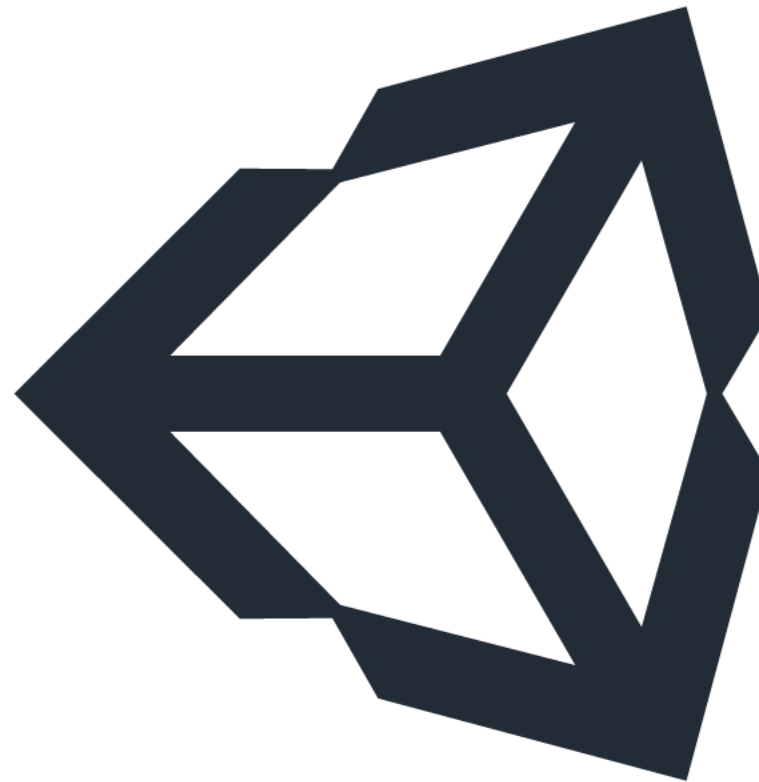




# UNITY 3D SNAKE GAME

CHO WOO HYUN  
major; software application



계획

01

진행과정

02

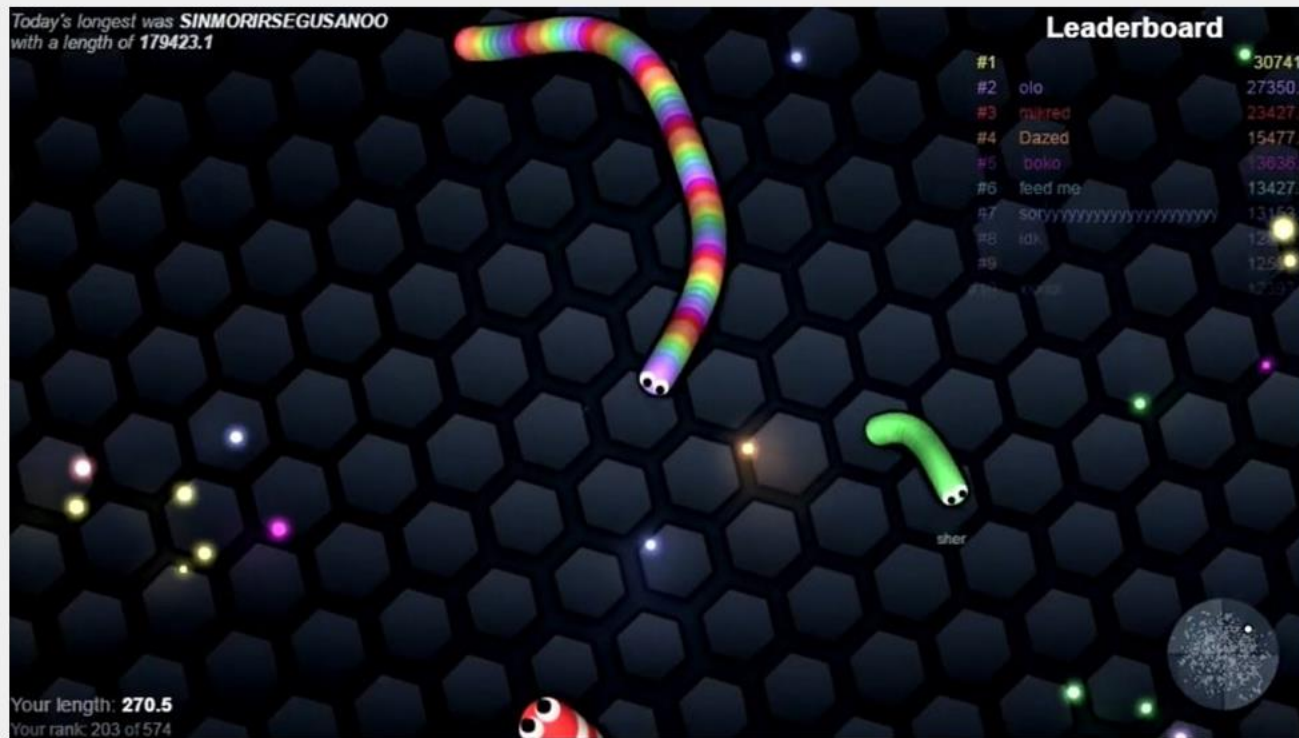
한계점

03

결과

04

## 계획



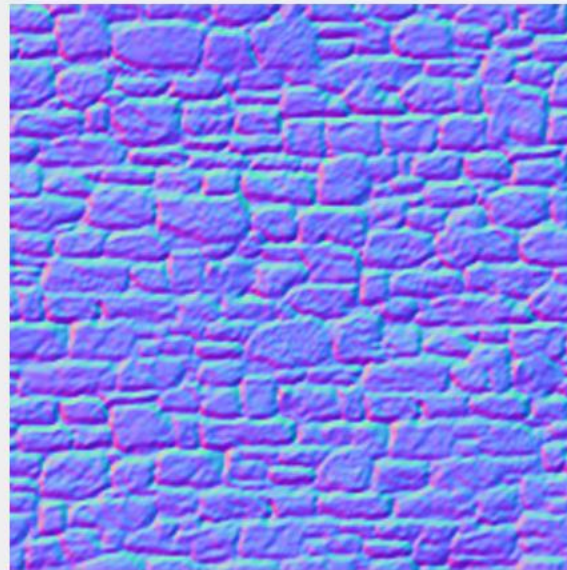
- > 한 때 선풍적인 인기를 끌었던  
지렁이 게임 모티브
- > 코인을 먹으면 꼬리가 점점 길어지고  
자기 꼬리에 머리가 닿거나  
벽에 닿으면 게임오버

## ■ 진행과정

— 맵 제작에서 Normal Map을 사용하여 입체감 구현



기본 텍스처



Normal Map Filter

## ■ 진행과정

### → 머리의 움직임

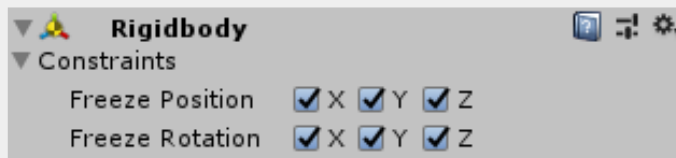
GetAxis("Horizontal")

A, D, ←. →

### → 꼬리의 움직임

선형 보간 알고리즘

Vector3.Lerp( )



```
float speedMove = 3f; //이동 속도  
float speedRot = 120f; //회전 속도
```

⋮

```
void MoveHead( ) { //머리이동  
    //이동  
    float amount = speedMove * Time.deltaTime;  
    transform.Translate(Vector3.forward * amount);  
  
    //회전  
    amount = Input.GetAxis("Horizontal") * speedRot;  
    transform.Rotate(Vector3.up * amount * Time.deltaTime);  
}  
  
void MoveTail ( ) { //꼬리이동  
    Transform target = transform;  
  
    foreach (Transform tail in tails) {  
        Vector3 pos = target.position;  
        Quaternion rot = target.rotation;  
  
        tail.position = Vector3.Lerp(tail.position, pos, 4 * Time.deltaTime);  
        tail.rotation = Quaternion.Lerp(tail.rotation, rot, 4 * Time.deltaTime);  
  
        target = tail;  
    }  
}
```

## ■ 진행과정

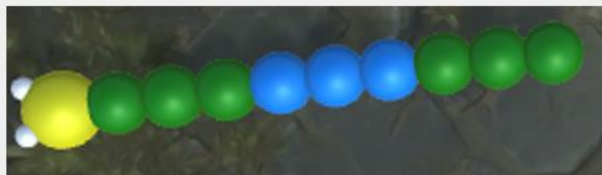
### — 꼬리 추가

Prefab 사용

List 구조 사용

첫번째 꼬리 Tag 제거  
(머리와 충돌 방지)

세 마디 단위로 색 변경



```
List<Transform> tails = new List<Transform>();
```

```
⋮
```

```
//꼬리 추가
void AddTail () {
    GameObject tail = Instantiate(Resources.Load("Tail")) as GameObject;
    Vector3 pos = transform.position;

    //처리
    int cnt = tails.Count;
    if (cnt == 0) {
        tail.tag = "Untagged";
    } else {
        pos = tails[cnt - 1].position;
    }
    tail.transform.position = pos;

    //색 변경
    Color[] colors = { new Color(0, 0.5f, 0, 1), new Color(0, 0.5f, 1, 1) };
    int n = (cnt / 3 % 2);
    tail.GetComponent<Renderer>().material.color = colors[n];
    tails.Add(tail.transform);
}
```



## ■ 진행과정

### → 충돌 처리

```
//동전처리
void MoveCoin() {
    coinCnt++;
    float x = Random.Range(-9f, 9f);
    float z = Random.Range(-4f, 4f);
    coin.position = new Vector3(x, 0, z);
}

//케이스별 충돌
void OnCollisionEnter(Collision other)
{
    switch (other.transform.tag)
    {
        case "Coin":
            MoveCoin(); //Coin이동
            AddTail(); //꼬리 추가
            break;
        case "Wall":
        case "Tail":
            isDead = true;
            PanelOver.SetActive(isDead);
            break;
    }
}
```

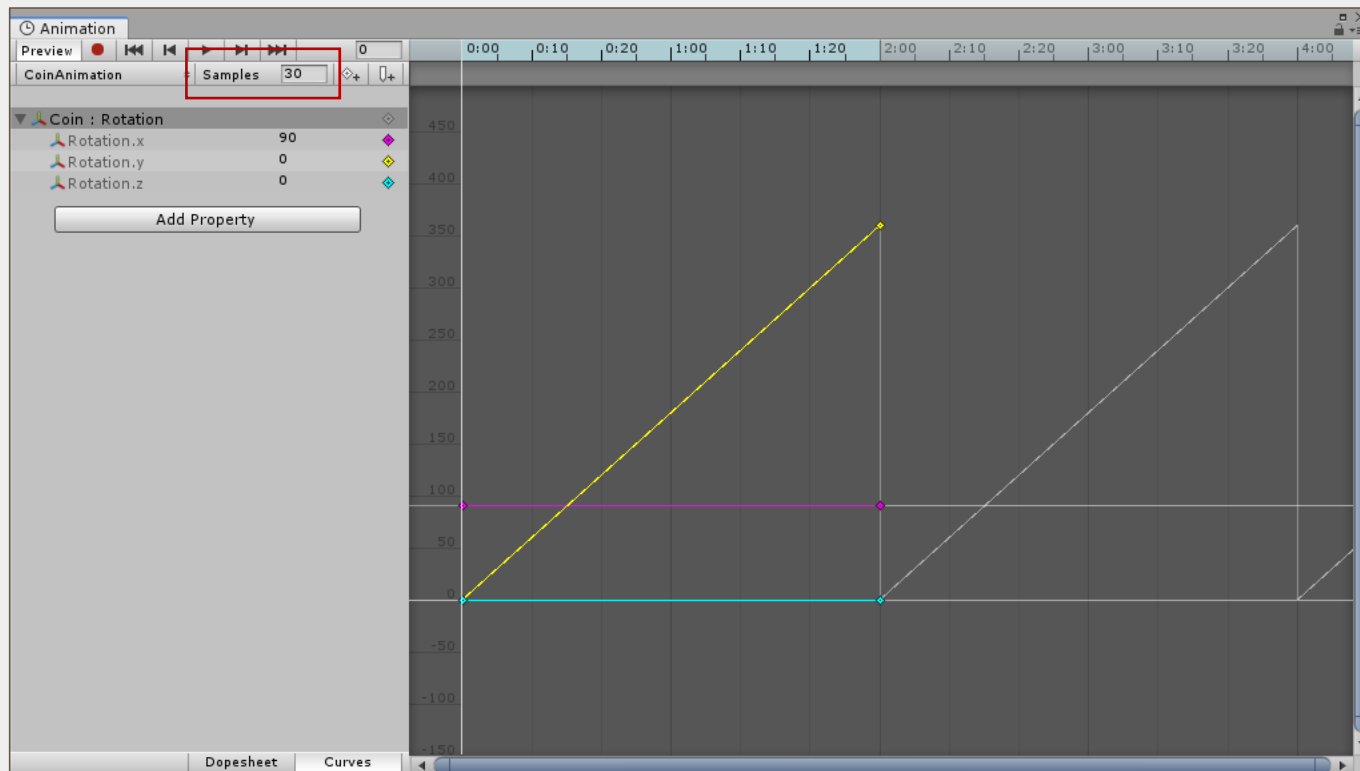
→ 동전에 충돌시 coinCnt 1씩 증가  
필드 내 랜덤 위치에 재생성

→ 코인 충돌시 코인 재생성 + 꼬리 추가

벽 또는 자기 꼬리에 충돌시 사망  
게임 오버 UI 오픈

## 진행과정

### ↳ 코인 애니메이션



Rotation.y값 0 ~ 360

Samples 프레임 30





## ■ 진행과정

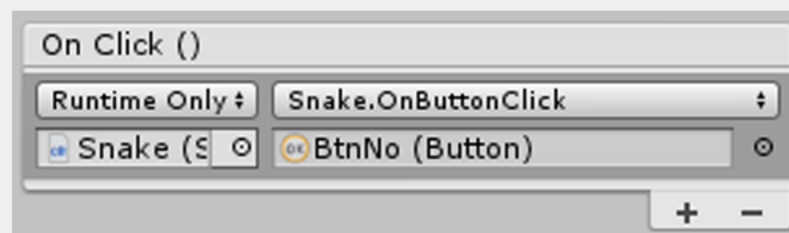
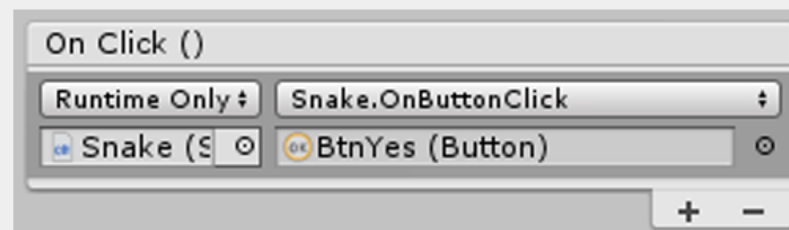
### → UI

```
//UI 표시
void SetTime()
{
    TxtCoin.text = coinCnt.ToString("획득 코인: 0");

    float span = Time.time - startTime;
    int h = Mathf.FloorToInt(span / 3600);
    int m = Mathf.FloorToInt(span / 60 % 60);
    float s = span % 60;

    TxtTime.text = string.Format("시간: {0:0}:{1:0}:{2:00}", h, m, s);
}

// UI 버튼클릭
public void OnButtonClick(Button button) {
    switch (button.name) {
        case "BtnYes":
            SceneManager.LoadScene(SceneManager.GetActiveScene().name);
            break;
        case "BtnNo":
            Application.Quit();
            break;
    }
}
```



## ■ 한계점

— 게임 시작 후 키보드 조작이 없을 시 코인을 먹어도 게임 오버로 처리



## ■ 결과



---

끝까지 봐주셔서 감사합니다.

과목 게임엔진기초

학과 소프트웨어응용학부

학번

이름 조우현

