Production Log

09/08/2019

* Went over the base code to get my bearings
  + Ship
    - Turns left/right when the corresponding key is pressed
    - Accelerates forward when the up key is pressed
    - Moves according to Velocity
    - Generates a shield when z is pressed
      * 3 per life
      * Lasts 3 seconds
      * Allows ship to collide without losing a life
    - Fires a missile
      * Created when player presses [space]
      * Velocity and position determined by location and rotation of the ship
      * Move according to velocity
  + Rocks
    - Random starting velocity and rotation speed
    - Move according to velocity
    - Rotate according to rotation speed
    - Have three different sizes
      * Big
      * Medium
      * Small
  + Collisions
    - Missile with rock
      * Original rock and missile removed
        + If “large” rock

Create 2 “medium” rocks

* + - * + If “medium”

Create 2 “small” rocks

* + - * + If “small

disappears

* + - Ship with rock
      * Rock
        + Same rules as if a missile collided
      * Ship
        + Ship is removes
        + Player has 3 lives
        + If not last life lives is depleted by one and new ship appears

Appears in the center of the screen after 2 seconds pass

* Changes
  + Replaced the booster graphic from a static image to a animation loop
    - Still Having Issues Implementing in game but it runs through the
  + Stopped Ship from wrapping
    - The Shipping being able to wrap can create a confusing setup for the player
      * This way the player has a clearer sense of where the ship is going to be
  + Added Rock Collision allowing rocks to “bounce” off each other
* Testing
  + Had trouble implementing booster in game due to inexperience with animate (YouTube helped).
  + The Rock to Rock collision uses the physics equation
    - v1 = (u1(m1-m2)+(2\*m2\*u2))/(m1+m2)
    - utilizes values set in the rocks array and a mass value decided by each rock’s radius
    - not sure if asteroids colliding with asteroids should damage both or not