Asteroids – Documentation

Programming Journey

Before we started to write any code we familiarized ourselves with the pygame library since it is essential to the game. We followed some tutorials from YouTube and some from the library's official website (https://www.pygame.org/wiki/tutorials). After we achieved a basic understanding we started with the code for the game.
First, we constructed the board in the main method and chose pictures for the ship the player will be controlling, background, bullets fired by the ship and for the asteroids. As suggested by the tutors we went with a single picture for all asteroids and only varied their appearance in size. Next we created the player class with the picture. At this point you can already start the game and see the ship in the middle of the board but can not move or shoot. Our next step was to create the asteroids, so we created the "enemy" class and defined the spawn points, direction, speed and size of the asteroids with some math. For variability we used the random library to choose a size for newly created asteroids, one of four possible spawn points which are all slightly outside the observable window to avoid witnessing their sudden creation and a small window of different possible speeds. The movement of the asteroids was achieved by choosing the direction and continuously updating the asteroid's position via a formula. We also added the score and lives to the displayed information during the game and defined our game over condition with the lives reaching 0. As the next step we implemented the movement of the player and the bullets shot. Since the pygame library enables inputs to be translated into actions we just had to choose values for the rotation and movement speed as well as any input for shooting. Figuring out a convenient way to update the player's rotation was a challenging task as this is handled by using a sine and cosine function for the player. But eventually this could be figured out after one to two hours. The shots fired by the player were implemented as "bullet" objects. By testing we decided to use a fire rate of three shots per second and chose a speed for the bullets which update their position in a straight line. As the last and biggest step we created the numerous collision conditions between the different objects, like if a bullet collides with an asteroid the score increases, the bullet disappears and the asteroid (if not already of smallest size) splits into more asteroids or an asteroid colliding with the player resulting in the player losing a life and the asteroid vanishing. After this the game was running as we intended so we experimented with our chosen spawn rates, asteroid speeds and decided to include different difficulties. Since the hitboxes of our objects are all rectangles we edited the original ship to look more like a square to make it easier to see if you will get hit by an approaching asteroid. Lastly, we tested different settings of the game (e.g. with the asteroids reentering the field on the opposite side when exiting the displayed window instead of just disappearing), further adapted values for speed, size, spawn rates etc. and decided for what we liked best.

Screenshots