

Capstone Design

Final Presentation

Team: Fugitive Or Murderer

Team Member: 김관년, 박영범, 이돈휘,
이현성, 최종호, 황준원

Fugitive

OR

MURDER

ER

목차



CONTENTS A

게임 배경

CONTENTS B

구현 기술

CONTENTS C

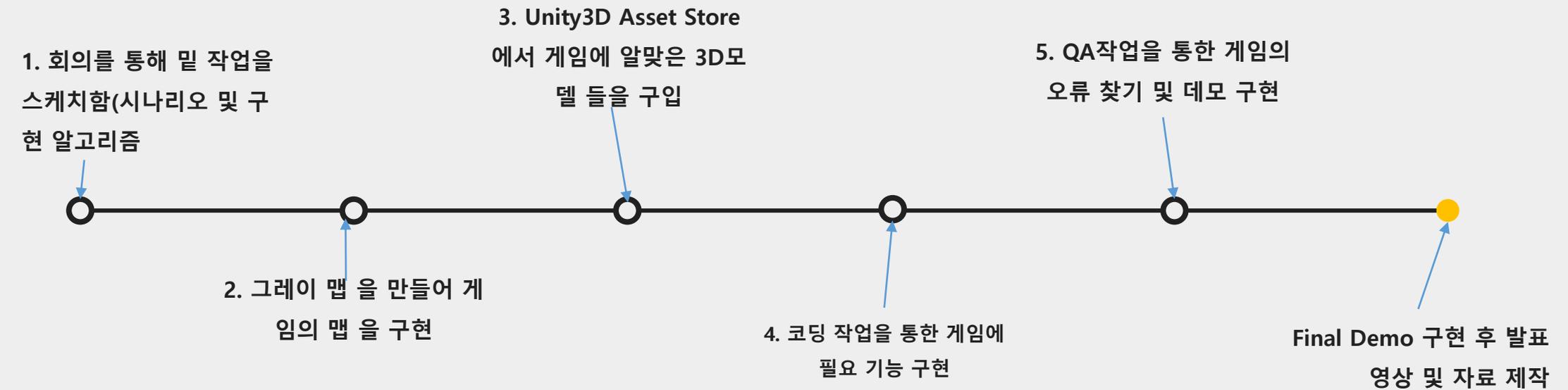
시장성

게임의 시나리오

실험에 참가한 죄수가 조작된 기억의 파편을 찾아서 자신이 살인자 또는 도망자로 기억이 조작되어 기억을 찾은 후 탈출구를 찾아 탈출을 해야함

CONTENTS B

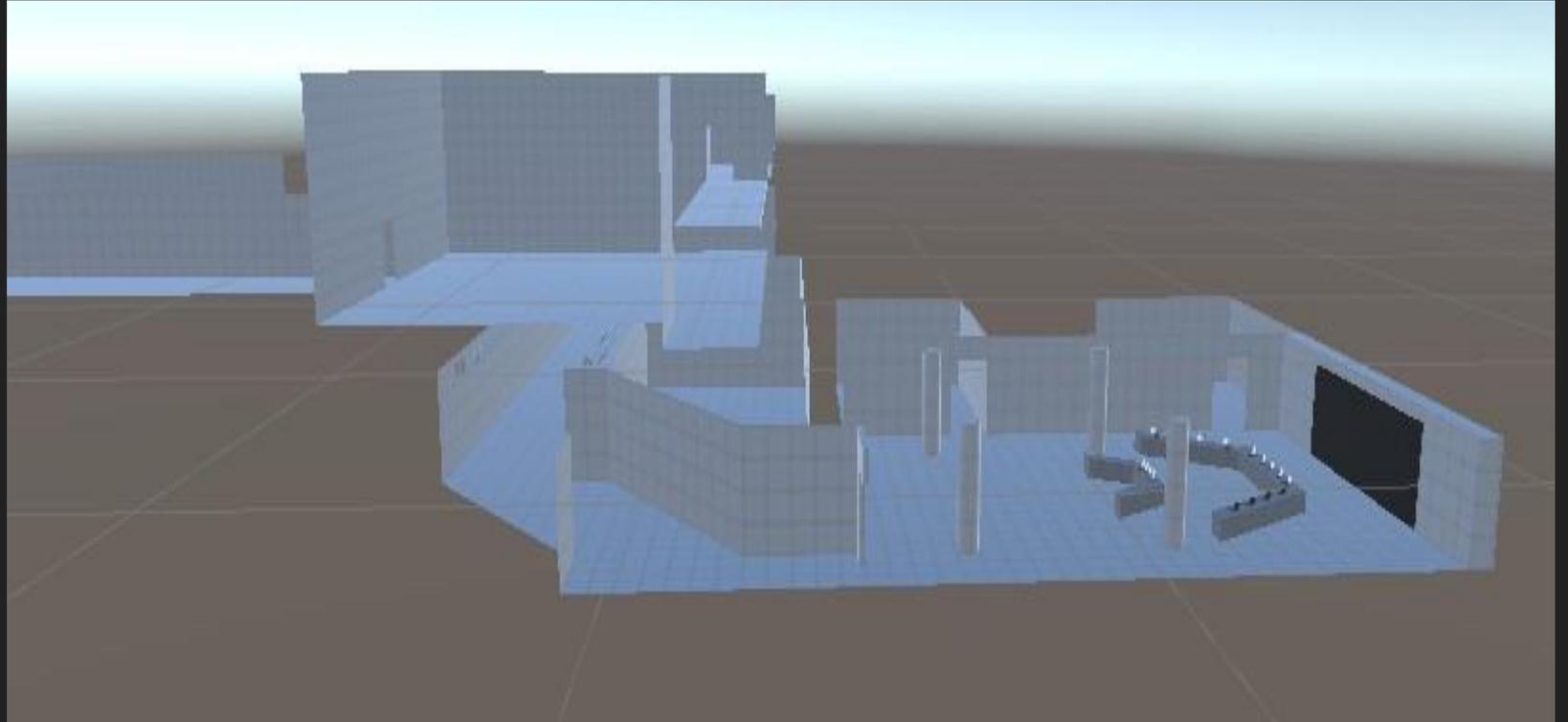
제작 과정



게임 구현 기술

10%

그레이 맵



Unity3D Probulider를 통한 게임의 맵
구상

게임 구현 기술

30%

Asset 구매

The image shows a screenshot of the Unity Asset Store interface. The top section displays a grid of asset packs for sale, including 'Hospital Bundle', 'Blood Spray', 'Toy Soldiers', 'Clocks collection', 'Kid Room', 'The Alleyway | HDRP Asset Pack', 'Modern Soldier Camo Pack', 'Horror Collection: Blood', 'Garage', 'Modular! Apartments', 'House Furniture Pack', 'PBR Wooden Furniture Pack', and 'School laboratory - interior and props'. Each asset pack includes a thumbnail, name, price, and a '삭제하기' (Delete) button.

The bottom section shows an 'Order Details' page for an order dated April 27, 2020. The order includes the following items:

Purchased Assets	Price	Qty	Total
Modular Prison Pack	\$85	x1	\$42.50
Office and Police Station Pack (Modu...	\$79	x1	\$39.50
Medicine Bottle	\$4.99	x1	\$4.99
Police Security Control Room	\$20	x1	\$20
Vintage First Aid Kit	\$12.99	x1	\$12.99
CCTV Cameras 3-Pack	\$4.99	x1	\$4.99
Tools and workplace	\$12	x1	\$12
Industrial Cabinets and Shelves	\$9.99	x1	\$9.99
Deer Trophy	\$9.99	x1	\$9.99
Electric Fuse Box 01	\$4.99	x1	\$4.99
Clothing set	\$7	x1	\$7
Electric Props	\$7	x1	\$7

Below the order details, there is a list of purchased assets with their respective prices, quantities, and total costs:

- Characters pack - guards and prisone. \$29.99 x1 \$29.99
- Characters pack - laboratory scientist. \$24.99 x1 \$24.99
- Garage Workshop \$30 x1 \$30
- City Bus \$10 x1 \$10
- Mansion Level Interior \$12.99 x1 \$12.99
- High Quality Server Room Pack \$12 x1 \$12

캡스톤 지원금을 통한 Asset을 구매

게임 구현 기술

70%

코딩 구현

코딩 부분: Raycast, NeviMesh, Inventory UI, AI, Animation,

RayCast

```
참조 0개
void Update ()
{
    Ray ray = cam.ViewportPointToRay (new Vector3 (0.5f, 0.5f, 0));
    RaycastHit hit;

    if (Physics.Raycast(ray, out hit, rayDistance))
    {
        if (hit.transform.GetComponent<Door>())
        {
            crossHairStatus = 2;
            if (Input.GetKeyDown(KeyCode.E))
            {
                hit.transform.GetComponent<Door>().InteractWithThisDoor();
            }
        }
        else if (hit.transform.tag.Equals("Item"))
        {
            crossHairStatus = 1;
            if (Input.GetKeyDown(KeyCode.E))
            {
                Item item = hit.transform.GetComponent<Item>();
                hit.transform.gameObject.SetActive(false);
                if(item.Memory_Key != null)
                {
                    item.Memory_Key.SetActive(true);
                    Destroy(item.Memory_Key.gameObject, 5.0f);
                }

                if (item.isMurderer)
                {
                    GameManager.Instance.Add_M_Point();
                }
                else if (item.isRunner)
                {
                    GameManager.Instance.Add_F_Point();
                }
                else if (item.isBoth)
                {
                    GameManager.Instance.Add_Both();
                }
            }
        }
        for (int i=0; i < inventory.item.Length; i++)
        {
            else if (hit.transform.tag.Equals("KeyPad"))
            {
                crossHairStatus = 1;
                if (Input.GetKeyDown(KeyCode.E))
                {
                    hit.collider.SendMessage("Elevator", SendMessageOptions.DontRequireReceiver);
                }
            }
            else if (hit.transform.tag.Equals("Computer"))
            {
                crossHairStatus = 1;
                if (Input.GetKeyDown(KeyCode.E))
                {
                    hit.collider.SendMessage("Computer On", SendMessageOptions.DontRequireReceiver);
                }
            }
            else if (hit.transform.tag.Equals("Key"))
            {
                crossHairStatus = 1;
                if (Input.GetKeyDown(KeyCode.E))
                {
                    hit.collider.SendMessage("GetKey", SendMessageOptions.DontRequireReceiver);
                    hit.transform.gameObject.SetActive(false);
                }
            }
        }
    }
}
```

게임 구현 기술

70%

코딩 구현

코딩 부분: Raycast, Ai, Inventory UI,
Animation

AI

```
참조 0개
private void OnTriggerEnter(Collider other)
{
    if (other.transform.tag.Equals("WayPoint"))
    {
        if (other.transform.GetComponent<WayPointID>().ID == -2)
        {
            Destroy(other.transform.gameObject);
            StartCoroutine(WaitThenMove());
        }
        else if (currentId != other.transform.GetComponent<WayPointID>().ID)
        {
            currentId = other.transform.GetComponent<WayPointID>().ID;
            StartCoroutine(WaitThenMove());
        }
    }
    else if (other.transform.GetComponent<Door>())
    {
        if (other.transform.GetComponent<Door>().isClosed)
            StartCoroutine(DoorOpen(other));
    }
}

참조 2개
IEnumerator WaitThenMove()
{
    na.speed = 0.0f;
    yield return new WaitForSeconds(2.0f);
    do
    {
        idx = Random.Range(0, points.Length);
    }
    while (idx == currentId);

    na.speed = 3.0f;
    goal = points[idx].transform;
}

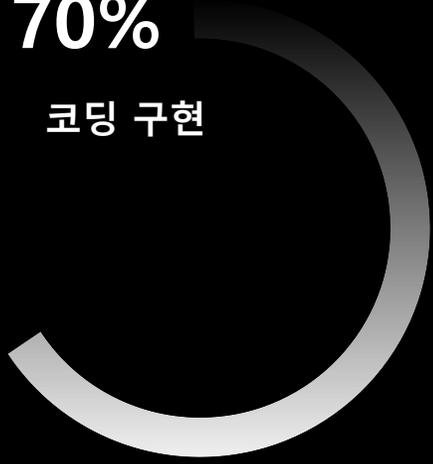
참조 1개
IEnumerator DoorOpen(Collider other)
{
    na.speed = 0.0f;
    anim.SetBool("Kick", true);
    yield return new WaitForSeconds(2.0f);
}
```

문제가 검색되지 않음

게임 구현 기술

70%

코딩 구현



코딩 부분: Raycast, Ai, Inventory UI,
Animation

Inventory

```
참조 0개
public void DropTheItem()
{
    for (int i = 0; i < item.Length; i++)
    {
        if (item[i].sprite.name.Equals(ObjectNa.text))
        {
            item[i].sprite = defaultImage;
            tempItems[i].SetActive(true);
            tempItems[i].transform.position = player.transform.position + new Vector3(0, 1.0f, 0);
            if (!tempItems[i].GetComponent<Rigidbody>())
                tempItems[i].AddComponent<Rigidbody>();
            ObjectEx.text = "";
            ObjectNa.text = "";
            break;
        }
    }
}

참조 0개
public void EquipItem()
{
    for (int i = 0; i < item.Length; i++)
    {
        if (item[i].sprite.name.Equals(ObjectNa.text) && tempItems[i].GetComponent<Item>().isWeapon)
        {
            player.GetComponent<CanAttect>().enabled = true;
            item[i].sprite = defaultImage;
            tempItems[i].SetActive(true);
            tempItems[i].transform.parent = rightHand.transform;
            tempItems[i].transform.localPosition = new Vector3(0, 0, 0);
            tempItems[i].transform.localRotation = Quaternion.identity;
            tempItems[i].transform.tag = "Weapon";
            ObjectEx.text = "";
            ObjectNa.text = "";
            if (!player.GetComponent<Animator>().runtimeAnimatorController.Equals(weaponAnim))
            {
                player.GetComponent<Animator>().runtimeAnimatorController = weaponAnim;
            }
            if (tempItems[i].GetComponent<Rigidbody>())
            {
                Destroy(tempItems[i].GetComponent<Rigidbody>());
            }
        }
    }
}
```

게임 구현 기술

70%

코딩 구현

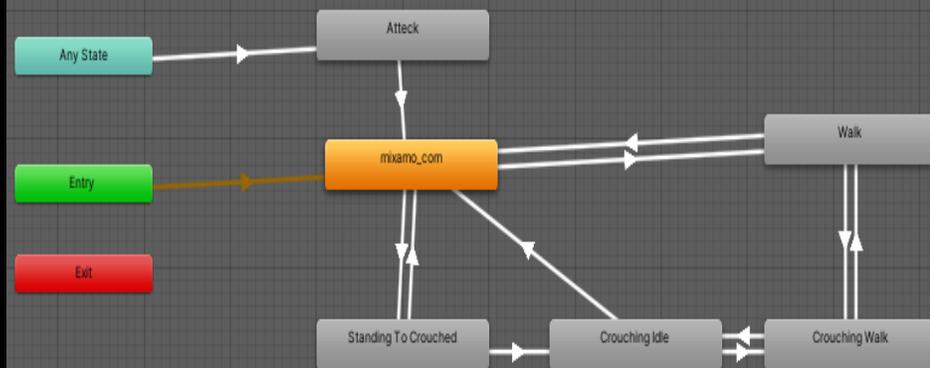
코딩 부분: Raycast, Ai, Inventory UI,
Animation

Animation

```
if ( Input.GetAxis("Horizontal") != 0 || Input.GetAxis("Vertical") != 0 )
{
    anime.SetBool("Crouch Walk", true);
    anime.SetBool("Walk", true);
}
else
{
    anime.SetBool("Crouch Walk", false);
    anime.SetBool("Walk", false);
}

if ( Input.GetKeyDown(KeyCode.C) )
{
    crouch = !crouch;
}

if (crouch)
{
    anime.SetBool("Stand", false);
    controller.center = new Vector3(0, 0.4f, 0);
    controller.height = 0;
}
else
{
    anime.SetBool("Stand", true);
    controller.center = new Vector3(0, 0.9f, 0);
    controller.height = 1.7f;
}
```



게임 구현 기술

100%

맵 완성

맵 구현 완성



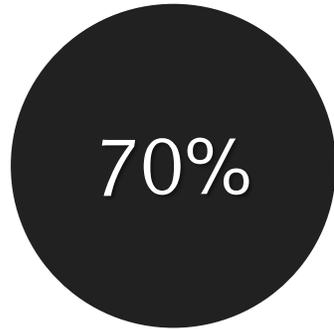
시장성

차별화



살인자와 도망자 라는 역할을 나눠 주면서 플레이어가 찾은 기억으로 플레이 할 수 있도록 구현

완성도



처음 만드는 게임이기에 미흡한 부분이 많아서 아직 완성도는 미흡

재미



싱글플레이의 한 방향성만 있는 것이 아닌 여러가지 변수와 가능성이 있기에 한번 게임을 끝을 보아도 다른 역할을 플레이를 궁금하게 함

다른 게임들과 차별성이 있기에 게임 시장에 출시를 목표로 함에 있어 기대 효과가 있다고 생각합니다.

80%

Final Demo 영상



감사합니다.