CS 3110: **Monopoly**
Authors: Kevin Cook (kjc244), Robert Villaluz (rcv37), Charlie Friedberg (cbf43), Gregory Schultz (gjs86)

***Vision**: In one paragraph, what is your current vision for the system you are building? How has it evolved from previous sprints?*

Our vision has evolved from previous sprints in that we committed to implementing advanced features such as loading/saving and an AI player. We agreed that including these features would not only result in a complete, standalone product, but it would improve upon the limitations of a real-life Monopoly game.

***Summary of Progress:***

As per our "Satisfactory" scope, all game functionality we listed has been completed. This includes Chance/Community cards, Jail, houses/hotels, and appropriate GUI renderings as well as thorough testing.

Additionally, we accomplished the game state saving and loading as per our "Good" scope. This enables players to save their current points in monopoly games to pick up and play later. This functionality doubled as a feature and a testing tool, as being able to see our game state at a certain point was invaluable as we tested edge cases throughout the development of the more advanced game functionality.

Finally, we were able to create an AI as per our "Excellent" scope. The two main challenges for this were (1) modifying the existing game infrastructure to support an AI and (2) implementing logic that would make the AI a challenging opponent. We certainly accomplished the first goal as an AI player is fully autonomous, requiring no human interaction. In fact, one can initialize a game consisting of only AI players and this will progress until one AI remains as the winner. The AI operates within the game's existing infrastructure just as a human player would. For instance, in order to mortgage or buy or sell a house, the AI can open the main menu and navigate to the appropriate option. Similarly, we accomplished the second goal for our AI by drawing from Tim Darling's "surefire" monopoly strategy as outlined in this article. More specifically, the AI prioritizes its monopolies over other properties, prioritizes investments in monopolies that are closer to the "Go" tile, and mortgages its cheapest available properties in order to raise cash.

***Activity Breakdown:***

**Kevin**:

- Implemented all aspects of the user interface (known as KUI)
- Worked with other team members to integrate the rest of the game with the UI
- Worked on parts of main with Bobby and Charlie

**Bobby:**

- Implement AI player
- Complete Main module
    - Game functionality (jail, buying houses, changing turns, mortgageing)
    - Modified game infrastructure to support an AI player

**Charlie:**

- Implement state module, main, and the AI
    - Help other teammates with their work
- Implement parts of game not related to json parsing.

**Greg:**

- Implemented JSON => OCaml conversion and getters in game.ml
    - Defined monopoly.json file as well as zelda.json (to test engine capability)
- Game, State, Command testing in test.ml
- Implemented game state Saving + Loading functionality

***Productivity Analysis:***

As a team, we believe we have successfully created a fully capable Monopoly game. At times it may have felt like we were not going to accomplish our stretch goals like the AI, but through the continued use of predefined roles and modular programming, we were able to accomplish all of our stated goals in an efficient manner. We also spent a lot of time on this.

***Coding Standards Grades:***

Documentation: Meets Expectations
Testing: Meets Expectations
Comprehensibility: Meets Expectations
Formatting: Meets Expectations

Our documentation meets expectations as we commented every type and name in the necessary .ml and .mli files. We ensure that comments in the .mli files do not assume anything about the implementation. More importantly, our comments are written using terms relevant to understanding how Monopoly works. Our testing file is complete and thorough, testing all core game functionality in a variety of combinations.

We believe that our code's comprehensibility meets the rubric's expectations as almost all of the code is quite readable.The notable exceptions are the "continue_game" loop in Main and GUI. At first glance, main might seem like a mess. However, monopoly is a game with lots of different scenarios and cases. There is no way to somehow make them all concise. We tried our best to keep main clean, and we think we did a good job doing so. Since GUI relies on the ANSITerminal library and positions within the terminal, it is difficult to understand at first glance. However, we still believe that even these functions are comprehensible and well-organized. As per discussion with the course staff, we have removed comments for public functions in all of our .ml files to reduce copying and increase clarity. Lastly, our formatting meets expectations as we have ensured that all of our relevant .ml and .mli files have lines less than 80 characters and are properly indented.

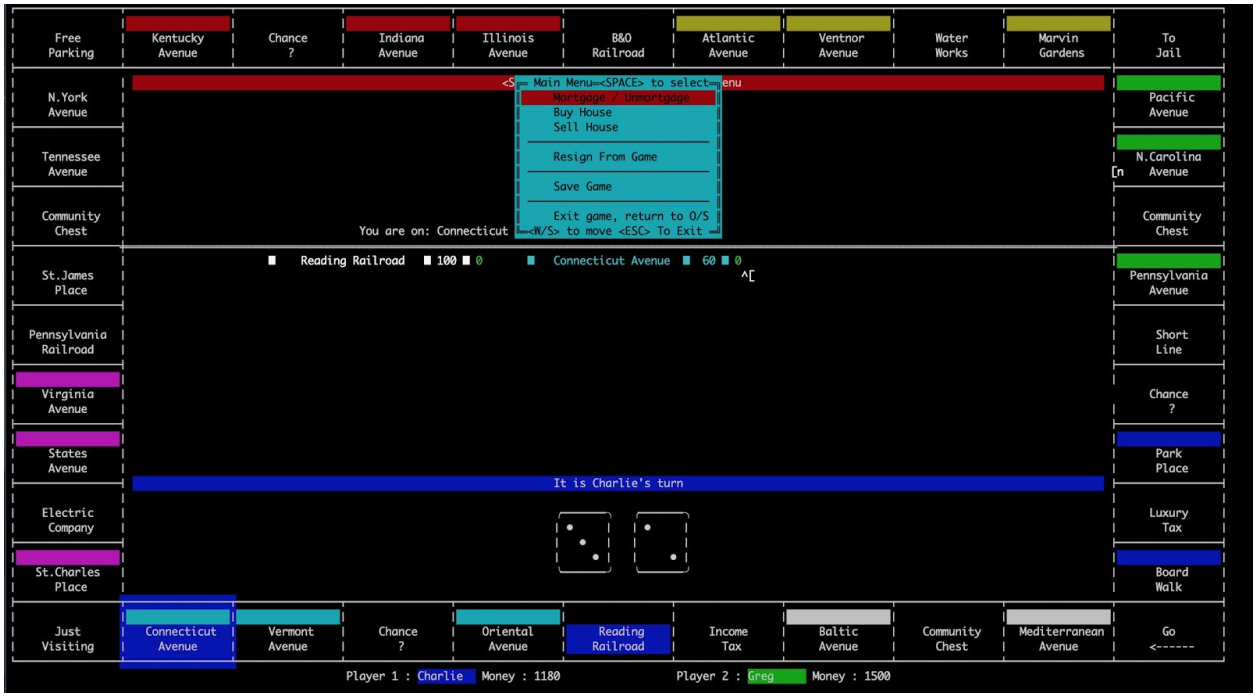
***Scope Grade:***

As we were able to successfully accomplish our stated goals for the project (surprisingly, we might add), we would assign a final scope grade of *Excellent*. Because of the way we structured our game logic, it wasn't painfully difficult to add more functionality once we got the

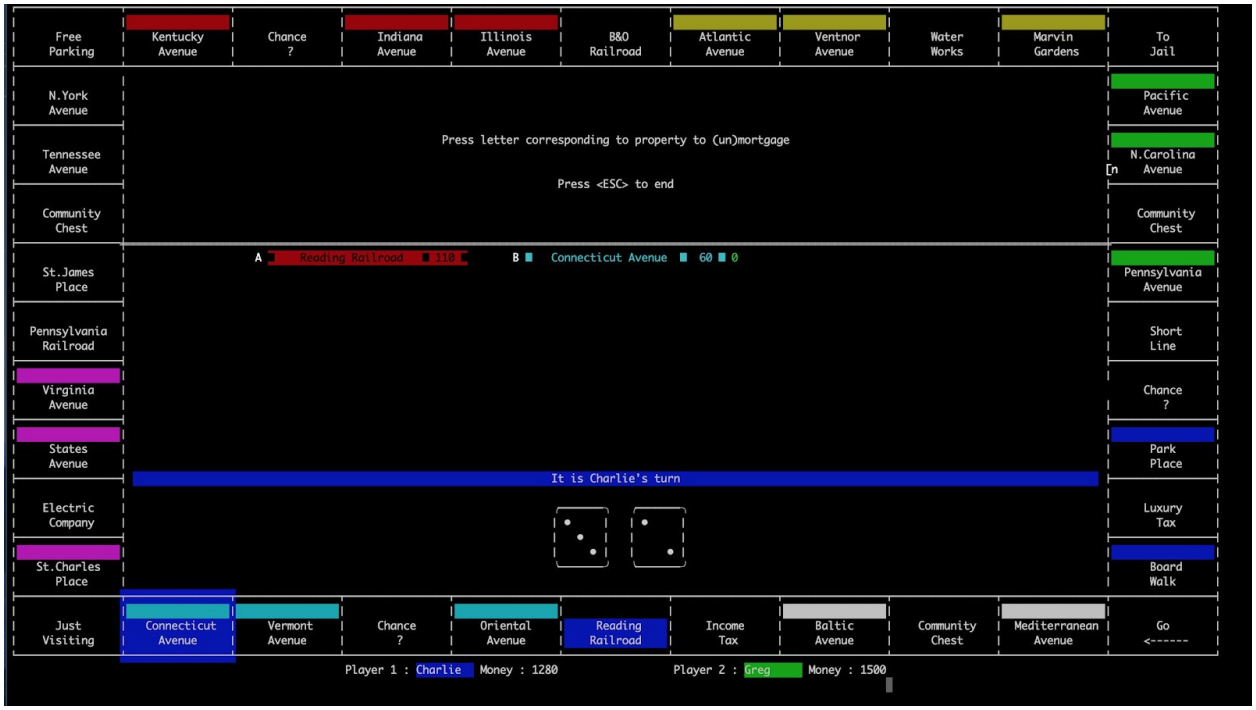
foundation done. For example, our cash menu took a while to make. But once it was made, it became trivial to employ in situations that should prompt it. In regards to overall user value, we feel that the AI, coupled with the saving and loading are features that augment our game in an immense way. These advanced features required research and thought beyond the scope of projects like A5, pushing our project to its maximal potential. It's important to note that these advanced features are in addition to the multitude of monopoly rules that we have implemented. Implementing these basics required much of our time, but we felt that by elevating the game of monopoly with additional features, we created a product that goes above and beyond the user value of a regular monopoly game.

***<u>Screenshots of our Completed Game</u>***

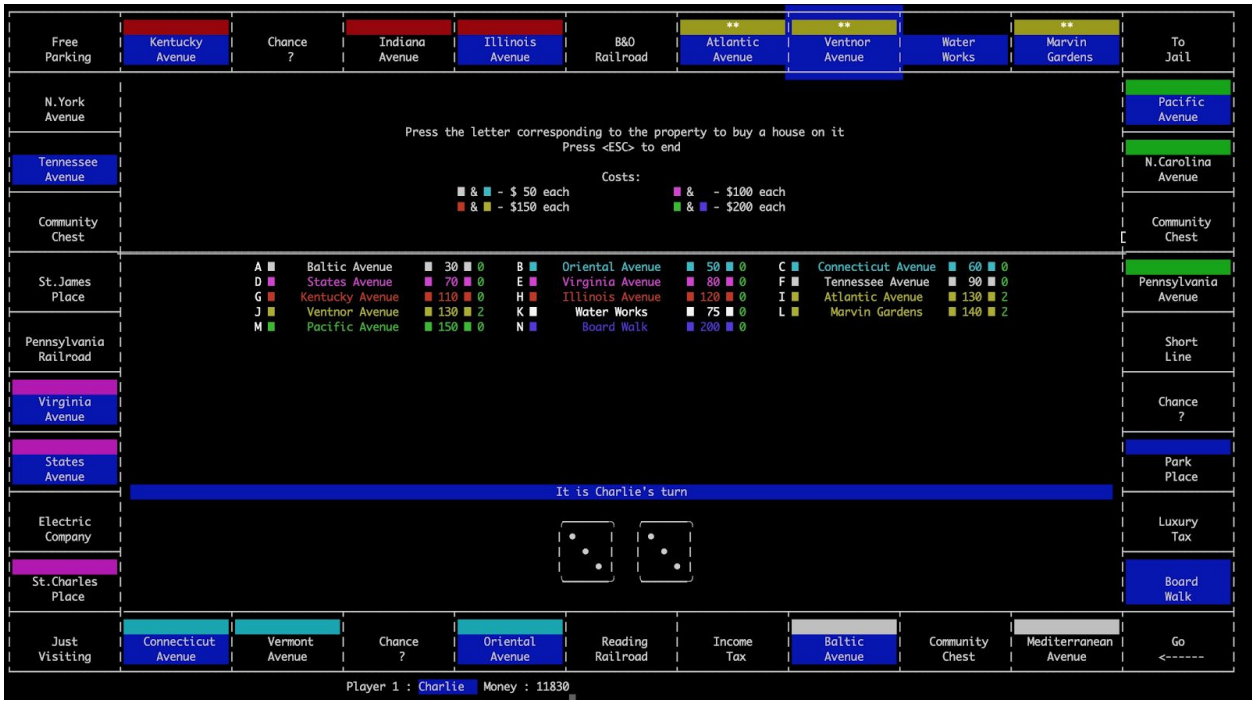
***New Menus***



***Mortgages (highlighted in red)***




***Displaying monopoly and houses (see the upper yellow color group)***




***Note that we are playing the above game with one player to get a monopoly sooner. The actual game doesn't let you play with one player.**