Assignment 4: Collaborative 2D Game Project - Process

Seth Rodger Patterson

TO DO:

- **Camera Panning/ Camera Follows Player**
- **Gravity Swap/Polarity**
- **Lives/Checkpoint management**
- **Enemy API**
    - **Spike Fall**
    - **Polarity switch**
    - **Boss**

Gravity Swap,

Gravity is just the constant movement in the Y-axis based on the player's position. But since gravity is swapping the feature will need to be static.

When the space bar is pressed Gravity is swapped, changing polarity and thus changing gravity for other enemies as well.

Camera Panning

It needs to be centered or at least following the player, but since the level is flat-ish the camera needs to be locked in the Y-axis.

Useful link for example raylib/examples/core/core_2d_camera_platformer.c at master · raysan5/raylib (github.com)

Lives,
To keep it in the theme health is represented by a battery decreasing in charge.

Originally wanted it to be timed based on the the checkpoints resetting the timer. I couldn't get multi-threading to work. Used simpler if statements to calculate. Once health becomes 0 player is sent to the last checkpoint and health is replenished.

Checkpoints have a collision box over them that adds to an external counter for where the player should respawn, Make sure that it only adds it once.

Enemy API,

Originally wanted a more class-based, with a combo of an array. I still think this would be a good idea if we had more time to work on levels. But for the scope of the project just complying with the functions is good enough.

Each Enemy type has a different AI,
The Falling Enemy with the spike on the bottom will only fall if the player is below it.
The Polarity Enemies will swap polarity with the player but are significantly slower to fall.
The Boss is just a giant death box if the player needs to race them to win the level.

Any time the player is touched by the enemy the player loses health.

Some Other Useful Examples:
nezvers/Raylib_RectangleCollision: Retro platformer physics with Rectangle collision (github.com)
chvia223/csharp-metroidvania: A game based on metroidvanias written in C#. (github.com)
Raylib-cs/Examples at master · ChrisDill/Raylib-cs (github.com)

Enemy Collision and Movement Saved Code:

```
 static void EnemyCollisionPlayer()
    {
    Vector2 GravityBasic = new Vector2(0, 10);
    bool TopEnemy = false;
    bool BottomEnemy = false;
    Rectangle PlayerRec = new Rectangle(PlayerPosition.X, PlayerPosition.Y, PlayerSize.X,
PlayerSize.Y);
    Rectangle EnemyRec = new Rectangle(EnemyPos.X, EnemyPos.Y, EnemySize.X,
EnemySize.Y);
    bool collision = Raylib.CheckCollisionRecs(PlayerRec, EnemyRec);

    if (collision == true)
    {
        // Top Collision Enemy
        if (PlayerPosition.Y - 5 < EnemyRec.Y)
        {
            GravityBasic = new Vector2(0, 0);
            TopEnemy = true;
            if (Raylib.IsKeyPressed(KeyboardKey.KEY_SPACE) && TopEnemy)
            {
                GravityBasic = new Vector2(0, 10);
                PlayerPosition = PlayerPosition + GravityBasic;
            }
        }
        // Bottom Collision Enemy
        if (PlayerPosition.Y + 5 > EnemyRec.Height)
        {
            GravityBasic = new Vector2(0, 0);
            BottomEnemy = true;
            if (Raylib.IsKeyPressed(KeyboardKey.KEY_SPACE) && BottomEnemy)
            {
                GravityBasic = new Vector2(0, 10);
```

```
            PlayerPosition = PlayerPosition - GravityBasic;
        }
    }
  }
}

static void EnemyDraw()
    {
        Vector2 GravityBasic = new Vector2(0, 15);
        //Draw enemey
        Raylib.DrawRectangle((int)EnemyPos.X, (int)EnemyPos.Y, (int)EnemySize.X,
(int)EnemySize.Y, Color.RED);

        // Enemy collisons
        bool TopWall = false;
        bool BottomWall = false;
        // Top Collisiom
        if (EnemyPos.Y - 5 < 0)
        {
            GravityBasic = new Vector2(0, 0);
            TopWall = true;
            if (Raylib.IsKeyPressed(KeyboardKey.KEY_SPACE) && TopWall)
            {
                GravityBasic = new Vector2(0, 10);
                EnemyPos = EnemyPos + GravityBasic;
            }
        }
        // Bottom Collision
        if (EnemyPos.Y + 15 > FloorBrickHeight)
        {
            GravityBasic = new Vector2(0, 0);
            BottomWall = true;
            if (Raylib.IsKeyPressed(KeyboardKey.KEY_SPACE) && BottomWall)
            {
                GravityBasic = new Vector2(0, 10);
                EnemyPos = EnemyPos - GravityBasic;
            }

        }

        if (Polarity == true)
        {
            PlayerPosition = PlayerPosition + GravityBasic;
        }
        else
        {
            PlayerPosition = PlayerPosition - GravityBasic;
        }
```