Gotta Moon

Group 21

Prompt: Gotta Move

Link to Slides to See Working Gifs:
https://docs.google.com/presentation/d/11Sfp272IT4i8KvldJN50tFopEJwjWMFvl9NQ5jxCAwc/edit?usp=sharing

Group 21

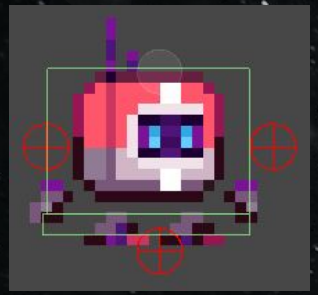
Members:

- Solomon Bell (Production, SFX, Video Editing/VO)
- Joe Frumenti (Ledge forgiveness, coyote time, better wall sliding, default dash forward, squash and stretch, turn acceleration)
- Casey Chen (Dash Controls, Backend polish)
- Malachi Maldonado (lerp, dash momentum)
- Jessica Huang (Video Scripting and Website Design)
- Benjamin Chavez (Art assets, parallax)
- Alex Xie (Camera Controls)
- Graydon Simons (Level Design)

Finished Product

https://jhuan221.itch.io/gotta-moon

Ledge Forgiveness

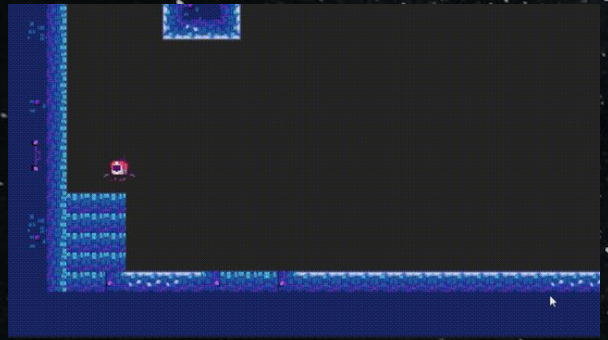


```
void Update()
{
    if (player.GetComponent<Movement>().moveset == 1)
        return;
    if (bottomTriggered && !topTriggered && !coll.onGround)
    {
        player.transform.Translate(new Vector2(0.01f * movement.side, 0.005f));
    }
}
```

```
private void OnTriggerEnter2D(Collider2D collision)
{
    cheater.topTriggered = true;
}

Unity Message | 0 references
private void OnTriggerExit2D(Collider2D collision)
{
    cheater.topTriggered = false;
}
```

```
private void OnTriggerEnter2D(Collider2D collision)
{
    cheater.bottomTriggered = true;
}

Unity Message | 0 references
private void OnTriggerExit2D(Collider2D collision)
{
    cheater.bottomTriggered = false;
}
```

Coyote Time

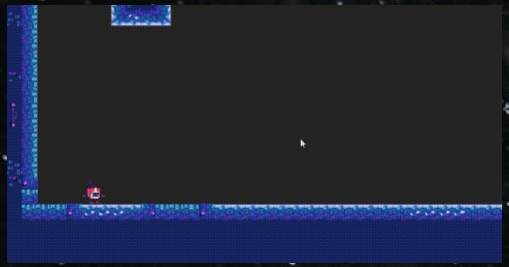


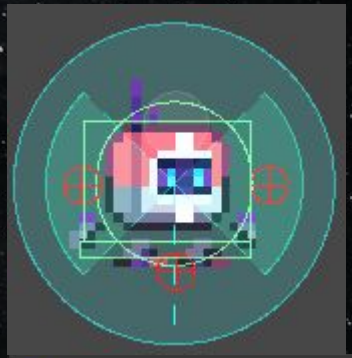
```
void checkCoyote()
{
    if(moveset == 1)
    {
        onGround = Physics2D.OverlapCircle((Vector2)transform.position + bottomOffset, collisionRadius, groundLayer);
        return;
    }

    else if (Physics2D.OverlapCircle((Vector2)transform.position + bottomOffset, collisionRadius, groundLayer))
    {
        onGround = true;

    }
    else
        StartCoroutine(coyote(coyoteTime));
}

1 reference
IEnumerator coyote(float x)
{
    yield return new WaitForSeconds(x);
    onGround = false;
}
```

Turn Acceleration

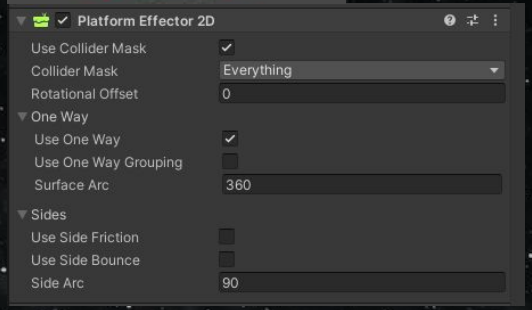
```
private void updateSpeed()
{

    currentSpeed += accel * Time.deltaTime * Input.GetAxisRaw("Horizontal");
    if(Input.GetAxisRaw("Horizontal") != Math.Sign(currentSpeed))
        currentSpeed += accel * Time.deltaTime * Input.GetAxisRaw("Horizontal");

    //if no input, approach zero
    if (Input.GetAxisRaw("Horizontal") == 0)
    {
        int speedSign = Math.Sign(currentSpeed);
        currentSpeed -= speedSign * accel * Time.deltaTime;
        if (speedSign != Math.Sign(currentSpeed))
            currentSpeed = 0;

    }

    currentSpeed = Mathf.Clamp(currentSpeed, -speed, speed);
    if (moveset > 1)
        anim.playerMoveSpeed = currentSpeed;
    else
        anim.playerMoveSpeed = Input.GetAxis("Horizontal");
}
```

Upwards Wall Sliding

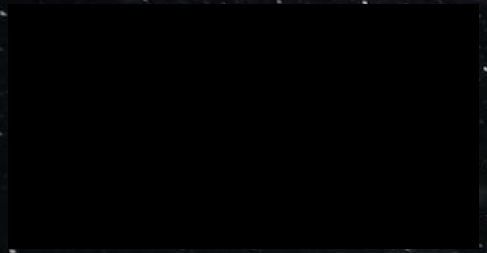




```
private void WallSlide()
{
    if (Input.GetKey("space") && rb.velocity.y > 0
        return;
```



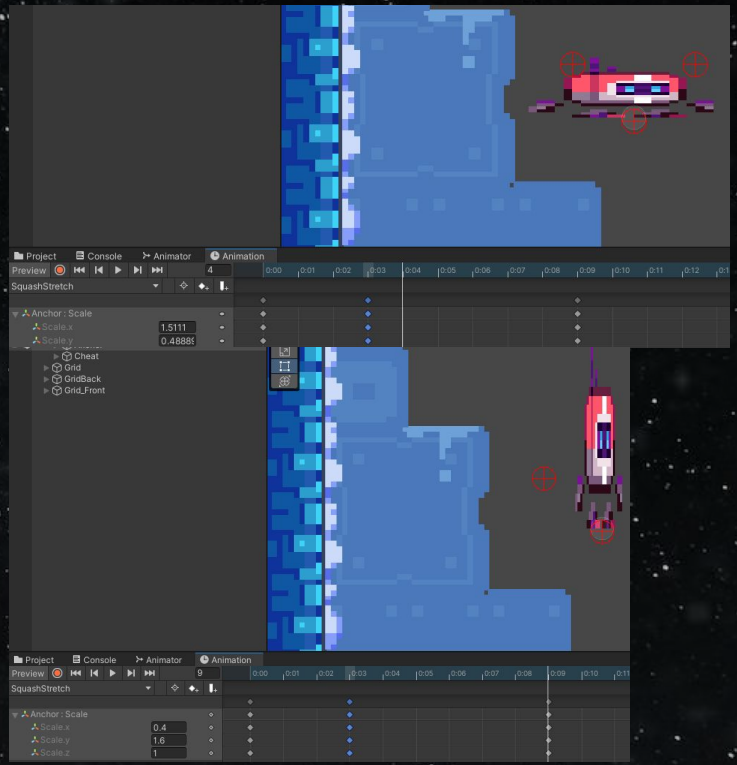
Dash Momentum



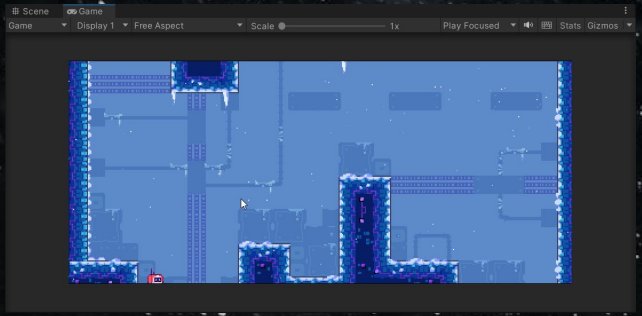
```
IEnumerator DashWait()
{
    FindObjectOfType<GhostTrail>().ShowGhost();
    StartCoroutine(GroundDash());
    DOVirtual.Float(dashMomentum, 0, .8f, RigidbodyDrag);
```

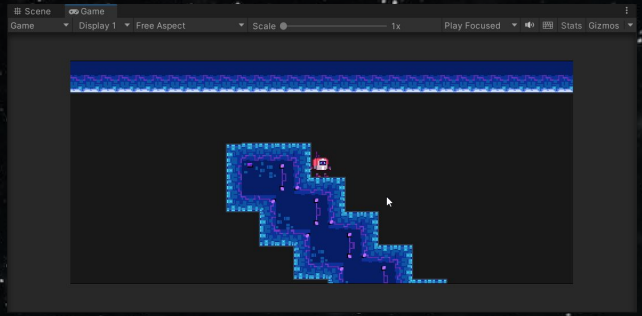
```
private void updateMoveset(uint newMoveset)
{
    moveset = newMoveset;
    coll.moveset = newMoveset;

    if (moveset > 2)
    {
        playerGrav = 2;
        dashMomentum = 4;
    }
    else
    {
        playerGrav = 3;
        dashMomentum = 14;
    }

}
```

Squash and Stretch



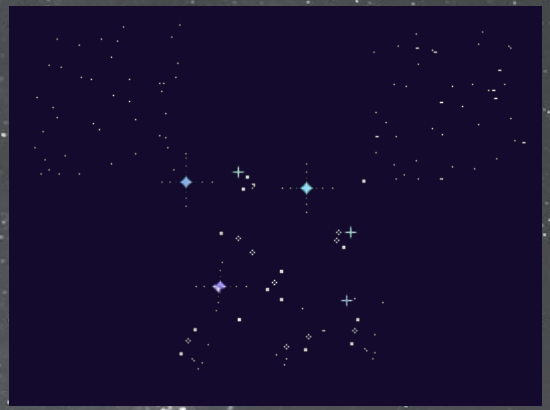
Camera Movement

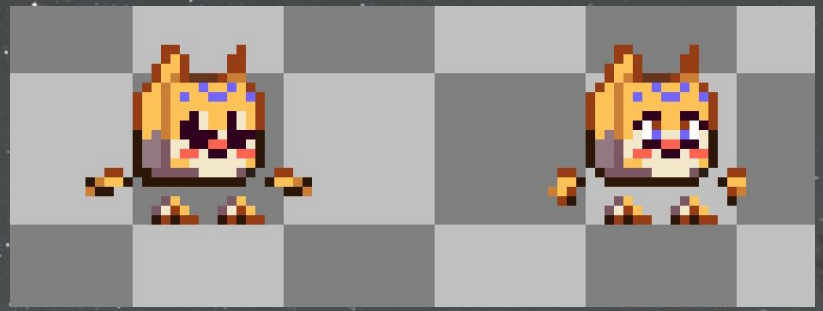





```
if(transform.position != target.position){
    Vector3 targetPosition = new Vector3(target.position.x,
                                         target.position.y,
                                         transform.position.z);

    targetPosition.x = Mathf.Clamp(targetPosition.x, minPosition.x, maxPosition.x);
    targetPosition.y = Mathf.Clamp(targetPosition.y, minPosition.y, maxPosition.y);


    transform.position = Vector3.Lerp(transform.position, targetPosition, smoothing);
}
```

Art Assets





Video Essay

Thank You!