**Final Project Documentation**

**Group Name: The Aliens**

**Group #: 3**

**Project Name: Spacetard**

**Design/Implementation:**

When it came to design, we began to think about a player and a chaser. We wanted it to be somewhat like a natural environment that could happen in nature, so we thought up different relations that happen in the wild between animals. For example, a killer whale and a seal, a deer and a bear, a squirrel and a pine marten, and things of that nature. Dylan came up with the idea of something space related, which instantly reminded me of tardigrades. Tardigrade are little aquatic creatures that can survive the vacuums of space by going into suspended animation. Julia thought it would be a cool idea to incorporate how tardigrades live in water with a space-like theme. We came up with the idea of an astronaut chasing a tardigrade to capture it, and the tardigrade collecting food to survive. When we realized the astronaut chaser would be too difficult, we decided we would incorporate a score and health. With the score, the score would increase after each collectable item was collected. Since the health needed to be decreased, but we couldn’t get the astronaut working, we decided to use the idea of bad collectables. Bad collectables were items (such as comets and asteroids) that would cause a negative effect if the player bumped into them. These collectables would also move around the screen, making it difficult to collect the good items (star leaves and star dust).

Essentially the game became a survival game, where the goal was for the tardigrade(player) to collect star-leaves and star-dust to increase the score, while avoiding bad items like comets and asteroids that would decrease the health. Upon health reaching 0, the game would be lost.

Good collectables and bad collectables were stored in separate lists to help with the organization. Keyboard movement used to let the player control the tardigrade. Communication between scenes via events (i.e. button pressed, score reaching 0, etc). COmmunication between objects (i.e. collision, pickup, etc). Used UI components to allow player to keep track of their score throughout the game. Communication between classes (SoundManager with PlayerMove), and used an array (for sound variation.)

**Programming issues:**

* Having the tardigrade wrap around the screen.
  + Solution: Changing the tardigrade’s position to start at the opposite end of the screen.
* Getting the collectibles to spawn randomly on the screen and within the screen size
  + Solution: Setting a variable to hold the random position for the good collectables. Deleting ViewportToWorldPoint because objects were not spawning within the random parameter.
* Getting the collectibles to not spawn on top of each other.
  + Solution: We couldn’t figure this out, so instead we decided that the asteroids and comets would always be moving so an asteroid or comet wouldn’t spawn on top of a good collectible and stay there, preventing you from collecting it.
* Hit Detection of player with collectibles
  + Solution: We needed a rigid body attached to each prefab and the player.
* Getting the asteroids and comets to move and wrap around the screen.
  + Solution: The asteroids were always moving towards the left because the Random.Range function was favoring the negative values. To fix we had to raise the second value by 1 so it would evenly distribute randomly in all directions.
  + Solution: Once the comets were off the screen an if statement was made so that if the y position was greater than -5, the comets were given a new position to start at y = 5, and a random x position.
* Scoring issue where hitting a collectible gave you more points then it should. Multiple prefabs had been instantiated on top of each other so when the player collided with it, it added the score for multiple collisions instead of one.
  + Solution: We had instantiated the prefabs twice. We took out the second instantiation.

**Each Person’s Accomplishment:**

**-Dina**: Helped get the tardigrade to move around the screen without getting keycodes and wrapping around. Figuring out how to get the collectibles to appear randomly on the screen when game starts. Was able to get the collectibles to not appear off the screen when instantiated. For the collision detection, helped with the rigid bodies. We wanted the comets to also wrap around the screen, so a translation was added and then an if statement to get the position of the comets. If statement made sure that if the comets passed a y direction of -5 the comets wrapped back to the top and a random x position. Helped with logic errors and meeting times to work on project and stay with the schedule we created.

**-Julia**: I designed the elements that would appear on the screen. I then created or modified them using Photoshop. This would be the background setting, the tardigrade, the Star-Dust, the Star-Leaves, the asteroid and the comet. I then created them into prefabs and helped with linking scripts to them and creating settings for the prefabs. I also designed and created the scene menus. I coded the implementation of the scenes as well using buttons and linked everything together(communication between scenes). Created and implemented UI elements. I also created the scene order in the build settings. Next, I added menu audio to work playing over the scenes and worked on making sure the audio didn’t overlap. I worked on the code to help transition from scene to scene depending on the event, such as the health reaching 0. I assisted helping my team and fixed code when there were logical errors, such as when the score was increasing rapidly due to having code instantiated multiple times. I was also responsible for a lot of the planning process and figuring out parts of the game logic. I created the SoundManager and wrote the code for that(some functions and an array of audio), and the implemented its communication with the playerMove class. The soundManager was responsible for collisions between the player and objects in which audio was produced.

**-Dylan**: Helped code the Tardigrade movement to get the tardigrade to move with the arrow keys and WASD. Helped figure out spawning of collectibles on screen randomly. Wrote the code for detecting collisions between the player and collectibles. I did the code for scoring and health based off of the collisions between good and bad collectibles. Handled the respawn of good collectibles once they are picked up. Made the rotating animations of the stardust and starleaf collectibles. I figured out the code to make the asteroids move and wrap around the screen. Helped with bugs and logic errors.

**Instructions:**

* **Movement**: Use arrow keys **OR** the W A S D keys to move the player around the screen.
* **Collect:** The star leaves and the star dust to increase the score while navigating the screen. Each item collected will increase your health by 2.
* **Avoid:** The asteroids and comets.