

PORTFOLIO

게임 개발자 정래규_포트폴리오

이름	정래규
생년월일	2000.06.12
이메일	wogns9769@naver.com
전화번호	010-8631-0253
주소	울산시 중구

PORTFOLIO

CONTENTS

01.

VR게임개발

대학교 과제로 Unity사용경험 및 숙련도, 협업 능력 향상을 위해 진행

02.

Ghost보드게임

Unity 사용경험 및 숙련도 향상, Photon서버 사용경험 및 숙련도 향상

03.

04.

05.

06.

Project

VR게임개발

About project

대학교 과제로 Unity사용경험 및 숙련도, 협업 능력 향상을 위해 진행하였습니다.

VR기기를 사용하여 진행하고 어그로 수치 관리를 하여 귀신을 피해서 병원을 탈출하는 게임입니다.

Introduce project

작업 기간	2023-05-2~2023-06-21(약 2개월)
인력 구성(기여도)	서버1명, 맵 디자인1명, 적 개발1명, VR개발1명(적 개발 100%)
프로젝트 목적	대학교 과제로 Unity사용경험 및 숙련도, 협업 능력 향상을 위해 진행
프로젝트 내용	VR기기를 사용하여 진행하고 어그로 수치 관리를 하여 귀신을 피해서 병원을 탈출하는 게임입니다.
주요 업무 및 상세 역할	1) 귀신 에셋과 애니메이션 넣기 2) 어그로 수치에 따른 귀신의 움직임 만들기 3) 플레이어 사망시 귀신의 모션 제작
사용언어 및 개발 환경	C#,Unity
참고 자료	Github 링크 : ChungRaeGyu (Chung Rae Gyu) (github.com)

Main work ① 귀신 에셋과 애니메이션 넣기

• 기능 소개

-귀신 에셋 찾기

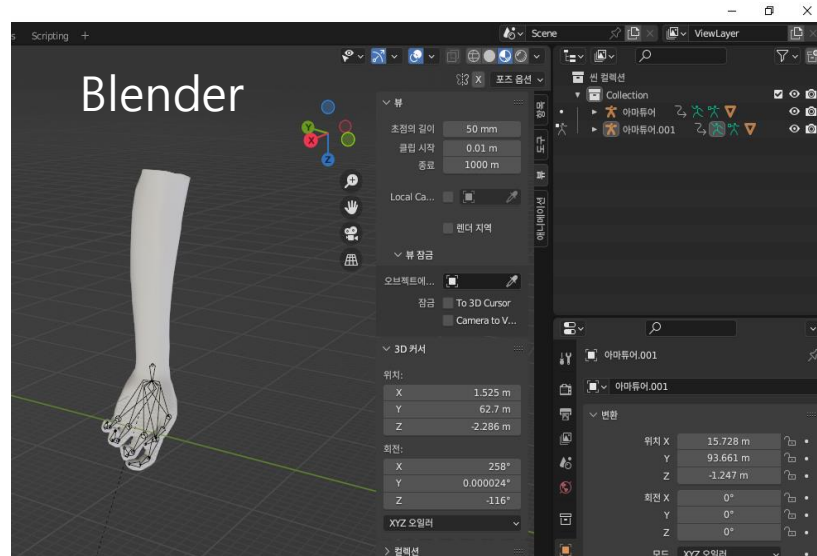
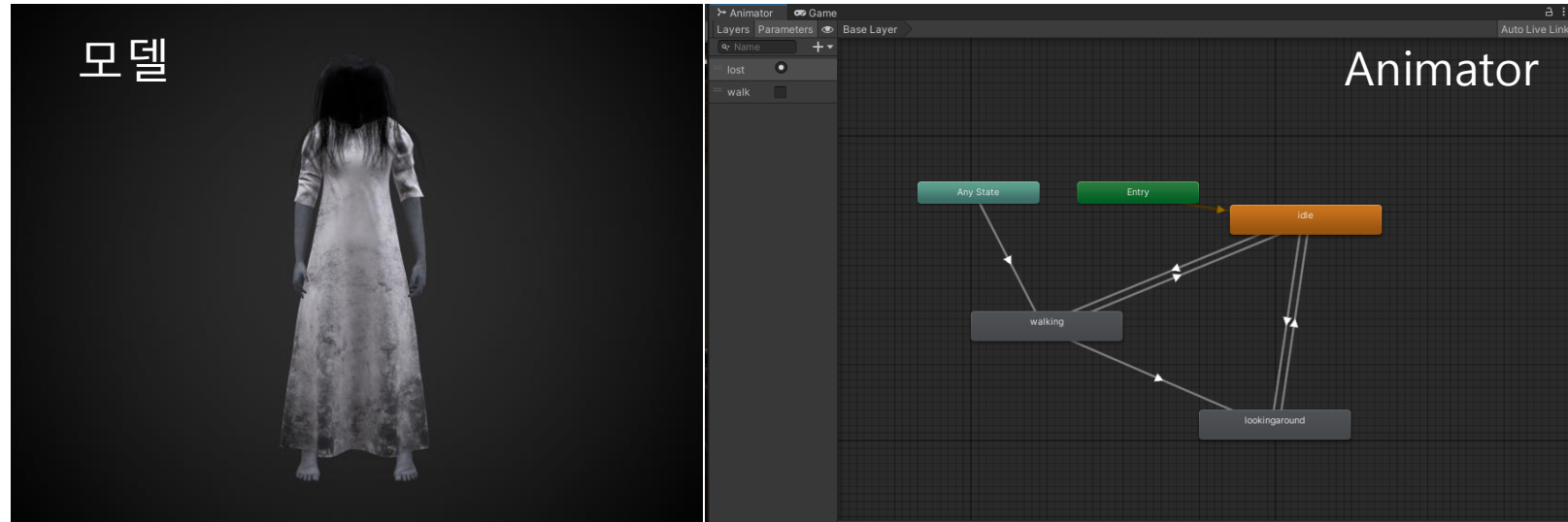
-애니메이션 적용

작업 내용

-AssetStore, Sketchfab, CGTrader에서 모델을 찾았다.

-Mixamo에서 동작 애니메이션을 찾았다.

- Blender를 사용하여 미리 받아놓은 손 에셋에 리깅을 하여 모션을 제작하였다.



Main work ②어그로 수치에 따른 귀신의 움직임 만들기

• 기능 소개

- 어그로 수치100 이하일때 5초마다 1씩 어그로 수치 감소
- 어그로 수치 50~100일 때 일정 확률로 플레이어 뒤를 지나간다.

• 작업 내용

- Time.deltaTime을 이용하여 시간감소를 만들었다.
- 귀신이 플레이어의 뒤쪽으로 위치하게 하기 위해서 플레이어가 바라보는 방향과 위치를 계산하는 식을 썼다. 확률 계산도 하였다.

```
public void over50per(){ //정신력 수치가 50퍼 이상일 때
    StartCoroutine(Fadein(1));
    nav.enabled=false;
    rigid.useGravity=false;
    over50perSound.Play();
    //플레이어의 뒤에 그리고 90도로 사라지기
    GameObject Playertr = GameObject.Find("OVRPlayerController");
    //Plyer포지션에 방향에서 12만큼만 빼준다.
    transform.position = Playertr.transform.position - Playertr.transform.forward * 4;
    transform.position -= new Vector3(0, 1.5f, 0);
    Quaternion newRotation = Quaternion.Euler(0f, Playertr.transform.rotation.y + 90, 0f);
    transform.rotation = newRotation;
    rigid.velocity = transform.forward * 1.5f;
    StartCoroutine(velocity0());
}
```

50~100일 때

어그로 수치 관련 코드

```
if (attention_level >= 100)
{
    if(GhostScript_NoPoint.diestart == false){
        GameObject.Find("HorrorGirl").GetComponent<GhostScript_NoPoint>().setDirection(transform.position);
        if(Audiobool){
            heartbit.Play();
            Audiobool=false;
        }
    }
}
else if(attention_level >=50){
    if(timer>settime){
        Debug.Log("체크");
        timer=0;
        if(Random.Range(1,10)%3==0){
            Debug.Log("실행");
            GameObject.Find("HorrorGirl").GetComponent<GhostScript_NoPoint>().over50per();
        }
    }
    timer+=Time.deltaTime;
}
if(attention_level>0&&attention_level<100){
    decreaseTimer -= Time.deltaTime;
    if(decreaseTimer<=0)
    {
        attention_level-=decreaseAmount;
        decreaseTimer=5f;
    }
}
```

Main work ③ 플레이어 사망시 귀신의 모션제작

• 기능 소개

-불이 켜질때 마다 점점 앞으로 점멸 하고 마지막에 손으로 눈을 가린다.

• 작업 내용

-좌표를 지정해서 귀신을 점멸하게 만들었다.



액션 시작

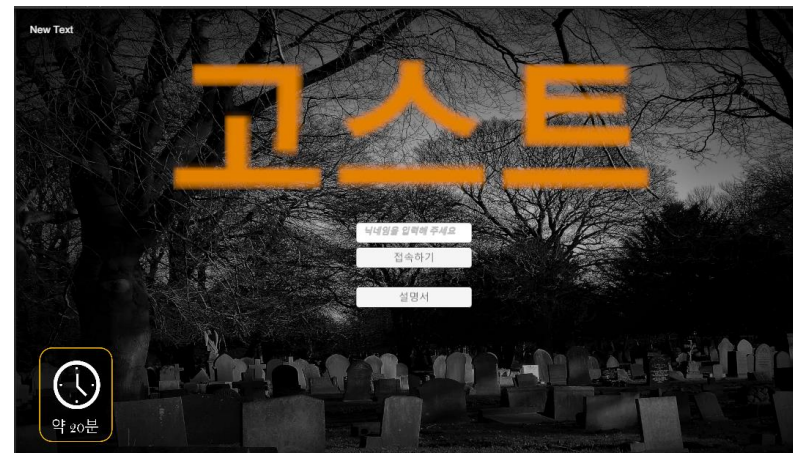
```
//PlayerScript에서 사용
public void PlayerDieAction(){
    diestart=true; //한번만 하기 위한 값
    nav.ResetPath();
    Debug.Log("네비 초기화");
    anim.SetBool("walk",false);
    StartCoroutine(Fadein(1));
    rigid.useGravity = false;
    nav.enabled = false;
    SpotlightController light = GameObject.Find("SpotlightController").GetComponent<SpotlightController>();
    light.StartCoroutine(light.lightout());
}
```

불과 귀신의 위치 코드

```
public IEnumerator lightout(){
    laughsound.Play();
    hand.SetActive(true);
    yield return new WaitForSecondsRealtime(7f);
    for(i =0;i<Spotlight.Length-1;i++){
        yield return new WaitForSecondsRealtime(1f);
        lightsound[i].Play();
        Spotlight[i].enabled = true;
        GameObject.Find("HorrorGirl").transform.position=position[i];
        yield return new WaitForSecondsRealtime(1f);
        Spotlight[i].enabled = false;
        GameObject.Find("HorrorGirl").transform.position = new Vector3(0,0,0);
    }
    Spotlight[i].enabled = true;
    handAnim.SetTrigger("Die");
    i=0;
}
```

Project 02.

GHOST(보드게임)



About project

Unity 사용경험 및 숙련도 향상, Photon서버 사용경험 및 숙련도 향상
상대방의 파란유령4개를 모두 잡거나, 나의 파란 유령이 상대방 모서리에 화살표 밖으로 내보내거나, 나의 빨간유령4개가 모두 잡히면 승리하는 보드게임입니다.

Introduce project

작업 기간	2023-09-01~2023-10-28 (약 2개월)
인력 구성(기여도)	개발1명
프로젝트 목적	Unity 사용경험 및 숙련도 향상, Photon서버 사용경험 및 숙련도 향상
프로젝트 내용	상대방의 파란유령4개를 모두 잡거나, 나의 파란 유령이 상대방 모서리에 화살표 밖으로 내보내거나, 나의 빨간유령4개가 모두 잡히면 승리하는 보드게임입니다.
주요 업무 및 상세 역할	Photon서버 연동 방 관리 유령잡기 유령놓기 유령의 배치가 올바른지 확인 차례 확인 적 죽이기 시작과 종료 다시하기
사용언어 및 개발 환경	C#,Unity
참고 자료	음향 : ✓ SFX provided by 셀바이뮤직 https://sellbuymusic.com/md/slpnfzf-acwfhwk

Main work ① Photon서버 연동

• 기능 소개

-유니티 에셋인 Pun2를 사용하여 멀티플레이가 가능하게 만들었다.

작업 내용

-photonengine.com사이트에서 서버를 생성 후 사용하였다.

-PhotonNetwork함수를 이용하여 서버연결을 했다.

```
void Update()
{
    StatusText.text = PhotonNetwork.NetworkClientState.ToString();
}

public void Connect(){
    PhotonNetwork.ConnectUsingSettings();
}

public override void OnConnectedToMaster()
{
    print("서버접속 완료");
    if(PhotonNetwork.LocalPlayer.NickName.IsNullOrEmpty()){
        PhotonNetwork.LocalPlayer.NickName = NickNameInput.text;
    }
    JoinLobby();
    PhotonNetwork.LoadLevel("Lobby");
}

public void JoinLobby()=>PhotonNetwork.JoinLobby();

public override void OnJoinedLobby(){
    print("입장");
    myList.Clear();
}
```

Main work ② 방 관리

- 기능소개

- 방이 만들어지거나 없어질 때마다 업데이트한다.

- 방을 클릭하면 그 이름의 방으로 바로 들어간다.

방 업데이트



```
public override void OnRoomListUpdate(List<RoomInfo> roomList)
{
    int roomCount = roomList.Count;
    for(int i=0; i<roomCount;i++){
        Destroy(GameObject.Find("RoomNameListBtn(Clone)"));
    }
    for (int i=0; i<roomCount;i++){
        if(!roomList[i].RemovedFromList){ //if RoomOut
            if (!myList.Contains(roomList[i])){
                myList.Add(roomList[i]); //if myList and roomList n
                print("방 추가 ");
            }
            else {
                myList[myList.IndexOf(roomList[i])] = roomList[i];
                //myList.IndexOf(roomList[i]) : Index return for that o
                print("방 정렬 " );
            }
        }
        else if (myList.IndexOf(roomList[i]) != -1) {
            myList.RemoveAt(myList.IndexOf(roomList[i]));
            print("방삭제");
        }
    }
    myListRefresh();
}
```

```
void myListRefresh(){
    int myListCount = myList.Count;
    for (int j = 0; j < myListCount; j++)
    {
        GameObject RoomNameList = Instantiate(RoomNameList_Txt);
        Text RoomBtnName = RoomNameList.GetComponentInChildren<Text>();
        RoomBtnName.text = myList[j].Name;
        RoomNameList.transform.SetParent(RoomListContent.transform, false);
        print("실행 " + j);
    }
}
```

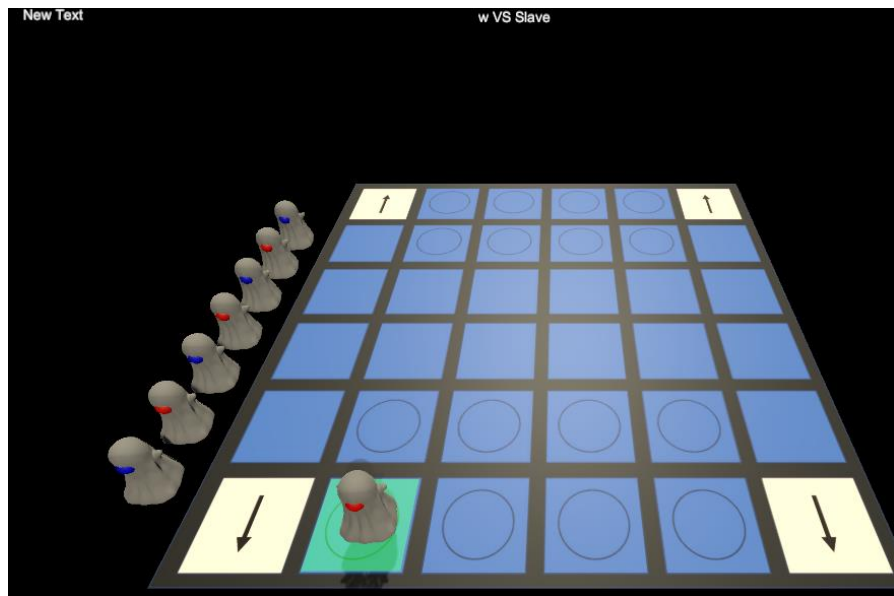
다시 방을 추가한다.

```
public void ButtonJoinRoom()
{
    print("이름 : " + EventSystem.current.currentSelectedGameObject.GetComponentInChildren<Text>().text);
    PhotonNetwork.JoinRoom(EventSystem.current.currentSelectedGameObject.GetComponentInChildren<Text>().text);
}
```

Main work ③유령집기

- 기능소개

- OnMouseDown() 함수를 이용하여 유령을 잡았을 때 유령의 현재 위치를 저장하고 놓고 싶은 위치를 볼 수 있게 판의 색 변화로 나타내었다.



마우스 클릭 시

```
void OnMouseDown()
{
    if(gamemanager.myturn){
        if(!gamemanager.Move){
            if(pv.IsMine){
                pickup = true;
                m_ZCoord = Camera.main.WorldToScreenPoint(gameObject.transform.position).z;
                m_Offset = gameObject.transform.position - GetMouseWorldPosition();
                pv.RPC("MouseDownRPC", RpcTarget.AllBuffered, Iname);
                nowLocation = Iname;
            }
        }else{
            announcement.text = "이미 유령을 움직였습니다. 차례를 넘겨주세요";
        }
    }
}
```

판의 색 변화

```
if(pickup){
    Ray ray = maincamera.ScreenPointToRay(Input.mousePosition);
    RaycastHit hit;

    //RayCast

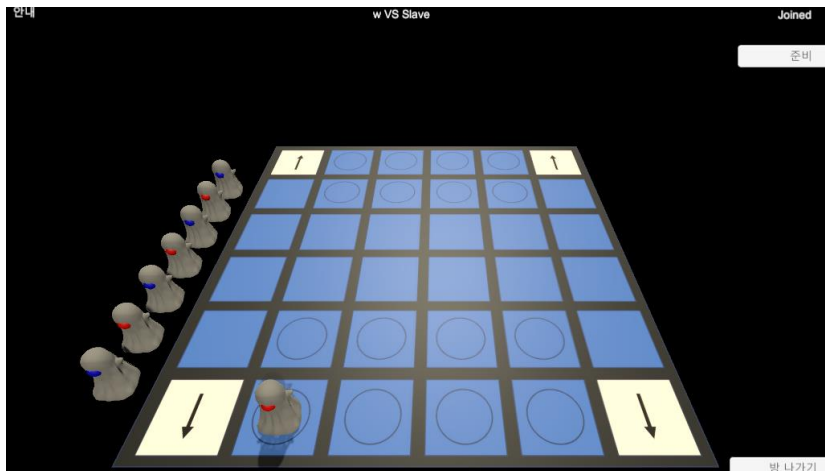
    if(Physics.Raycast(ray,out hit,10000f,layerMask)){

        //Point Color

        if(lastpointRenderer==null){
            point = hit.collider.gameObject;
            name = point.name;
            Iname = Convert.ToInt32(name);
            lastpointRenderer = point.GetComponent<Renderer>();
            lastpointColor = lastpointRenderer.material.color;
            hit.collider.GetComponent<Renderer>().material.color = twinkleColor;
        }
        if(name!=hit.collider.name){
            lastpointRenderer.material.color = lastpointColor;
            lastpointRenderer=null;
        }
    }
}
```

Main work ④유령놓기

- 기능소개
 - 게임 중일때는 내 차례인지 확인한다.
 - 유령을 원하는 장소에 배치하고 판의 색을 원상복구한다.
 - 몇 번째 판에 배치됐는지 체크한다.
 - 현재 위치가 어딘지 저장한다.



```
void OnMouseUp() {  
    if (gamemanager.myturn)           마우스 클릭 놓기  
    {  
        if (!gamemanager.Move)  
        {  
            if (pv.IsMine){  
                gamemanager.checkempty[0] = true; //for Reset  
                lastpointRenderer.material.color = basicColor;  
                pickup = false;  
                afterLocation = Iname;  
                putdownGhost(gamemanager.gamestart);  
            }  
        }  
        else  
        {  
            announcement.text = "이미 유령을 움직였습니다. 차례를 넘겨주세요";  
        }  
    }  
}
```

Main work ⑤유령의 배치가 올바른지 확인

- 기능소개

- 게임 중일 때

상하좌우 한 칸씩 움직였는지 확인한다.

다른 유령이 있다면 적인지 아군인지 판단 후 적이면 잡고 아군이면 이전 위치로 돌아간다.

- 게임 시작 전일 때

시작 지점에 올바르게 났는지 확인한다.

```
void putdownGhost(bool IsStart){
    if(!gamemanager.checkempty[Iname]){           유령을 났을 때
        if(IsStart)
        {
            if(Math.Abs(nowLocation-afterLocation)==1|| Math.Abs(nowLocation - afterLocation) == 6){
                moveghost(IsStart);
                GoalInCheck();
            }
            else{
                announcement.text="상,하,좌,우 한칸씩 가능합니다.";
                transform.position = Beforeposition;
                Iname = nowLocation;
                pv.RPC("putdownGhostRPC",RpcTarget.AllBuffered,nowLocation);
            }
        }
    }
    else
    {
        if (point.CompareTag("startposition"))
        {
            moveghost(IsStart);
        }
        else{
            announcement.text = "시작 지점에 배치해 주세요";
            transform.position = startposition;
            afterLocation = nowLocation;
            transform.name=gname;
        }
    }
}
```

```
else{
    if(GameObject.Find(gname+Iname).GetComponent<PhotonView>().IsMine){           유령을 났을 때
        announcement.text = "빈곳에 놔주세요";
        if (IsStart){
            transform.position = Beforeposition;
            Iname = nowLocation;
            pv.RPC("putdownGhostRPC", RpcTarget.AllBuffered, nowLocation);
        }else{
            Iname = 0;
            transform.position = startposition;
        }
        afterLocation = nowLocation;
    }
    else{
        if (IsStart){
            if (Math.Abs(nowLocation - afterLocation) == 1 || Math.Abs(nowLocation - afterLocation) == 6)
            {
                if (GameObject.Find(gname+Iname).CompareTag("Red")){
                    gamemanager.ColorCheck("Red");
                }else
                {
                    gamemanager.ColorCheck("Blue");
                }
                pv.RPC("DestoryGhost",RpcTarget.AllBuffered,Iname);
                moveghost(IsStart);
                GoalInCheck();
                print("잡음");
            }
        }
        else{
            announcement.text = "상,하,좌,우 한칸씩 가능합니다.";
            transform.position = Beforeposition;
            Iname = nowLocation;
            pv.RPC("putdownGhostRPC", RpcTarget.AllBuffered, nowLocation);
        }
    }
}
```

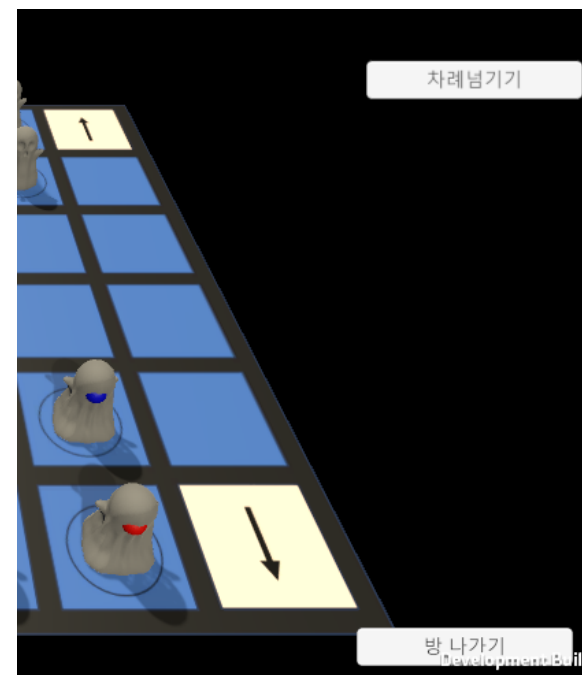
Main work ⑥차레 확인

- 기능소개
 - 차례넘기기 버튼을 눌렀을 때 차례가 넘어간다
 - PunRPC를 이용하여 모든 플레이어에게 내용을 전달합니다.

```
if(gamestart){  
    if(time> timer){  
        timer += Time.deltaTime;  
    }else{  
        pv.RPC("GameStartTxt0ffRPC",RpcTarget.AllBuffered);  
    }  
    if (myturn)  
        turnTxt.text = "내 차례";  
    else  
        turnTxt.text = "상대 차례";  
    if(Red==4){  
        GameOver("패배");  
    }  
    if(Blue==4){  
        GameOver("승리");  
    }  
}
```

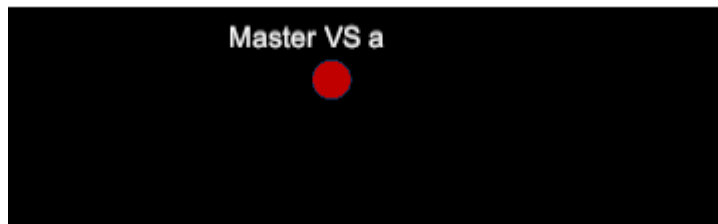
차례확인

```
[PunRPC] 시작 버튼 끄기  
void GameStartTxt0ffRPC(){  
    GameStartTxt.SetActive(false);  
}
```



Main work ⑦ 적 죽이기

- 기능소개
- 적을 잡았을 때 색을 확인한다.



```
if (GameObject.Find(gname+Iname).CompareTag("Red")){
    gamemanager.ColorCheck("Red");
}else
    gamemanager.ColorCheck("Blue");
pv.RPC("DestoryGhost",RpcTarget.AllBuffered,Iname);
moveghost(IsStart);
GoalInCheck();
print("잡음");
```

잡았을 때 색 체크

```
public void ColorCheck(string a){
    if(a=="Red"){
        Red++;
        RedImage[Red-1].SetActive(true);
    }else{
        Blue++;
        BlueImage[Blue-1].SetActive(true);
    }
}
```

맞는 색 개수 올리기

```
if(Red==4){
    GameOver("패배");
}
if(Blue==4){
    GameOver("승리");
}
```


Main work ⑧ 시작과 종료

- 기능소개
- 상대방 모서리에 들어갔을 때 게임을 승리하게 한다.
- 준비완료가 되면 시작버튼이 나온다.
- 적을 잡았을때

```
public void ReadyBtn(){
    if(PhotonNetwork.IsMasterClient)
    {
        if(checkempty[2]&& checkempty[3]&& checkempty[4]&& checkempty[5]&&
            checkempty[8]&& checkempty[9]&& checkempty[10]&& checkempty[11])
        {
            ready.GetComponentInChildren<Text>().text = "준비완료";
            ready.GetComponent<Button>().interactable = false;
            announcement.text = "준비완료";
            pv.RPC("readyRPC", RpcTarget.AllBuffered);
        }
        else{
            announcement.text = "시작 위치에 유령을 배치해 주세요";
        }
    }
    else
    {
        if (checkempty[26] && checkempty[27] && checkempty[28] && checkempty[29] &&
            checkempty[32] && checkempty[33] && checkempty[34] && checkempty[35])
        {
            ready.GetComponentInChildren<Text>().text = "준비완료";
            ready.GetComponent<Button>().interactable = false;
            announcement.text = "준비완료";
            pv.RPC("readyRPC", RpcTarget.AllBuffered);
        }
        else{
            announcement.text = "시작 위치에 유령을 배치해 주세요";
        }
    }
}
```

준비 버튼

```
void GoalInCheck(){
    print("Iname : " + Iname);
    print("point : "+point.name);
    switch(Iname){
        case 31: if(PhotonNetwork.IsMasterClient && point.CompareTag("Goal") && transform.CompareTag("Blue")){
            gamemanager.GoalIn();
        }
        break;
        case 36:
            if (PhotonNetwork.IsMasterClient && point.CompareTag("Goal") && transform.CompareTag("Blue"))
            {
                gamemanager.GoalIn();
            }
            break;
        case 1 :
            if (!PhotonNetwork.IsMasterClient && point.CompareTag("Goal") && transform.CompareTag("Blue"))
            {
                gamemanager.GoalIn();
            }
            break;
        case 6:
            if (!PhotonNetwork.IsMasterClient && point.CompareTag("Goal") && transform.CompareTag("Blue"))
            {
                gamemanager.GoalIn();
            }
            break;
    }
}
```

모서리에 들어갔을 때

```
if(Red==4){
    GameOver("패배");
}
if(Blue==4){
    GameOver("승리");
}
```

```
public void GameOver(string a){
    pv.RPC("GameReset",RpcTarget.AllBuffered);
    GameResult.text = "a";
    if(a=="승리")
        pv.RPC("GameOverRPC", RpcTarget.OthersBuffered, "패배");
    else
        pv.RPC("GameOverRPC", RpcTarget.OthersBuffered, "승리");
}
```

PunRPC를 사용하여 모든 플레이어에게 전달

Main work ⑨ 다시하기

- 기능소개

- 종료패널에서 다시하기를 눌렀을 때 다시 할 수 있도록 관련 변수와 유령의 위치를 초기화 시킨다.

```
void RestartBtn(){
    EndingPanel.SetActive(false);
    timer=0;
    turn.gameObject.SetActive(false);
    startpointCheck = 0;
    gamestart = false;
    readycheck = 0;
    ready.interactable=true;
    announcement.text = "안내";
    Move = false;
    turnoverBtn.gameObject.SetActive(false);
    myturn=true;
    for (int i = 0; i < checkempty.Length; i++)
    {
        if(checkempty[i]&& i!=0){
            checkempty[i] = false;
            Destroy(GameObject.Find("Ghost" + i));
        }
    }
    AddGhost();
}
```

다시시작

```
void AddGhost()
{
    print("유령 생성");
    if (PhotonNetwork.IsMasterClient)
    {
        print("마스터");
        x = -6;
        for (int i = 0; i < 8; i++)
        {
            if (i % 2 == 0)
            {
                PhotonNetwork.Instantiate("Ghost_red", new Vector3(x, 0, Z + i), Quaternion.identity, 0);
            }
            else
            {
                PhotonNetwork.Instantiate("Ghost_blue", new Vector3(x, 0, Z + i), Quaternion.identity, 0);
            }
        }
    }
    else
    {
        print("슬레이브");
        Vector3 camera = new Vector3(44.237f, 180, 0);
        mainCamera.transform.rotation = Quaternion.Euler(camera);
        mainCamera.transform.position = new Vector3(0, 7.45f, 7.5f);
        x = 6;
        for (int i = 0; i < 8; i++)
        {
            if (i % 2 == 0)
            {
                PhotonNetwork.Instantiate("Ghost_red", new Vector3(x, 0, Z + i), Quaternion.Euler(0, 180, 0), 0);
            }
            else
            {
                PhotonNetwork.Instantiate("Ghost_blue", new Vector3(x, 0, Z + i), Quaternion.Euler(0, 180, 0), 0);
            }
        }
    }
}
```

유령생성

End of Document
