
| 4 | Sign up | As a user I want to be able to sign up for an account | High | 3 pts | 3 pts | Done | Sabrina |
| 5 | Front end:VUE research spike | Research spike | Research | No pts | No pts | Done | Zeal |
| 6 | Look into Django Views research spike | Research spike | Research | No pts | No pts | Done | Zeal/ Sabrina |
| 7 | Initial Board built | As a user I want to see checkers on a simple board | Medium | 3 pts | 3 pts | Done | Bayard /Usman |
| 8 | Initial Game built | As a user I want to see the other aspects of the game like whose turn, color is assigned to the player, is game ended, winner, assign unique id to game. | High | 13 pts | 13 pts | Done | Usman/Ergin |
| 9 | Evaluate communication channels | Research spike | Research | No pts | No pts | Done | Zeal |
| 10 | Initializing Pieces on board | As a user I want pieces to appear on board | Medium | 3 pts | 3 pts | Done | Usman/Ergin |
| 11 | Game logic | As a user I want the rules of the game to be enforced | High | 10 pts | 10 pts | Done | Usman |
| 12 | Rules page | As I user I want a page where I can see the game rules | Low | 3 pts | 3 pts | Done | Bayard |
| 13 | Player stats | As a user I want to see player statistics | Low | 3 pts | 3 pts | Done | Sabrina, Usman |
| 14 | Routing defect | As a user I want the routing fixed | Low | 1 pts | 1 pts | Done | Zeal |
| 15 | See network issues | As A user I want to be able to play against players on other machines | Medium | 5 pts | 5 pts | Done | Zeal |
| 16 | Game end | As a user I want to see when game ends | High | 2 pts | 2 pts | Done | Zeal |
| 17 | Show possible moves | As a user I want see where my selected piece can possibly move | Medium | 2 pts | 2 pts | Done | Zeal, Usman |
| 18 | Change board | As a player I would like to change board orientation as per pieces color(default is dark side at the bottom). | Low | 1 pts | 1 pts | Done | Usman/Bayard/Ergin |
| 19 | Display turn | As a user, the player wants to see whose turn it is. | Medium | 2 pts | 2 pts | Done | Bayard |
| 20 | Auto Reconnect on network issue | As a user, I want to reconnect to the game automatically without refreshing the page. | Low | 3 pts | 3 pts | Done | Zeal |
| 21 | Network issues | As a user I want to know when my game has some network issues. | Low | 2 pts | 2 pts | Done | Ergin |
| 22 | Game synchronization | As user boards need to be updated when there is change in game | High | 3 pts | 3 pts | Done | Zeal |
| 23 | Navigation to create and join game | As a user, I want to see interactive visualization on the game home page | Low | 3 pts | 3 pts | Done | Zeal, Sabrina |
| 24 | Must capture game logic | As a User I want must capture rule logic to game.py class | High | 3 pts | 3 pts | Done | Zeal  Sabrina |
| 25 | UI for to play game against computer | As a user I want to see button from where I can start game against computer | High | 3 pts | 3 pts | Done | Zeal |
| 26 | AI player logic | As a user I want to play against an AI which moves according to human opponent move | High | 8 pts | 8 pts | Done | Bayard |
| 27 | Game history | As a user I want to see game history step by step of game play | High | 8 pts | 8 pts | Done | Zeal |
| 28 | Live viewers count | As a user, I want to see how many users are watching the game in real time. | Low | 2 pts | 2 pts | Done | Zeal |

1. **Updated Acceptance Criteria (AC)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User Story ID and Name** | **AC**  **ID** | **Description of Acceptance Criterion** | **Status (Done, To Do, in Progress)** | **Developer Names** |
| 1 Research spike | 1.1 | Given the two different frameworks we are looking into. When a dev has Research them and then presents the finding to the team | Done | Bayard/ Usman |
| 2 Save games | 2.1 | Given a mySQL db when the app starts up then it should Connect to the db and save information there | Done | Sabrina |
| 3 User Login | 3.1 | Given a user arrives at the sign up page then the user should be able to login | Done | Sabrina |
|  | 3.2 | **Given** a user arrives at the app **then** the user is redirected to the login page | Done | Sabrina |
|  | 3.3 | **Given** a Registration page renders. **When** a user registered successfully **then** they should be redirected to the login page. | Done | Sabrina |
| 4 User Sign up | 4.1 | **Given** a user when a user arrives at the app **then** the user should be able to sign up or sign in. | Done | Sabrina |
|  | 4.2 | **Given** a new user, **when** the user doesn't have a set up account, **then** he has to register(sign up) in order to create one and access his playing space | Done | Sabrina |
|  | 4.3 | **Given** a non existent valid user name, **When** the user sign up with this user name, a valid and confirmed password, and a valid email address, **Then** the new player has been created and added to the db successfully | Done | Sabrina |
|  | 4.4 | **Given** a user uses the Registration page when the account has been created for this user **then** it should connects to DB and Django app and save this user into Django-users table and django send a signal to the player table to communicate the new user informations to the player table and this user will become player | Done | Sabrina |
|  | 4.5 | **Given** a username that already exists, **When** the user try to register with this existent username, a valid and confirmed password, and a valid email address, **Then** the new player account is not created | Done | Sabrina |
|  | 4.6 | **Given** a non existent valid username, **When** a user tries to register with this username, an invalid password, same confirmed password, and a valid email address... **Then** the new account is not created | Done | Sabrina |
|  | 4.7 | **Given** an non-existent valid username, **When** a user creates an account with this username, a valid email address, a valid password, but different with a confirmed password, **Then** the new player account is not created | Done | Sabrina |
|  | 4.8 | **Given** an non-existent valid username, **When** a user creates an account with this username, an invalid email address, a valid password, same confirmed password, **Then** the new player account is not created | Done | Sabrina |
|  | 4.9 | **Given** empty input, **When** a user register with any empty input **Then** the new player account is not created and a message to fill that field will be displayed | Done | Sabrina |
|  | 4.10 | **Given** the new player **when** this player rendered to the login page and the input credentials are verified with the DB **then** he will be able to access his account | Done | Sabrina |
|  | 4.11 | **Given** an empty username and password, **When** a player try to login, the player is redirected to the login page(can not access to his space) | Done | Sabrina |
|  | 4.12 | **Given** a valid user name and an invalid password, **When** a player try to login, the player is redirected to the login page(can not access to his space) | Done | Sabrina |
|  | 4.13 | **Given** an invalid username and an invalid password, **When** a player try to login, the player is redirected to the login page(can not access to his space) | Done | Sabrina |
|  | 4.14 | **Given** an invalid username and a valid password, **When** a player try to login, the player is redirected to the login page(can not access to his space) | Done | Sabrina |
|  | 4.15 | **Given** a valid username and a valid password, **When** a player try to login, the player is redirected into his session page and can start playing. | Done | Sabrina |
| 5 VUE research spike | 5.1 | Given the different approaches we are looking into. When a dev has Research them and then presents the finding to the team | Done | Zeal |
| 6 Look into Django Views spike | 6.1 | Given the different approaches we are looking into. When a dev has Research them and then presents the finding to the team | Done | Zeal/  Sabrina |
| 7 Initial Board built | 7.1 | When a player start a game, then a board of 8\*8 grid is generated | Done | Bayard Usman |
|  | 7.2 | When a Board generated, the DARK and LIGHT pieces get positioned in specific cells of the board | Done | Bayard Usman |
| 8 Initial Game built | 8.1 | Given a new game when a user playing then the game should start with its start point (initial piece placement) | Done | Usman |
|  | 8.2 | Given a game when a user tries to make an illegal move then the game should not update its state. | Done | Usman |
|  | 8.3 | Given game is created, when a user requests to create a game then initialize the game with an opponent player and winner as an empty string. | Done | Usman |
|  | 8.4 | Given a new game is created, when a player creates a game then the game is initialized with the player who created the game gets the dark side with the board in its initial state. | Done | Usman |
|  | 8.5 | Given a player willing to join, When a player join an existing game, then the player get the LIGHT pieces and the second turn | Done | Usman |
| 9 evaluate communication channels | 9.1 | Given the different approaches we are looking into. When a dev has Research them and then presents the finding to the team | Done | Zeal  Usman |
| 10 Initialize pieces on board | 10.1 | Given a new game when a user is playing then pieces should operate under the same rules as the original game. | Done | Usman |
|  | 10.2 | Given a user is playing when the pieces are moved then they should be controlled by the same game logic. | Done | Ergin |
| 11 Game logic | 11.1 | Given a game when the game is being played it should follow the rules of the game. | Done | Usman |
|  | 11.2 | Given a game when a user tries to move a piece that is not theirs then the game should not allow it. | Done | Usman |
|  | 11.3 | Given an Ongoing game, When a player DARK / LIGHT pieces start the game , I need to make diagonal moves and jumps | Done | Usman |
|  | 11.4 | Given an Ongoing game, When a player has one adjacent diagonal piece with different color and the turn to move, then the player needs to make jump and capture the piece | Done | Usman |
|  | 11.5 | Given an Ongoing game, When the player make a jump to captures the opposite piece, then the piece will be removed from the board | Done | Usman |
|  | 11.6 | Given an Ongoing game, When the player after the first jump get another opposite piece to capture, then the player has to make double jump to capture the other piece and the piece will be removed from the board | Done | Usman |
|  | 11.7 | Given an Ongoing game, When the player after the double jump get another piece to capture, the this piece need to be captured with the 3rd jump and needs to be removed from the board | Done | Usman |
|  | 11.8 | Given an Ongoing game, when the player BLACK Pieces/LIGHT pieces can not make a diagonal move or jump, then the game is Over and the Black pieces player win the game | Done | Usman |
|  | 11.9 | Given an Ongoing game, when the player BLACK Pieces has no pieces to move then the game is Over and the LIGHTpieces player win the game | Done | Usman Ergin |
| 12 Rules page | 12.1 | Given a user is logged in when the user goes to the rules page then they should see the rules | Done | Bayard |
| 13 Player stats | 13.1 | Given a user is logged in when the user goes to the stats page then they should see the stats of all game played by this logged in user | Done | Sabrina |
|  | 13.2 | Given a user can see previous game results when the user requests player stats, then the user should also see a button which can navigate to that game history. | Done | Sabrina |
| 14 Routing defect | 14.1 | Given a user is logged in when they try moving to other pages then they should be rerouted properly | Done | Zeal |
| 15 See network issues | 15.1 | Given a user is logged in, when a network issues happen, then the game should let the user know | Done | Zeal |
| 16 Game end | 16.1 | Given a user is loser player in a game when the game is over then the user should be notified user won | Done | Zeal |
|  | 16.2 | Given a user is winner player in a game when the game is over then the user should be notified user lost | Done | Zeal |
|  | 16.3 | Given a user is viewer in a game when the game is over then the user should be notified winner username | Done | Zeal |
| 17 Show possible moves | 17.1 | Given a user is playing a game, when player selects piece then user should see update board with yellow circles as possible moves of selected piece | Done | Usman/Zeal |
| 18 Change board orientation | 18.1 | Given user is either player1 or player2, when game is loaded, then board should orient dark side down for player1 and light side down for player 2 | Done | Usman/Bayard/Ergin |
| 19 Display Turn | 19.1 | Given a user is playing when a move is completed then the game should hand off control to he next player | Done | Usman/Bayard/ |
| 20 Auto Reconnect on network issue | 20.1 | Given a user is playing when they get disconnected then the app should reconnect automatically without any user interference. | Done | Zeal |
| 21 Network issues | 21.1 | Given a user is logged in when the user has network issues then the app should informer the user of these issues | Done | Ergin |
| 22 Game Synchronization | 22.1 | Give a user is player when there turn is over then the board should update with the new game state | Done | Bayard/Sabrina/Ergin |
|  | 22.2 | Given a user is playing a game, when the user selects or moves a piece, then update of the board to be sent out to everyone connected to that game. | Done | Zeal |
| 23 Navigation to create and join game | 23.1 | Given a user is logged in when they go to start a game then the UI is easy to use | Done | Sabrina |
|  | 23.2 | Given a user is logged in when they go to join a game then the UI is easy to use | Done | Sabrina |
| 24 Must capture game logic | 24.1 | Given a user is playing a game when a capture is available then that game rule should be enforced | Done | Zeal  Sabrina |
| 25 UI for to play game against computer | 25.1 | Given a user is logged in when they start a game then they should have the option to play against an ai | Done | Zeal |
| 26 AI player logic | 26.1 | Given a user has logged in when they select ai game then they should be able to play against the apps ai player | Done | Bayard |
|  | 26.1 | Given an user is playing the ai should be able to beat a really bad human player | Done | Bayard |
| 27 Game history | 27.1 | Given a user, when game is created, then create history file with initial board | Done | Zeal |
|  | 27.2 | Given a user and already created game, when user successfully completes the turn, then save the next state of game to file. | Done | Zeal |
|  | 27.3 | Given a user, when game history is requested, then get file game game states and load last state to screen | Done | Zeal |
|  | 27.4 | Given a user, when game history is opened, then show a button to change game state to forward, backward, first, last and autoplay forward. | Done | Zeal |
| 28 Live viewers count | 28.1 | Given a user, when game link is opened, then show total number of users connected to same game id group (players and viewers all included) | Done | Zeal |

1. **Updated Implementation Tasks**

Include the tasks from the previous report and highlight the new tasks with a different color.

Summary of production code

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Story ID and Name** | **AC ID** | **Class Name(s)** | **Method Name(s)** | **Developer Name(s)** | **Status** | **Notes (optional)** |
| 2 Save games | 2.1 | migrations/ | Django migrations | Sabrina | Done |  |
| 3 Login | 3.1 | Django user authentication | authenticate()  login() | Sabrina | Done |  |
|  | 3.2 | Checking for credential | authenticate()  login() | Sabrina | Done |  |
|  | 3.3 | Class login view | post() | Sabrina | Done |  |
|  | 3.4 | Class login view  urls.py | post() | Sabrina | Done |  |
|  | 3.5 | Class logout | logout() |  |  |  |
| 4 Sign up | 4.1 | HTML sign up page  Class Sign up view | post() | Sabrina | Done |  |
|  | 4.2 | Used Django models | get() | Sabrina  /Usman | Done |  |
|  | 4.3 | Html page | get() | Sabrina | Done |  |
|  | 4.4 | Used dragon models | migrations | Sabrina | Done |  |
|  | 4.5 | Html page |  |  |  |  |
|  | 4.6 | Django validation check | is\_valid() | Sabrina | Done |  |
|  | 4.7 | UserCreationForm | Form with predefined field for the user coming with django | Sabrina | Done |  |
|  | 4.8 | Store the user into the mysql DB | save() | Sabrina | Done |  |
|  | 4.9 | Djagno validation | is\_valid() | Sabrina | Done |  |
|  | 4.10 | Store in Db | Db compare | Sabrina | Done |  |
|  | 4.11 | Login view | Post() | Sabrina | Done |  |
|  | 4.12 | Loginview | Post() | Sabrina | Done |  |
|  | 4.12 | Login view | Post() | Sabrina | Done |  |
|  | 4.13 | Login view | Post() | Sabrina | Done |  |
|  | 4.14 | Login view | Post() | Sabrina | Done |  |
|  | 4.15 | Login view | post() | Sabrina | Done |  |
| 7 Initial Board built | 7.1 | Initialize matrix variable | constructor of Game class | Bayard | Done |  |
|  | 7.2 | Game | new\_board() | Usman | Done |  |
| S:8 Initial game built | 8.1 | Class Game (Board) | new\_board() | Usman | Done |  |
|  | 8.2 | Class game | update\_game\_object() | Usman | Done |  |
|  | 8.3 | Class game | Player2\_username, winner in \_\_init\_\_() | Usman | Done |  |
|  | 8.4 | Class game | Player1\_username in \_\_init\_\_() | Usman | Done |  |
|  | 8.5 | Class game(view) | join\_game() | Usman | Done |  |
| 10 Initialize pieces on board | 10.1 | Class Board()  Class piece() | new\_board() | Usman | Done |  |
|  | 10.2 | Class Board() | new\_board() | Ergin | Done |  |
| 11 Game logic | 11.1 | Board() | location()  blind\_legal\_move()  chekc\_jumps\_avaialbe()  legal\_moves()  adjacent()  location()  Check\_jumps\_a  vialiable ()  remove\_piece()  move\_piece() | Zeal | Done |  |
|  | 11.2 | Baord() | move\_piece() | Zeal | Done |  |
|  | 11.3 | Board() | Check\_jumps\_a  vialiable ()  remove\_piece()  move\_piece() | Zeal | Done |  |
|  | 11.4 | Board() | remove\_piece()  move\_piece() | zeal | Done |  |
|  | 11.5 | Board() | remove\_piece()  move\_piece() | Zeal | Done |  |
|  | 11.6 | Board() | remove\_piece()  move\_piece()  Check\_jumps\_a  vialiable () | Zeal | done |  |
|  | 11.7 | Board() | remove\_piece()  move\_piece()  Check\_jumps\_a  vialiable () | Zeal | Done |  |
|  | 11.8 | Game() | end\_turn() | Zelal | Done |  |
|  | 11.9 | Game() | end\_turn() | Zeal | done |  |
| 12 Rules page | 12.1 | Class rulesview(view) | get() | Bayard | done |  |
| 13 Player stats | 13.1 | Class player\_view(view) | Db backend ,get() | Sabrina | Done |  |
|  | 13.2 | Class player\_view(view) | Db backend , get() | Sabrina | Done |  |
| 14 Routing defect | 14.1 | Redirecting in routing.py | re\_path() | Zeal | Done |  |
| 15 See network issues | 15.1 | Frontend (JavaScript) | game\_socket.disconnect() | Zeal | Done |  |
| 16 Game end | 16.1 | Class game() | end\_turn(), check\_for\_endgame() | Zeal | Done |  |
|  | 16.2 | Class game() | end\_turn(), check\_for\_endgame() | Zeal | Done |  |
|  | 16.3 | Class game() | end\_turn(), check\_for\_endgame() | Zeal | Done |  |
| 17 Show possible moves | 17.1 | Class game() | update\_legal\_moves(), update\_game\_object() | Zeal, Usman | Done |  |
| 18 Change board orientation | 18.1 | Front end (Javascript) | change\_board\_orentation() | Usman/Bayard/Ergin | Done |  |
| 19 Display Turn | 19.1 | game consumer() | send\_update\_message()  connect() | Usman/Bayard | Done |  |
| 20 Auto Reconnect on network issue | 20.1 | Front end (javaScript) | Changed reconnect library | Zeal | Done |  |
| 21 Network issues | 21.1 | Frontend (JavaScript) | game\_socket.disconnect() | Zeal | Done |  |
| 22 Game Synchronization | 22.1 | Gameconsumer() | send\_update\_message() | Zeal | Done |  |
|  | 22.2 | Gameconsumer() | send\_update\_message()  connect() | Zeal | Done |  |
| 23 Navigation to create and join game | 23.1 | View class game | get() | Bayard | Done |  |
|  | 23.2 | View class game | get()  create\_game() | Bayard | Done |  |
| 24 Must capture game logic | 24.1 | Class Game(Board), | set\_jump\_available(),  checks\_for \_jumps\_available(), | Sabrina | Done |  |
| 25 UI for to play game against computer | 25.1 | ai\_game(View) | get() | Zeal | Done |  |
| 26 AI player logic | 26.1 | Aiplayer() | \_\_init\_\_()  minmax()  pieces\_update() | Bayard | Done |  |
|  | 26.1 | Aiplayer() | minmax() | Bayard | Done |  |
| 27 Game history | 27.1 | Create text file/ | open(filename, a) | Zeal | Done |  |
|  | 27.2 | DB | Db table | Zeal | Done |  |
|  | 27.3 | Html | Html , javascript | Zeal | Done |  |
|  | 27.4 | Html | Html, javascript | Zeal | Done |  |
| 28 Live viewers count | 28.1 | Game\_consumer.py | consumer , javascript | Zeal | Done |  |

Summary of automated test code (directly corresponding to some acceptance criteria)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Story ID and Name** | **Acceptance Criterion ID** | **Class Name (s) of the Test Code** | **Method Name(s) of the Test Code** | **Description of the Test Case (input & expected output)** | **Status** | **Developer Name(s)** |
| player class |  | Using pytest not specific classes | test\_player\_build()  test\_player\_new\_game()  test\_player\_get\_functions()  test\_player\_reset() | player\_build test that the class is being built correctly.  new\_game test that once a player starts a game they attributes are updated correctly  get\_functions()  Test the return values of the getter functions  reset()  Is a test to reset the player after a game | Done | Ergin  Sabrina  Bayard |
| S:18 unit testing for game class | 18.1 | Pytest does not need a class | test\_game\_build()  test\_game\_turn()  test\_game\_functions()  test\_game\_chekc\_end\_game() | Test the game is building correctly  Function are rusting properly,  Turns are not being violated  And end game status is returning correctly | Done | Zeal |
| S:7 Unit testing for board class | 7.1 | test\_board\_class(testCase) | test\_matrix\_length() | Test board matrix dimensions | Done | Zeal |
| S:7 Unit testing for board class | 7.2 | test\_board\_class(testCase) | test\_matrix\_type() | Test data type of matrix | Done | Zeal |
| S:7 Unit testing for board class | 24.1 | test\_board\_class(testCase) | test\_jump\_available() | Test jump available function | Done | Zeal |
| 26 AI player logic | 26.1 | AIplayer | test\_ai\_setup() | Test the creation of the ai player, makes sure all instance variables are set correctly for the start of a game. | Dones | Bayard |
| 26 AI player logic | 26.2 | Aiplayer | test\_ai\_minmax() | Test that the minimax algorithm is running properly. And checking enough states to make good moves. | Done | Bayard |

Summary of manual test cases (directly corresponding to some acceptance criteria)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Story ID and Name** | **Acceptance Criterion ID** | **Test Case Input** | **Test Oracle (Expected Output)** | **Status** | **Notes** | **Developer Name(s)** |
| S:3 Set up MVP for app back end | 3.1 | App runs in dev env |  | Done |  | Bayard |
| S:2 MYSQL | 2.1 | Check read and writes in DB |  | Done |  | Sabrina |
| S:4 registration page view and logic | 4.1 | Page renders |  | Done |  | Sabrina |
| S:8 build game class | 8.1 | Game is playable from command line |  | Done |  | Usman/Ergin |
| S:14 connect board to front end UI | 14.1 | Page renders |  | Done |  | Bayard |
| S:24 Must Capture Logic | 24.1 | Player selects piece which has not capture | No possible moves shown because must capture is available | Done |  | Zeal |
| S:28 live viewers count | 28.1 | Shared game link to friends to open from multiple locations | Increase in number of Viewer count in front end | Done |  | Zeal |

Summary of other automated or manual tests (not corresponding to the acceptance criteria)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Number** | **Test Input** | **Expected Result** | **Class Name of the Test Code** | **Method Name of the Test Code** | **Status** | **Developer Name(s)** |
| 1 | Click on ‘Create Game button’ | Generate a unique 16 character alphanumeric string and save it to the database as a game id. Also add player username (who created game) in player1 field | Class game(view); | create\_game() | Test successful | Zeal |
| 2 | Click join button after selecting game for radio list | Get game record from game\_session table and add username in player2\_username field | Class game(view) | join\_game() | Test successful | Zeal |
| 3 | Click resume button after selecting game for radio list | Redirect request to room method after appending game\_id to /game/ link | Class game(view) | resume\_game() | Test successful | Zeal |
| 4 | Click game History button | Redirect request to room method after appending game\_id to /room/ link | Class room (view) | room.history\_room() | Test Successful | Sabrina |
| 5 | Click on the link player stats in the navbar redirect me into the player\_stats page and show all the played game by the logged user | Redirect link into href={%url 'player\_stats'%} page | Class player\_statsview(View) | player\_statsview() | Test Successful | Sabrina |
| 6 | Click on the link player rules in the navbar redirect me into the rules page and show the rules of the game | Redirect link into href={%url 'rules'%} page | Class player\_rules(View) | player\_rulesview() | Test successful | Bayard |
| 7 | Click join button after selecting ai game | Get game record from game\_session table and player agani ai player as player 2 | Class aigame(view) | join\_game() | Test successful | Bayard |

**IV Design Documentation**

**1 User Interface Design**

List the names of the team members who contributed to this section.

Summarize the user-interface design, using a combination of screenshots and textual descriptions.

Team member: Zeal Sabrina

NOTE: Added frontend User Interface images at end of this report.

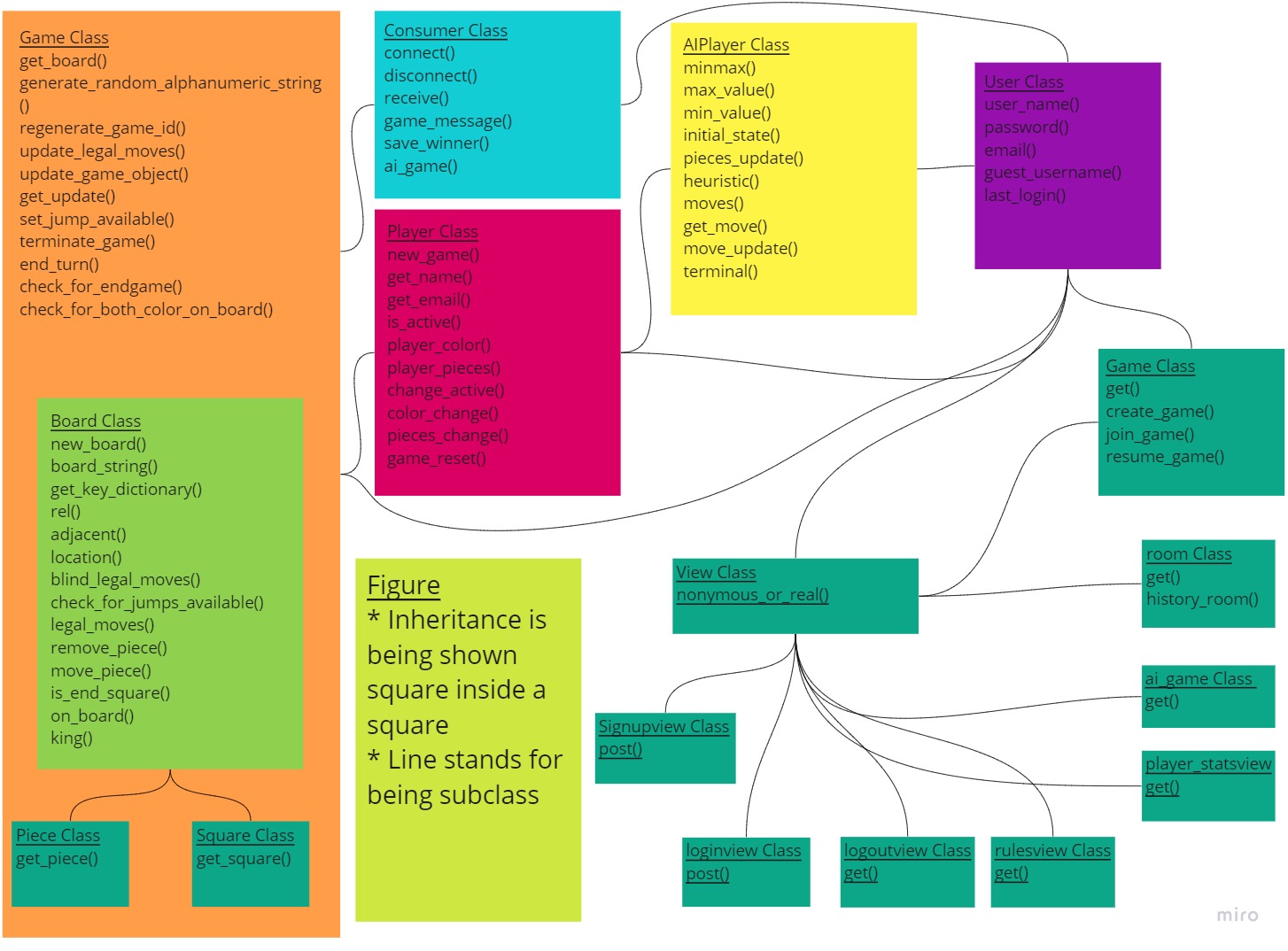
The UI design used the django framework as well as html css and javascript. The point of this design is to be simple so that new user can easily use the app. There is a bar at the top of the page that allows routing to the different pages. Like rules home page. State page. On the home page there are two main buttons for starting a person to person game. Or starting an AI game. There will also be a list of unfinished games. In the top right connor there is a logout button

**2 Software Architecture**

List the names of the team members who contributed to this section.

Provide a class diagram that captures the main classes and their relationships in your final program.

Team members: Ergin, Usman



**3 Algorithm Design**

List the names of the team members who contributed to this section.

Describe the algorithm design of the computer opponent (e.g., using pseudo code). The description should be understandable without referring to the source code.

Team members: Sabrina, Usman, Bayard, Zeal, Ergin

Using the django framework the app is launched from the mange.py. The back end of django runs the server. The server serves up the template pages starting with the log in page. Once a user is logged into the app the server will allow for routing to any of the apps pages. When a game is started django will go to the checkers all the load in the game.py file that holds most of the game logic. As the game is played through game.py and javascript from the templates file will handel passing the game back and fourth to player. In the case of the ai player. That file is also in checks/ so the consumer.py file will pass game to whichever players turn it is.

**4 Extensibility**

List the names of the team members who contributed to this section.

Discuss how your code can be extended for the variants of Nine Men's Morris, including Six Men’s Morris and Twelve Men’s Morris. What classes and methods need to be changed, and how? How was the Open-Closed Principle applied (i.e., which functions or classes are open for extension, but closed for modification)?

Team members: zeal, Bayrd

This code can be extended by modifying the board class sense both game and ai player draw from this class Game class inherited from board. While ai player pulls relivelent information that it needed to make moves. Once the board class is adjusted to have the desired number of squared the rest of the game is set up so that it should work fine.

**V Findings from the Code Review Exercise**

Use the following template to document the findings from the code review of **each** class.

Participant names: Bayard, Zeal, Ergin, Sabrina and Usman

Class that was reviewed: Game class

|  |  |  |  |
| --- | --- | --- | --- |
| **Checklist** | **Checklist Item** | **Findings** | |
| Coding Standards | Naming conventions | Naming looks good for pep8 standards | |
| Ordering convention of method arguments | Could be reorder to be more optimized but overall pretty good | |
| Meaningful and valid comments | Good and up to pep8 standards | |
| Consistent style of code blocks | Coding is consistent. Not 100% in line with pep8  For example some lines are two long | |
| Consistent indentation | Spacing is good python enforces this | |
| Design Principles | Good class abstraction and interface | Game class inherits from the board. This was a solid design decision. There is space for improvements. | |
| Appropriate visibility of each variable, method, and class | Yes python handles most of that for the developer. | |
| Any violation of the command-query separation principle | Yes this principle was violated in few methods. | |
| Design by contract (pre/post-conditions) | Design by contract looks ok for the most part but there are a few violations | |
| Is the Open-Closed Principle violated? | This principle is implemented correctly | |
| Is the Single Responsibility Principle violated? | Yes this principle is implemented correctly. There are few methods which violate it. | |
| Code Smells | Magic numbers | None | |
| Unnecessary global / class variable | None | |
| Duplicate code | None | |
| Long methods | yes | |
| Long parameter list | yes | |
| Over-complex expression | Yes. mostly in board.py because many complex conditions to check for multiple jumps and other things. | |
| Switch or if-then-else that needs to be replaced with polymorphism | Maybe the team is split on this | |
| Variable or method name whose intent is unclear | None | |
| Any similar methods in other classes? | No | |
| Secure Coding | Strong password policy | Yes django enforces this. | |
| Prevention of SQL injection | Yes django support this on its backend | |
| Integer overflow | We don’t think so but this type of error is possible in python. We have not seen this error come up. However in certain edge causes it could occur. Also there is no uncontrolled data backend receives which can occur such problem | |
| **Bugs** | **Buggy code snippet** | **What is the bug?** | **Why is it a bug?** |
| Joining the game. | If two player trying to join a game player will not know that other player is joined first | There is no sockets implement to notify that game is not open. First join is final. |
|  | Three jump in a row | UI will not automatically shows third possible move, you have to click on piece again in order to see move | This bug does not affect game play, it just small bug which can easily solved. |

1. **Meeting Minutes**

Report the minutes of all meetings, including, but not limited to: project/sprint planning meeting, stand-up meeting, backlog grooming, retrospective meeting, and pair programming session.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Time and Duration** | **Place** | **Participant Names** | **Purpose of the Meeting** | **Specific Action Items** |
| 8/27 | 45 min | Zoom | Bayard, Zeal, Sabrina | Team formation | Set up a base team. Talked about possible approaches and high-level details like languages |
| 8/27 | 45 min | In person | Bayard, Ergin | Team formation | Set up a base team. Talked about possible approaches and high-level details like languages |
| 8/30 | 1.5 hours | Zoom | Full team | First team meeting | Introductions, talked about high-level project requirements, set up Trello board and added initial stories, set up GIT repo |
| 9/6 | 1 hour | Zoom | Full team | Week meeting time | Checked in on everyone progress. Added a few stories. Decided to use Django as main framework and MYSQL as DB |
| 9/13 | 2 hours | Zoom | Full team | Week meeting time | Weekly meetings decided not to use VUE for the front end. Base app set up |
| 9/20 | 1.5 hours | Zoom | Full team | Week meeting time | To be more agile we followed agile practices listed in meeting agenda doc added testing library, walked thru UI updates and how Django connects to MYSQL |
| 9/20 | 1 hour | In person student union | Bayard, Ergin | Paired programming | Talking about project overview and worked on local set up for MYSQL and setting up player class |
| 9/27 | 2 hours | Zoom | Full team | Week meeting time | Followed agenda doc. Talked about implemented stories and board class focused on OOP and unit testing |
| 10/4 | 1 hour | Zoom | Full team | Week meeting time | Focused on updating routing and UI. Added and pointed to new stories. |
| 10/8 | 45 min | Zoom | Bayard,  Usman | Code review | Review of game class and talk about next steps |
| 10/16 | 30 minutes | Zoom | Full team | Demo review | Talked about spring one report and planning for sprint 2 |
| 10/18 | 45 min | Zoom | Full team | Weekly meeting | Spring 2 planning and backlog refinement |
| 10/25 | 30 min | Zoom | Full team | Weekly meeting | Stand up and weekly check in meeting |
| 10/29 | 1 hour | Zoom | Full team | Working session | Everyone worked on assigned stories |
| 11/1 | 45 min | Zoom | Full team | Weekly meeting | Backlog refinement and stand up and planning |
| 11/8 | 45 min | Zoom | Full team | Week meeting time | Stand up, troubleshooting defects and code updates |
| 11/15 | 30 minutes | Zoom | Full team | Week meeting time | Stand up working on sprint 2 report |
| 11/22 | 1 hour | Zoom | Full team | Start of spring 3 planning | Discussed sptring 3 requirement and whats needed for finishing up the project |
| 11/29 | 30 min | Zoom | Full team | Weekly meeting | Had stand up and provided updates on progress for spring 3 |
| 12/6 | 1 hour | Zoom | Full team | Weekly meeting | Stand up, code review, preparation for presentation |
| 12/8/8 | 1 hour | Zoom | Full team +professor | Report update | Talked with the professor about how to improve our sprint 3 report. Also talked about how to run the demo. |
| 12/9 | 1 hour | Zoom | Zeal, Sabrina, Bayard | Working session | Worked on sprint 3 report and last minute code updates |
| 12/10 | 1 hour | Zoom | Full team | Demo walk thru | Ran thru the demo and made sure everyone was ready to present |

1. **Buddy Ratings**

If you don’t feel comfortable to include your ratings in this report, you may email your ratings to the instructor or grader.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *Rating giver* | *Rating receiver* | | | | |  |
|  | Bayard Rucker | Muhammad Usman | Zeal Patel | Ergin Bostanci | Sabrina Djeddi |
| Bayard Rucker | X | 1 | 1 | 1 | 1 |
| Muhammad  Usman | 1 | X | 1 | 1 | 1 |
| Zeal Patel | 1 | 1 | X | 1 | 1 |
| Ergin Bostanci | 1 | 1 | 1 | X | 1 |
| Sabrina Djeddi | 1 | 1 | 1 | 1 | X |
|  | ***Average*** | **1** | **1** | **1** | **1** | **1** |

