**See the Whole** (10/02)

This week we scoped out weeks to come. Basically planning tentative weekly goals/ meet up schedules and division of tasks within the group.

Tasks accomplished this week:

* Each of us picked a UML diagram and started uploading their rough draft versions to GitHub. More than accuracy, the goal was to brainstorm ideas for our game. These would be eventually improvised after reviewing them as a team and after going over in class.
* To kick start, we first drew out our wireframe that we plan to follow closely in future. For more specifics, please check our Team GitHub directory.
* Another item was setting the stage ready for all of us to start coding. We brainstormed various elements that we might require in our world:
  + Finalizing entities/ graphics or images for the world: researching images required to set up our world
  + Creating multiple worlds with random graphs: finding a maximum of 3 graphs that we plan to solve for players to actually learn the algorithm while playing than memorizing patterns.
  + Selecting appropriates audios: to add being user interaction, we are looking into recording and downloading some audios for our game.
  + Planning scoring mechanism: we were having couple of ideas on implementing scoring system but need to try implementing and visualizing how it would work out.
  + Displaying result screen: it’s highly unlikely a user might lose our game as our main intention to leave them on a positive note therefore we would most like have a WON screen with final path highlighted.
* Currently our goal is to get our base version of the code ready so that we can scale it up to multi-player later on receiving further instruction during upcoming classes.

In the upcoming weeks, we shall look into:

* Integrating all of our pieces into the world and having some minor functionalities working.
* Design patterns that might work with our game – while giving it some room for growth in future by loosely coupling.
* Clarify and conceptualize how to implement multiplayer into the game.
* Come up with use cases/ test cases to validate functionality after development.
* Reuse and recycle code to maximize OOP