

1.0 GUI Report

1.1 Process

The GUI design process consisted of three distinct phases: group discussion with mind maps, initial prototyping and finally a group based iterative critiquing method.

To come up with initial design ideas for each stage in our game, each member of the group made some basic designs on paper. We then brought our ideas together in a group discussion, where each member presented their design to the group. After all the designs for an aspect of the game had been presented, we picked which elements of each design we wanted to bring forward. This process provided us with the base of the design and its interaction model in the form of paper prototypes.

Following this we proceeded to critique the design using an iterative process, suggesting and agreeing upon changes at each iteration until we were happy with the final design. During this design process we consulted the customer on any major decisions to ensure that their requirements were achieved and were of a deliverable standard. This design process was in place to complement our development methodology. Its iterative nature along with constructive feedback has parallels with a SCRUM sprint, meaning that it could be easily incorporated into our team, and the designs could evolve substantially in a relatively short amount of time.

1.2 Design (*The designs can be viewed [here](#) [1]*)

In order to fulfill our requirement to create a functional and enjoyable game we designed the GUI in order to provide a visually appealing, efficient and intuitive user experience. We decided that colour coding the GUI based on the player's selected college would provide an intuitive way for the player to identify game elements which belong to them, such as map tiles and market trades. This also provides a sense of personalisation for the player.

One of the first designs we looked at was the heads up display (HUD), which is present at most stages of the game. The HUD has been influenced mainly by requirements; F3, N4, F15. These requirements have directly impacted on our choice to include resource information in the HUD. This ensures the player can make quick, informed decisions during timed phases (N1). The HUD will contain all the high level information the player will need, meaning there are few superfluous clicks, saving time. This removes the need for a separate window for displaying resource/tile statistics, keeping the player immersed and reducing the distance the cursor and eyes have to travel.

The market is presented in a tabular format such that all information regarding an individual resource is grouped (buy/sell price and trade history), this distinction allows for users to focus on a singular task and allow them to quickly reach informed decisions and devise strategies.

In accordance with performance requirement P1 a box overlay containing tile details will be displayed on tile selection. Being able to click directly on a tile and receive its information makes this action intuitive and easy to use.

Throughout we have ensured consistency in the use of colours and fonts for GUI elements such as menus, information panes and overlays. There are 2 cases where the use of colour can be directly attributed to the effect of requirement F12; font colour in trading listing and tile boundaries. The colours of these aspects allow for the players to easily distinguish themselves from each other and provide a visual indicator of the current state of the game, thus providing an intuitive means of monitoring each other's progress.

The inclusion of graphs was inspired by the functional requirement F7 so that users may track historical prices and make strategies. The graph colouring was designed based on the assumption that information of all resources would be viewed on a singular plane, meaning that complementary colours would have to be selected to make easy viewing.

Bibliography

[1] SEPR, "GUI Designs". [Online]. Available:

Whiteboard (Initial Mockups):

https://seprated.github.io/Assessment2/whiteboard_mockups.zip [Accessed 22nd January 2017].

Photoshop Mockups: