**Team Project Sprint #1**

Instructions

Please read the instructions carefully. All members of your team should discuss the instructions together to ensure that everyone is on the same page.

**Objectives**

Create a brief project description, specify all software requirements as user stories and acceptance criteria, and implement the primitive functions (i.e., user management, board object and visualization, and piece placement for both players). Each team should meet at least once a week.

**Deliverable and Grading Policy**

1. Project Report (**15 points**)

The project report should include the following sections:

* 1. Project description (micro-charter), which should result from group discussion **(1 point)**.
  2. User stories using the template discussed in class. **(2 points)**

Provide a complete list of user stories and estimated efforts for the target software.

* 1. Acceptance criteria using the template discussed in class. **(8 points)**

Provide complete acceptance criteria for each of the user stories. Note that, although some of the user stories will be implemented in the future sprints, their acceptance criteria need to be defined in the first sprint. You may continue to improve the user stories and acceptance criteria in the next sprint.

* 1. Implementation tasks **(2 points)**

Describe the production code, automated test code or manual test case for each user story and acceptance criterion related to the implementation of the primitive functions, including **user registration**, **login**, **logout**, **board visualization**, and **piece placement**. For each acceptance criterion of every user story for the primitive functions, you need to implement at least one test (either test code or manual test case).

* 1. Minutes of ALL meetings, including, but not limited to: project/sprint planning meeting, stand-up meeting, backlog grooming, retrospective meeting, and pair programming (or development) session. **(2 points)**
  2. A table of buddy ratings. Individual members may email their buddy ratings to the instructor or teaching assistant.

Each team only needs to submit one report. For an individual member to receive the credit for this part of the project, the team’s project report must include explicit evidence of his/her contribution (e.g., his/her name is listed as a developer).

2. Demonstration **(5 points)**

Within 15 minutes, clearly demonstrate that:

1. your project has implemented the working software for the primitive functions, i.e., board object and visualization, and piece placement for both players.
2. for each acceptance criterion of every user story for the primitive functions, your project has implemented either an automated test method or performed an acceptance test manually.
3. your project has some unique features or enhancements (optional).

Grading of the demonstration: completion of the required functions (**2 points**), and overall presentation (**3 points**) using the following evaluation rubric:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Poor | Fair | Good | Very Good | Excellent |
| Was the demonstration logically organized |  |  |  |  |  |
| Were points made clearly and concisely |  |  |  |  |  |
| Were the instructor’s questions answered satisfactorily |  |  |  |  |  |

**Team Project Sprint #1**

Report Template

**Team Name:** Delta

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

1. **Project Micro-Charter (no more than one page)**

Provide a brief description about the project, including the following elements:

Project name

Vision statement: describe the future that you are trying to create

Mission statement /project purpose

Elevator pitch: no more than a few sentences

Business value

Customers and users: people who will make buying decisions or actually interact with your product

Metrics: how to measure the business value

Milestones: important points in time

Risks: things that may threaten or derail your project

Authors of this micro-charter

1. **User Stories**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **User Story Name** | **User Story Description** | **Priority** | **Estimated effort (hours)** | **Actual effort (if completed)** | **Status (completed, toDo, inProgress)** | **Developer names** |
| 1 | Flask and Django spike | As a dev I want to know if flask or Django will be a better fit for this project | Research | 3 points | Points 3 | Done | Bayard |
| 2 | MYSQL | As a developer I want this project to have a MYSQL back end | Set up DB | Points 3 | Points 3 | Done | Sabrina |
| 3 | set up mvp for app back end | As a dev I want to have a Django back end running | Set up | Points 3 | Points3 | Done | Bayard |
| 4 | registration page view and logic | As a user I want a page I can log into or set up a new account with | Front end and db | Points 8 | Points 13 | Done | Sabrina |
| 5 | VUE research spike | Research spike | Research | No points | No points | Done | Zeal |
| 6 | Look into Django Views research spike | Research spike | Research | No points | No points | Done | Zeal & Sabrina |
| 7 | build board class | As a user I want a back-end board class | OOD class | Points 3 | Points 3 | Done | Bayard &Usman |
| 8 | build game class | As a user I want a back-end game engine | OOD class | Points13 | points13 | Done | Usman |
| 9 | evaluate communication channels | Research spike | Research | No points | No points | Done | Zeal |
| 10 | build piece class | As a user I want a back-end pieces class | OOD class | Points 3 | Points 3 | Done | Usman |
| 11 | transfer prototype into Django | As a User I want the prototype UI transfer to the Django app | Transfer | Points 5 |  | In progress | Zeal |
| 12 | build player class | As a dev I want to have a class to track active players | OOD design | Points 3 |  | In progress | Ergin |
| 13 | add testing library | As a dev I want a centralized testing library | Testing | Points 2 | Points 2 | Done | Bayard |
| 14 | connect board to front end UI | As a user I want a ui page that the game will be played on | UI | Points 3 | Points 3 | Done | Bayard |
| 15 | change Django view functions to classes | Change view functions into classes | Refactor | Points 3 |  | In progress | Sabrina |
| 16 | add rules page | As I user I want a page where I can see the game rules | UI | Points 3 |  | In progress | Bayard |
| 17 | Player stats | As a user I want to see player stats | UI | Points 3 |  | To do |  |
|  |  |  |  |  |  |  |  |

1. **Acceptance Criteria (AC)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User Story ID and Name** | **AC**  **ID** | **Description of Acceptance Criterion** | **Status (completed, toDo, inPprogress)** | **Developer Names** |
| 1 flask and Django spike | 1.1 | Research are present finding to team | Done | Bayard |
| S:2 MYSQL | 2.1 | Connect MYSQL to Django app | Done | Sabrina |
| S:3 set up mvp for app back end | 3.1 | Set up base dragon app | Done | Bayard |
| S:4 registration page view and logic | 4.1 | Registration page renders | Done | Sabrina |
|  | 4.2 | Registration page connects to DB and Django app | Done | Sabrina |
|  | 4.3 | Sign up pages renders | Done | Sabrina |
|  | 4.4 | Sign up page connects to DB and re routs to loving page | Done | Sabrina |
| S:5 VUE research spike | 5.1 | Research are present finding to team | Done | Zeal |
| S:6 Look into Django Views research spike | 6.1 | Decide on whether to integrated VUE into Django present finding to team | Done | Zeal, Sabrina |
| S:7 build board class | 7.1 | Build a board class that generates a board data structure | Done | Bayard & Usman |
|  | 7.2 | add basic read and write functions for board | Done | Bayard |
|  | 7.2 | Add unit test | Done | Bayard |
| S:8 build game class | 8.1 | Build back in class that rules game logic | Done | Usman |
|  | 8.2 | Unite testing for this story has been reassigned to story S:18 | To do | Usman |
| S:9 evaluate communication channels | 9.1 | research communication needed for front end | Done | Zeal & Usman |
| S:10 build piece class | 10.1 | Build basic pieces class | To do | None |
|  | 10.2 | add basic read and write functions for board | To do | None |
|  | 10.3 | Add unit test | To do | None |
| S:11 transfer prototype into Django | 11.3 | See what code can be ported over to Django app | Done | Zeal |
| S:12 build player class | 12.1 | Build a board class that generates a board data structure | In progress | Ergin |
|  | 12.2 | add basic read and write functions for board | In progress | Ergin |
|  | 12.3 | Add unit test | In progress | Ergin |
| S:13 add testing library | 13.1 | Integrated purest with Django | Done | Bayard |
| S:14 connect board to front end UI | 14.1 | Set up routing page for where game will render | Done | Bayard |
|  | 15.1 | Visual test to make sure pages render properly | Done | Bayard |
| S:15 change Django view functions to classes | 15.1 | Change over all view functions to class form | In progress | Sabrina |
|  | 15.1 | Visual test to make sure pages render properly | In progress | Sabrina |
| S:16 add rules page | 16.1 | Render page with game rules on it | In progress | Bayard |
| S:17 add player states page | 17.1 | Render page that shows player stats | To do | None |
| S:18 unit testing for game class | 18.1 | Write unite test for game class | To do | Usman |

1. **Implementation Tasks**

Summary of production code

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Story ID and Name** | **AC ID** | **Class Name(s)** | **Method Name(s)** | **Developer Name(s)** | **Status** | **Notes (optional)** |
| S:3 set up mvp for app back end | 3.1 | Code generated in app set up | Code generated in app set up | Bayard | Done |  |
| S:2 MYSQL | 2.1 | Db connects to app |  | Sabrina | Done |  |
| S:4 registration page view and logic | 4.1 | Html page |  | Sabrina | Done |  |
|  | 4.2 | Used Django models |  | Sabrina | Done |  |
|  | 4.3 | Html page |  | Sabrina | Done |  |
|  | 4.4 | Used dragon models |  | Sabrina | Done |  |
| S:7 build board class | 7.1 | Board class | init()  Generate\_board()  get\_spaces()  space\_swap() | Bayard | Done |  |
|  | 7.2 | Board class | init()  Generate\_board()  get\_spaces()  space\_swap() | Bayard | Done |  |
|  | 7.3 |  | test\_board\_build()  test\_board\_generate()  test\_borad\_get\_spaces | Bayard | Done |  |
| S:8 build game class | 8.1 | Moving errors  State | ner()  move()  Errors  did\_end()  simple\_move()  jump\_avaialbe()  farther()  pieces\_after\_simple\_moce()  Piece\_after\_jump() | Usman | Done | This story has multiple class and functions and has stories for refactor added unit testing |
| S:9 evaluate communication channels | 9.1 |  |  | Zeal | Done |  |
| S:13 add testing library | 13.1 | Integration with Django tuning purest in .venv run successfully |  | Bayard | Done |  |
| S:14 connect board to front end UI | 14.1 | Html page renders |  | Bayard | 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)** |
| S:7 build board class | 7.1 | No need for class test yet | test\_board\_build test\_board\_game test\_board\_getspaces | Simple unite test, check data type, input files, and object length | Done | Bayard Rucker |
| S:18 unit testing for game class | 18.1 | Pytest does not need a class | To do | To do | To do | Usman |

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 play able from command line |  | Done |  | Usman |
| S:14 connect board to front end UI | 14.1 | Page renders |  | Done |  | Bayard |

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. **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 base team. Talked about possible approaches and high level details like languages |
| 8/27 | 45 min | In person | Bayard, Ergin | Team formation | Set up base team. Talked about possible approaches and hight 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 meeting decided not to use VUE for the front end. Base app set up |
| 9/20 | 1.5 hours | Zoom | Full team | Week meeting time | In a effort to be more agile we followed agile practiced listed in meeting agenda doc added testing lib, 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 new stories |
| 10/8 | 45 min | Zoom | Bayard,  Usman | Code review | Review of game class and talk about next steps |

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 |
|  |  |  |  |  |  |  |