**Client-Side Work – Josiah and Tom**

1. Design layouts for the playback page and the test arena page - Tom
2. Display background in phaser - Josiah
3. Set animation loop that calls stub methods to animate each action in the change specification object, and respects the current timestep, as well as the playback controls – Josiah
4. Implement all of the playback controls – Tom + Josiah
5. Be able to initialize the display of text entities, objects, and sprites - Josiah
6. For each of the sprites, get the different animations to work for the 4 verbs -Josiah
7. Support changes for text entities - Josiah
8. Get the IDE textarea to work - Tom
9. Incorporate compiler errors into the IDE display -Tom
10. Client-side Bot Upload - Tom

I believe all server-side code will be written in Node JS, <https://nodejs.org/en/>

**Server-Side Work – Lee, Sawyer, Taha**

1. Server side code to pull a game from the database (for playback mode)
2. Server side code to pull a test-instance game (for test arena)
3. Server side code to upload a bot to the Bot Database
4. Server side code to upload a new turn request from the test arena
5. Server side code to allow polling to check for the completion of the turn request
6. Create client side code that polls for the completion of the turn request