



TEAM ESC

201919043 김민제
201919069 정준호
201919071 채대승

개요

옛 레트로 감성 게임을 하고싶은 여러분들을 위해
유니티를 이용한
귀여운 캐릭터와 그에 맞는 배경설정, 그리고
재미있게 플레이 할 수 있도록 적당한 난이도있는
게임을 준비했습니다.



목차

1. 이미지 소개
2. 타일 맵
3. 캐릭터 조작
4. 적 AI
5. 피격체
6. UI
7. Q&A

개발환경



Visual Studio Code



1. 이미지 소개



귀여운 플레이어 캐릭터



플레이어를 방해하는 몬스터

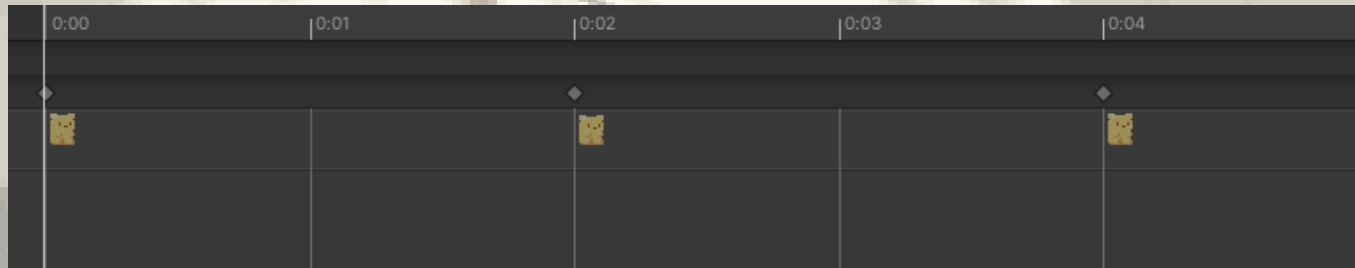


배경과 맞는 타일 맵

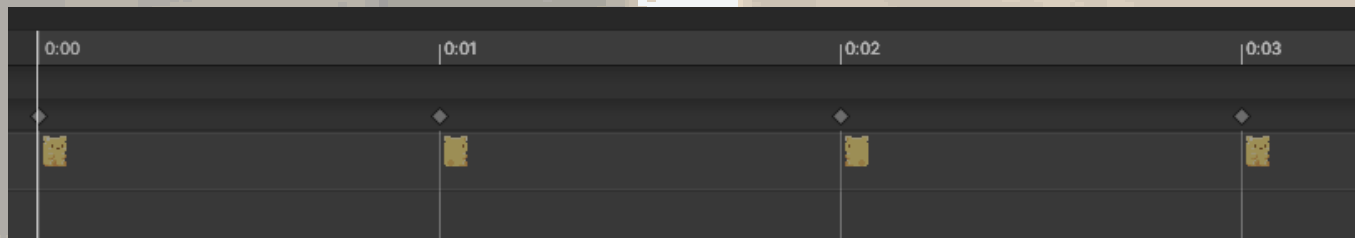


플레이어를 방해하는 오브젝트

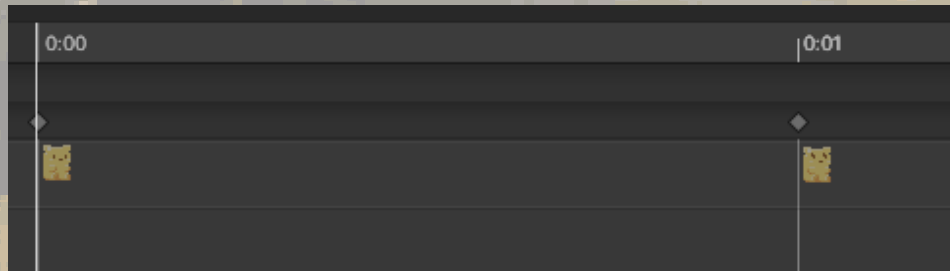
1-2. 애니메이션



기본

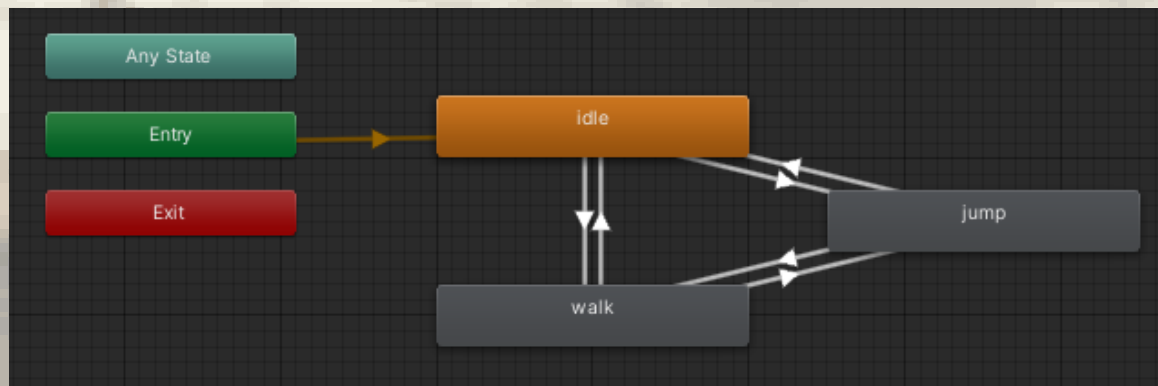


이동



점프

1-2. 애니메이션



```
if(Mathf.Abs (rigid.velocity.x) < 0.3)
    anim.SetBool("isWalking", false);
else
    anim.SetBool("isWalking", true);
```

SetBool을 이용하여
애니메이션 전환을
자연스럽게

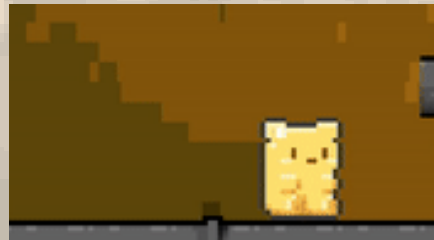
```
if(Input.GetButtonDown("Jump") && !anim.GetBool("isJumping")) {
    rigid.AddForce(Vector2.up * jumpPower, ForceMode2D.Impulse);
    anim.SetBool("isJumping", true);
    anim.SetBool("isJumping", false);
}
```

점프에도 마찬가지로
SetBool 사용

1-2. 애니메이션



기본

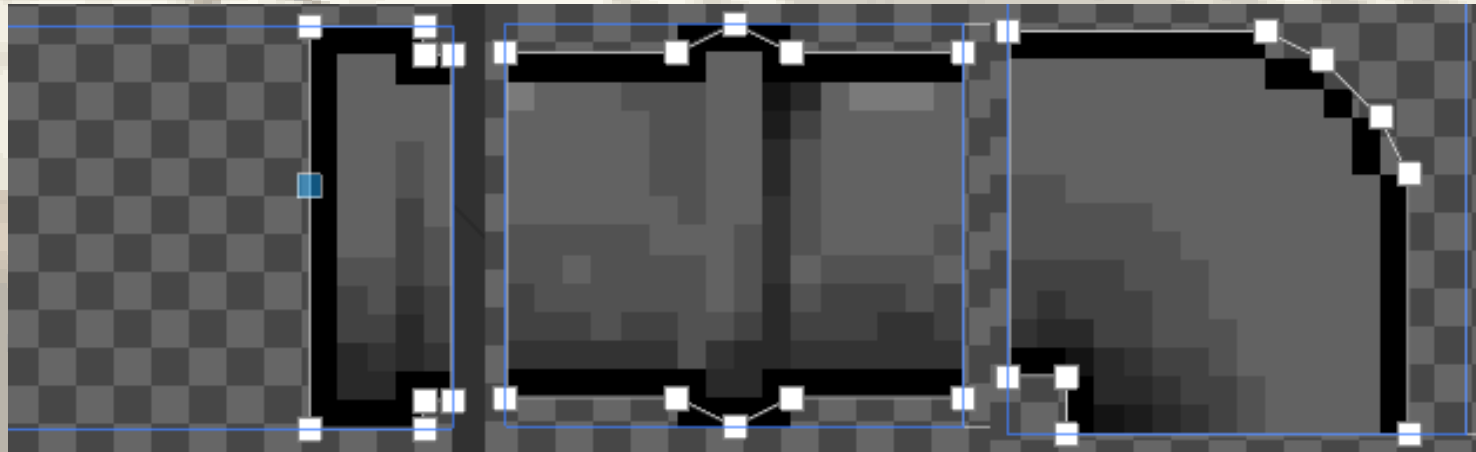


캐릭터 이동



캐릭터 점프

2. 타일 맵



타일 당 물리 범위를 적용 시켜 캐릭터의 딱딱한 움직임에도 이동을 할 때 역동적인 느낌을 살렸습니다.

2. 타일 맵



또한 캐릭터가 이동을 잘못할 시
게임 진행 방해 요소가 될 수도 있습니다.

2. 타일 맵



3. 캐릭터 조작



수평 조작 코드

```
void FixedUpdate()
{
    float h = Input.GetAxisRaw("Horizontal");
    rigid.AddForce(Vector2.right * h, ForceMode2D.Impulse);

    if (rigid.velocity.x > maxSpeed)
        rigid.velocity = new Vector2(maxSpeed, rigid.velocity.y);
    else if (rigid.velocity.x < maxSpeed*(-1))
        rigid.velocity = new Vector2(maxSpeed*(-1), rigid.velocity.y);
}
```

기본 버튼: 방향키

캐릭터의 속도를 제한하는 코드

Max Speed

4

```
if(Input.GetButton("Horizontal"))
    spriteRenderer.flipX = Input.GetAxisRaw("Horizontal") == -1;
```

3. 캐릭터 조작



점프 코드

```
void Update()  
{  
    if(Input.GetButtonDown("Jump")) {  
        rigid.AddForce(Vector2.up * jumpPower, ForceMode2D.Impulse);  
    }  
}
```

Jump Power

22

기본 버튼: 스페이스바



4-1. 몬스터 AI

```
void Think()  
{  
    nextMove = Random.Range(-1, 2);  
  
    anim.SetInteger("WalkSpeed", nextMove);  
  
    if(nextMove != 0)  
        spriteRenderer.flipX = nextMove == 1;  
  
    float nextThinkTime = Random.Range(2f, 5f);  
    Invoke("Think", nextThinkTime);  
}
```

`Random.Range(-1, 2)` : -1 ~ 2 사이의 수를 생성하여
몬스터의 진행방향을 설정

`Invoke("Think", nextThinkTime)` : nextThinkTime의 시간이 흐른뒤
Think함수를 실행해서 몬스터의 진행방향설정을 다시 설정함



4-1. 몬스터 AI

`Random.Range(-1, 2)` : -1 ~ 2 사이의 수를 생성하여
몬스터의 진행방향을 설정

`Invoke("Think", nextThinkTime)` : nextThinkTime의 시간이 흐른뒤
Think함수를 실행해서 몬스터의 진행방향설정을 다시 설정함

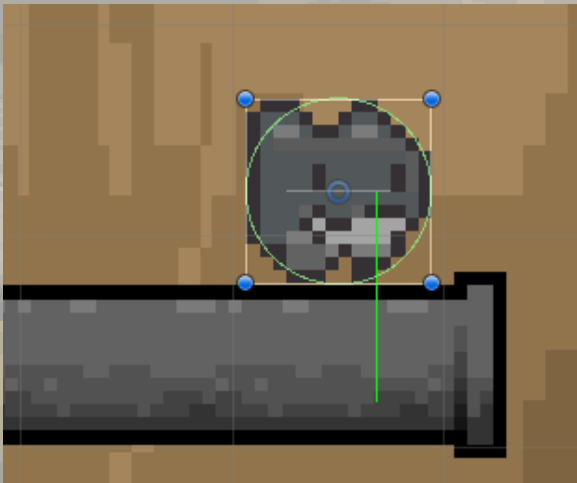




4-2. 몬스터 AI

```
void FixedUpdate()
{
    rigid.velocity = new Vector2(nextMove, rigid.velocity.y);

    Vector2 frontVec = new Vector2(rigid.position.x + nextMove*0.2f, rigid.position.y);
    Debug.DrawRay(frontVec, Vector3.down, new Color(0,1,0));
    RaycastHit2D rayHit = Physics2D.Raycast(frontVec, Vector3.down, 1, LayerMask.GetMask("Platform"));
    if(rayHit.collider == null) {
        Turn();
    }
}
```



`DrawRay` 를 사용해 몬스터 앞쪽에 선을 그려놓고

`rayHit` 로 몬스터 앞쪽에 "Platform"의 유무를 판단

`If(rayHit.collider == null)` : 앞쪽에 "Platform"이 없으면 Turn함수를 호출



4-3. 몬스터 AI

```
void Turn()  
{  
    nextMove = nextMove * -1;  
    spriteRenderer.flipX = nextMove == 1;  
  
    CancelInvoke();  
    Invoke("Think", 2);  
}
```



Turn 함수는 몬스터의 움직임을 진행반대방향으로 전환함

5-1. 피격체

```
void OnCollisionEnter2D(Collision2D collision)
{
    if(collision.gameObject.tag == "Enemy") {
        OnDamaged(collision.transform.position);
    }
}

void OnDamaged(Vector2 targetPos)
{
    GameManager.HealthDown();

    gameObject.layer = 9;

    spriteRenderer.color = new Color(1,1,1,0.4f);

    int dirx = transform.position.x - targetPos.x > 0 ? 1 : -1;
    rigid.AddForce(new Vector2(dirx, 1)*7, ForceMode2D.Impulse);

    Invoke("OffDamaged", 1);
    PlaySound("DAMAGED");
}
```

플레이어가 적에게 닿으면 OnDamaged 함수를 호출

`SpriteRenderer.color = new Color` 를 사용해
플레이어를 반투명한 상태로 바꿈

`AddForce` 를 사용해
플레이어를 적으로부터 튕겨나가게 만들

5-1. 피격체



```
int dirc = transform.position.x-targetPos.x > 0 ? 1 : -1;
rigid.AddForce(new Vector2(dirc, 1)*7,ForceMode2D.Impulse);

Invoke("OffDamaged", 1);
PlaySound("DAMAGED");
}
```

플레이어가 적에게 닿으면 OnDamaged함수를 호출

`SpriteRenderer.color = new Color` 를 사용해
플레이어를 반투명한 상태로 바꿈

`AddForce` 를 사용해
플레이어를 적으로부터 튕겨나가게 만듦

6-1. 체력

Health

3

```
void OnDamaged(Vector2 targetPos)
{
    gameManager.HealthDown();

    gameObject.layer = 9;

    spriteRenderer.color = new Color(1,1,1,0.4f);

    int dirx = transform.position.x-targetPos.x > 0 ? 1 : -1;
    rigid.AddForce(new Vector2(dirx, 1)*7,ForceMode2D.Impulse);

    Invoke("OffDamaged", 1);
    PlaySound("DAMAGED");
}

public void HealthDown()
{
    if(health > 1){
        health--;
        UIhealth[health].color = new Color(1, 1, 1, 0.2f);
    }
    else {
        UIhealth[0].color = new Color(1, 1, 1, 0.2f);
        player.OnDie();

        Debug.Log("죽었습니다.");
        UIRestartBtn.SetActive(true);
    }
}
}
```

```
public void OnDie()
{
    spriteRenderer.color = new Color(1, 1, 1, 0.4f);

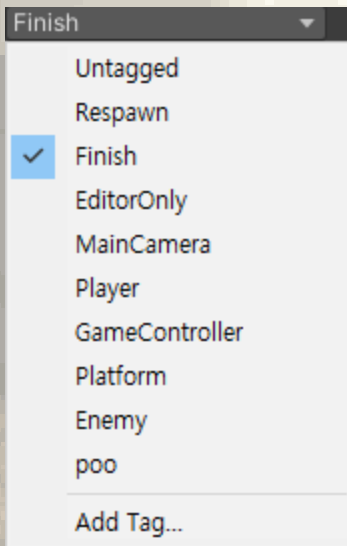
    spriteRenderer.flipY = true;

    capsuleCollider.enabled = false;

    rigid.AddForce(Vector2.up * 5, ForceMode2D.Impulse);
    PlaySound("DIE");
}
```



6-2. 클리어



```
void OnTriggerEnter2D(Collider2D collision)
{
    if(collision.gameObject.tag == "Finish"){
        gameManager.FinishStage();
        PlaySound("FINISH");
    }
}

public void FinishStage()
{
    Time.timeScale = 0;
    Debug.Log("게임 클리어");

    Text btnText = UIRestartBtn.GetComponentInChildren<Text>();
    btnText.text = "Game Clear";
    UIRestartBtn.SetActive(true);
}
```



게임 시연 영상



Q & A



다운로드 링크

<https://drive.google.com/file/d/1wqauvBHsGJ5OS26xBpiYRiJL5m7BqzQe/view?usp=sharing>

경청해 주셔서 감사합니다

