Requirements Update

Summary: the main focus was to get most of the functional requirements in place before moving onto the non-functional that main reason was to allow for testing before moving to design. Using the 80 20 rule this was better than implementing the non functional because most of those requirements aren’t needed for it to be a “game”.

Functional

* 1. “Game Basics and Rules” This concept was almost fully complete in the prototype stage and only needs some tidying up and more testing done before it is done. This was close to being #1 priority because it defines the game itself.
  2. “Left Click Action” This requirement was implemented during the prototyping and seemed to work well. Slight tweaks are still needed to make it more robust and failsafe. The prototype implementation was more or less “Get it working” not get it working the best way it can. I would like to revisit and factorise a lot of the code before moving forward.
  3. “Right Click Action” Similar to left click …..
  4. “Game Variants” \*\*talk about hexagons functionality being done in prototyping but need to implement it to be more solid and implement a menu to switch between the modes.\*\*
  5. “Timer and Scoring” \*\*start implementing this when the menu is done and this may lead to persistent score keeping\*\*

Non-Functional

2.1 “Speed” This hasn’t been the main focus of the game but definitely been monitored throughout the development it will be revised during every major requirement addition. This is something hard to test until a rough Alpha game is completed/

2.2 “Customisability” Because design is completed, and the main focus is functionality this requirement cannot be implement until the end phase of development. This stage will be implemented towards the end once the core game is done.

2.3 “High Scores” Linking back to the timer and scoring this won’t be implemented until the timer and scoring is done but this will allow for persistent scores to be kept using SQLite

2.4 “Visually pleasing” one of the last things to be done along with the speed

2.5 “Colour coding” similar to the previous requirement

Product Use Cases

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Summary of software architecture

Summary of design

Design goals are mainly outlined in non functional requirements might use some wireframes to demo the idea and some concept art found online as inspiration. Revisit the class diagrams from milestone 1 and add ui class diagrams (think it means UML but double check) also look into the other diagrams.

Discussion of the level of sophistication regarding Persistent Data management

Talk about DB design and how the scoring will be kept.

Describe what sections of the code have been tested

Use a testing thingo like in systems to describe the testing used. Throw in some TDD.

Extract logs of version control commits.  
Bullshit this part kinda since all the work has been done in single days 😊

Look into other software patterns