긴장푸는순간 RETRY(SCHOOL MAZE)

- VR의 이점을 살리고싶어 제작(컨트롤러)
- 게임을 하며 손에땀이찬다 라는말을 느끼게 해주는 게임
- 분위기는무섭지만 게임내용은 가벼운 겉과속이다른게임
- 애니메이션을통한 게임스토리 설명

플레이방식

- 컨트롤러를 이용해 캐릭터를 움직인다.
- 쫓아오는 거미와,벽에 부딪치지않고 GOAL을 향해간다.
- 잡히거나 부딪치면 처음부터 다시시작
- 클리어 할 시 플레이어이름 기록가능

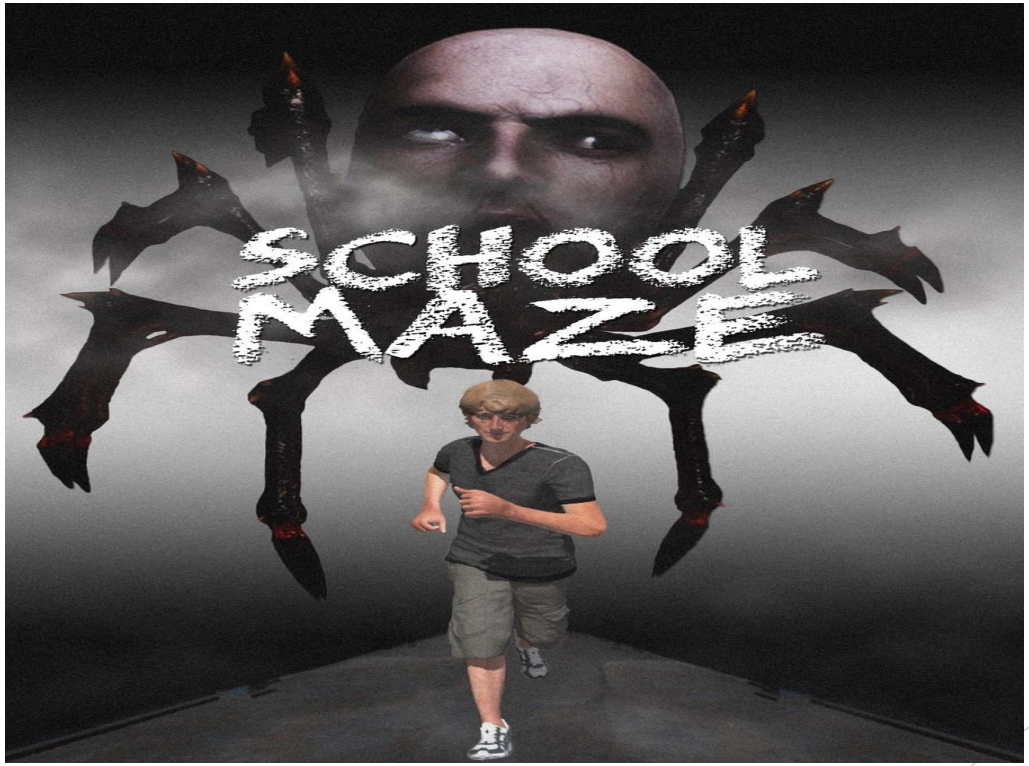
SCHOOL MAZE(2019G-STAR출품)

- 장르 : 캐주얼
- 플랫폼 : PC(VR)
- 필요 장비 : Vive
- 연령층 : 전체이용가
- 제작툴 : Unity3D
- 언어 : C#
- 프로젝트 인원 : 2명

(프로그래머1명,그래픽1명)

담당 분야 : 게임기획,게임개발

- 몬스터vr계약 제의



영상 자료 :

https://drive.google.com/drive/folders/18iOaNwuMuReUuAoG-QwCaTvTuS1Ft-n

게임의 특징

- VR의 이점을 살려 시야와 손으로 캐릭터를 움직이며 집중력을 발휘하며 플레이

- 다양한 애니메이션으로 게임스토리를 설명

게임의 특징

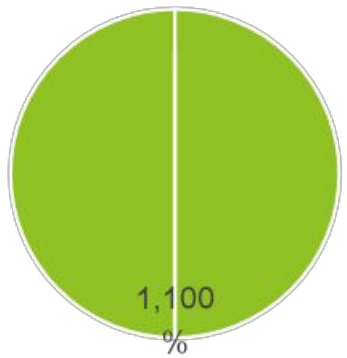
- 클리어 할 시 자기이름을 기록할 수 있으며 승부욕 증가

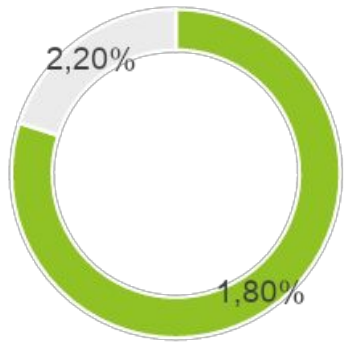
게임플레이

- 플레이어가 칠판에 있는 미로에 손전등을 이용하여 끝까지 가야한다.
- 미로의 벽에 부딪치거나 쫓아오는 거미에 잡히면 처음부터 시작한다.
- 제한시간 안에 모든 스테이지를 클리어 해야한다.

프로젝트 개발

- 개발기간 : 2019.09~2019.11(2달)
- 담당부분 : 기획,개발





중요소스(거미 움직이기)

```
void OnDrawGizmos()
{
    Gizmos.color = rayColor;
    theArray = GetComponentsInChildren<Transform>();
    path_objs.Clear();

    foreach(Transform path_obj in theArray)
    {
        if(path_obj !=this.transform)
        {
            path_objs.Add(path_obj);

        }
    }
    for(int i=0;i<path_objs.Count;i++)
    {
        Vector3 position = path_objs[i].position;
        if(i>0)
        {
            Vector3 previous = path_objs[i - 1].position;
            Gizmos.DrawLine(previous, position);
            Gizmos.DrawWireSphere(position, 0.3f);
        }
    }
}
```

```
if(end)
{
    CurrentWayPointID = 0;
    this.transform.position = pathToFollow.path_objs[CurrentWayPointID].transform.position;
    end = false;
}


    float distance = Vector3.Distance(pathToFollow.path_objs[CurrentWayPointID].position, transform.position);
    transform.position = Vector3.MoveTowards(transform.position, pathToFollow.path_objs[CurrentWayPointID].position, Time.deltaTime * speed);

var rotation = Quaternion.LookRotation(pathToFollow.path_objs[CurrentWayPointID].position - transform.position);
transform.rotation = Quaternion.Slerp(transform.rotation, rotation, Time.deltaTime * rotationSpeed);


if (distance <= reachDistance)
    {
        CurrentWayPointID++;
    }
    if (CurrentWayPointID >= pathToFollow.path_objs.Count)
    {
        CurrentWayPointID = pathToFollow.path_objs.Count - 1;
    }
}
```

중요소스(화면 전환)

```
void Start () {
    beHaviour = GetComponent<PostProcessingBehaviour>();
    beHaviour.profile.vignette.settings = VigModel;
    VigModel.intensity = 0f;
    VigModel.center.x = 0.5f;
    VigModel.center.y = 0.5f;
    VigModel.roundness = 1f;
    VigModel.rounded = true;
    VigModel.smoothness = 1f;
    beHaviour.profile.vignette.enabled = false;
}

// Update is called once per frame
void Update () {
    if (GameManager.instance.gamestage>=6)
    {
        timer.instance.start = false;
        maze.SetActive(false);
         number += 0.001f;
        beHaviour.profile.vignette.enabled = true;
        GameManager.instance.gamestart = false;
        timer.instance.start = false;
    }
    if (beHaviour.profile.vignette.enabled == true)
    {
        VigModel.intensity += (0.1f + number) * Time.deltaTime;
        beHaviour.profile.vignette.settings = VigModel;
        if (VigModel.intensity >= 0.87f)
        {

            GameManager.instance.gamestage = 0;
            ani.SetActive(true);
            player.SetActive(false);
            VigModel.intensity = 0;
            beHaviour.profile.vignette.enabled = false;
        }
```

중요소스(키보드구현)

```
if (Physics.Raycast(transform.position, transform.forward, out hit, maxdis, key))
{
    if (changeobj != null)
    {
        if (changeobj != hit.transform)
        {
            changeobj.GetComponent<Image>().color = new Color(255, 255, 255, 0.3f);
            Debug.Log(changeobj.GetComponent<Image>().color);
        }
    }
    hit.transform.gameObject.GetComponent<Image>().color = new Color(255, 255, 255, 0);
    changeobj = hit.transform;
    if (Input.GetKeyDown(KeyCode.A))
    {
        data = data + (hit.transform.GetChild(0).GetComponent<Text>().text);
        ip.text = data;
    }
}
if (Physics.Raycast(transform.position, transform.forward, out hit, maxdis, enter))
{
    if (changeobj != null)
    {
        if (changeobj != hit.transform)
        {
            changeobj.GetComponent<Image>().color = new Color(255, 255, 255, 0.3f);
            Debug.Log(changeobj.GetComponent<Image>().color);
        }
    }
    hit.transform.gameObject.GetComponent<Image>().color = new Color(255, 255, 255, 0);
    changeobj = hit.transform;
    if (Input.GetKeyDown(KeyCode.A))
    {
        PlayerPrefs.SetString("N", data);
        GameOver.SetActive (true);
        board.SetActive (false);
    }
```

```
private void inputkey()
{
    if (Physics.Raycast(transform.position, transform.forward, out hit, maxdis, key))
    {
        if (numberindex == 0)
            data = "";
        if (numberindex <= 5) {
            numberindex++;
            data = data + (hit.transform.GetChild (0).GetComponent<Text> ().text);
            ip.text = data;
        }
    }
    if(Physics.Raycast(transform.position, transform.forward, out hit, maxdis, enter))
    {
        name.text = data;
        PlayerPrefs.SetString("N", data);
        GameOver.SetActive (true);
        board.SetActive (false);
    }
}
```