

AGILE SPRINT 4 BACKLOG AND BURNDOWN CHART

Add rows and columns as needed.

| BACKLOG TASK & ID   | STORY POINTS | ASSIGNED TO       | STATUS      | ORIGINAL ESTIMATE | 4/12  | 4/13  | 4/14  | 4/15 | 4/16 | 4/17 | 4/18 | 4/19 | 4/20 | 4/21 | 4/22 | 4/23 | 4/24 | 4/25/ | 4/26 | 4/27 | 4/28 | SPRINT REVIEW |
|---|--------------|-------------------|-------------|-------------------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|-------|------|------|------|---------------|
| <b>User Story #1 The winner will be the wealthiest player through buying, selling and selling property. If all Players agree to end game, the winner is the wealthiest Player. If all but one player has gone bankrupt then that Player is the winner</b> |              |                   |             |                   |       |       |       |      |      |      |      |      |      |      |      |      |      |       |      |      |      |               |
| Task: Add action for players to end the game. Add separate quit operation so players can quit when convenient.  | Small        | Sydney            | Complete    | 1                 | 1     | 1     | 1     | 1    | 1    | 1    | 1    | 1    | 0    | 0    | 0    | 0    | 0    | 0     | 0    | 0    | 0    |               |
| Task: Add quit option for HumanPlayer to quit game.   |              | Sydney            | Complete    | 1                 | 1     | 1     | 1     | 1    | 1    | 1    | 1    | 1    | 0    | 0    | 0    | 0    | 0    | 0     | 0    | 0    | 0    |               |
| Task: Automate ComputerPlayer, so when all HumanPlayer quit they will quit.   |              | Sydney            | Complete    | 2                 | 2     | 2     | 2     | 2    | 2    | 2    | 2    | 2    | 0    | 0    | 0    | 0    | 0    | 0     | 0    | 0    | 0    |               |
| Task: Add component so that game tracks the wealthiest player.  |              | Sydney            | Complete    | 1.5               | 1.5   | 1.5   | 1.5   | 1.5  | 1.5  | 1.5  | 1.5  | 1.5  | 1.5  | 1.5  | 1.5  | 1    | 0.5  | 0     | 0    | 0    | 0    |               |
| Task: Add Tests to make sure logic is correct.  |              | Sydney            | Complete    | 1.5               | 1.5   | 1.5   | 1.5   | 1.5  | 1.5  | 1.5  | 1.5  | 1.5  | 1.5  | 1.5  | 1.5  | 1    | 0.5  | 0     | 0    | 0    | 0    |               |
| <b>US #1 One of the 2 to 4 players must be a Player simulated by the computer</b>   |              |                   |             |                   |       |       |       |      |      |      |      |      |      |      |      |      |      |       |      |      |      |               |
| Task: make startup screen   | Large        | Sean              | Complete    | 1.5               | 1.5   | 1.5   | 1.5   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     | 0    | 0    | 0    | 0             |
| Task: Startup screen must get player names and tokens, as well as number of CPU and human players   |              | Sean              | Complete    | 0.5               | 0.5   | 0.5   | 0.5   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     | 0    | 0    | 0    | 0             |
| Task: Startup screen must communicate to gameboard and gamestate how many human and CPU players there are and resolve accordingly.  |              | Sean              | Complete    | 1                 | 1     | 1     | 1     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     | 0    | 0    | 0    | 0             |
| Task: make button that leads to gameboard   |              | Sean              | Complete    | 0.25              | 0.25  | 0.25  | 0.25  | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     | 0    | 0    | 0    | 0             |
| <b>US 23: The Controller is a set of classes that implement the behaviors for the components displayed by the GUI</b>   |              |                   |             |                   |       |       |       |      |      |      |      |      |      |      |      |      |      |       |      |      |      |               |
| Task: Every GUI component with a text file associated to it has an interactive controller class. Task: Controller classes all have button listener methods appropriate to functionality.  | Large        | Jon               | complete    | 2                 | 2     | 1     | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     | 0    | 0    | 0    | 0             |
| Task: Controller class methods are accessible from the Controller interface for me.   |              | Jon               | complete    | 2                 | 2     | 2     | 2     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     | 0    | 0    | 0    | 0             |
| Task: All controllers have a purpose and all methods are useful without code repetition.  |              | Jon               | complete    | 3                 | 3     | 3     | 3     | 3    | 3    | 3    | 2    | 0    | 0    | 0    | 0    | 0    | 0    | 0     | 0    | 0    | 0    | 0             |
| Task: All controllers have a purpose and all methods are useful without code repetition.  |              | Jon               | complete    | 1                 | 1     | 1     | 1     | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1     | 0    | 0    | 0    | 0             |
| <b>US 18: The Game State should provide an interface to the Controller and the View so that they can modify the Game State and get the current status of the Game State.</b>  |              |                   |             |                   |       |       |       |      |      |      |      |      |      |      |      |      |      |       |      |      |      |               |
| Task: Update player position on board and return dice.  | Medium       | Jon               | complete    | 2                 | 2     | 0     | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     | 0    | 0    | 0    | 0             |
| Task: Displaying messages on the gameboard  |              | Sydney            | Complete    | 1.5               | 1.5   | 1.5   | 1.5   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     | 0    | 0    | 0    | 0             |
| Task: Display the accurate number of houses on each property.   |              | Sydney            | Complete    | 1                 | 1     | 1     | 1     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     | 0    | 0    | 0    | 0             |
| Task: Display Banker information on GUI   |              | Sydney            | Complete    | 3                 | 3     | 3     | 3     | 1    | 0.5  | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     | 0    | 0    | 0    | 0             |
| Task: Starting a trade/auction  |              | Sydney            | Complete    | 2.5               | 2.5   | 2.5   | 2.5   | 2.5  | 2.5  | 2.5  | 2.5  | 2.5  | 2.5  | 2    | 1    | 1    | 0    | 0     | 0    | 0    | 0    | 0             |
| Task: Add Tests to make sure logic for trade is correct.  |              | Sydney            | Complete    | 1.5               | 1.5   | 1.5   | 1.5   | 1.5  | 1.5  | 1.5  | 1.5  | 1.5  | 1.5  | 1.5  | 1    | 0.5  | 0    | 0     | 0    | 0    | 0    | 0             |
| <b>Miscellaneous/Cosmetic Items</b>   |              |                   |             |                   |       |       |       |      |      |      |      |      |      |      |      |      |      |       |      |      |      |               |
| Add Styles/Themes to the Interface  | Medium       | Sean              | Complete    | 1.5               | 1.5   | 1.5   | 1.5   | 1.5  | 1.5  | 1.5  | 1.5  | 1.5  | 1.5  | 1.5  | 1.5  | 1.5  | 1.5  | 0     | 0    | 0    | 0    | 0             |
| Add Music to the Game   |              | Sean              | Incomplete  | 1.5               | 1.5   | 1.5   | 1.5   | 1.5  | 1.5  | 1.5  | 1.5  | 1.5  | 1.5  | 1    | 1    | 1    | 1    | 1     | 1    | 1    | 1    | 1             |
| Banker Info Screen Configured   |              | Sean              | Complete    | 2                 | 2     | 2     | 2     | 2    | 2    | 2    | 2    | 2    | 2    | 2    | 2    | 2    | 2    | 2     | 2    | 2    | 0    | 0             |
| Quit/End Game Feature w/ Tests  |              | Sydney            | Complete    | 2                 | 2     | 2     | 2     | 2    | 2    | 2    | 2    | 2    | 2    | 2    | 2    | 2    | 2    | 2     | 2    | 2    | 2    | 2             |
| <b>Refactoring Sprint 3</b>   |              |                   |             |                   |       |       |       |      |      |      |      |      |      |      |      |      |      |       |      |      |      |               |
| Task: Refactor Computer Player  | Extra Large  | Sydney, Sean, Jon | Complete    | 1.5               | 1.5   | 1.5   | 1.5   | 2    | 3    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     | 0    | 0    | 0    | 0             |
| Task: HumanPlayer refactor for individual actions   |              | Sydney, Sean, Jon | In-Progress | 1                 | 1     | 1     | 1     | 2    | 2    | 2    | 2    | 2    | 2    | 2    | 4    | 4    | 4    | 3     | 2    | 2    | 1    | 1             |
| Task: Refactor auction process with gui implementation.   |              | Sydney, Sean, Jon | In-Progress | 1                 | 1     | 1     | 1     | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 3     | 3    | 3    | 3    | 3             |
| Task: TileOwnedCards data is updated dynamically.   |              | Sydney, Sean, Jon | Complete    | 0.5               | 0.5   | 0.5   | 0.5   | 0.5  | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0     | 0    | 0    | 0    | 0             |
| TOTAL   |              |                   |             | 39.75             | 39.75 | 36.75 | 35.75 | 27.5 | 27.5 | 24   | 23   | 21   | 18   | 17   | 14.5 | 15   | 12.5 | 11    | 10   | 10   | 7    | 7             |

