**Team Project Sprint #2**

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**

1. Update the user stories and acceptance criteria of the target software that allows a human player to play against either a human opponent.
2. Specify the requirements (user stories and acceptance criteria) of the target software that allows a human player to play against the computer opponent (Do not call it AI). The user stories and acceptance criteria must cover the following features separately:

* Placing a piece
* Moving a piece
* Flying a piece
* Removing an opponent’s piece

1. Implement all the user stories for a human player to play complete Mill games against a human opponent (including all improvements on the previous sprint). Every member should contribute to the coding.
2. Conduct a full retrospective meeting (refer to the lecture notes) and report the meeting minutes.

**Deliverables and Grading Policy**

Please zip the project report, the demonstration video, and all source code into one .zip file before submitting.

Please don’t upload rar files and don’t submit links.

1. Project Report **(25 points)**

The project report should include the following sections. Please use the attached template.

* 1. Updated complete user stories using the template discussed in class. **(1 points)**

Provide a complete list of user stories and estimated efforts for the target software that allows a human player to play against either a human or a computer opponent.

* 1. Updated complete acceptance criteria using the template discussed in class. **(10 points)**

Provide complete acceptance criteria for all the user stories. If your acceptance criteria in sprint 1 are not good, you must fix them; otherwise, you will lose points.

* 1. Implementation tasks **(12 points)**

Provide source code metrics (number of classes/methods, lines of code) and describe the production code, automated test code or manual test cases for all the user stories for a human player to play complete Mill games against a human opponent. For each acceptance criterion of every user story, you need to implement at least one test (either test code or manual test case). Some automated tests using xUnit or a similar tool are required.

* 1. Source Code Summary **(1 points)**

Provide a zip file of all source code and summarize each member’s contribution. You will receive no credit if the source code is not submitted. Make sure your project report is consistent with the source code.

* 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).

1. Demonstration **(6 points)**

Submit a 5-minute video, clearly demonstrating that:

1. your project has implemented the following features for a human player to play complete Mill games against a human opponent.

|  |  |  |
| --- | --- | --- |
|  | **Feature** | **Score** |
| 1 | Place a piece at a vacant point and attempt to place a piece at an occupied point |  |
| 2 | Remove a piece that is not part of a mill, remove a piece from an opponent's mill, attempt of an invalid removal from an opponent's mill |  |
| 3 | Move a piece to a vacant point and attempt to move a piece to an occupied point |  |
| 4 | Fly a piece |  |
| 5 | Demonstrate a complete game (win without fly) |  |
| 6 | Demonstrate a complete game (win with fly) |  |
| 7 | Demonstrate other features or enhancements (optional) |  |

1. for each acceptance criterion of an implemented user story, your project has implemented either an automated test method or performed an acceptance test manually.

* Automation of unit tests (10+) is required.

1. overall presentation (**2 points**) using the following evaluation rubric:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Poor | Fair | Good | Very Good | Excellent |
| Was the demonstration logically organized |  |  |  |  |  |
| Were points made clearly and concisely |  |  |  |  |  |

**Team Project Sprint #2**

Report Template

Team Name:

|  |  |  |  |
| --- | --- | --- | --- |
| **Information provided by the student team** | | **To be used by grader** | |
| **Student name** | **Specific contributions to this sprint** | **Team Score** | **Individual Score** |
| Student 1 |  |  |  |
| Student 2 |  |  |  |
| Student 3 |  |  |  |
| Student 4 |  |  |  |

A student without specific contributions shall not receive credit.

1. **Updated User Stories**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **User Story Name** | **User Story Description** | **Priority** | **Estimated effort (hours)** | **Actual effort (if completed)** | **Status (completed, toDo, inProgress)** | **Developer names** |
| 1 |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |
| .. |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User Story ID and Name** | **AC**  **ID** | **Description of Acceptance Criterion** | **Status (completed, toDo, inPprogress)** | **Developer Names** |
| 1 story one | 1.1 | AC 1.1 <scenario description>  Given  When  Then |  |  |
|  | 1.2 | AC 1.2 <scenario description>  Given  When  Then |  |  |
|  | … |  |  |  |
| 2 story two | 2.1 | AC 2.1 <scenario description>  Given  When  Then |  |  |
|  | … |  |  |  |

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)** |
| 1 | 1.1 |  |  |  |  |  |
|  | 1.2 |  |  |  |  |  |
|  | … |  |  |  |  |  |
| 2 | 2.1 |  |  |  |  |  |
|  | … |  |  |  |  |  |

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)** |
| 1 | 1.1 |  |  |  |  |  |
|  | 1.2 |  |  |  |  |  |
|  | … |  |  |  |  |  |
| 2 | 2.1 |  |  |  |  |  |
|  | … |  |  |  |  |  |

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)** |
| 1 | 1.1 |  |  |  |  |  |
|  | 1.2 |  |  |  |  |  |
|  | … |  |  |  |  |  |
| 2 | 2.1 |  |  |  |  |  |
|  | … |  |  |  |  |  |

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. **Summary of Source Code**

|  |  |  |  |
| --- | --- | --- | --- |
| Production or test code | Source code file name | # lines of code | Developer names and contributions (% of the source code) |
|  |  |  |  |
|  |  |  |  |
| Total | |  |  |

1. **Meeting Minutes (only during this sprint)**

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** |
| 03/17/21 | 7pm - 8pm | Zoom | Susma Gajmer  Irvin Jimenes-Solis |  |  |
|  |  |  |  |  |  |

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* | | | | |
|  | Full Name 1 | Full Name 2 | Full Name 3 | Full Name 4 |
| Full Name 1 | X |  |  |  |
| Full Name 2 |  | X |  |  |
| Full Name 3 |  |  | X |  |
| Full Name 4 |  |  |  | X |
|  | *Average* |  |  |  |  |