BROTHER

By Anika, Kyra, Tanisha

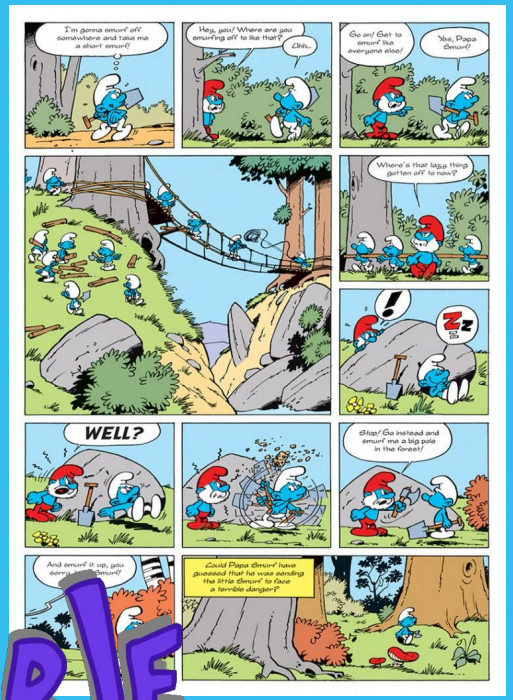
References & Inspiration

GAMES

- Classic Mario
- Prince of Persia (1991)
- Undertale (2015), Geometry Dash (2013)
- The Impossible Quiz (2007)

Honorable mentions:

- *Fez*
- *Monument Valley*
- *Nidhogg II*
- *Baba Is You*



VISUALS

- Peyo's Smurfs
- Adventures of TinTin
- Prince of Persia (1991)

MECHANICS

→ Movement with Arrow Keys (Stationary, Run, and Jump)
→ Click-and-Drag
→ One-sided platforms
→ Portal Transportation

DYNAMICS

→ Interactive environment in more than 1 way
→ Unexpected “abilities” and places (through portal)

AESTHETICS

→ Submissive
→ Challenge
→ Discovery

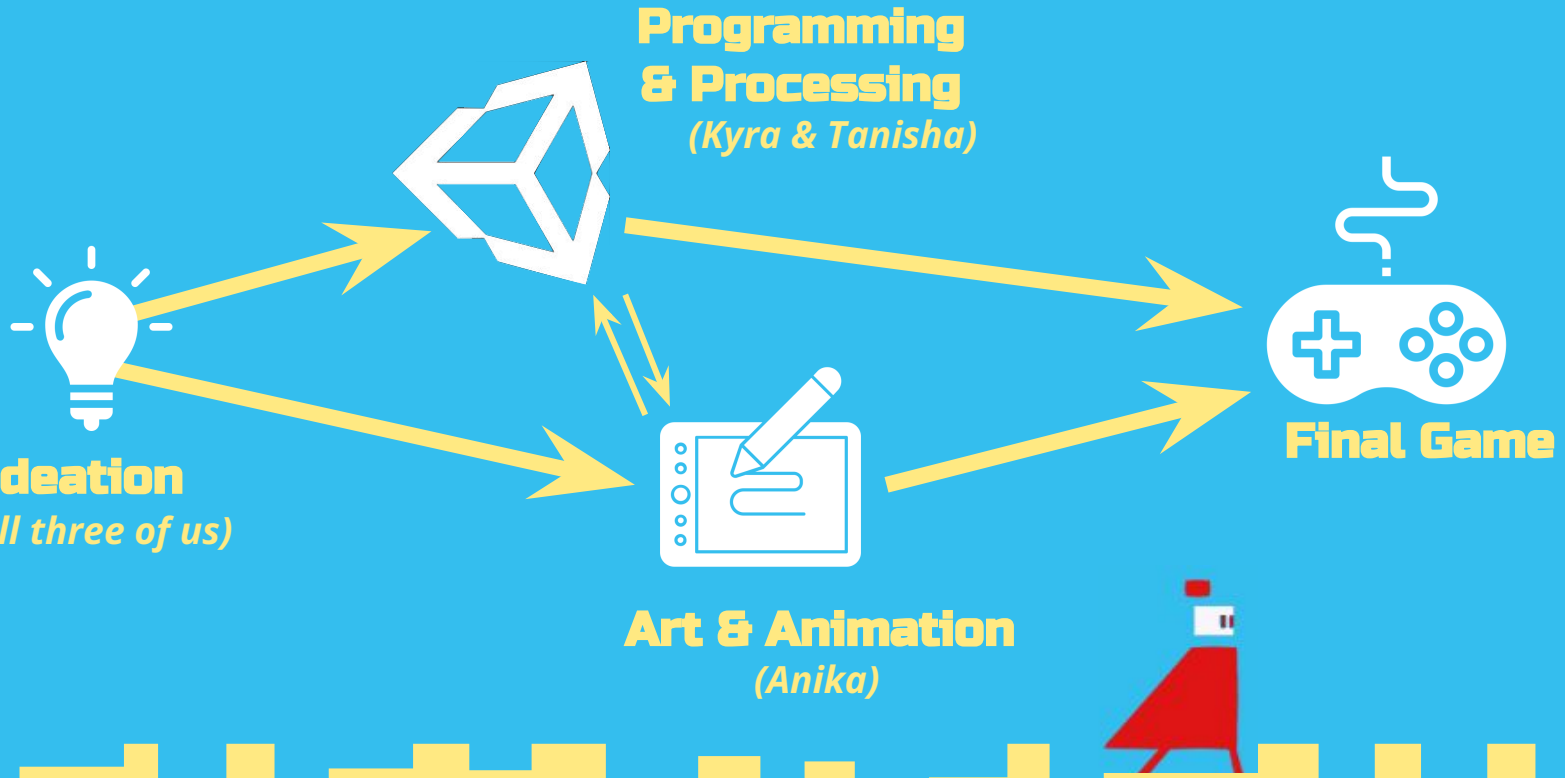
NEED

- Basic background ✓
- Drag and Click ✓
- Keyboard controls ✓
- Platform collisions ✓
- One level ✓
- Animation ✓

WANT

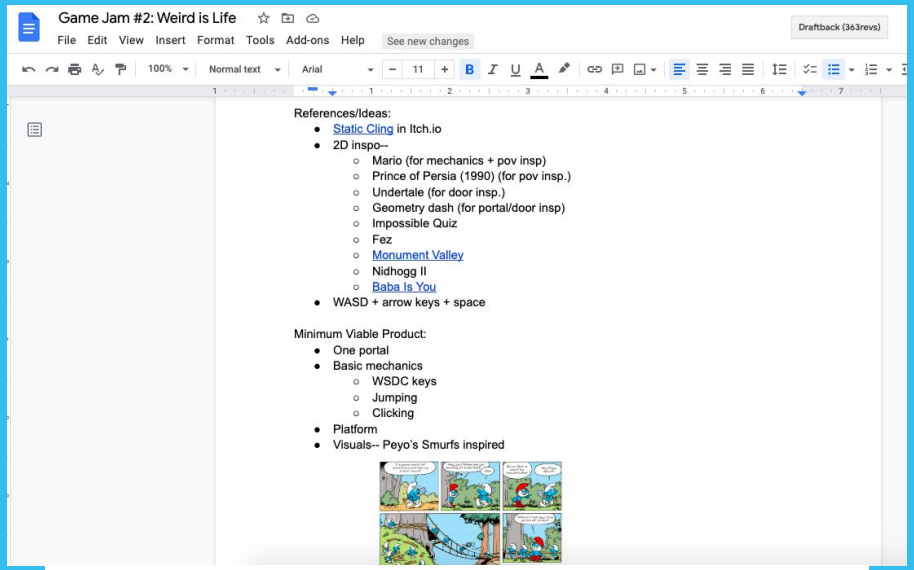
- Animation states (moving vs. still) ✓
- Camera movement ✗
- Multiple backgrounds ✓
- Portal collisions ✓
- One way collision platform ✓
- Music ✗
- Multiple levels ✗
- Two different jump mechanics (short & long) ✗
- Retry button ✗

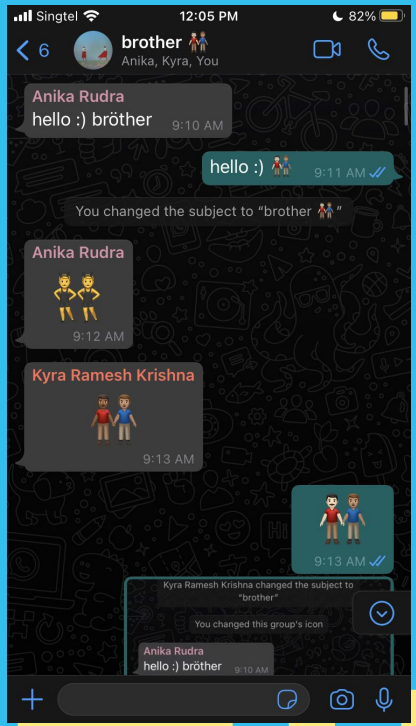
PROCESS


Programming
& Processing
(Kyra & Tanisha)
Ideation
(All three of us)
Art & Animation
(Anika)
Final Game

PLANNING & COMMUNICATION

DOCUMENTS AND WHATSAPP GROUP





- **Kyra - Programmer**
- **Tanisha - Programmer**
- **Anika - Art Director :) King. Ruler 你好唐妮莎**

References/Ideas:

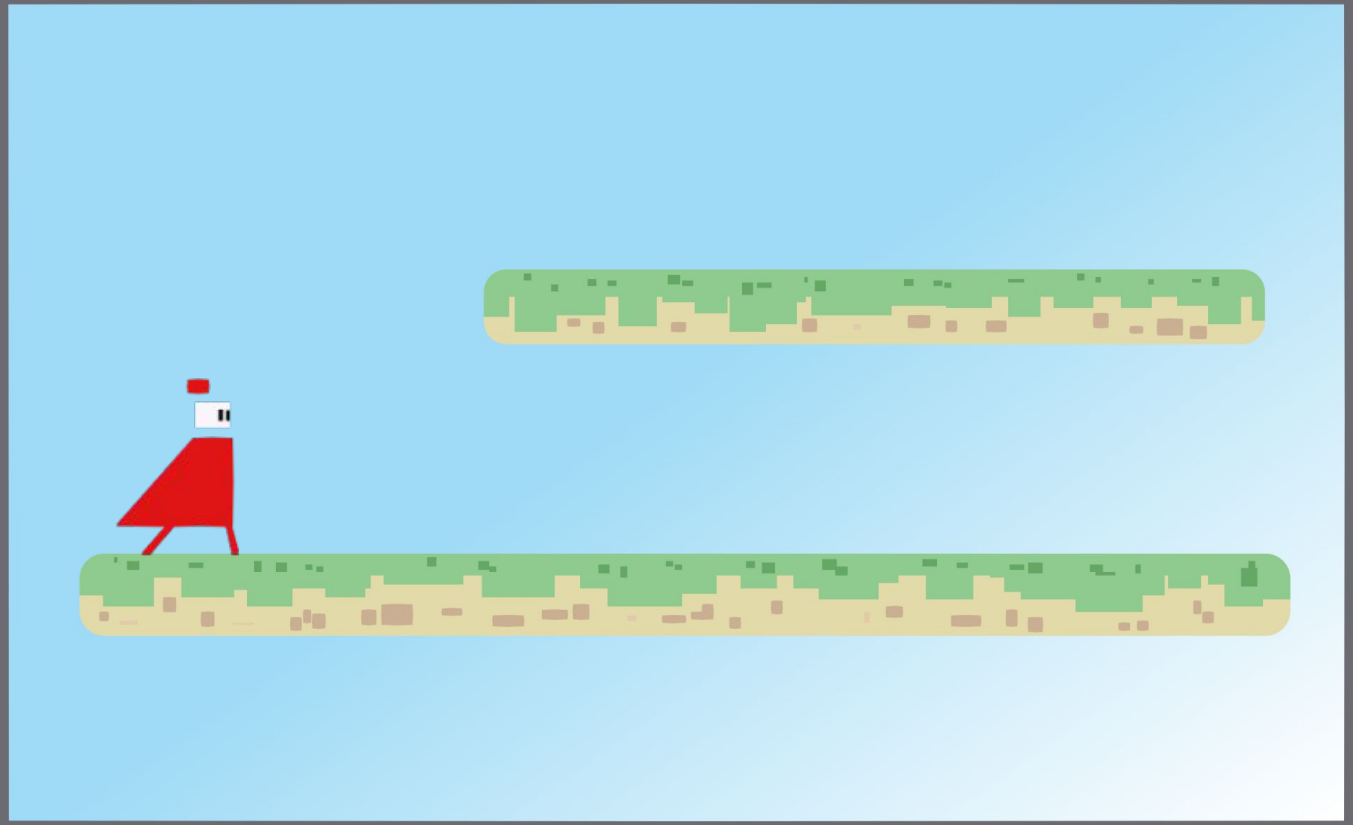
ART & ANIMATION

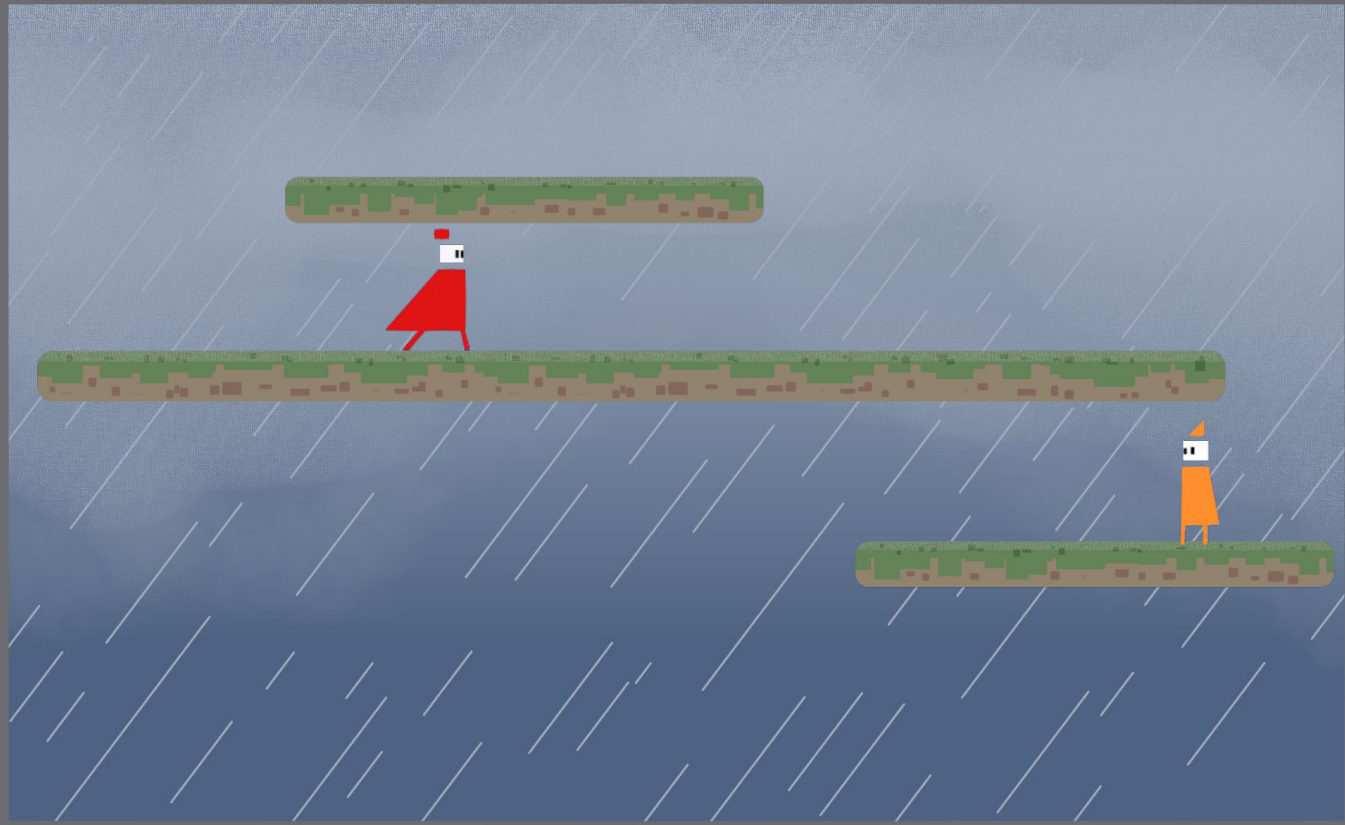
GOAL:

- Fake pixel art look + simple cel shading (if needed)
- Keep it simple so it works with the timeline and streamlines the process

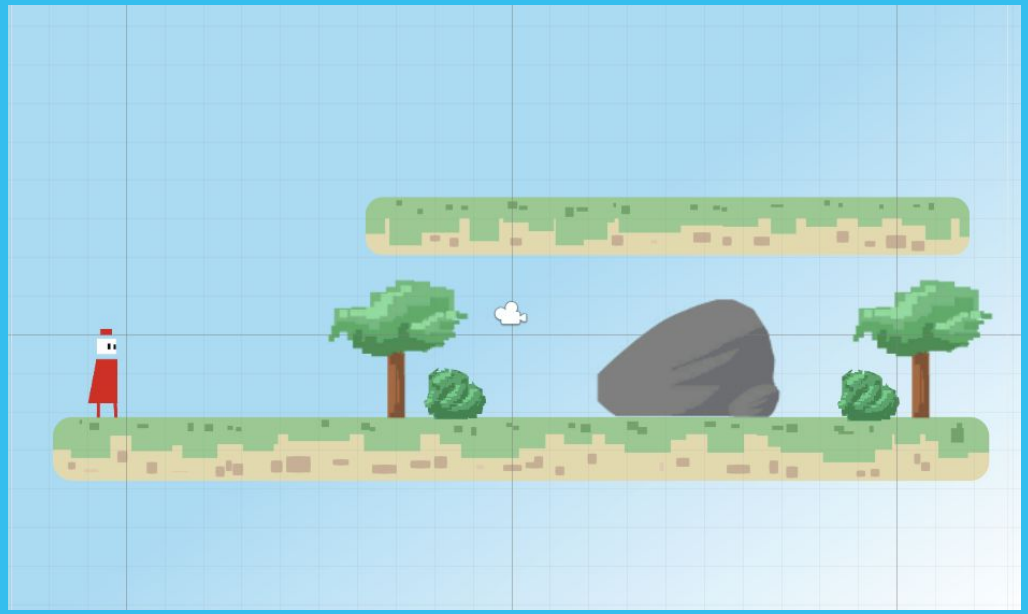
CHALLENGES:

- Making believable movement with as less frames as possible
- Keeping a consistent style
- Making sure what I made could actually be used

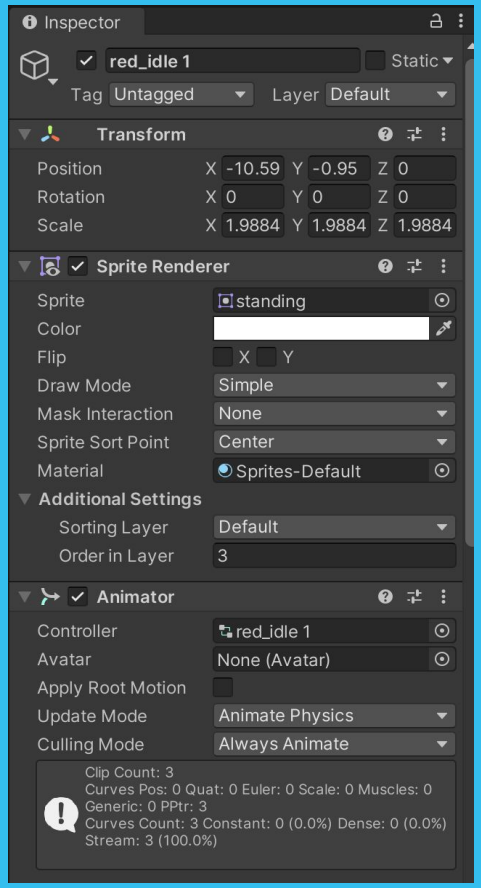




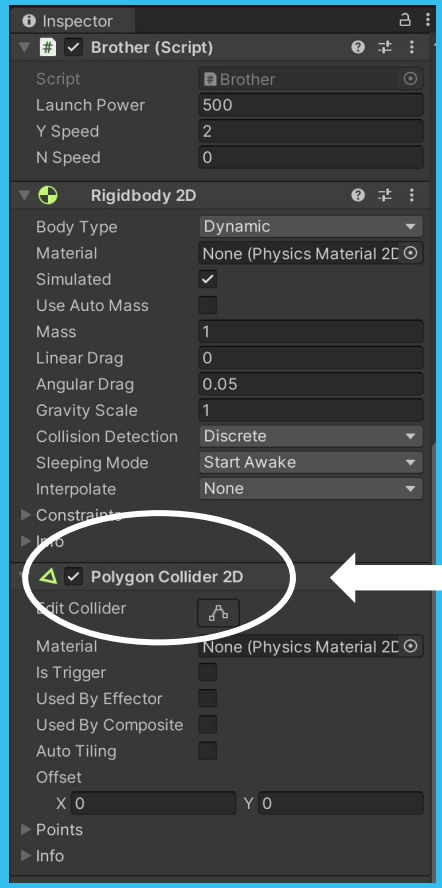
PROGRAMMING & PROCESSING



Step 1: setting up the background and the sprite

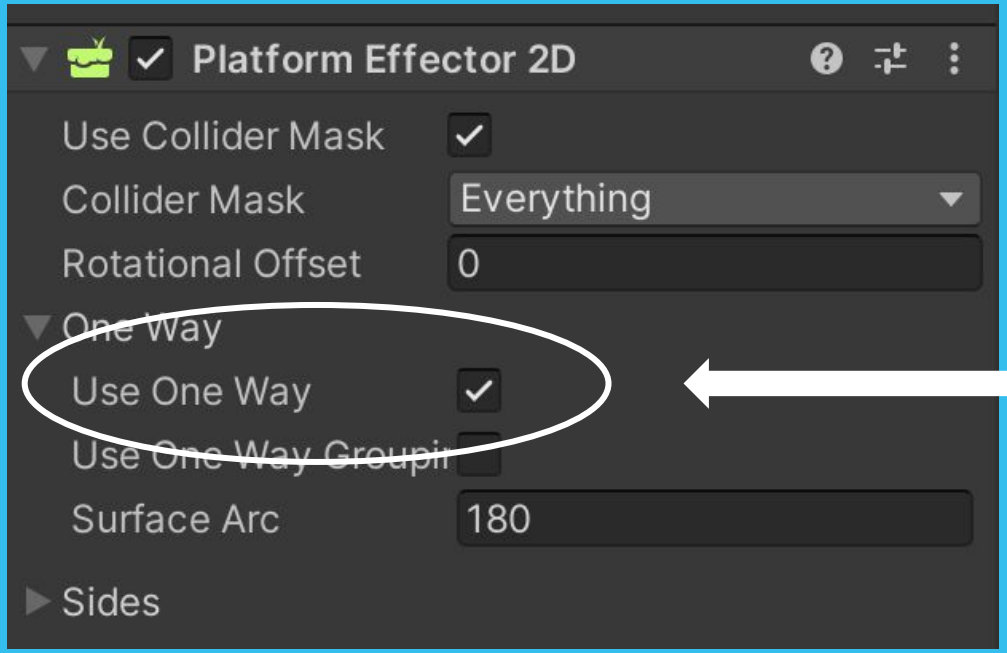


+



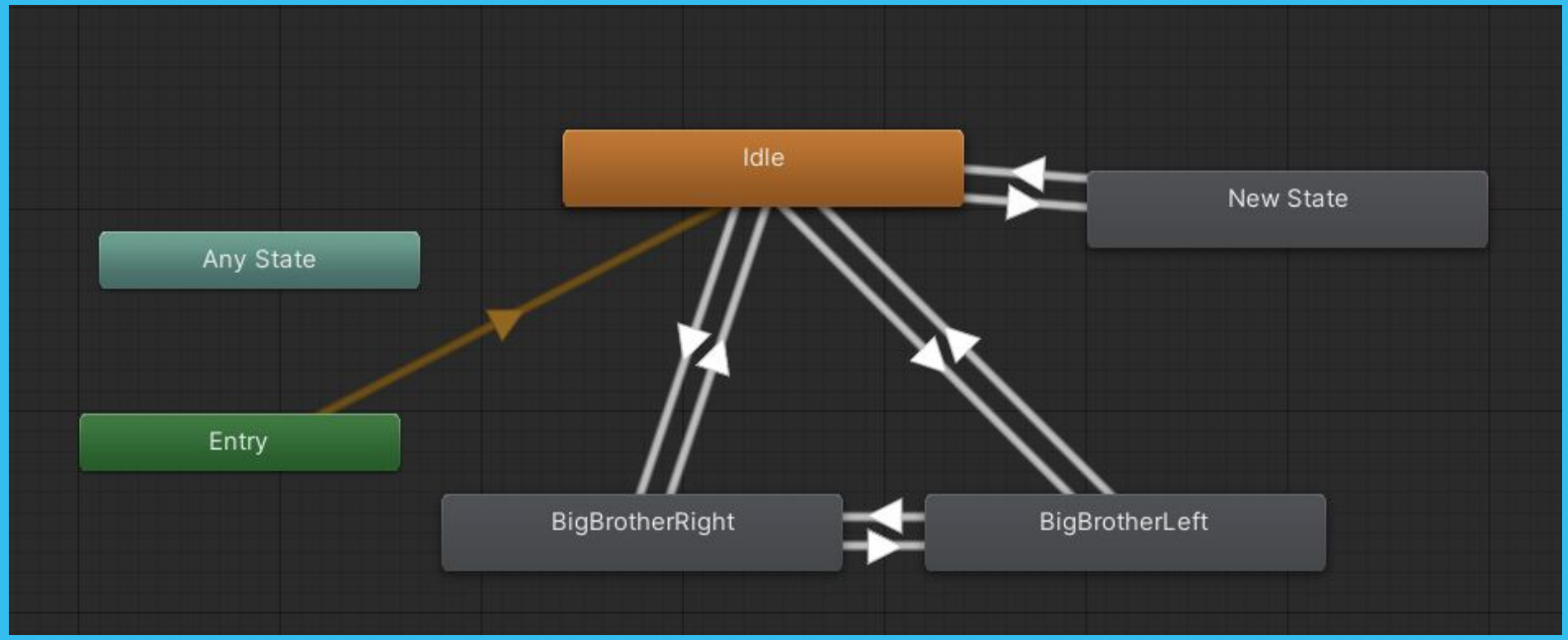
Step 2: adding physics components to our objects / sprites

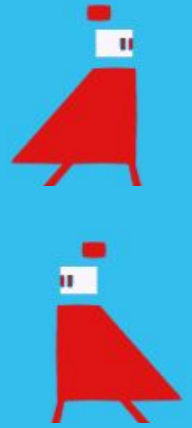
Use polygon collider for sprites



Step 3: Working on our platforms

Adding Platform Effector to have one-way collision





Setting up bool variables between states to differentiate stationary with motion + left/right direction.

Animation Boolean Code -- Reset Scene

```
            m_animator.SetBool("isLeft", false);
            m_animator.SetBool("isRunning",false);
        }
        if (Input.GetKeyDown(KeyCode.UpArrow))
        {
            m_animator.SetBool("isUp", true);
        }
        else if (Input.GetKeyUp(KeyCode.UpArrow))
        {
            m_animator.SetBool("isUp", false);
        }

        if (transform.position.y > 10.58)
            SceneManager.LoadScene(SceneManager.GetActiveScene().name);
        if (transform.position.x < -13.97)
            SceneManager.LoadScene(SceneManager.GetActiveScene().name);
        if (transform.position.x > 13.77)
            SceneManager.LoadScene(SceneManager.GetActiveScene().name);
        if (transform.position.y < -9)
            SceneManager.LoadScene(SceneManager.GetActiveScene().name);

        if(playAgain.retry)
        {
            SceneManager.LoadScene(SceneManager.GetActiveScene().name);
        }
        if(transform.position.y<-0.19 && transform.position.y>-0.81 && transform.position.x>5.3 && transform.position.x<6.06)
        {
            changeBackground = true;
            if(!isTransfered)
            {
                transform.position = new Vector3(-10.4f, 3.62f, 0f);
                isTransfered = true;
            }

        }
```

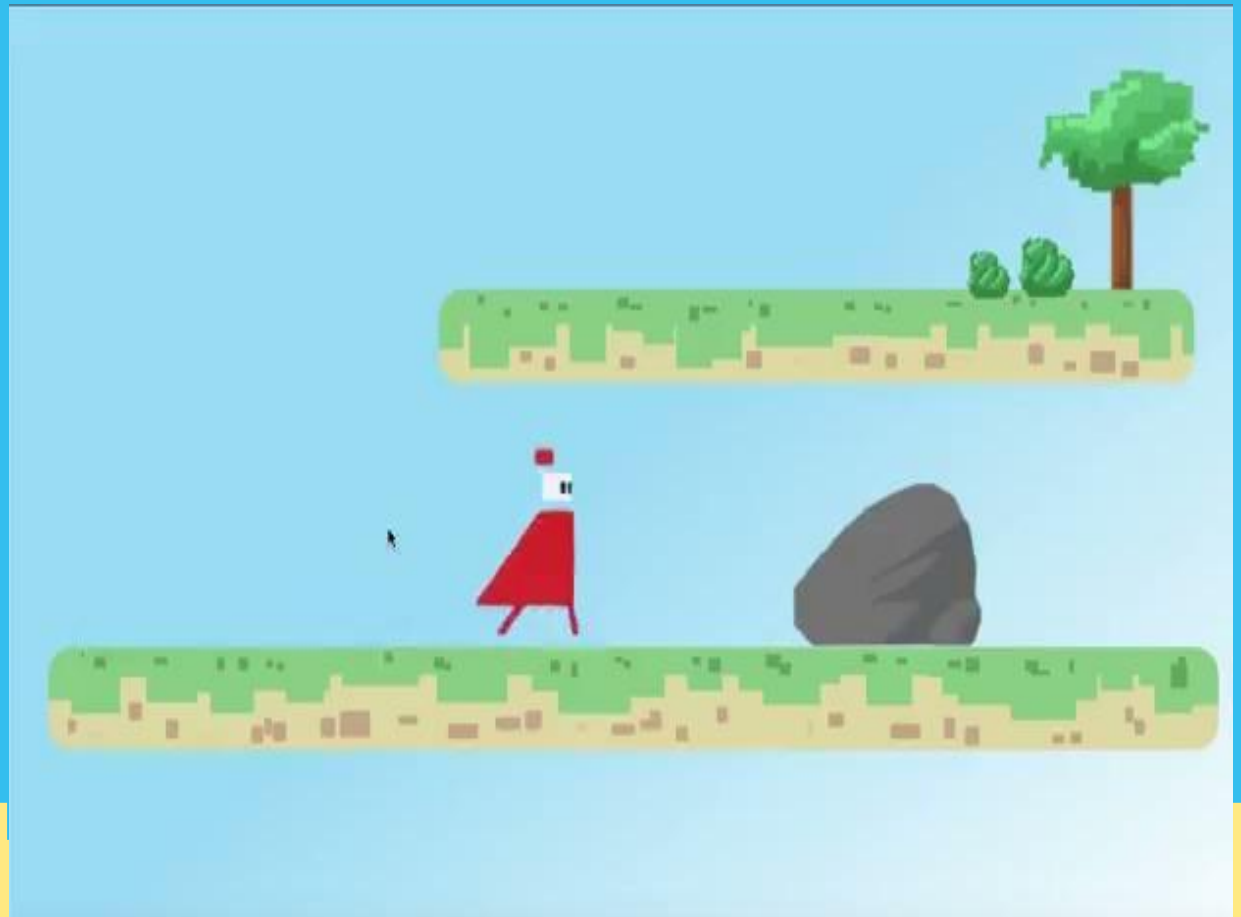
Wanted two different platforms, but not the collisions... Ended up simply moving the second platform out of the frame and out of reach of the sprite.

“click & drag” code + visibility

```
    private void OnMouseDown()
    {
        GetComponent<SpriteRenderer>().color = Color.black;

        Vector2 directionToInitialPosition = _initialPosition - transform.position;
        //GetComponent<Rigidbody2D>().AddForce(directionToInitialPosition * _launchPower);
        GetComponent<Rigidbody2D>().gravityScale = 1;
    }

    private void OnMouseUp()
    {
        GetComponent<SpriteRenderer>().color = Color.white;
    }

    private void OnMouseDrag()
    {
        Vector3 newPosition = Camera.main.ScreenToWorldPoint(Input.mousePosition);
        transform.position = new Vector3(newPosition.x, newPosition.y);
    }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class newPlatforms : MonoBehaviour
{
    public Renderer rend;
    public CompositeCollider2D coll;
    // Start is called before the first frame update
    void Start()
    {

        transform.position = new Vector3(-0.26f, -14.35f, 0f);


    }
    private void Awake()
    {
        rend = GetComponent<Renderer>();
        coll = GetComponent<CompositeCollider2D>();
        rend.enabled = false;
    }

    // Update is called once per frame
    void Update()
    {
        if (Brother.changeBackground)
        {
            transform.position = new Vector3(0f, 0f, 0f);

            coll.enabled = true;
            rend.enabled = true;


        }

    }
}
```

“Weird is Life” -- flailing movement



LEVELING UP -- ANIKA

In terms of Art --

- Lvl 4 → Lvl 5

In terms of Animation --

- Lvl 3 → Lvl 4

What I learned:

- Make the art in a way that makes the programmers' lives easier lol
- Simplicity is key
- Good to work with what you're familiar with sometimes, to make the creation process more efficient

LEVELING UP -- TANISHA & KYRA

In terms of Programming --

- Lvl 4 → Lvl 4

In terms of Unity --

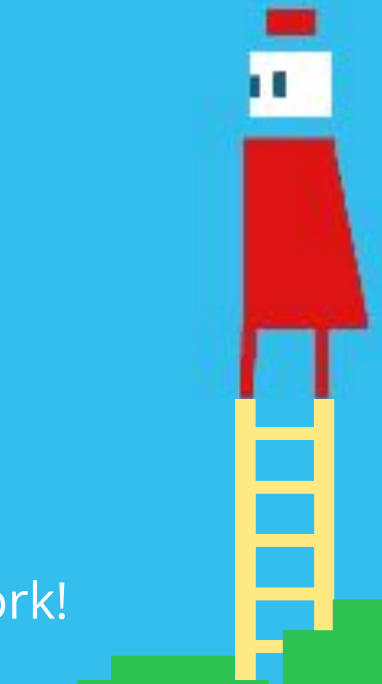
- Lvl 1 → Lvl 2.5 (I believe there is still a lot more to learn)

In terms of Animation --

- Lvl 1 → Lvl 2

Main Takeaways

- Art and Animation
  - Keep it simple!
  - Know exactly what you're aiming for
- Programming
  - Booleans are the best!
  - Don't be afraid to move things around!
- General
  - Communicate with your team members consistently; keep them in the loop on your work!



Credits

- Art
  - N/A
- Programming – helpful websites
  - Unity Manual and API
  - Beginner Tutorial

Thank You!