Platform class Team Project Assignment 4 Sebastian Pineda

- Platforms must appear at specific positions on the screen.
- Some platforms need to move horizontally
- All platforms must scroll upward with the world.
- Platforms must have breaking mechanics.
- Player charachter should be able to land on platforms.
- Platforms will integrate PNG images so Riley can plug in his art.

Platform Structure and movement Implementation

Needed elements for platforms include both position and size, velocity, movement, breakable behaviour and two discernable textures to differentiate between a broken platform and a normal one.

One of the initial problems I had was handling both scrolling and movement at the same time

The first try I did to add scroll and movement both independently, but this made it so that platforms drifted wrong since its movement origin wasnt updating with the scroll of the game.

The issue is that "startPos" was used for horizontal movement which wouldnt scroll with the world. To fix this I changed "startPos" to "moveOrigin" and updated every frame while scrolling. This fixed the problem and ensured platform movement even while scrolling.

Player Collision was a challenge since its' needed to specify separating actually landing on the platform from just hitting its side or bottom. Issues are that first attempts triggered collision even if the player wasnt falling.

So to fix this I added multiple collision checks like the player has to be falling downward, that there is no horizontal overlap between the player model and the platform. This made collision accurate and prevents accidental triggering.

Adding Texture Support

As a baseline all platforms were draw as rectangles just to see if they functioned as intended, but then it was needed that they could support PNGs to better do this and avoid issues I added scaling to the "Graphics.Draw()" so the textures would stretch to match their size and not be too small or large. So consistency was key through gameplay.