Sprint Planning

|  |  |  |
| --- | --- | --- |
| Team Name | Sprint Start Date | Sprint End Date |
| Chess Masters | 2020-08-04 | 2020-08-10 |

|  |  |  |
| --- | --- | --- |
| Stories Committed To | Points Committed To | Estimated Hours |
| 5 | 10 | 8.5 h |

# Sprint Backlog

*Replace this text with the User Stories and Tasks the team commits to complete in this sprint. Include the name of the team member responsible for completing each task and an estimate on the hours it will take them to complete it.*

Dylan Roberts

User input (2 h)

Validate commands, parse them, and pass them into the board

Joe Reed

Print board to console (0.5 h)

Initial board setup (2 h)

Initialize board with pieces (0.5 h)

Validate a given move according to the layout of the board (go/no go) (1.5 h)

Pieces in the way

Capturing

Don’t worry about check yet

Travis Eggett

Model pieces (2 h)

Store position

Validate a given move according to the piece’s rules (go/no go)

Josh Conlon

Set up two-player system, alternating turns (2 h)

Sprint Retrospective

|  |  |  |
| --- | --- | --- |
| Stories Completed | Points Completed | Actual Hours |
| 4 | 7 | 9 h |

# What was good?

* Communication was effective
* No conflicts or disputes
* Roles and tasks were clearly understood

# What was bad?

* Some tasks in a dependency chain took too long to complete, preventing the next steps from being taken
* Code on GitHub wasn’t easy to keep up-to-date with our local repositories
* Time tracking was ineffective, and the resultant data does not help us analyze our work patterns

# Ideas

* Come up with a way to test everything easily and visually
* Get everyone on the same page codebase-wise

# Actions

* Tie together individual components of the code
  + From here it should be easier to manage version control
* Develop a simple test harness that demonstrates unit tests visually