

공모전이 꿈꾸게 한

게임 개발자 이현희 포트폴리오

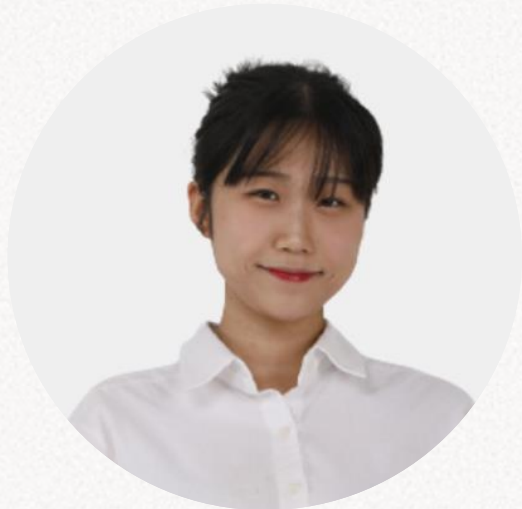


이현희 LEE HYUN HEE

010-2726-4991

42.4.hyunhlee@gmail.com

About Me



Lee Hyun Hee

EDUCATION

2023.2 덕성여자대학교 IT미디어공학과 졸업
2018.3 덕성여자대학교 IT미디어공학과 입학
2017.2 서울문현고등학교 졸업

STACK

- C++ : C++를 활용하여 알고리즘 문제를 해결할 수 있습니다.
- C# : 주 개발 프로그램인 unity에서 다양한 구조를 만들어 사용해본 적 있습니다.
- Unity : 외부 플러그인과 연결하여 AR 서비스를 만들 정도의 이해도를 가지고 있으며, 구글 플레이 스토어 출시 경험이 있습니다.
- Linux : 42SEOUL 개발자 양성 교육 프로그램에서 1년 이상 Linux를 이용한 개발을 하였습니다.

PROJECT

- <드리밍 펫> : UNITY를 이용, 광고를 넣어 수익 창출할 수 있게 하여 플레이스토어에 출시한 하이퍼캐주얼 게임
- <NAVATAR> : Unity와 pose tracking을 접목시킨 AR 기반 가상 옷 입히기 서비스
- <Fill IT!> : Unity를 이용하여 2주간의 짧은 시간동안 빠르게 개발한 곰인형에 숨을 채워 넣는 하이퍼캐주얼 게임
- <코엑스 꾸미기> : MAXST 회사 주관 공모전을 위해 Unity를 이용해 개발한 VPS 기반 코엑스 꾸미기 앱

AWARDS

2022 ICT 멘토링 한국정보처리학회 학술대회
최우수 논문상 동상

LICENSE

2022.9 정보처리기사
2021.3 리눅스 마스터 1급



PROJECT

1. <드림িং 펫>: 하늘로 간 강아지를 지상으로 데려오는 플레이스토어 출시 게임
2. <Fill IT!>: 곰인형에 솜을 채워 넣는 하이퍼 캐주얼 게임
3. <코엑스 꾸미기>: VPS 기반 코엑스 꾸미기 어플
4. <나바타>: AR 기반 가상 옷입히기 서비스

1. 드림펫 : 하늘로 간 강아지를 지상으로 데려오는 플레이스토어 출시 하이퍼 캐주얼 게임

2023 간단한 조작만으로 즐길 수 있는 힐링 게임이다. Android 플레이스토어에 출시하였다. 광고를 넣어 수익을 창출을 할 수 있도록 만들었다.

- 개발 기간: 2023.2 ~ 2023. 4 (2달)
- 개발 도구: Unity Engine
- 개발 언어: C#

기여도

- 개발 50

팀원 구성

디자인 1, 개발 2, 아트 1

맡은 부분

- 커신 동작 구현
- 전체 Manager 코드 틀 구현
- 무한 배경 움직임 구현
- Level Manager를 통한 레벨디자인 구현
- Object pooling을 활용한 적 생성, 삭제 구현
- 피버타임 구현

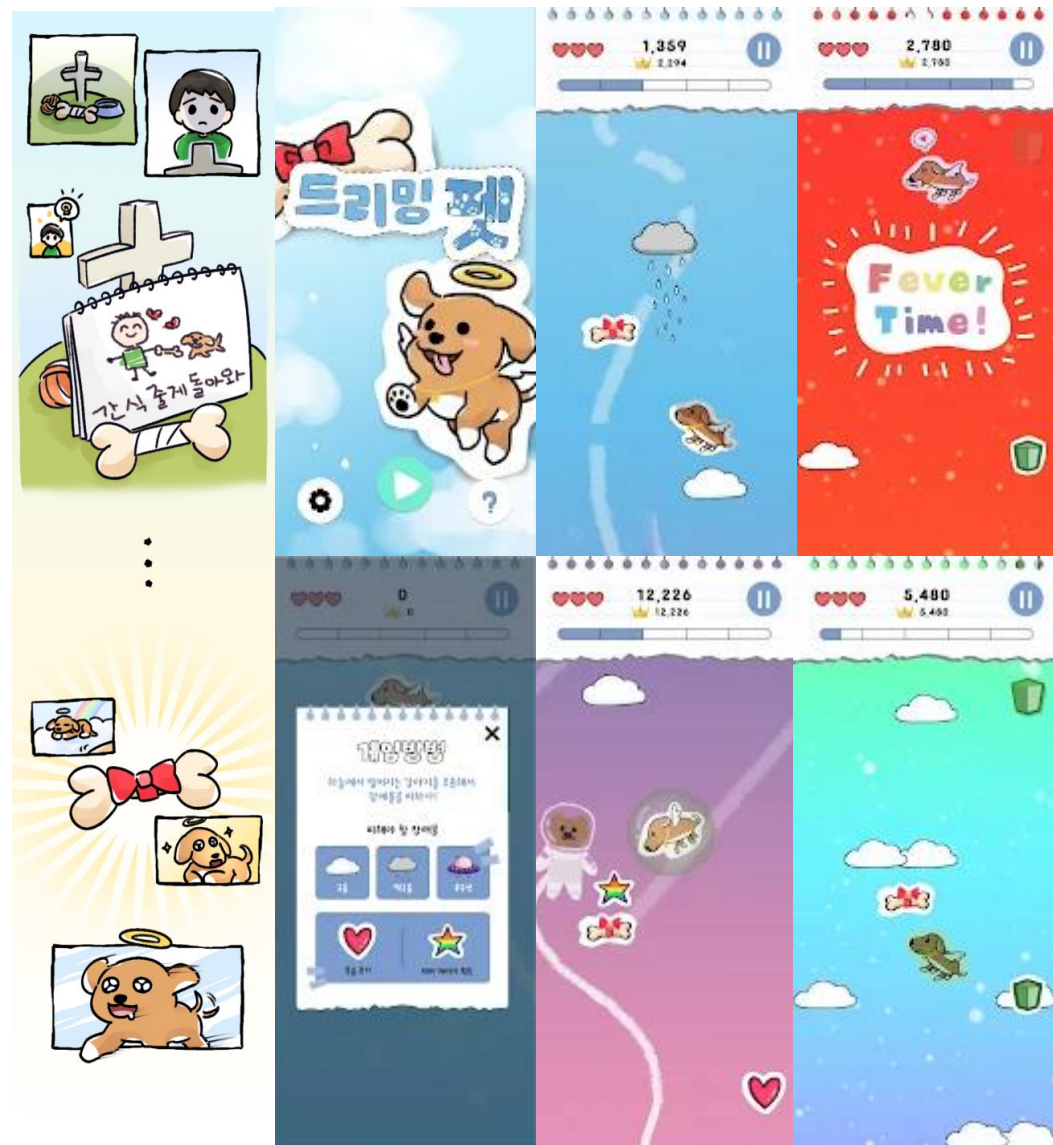
ETC

게임 연합 동아리 'GAMEMAKERS'에서 팀원들과 함께 만들었습니다.
처음으로 기획자, 디자이너, 개발자의 역할이 명확히 나누어졌습니다.

[<Github 주소>](#)

[<구글스토어>](#)

[<시연 영상 보러가기>](#)



- Object Manager에서 prefab을 최초 생성하고 prefab의 active상태를 끄고 키는 방식을 이용하여 enemy의 상태를 관리하였습니다.

```
public GameObject MakeObj(string type)
{
    switch(type)
    {
        case "enemy":
            targetPool = enemy;
            break;
        case "fever":
            targetPool = fever;
            break;
        case "rain":
            targetPool = rain;
            break;
        case "spaceship":
            targetPool = spaceship;
            break;
        case "heart":
            targetPool = heart;
            break;
        case "shieldpiece":
            targetPool = shieldpiece;
            break;
    }

    for (int index = 0; index < targetPool.Length; index++)
    {
        if (!targetPool[index].activeSelf)
        {
            targetPool[index].SetActive(true);
            return targetPool[index];
        }
    }
}
```

```
public class LevelManager : MonoBehaviour
{
    void Update()
    {
        if(!stop)
        {
            if (t[i] > 0)
            {
                t[i] -= Time.deltaTime;
                if (GetComponent<GameManager>().feverState) realSpawnDelay = 0.6f;
                else realSpawnDelay = maxSpawnDelay[i];

                if (curSpawnDelay > realSpawnDelay)
                {
                    enemyCnt++;
                    if (enemyCnt % 3 == 0 && GetComponent<GameManager>().feverState) SpawnShieldPiece();
                    if (enemyCnt % 9 == 0 && !GetComponent<GameManager>().feverState && level >= 3) SpawnRain();
                    else if (enemyCnt % 13 == 0 && !GetComponent<GameManager>().feverState && level >= 4) SpawnSpaceShip();
                    else if (enemyCnt % 6 == 0 && !GetComponent<GameManager>().feverState) SpawnFever();
                    else if (enemyCnt % 11 == 0 && !GetComponent<GameManager>().feverState) SpawnHeart();
                    else SpawnEnemy();

                    curSpawnDelay = 0;
                }

                DeleteRain();
                DeleteEnemy();
                DeleteFever();
                DeleteSpaceShip();
                DeleteHeart();
                DeleteShieldPiece();
            }
            else
            {
                if (i < t.Length - 1)
                {
                    void Start()
                    {
                        //Level Design
                        t = new float[8] { 0.0f, 5.0f, 15.0f, 30.0f, 30.0f, 40.0f, 40.0f, 40.0f };
                        maxSpawnDelay = new float[8] { 5f, 3f, 1.5f, 1f, 1f, 0.8f, 0.8f, 0.6f };
                        speed = new float[8] { 2.4f, 2.7f, 3.0f, 3.3f, 3.6f, 3.9f, 4.2f, 4.5f };

                        maxT = t[t.Length - 1];
                        level = 0;
                    }
                }
            }
        }
    }
}
```

- Level Manager에서 Start에서 레벨이 바뀌는 시간, 스폰 딜레이, enemy speed를 변화시키도록 설정하여 일정 시간마다 몬스터가 생성 및 삭제되도록 구현하였습니다.

```

void Update()
{
    Vector3 curPos = transform.position;
    Vector3 nextPos = Vector3.up * speed * Time.deltaTime;
    transform.position = curPos + nextPos;
    if ((transform.position.y+40+200) % 202 <= 1 && transform.position.y / (viewHeight % 3) >= 1)
    {
        if (tmp == 0)
        {
            cyclecnt++;
            cycle.text = string.Format("\'\'{iÄÖ {0}1ÜÄü\'\'", cyclecnt);
            gm.soundManager2.EffectSoundPlay(audioClip);
            StartCoroutine(cycles());
            tmp = 1;
        }
    }
    else
        tmp = 0;

    if (sprites[endIndex].position.y > 0)
    {
        Vector3 downSprite = sprites[endIndex].localPosition;
        sprites[startIndex].transform.localPosition = downSprite + Vector3.down * viewHeight;

        //#.Cursor Index Change
        int endIndexSave = endIndex;
        endIndex = startIndex;
        startIndex = (endIndexSave - 1 == -1) ? sprites.Length - 1 : endIndexSave - 1;
    }
}

```

● 3개의 배경 이미지가 위로 계속해서 이동하는 것처럼 보이도록 하기 위해서 sprites 배열에 배경 이미지들을 세팅하고, 3개의 index들을 계속 바꿔가며 배경 이미지가 무한으로 흐르면서 메모리 사용량이 최소화되도록 구현하였습니다.

2. Fill IT! :곰인형에 타이밍에 맞춰 솜을 채워넣는 하이퍼 캐주얼 모바일 게임

2022 슈퍼센트 주관하이퍼 캐주얼 게임 공모전 출품작
적절한 타이밍에 인형에 솜을 채워 넣는 간단한 방식의 게임입니다.

- 개발 기간: 2021.12 ~ 2022. 1 (2주)
- 개발 도구: Unity Engine, 3DS MAX, Blender
- 개발 언어: C#

기여도

- 기획 30
- 개발 25
- 디자인 30

팀원 구성

디자인 1, 개발 4

맡은 부분

- Unity 내 UI/UX 동작 적용
- 레벨업 처리 및 점수 산정, 점수 저장
- 컨베이어 벨트 움직임 구현

ETC

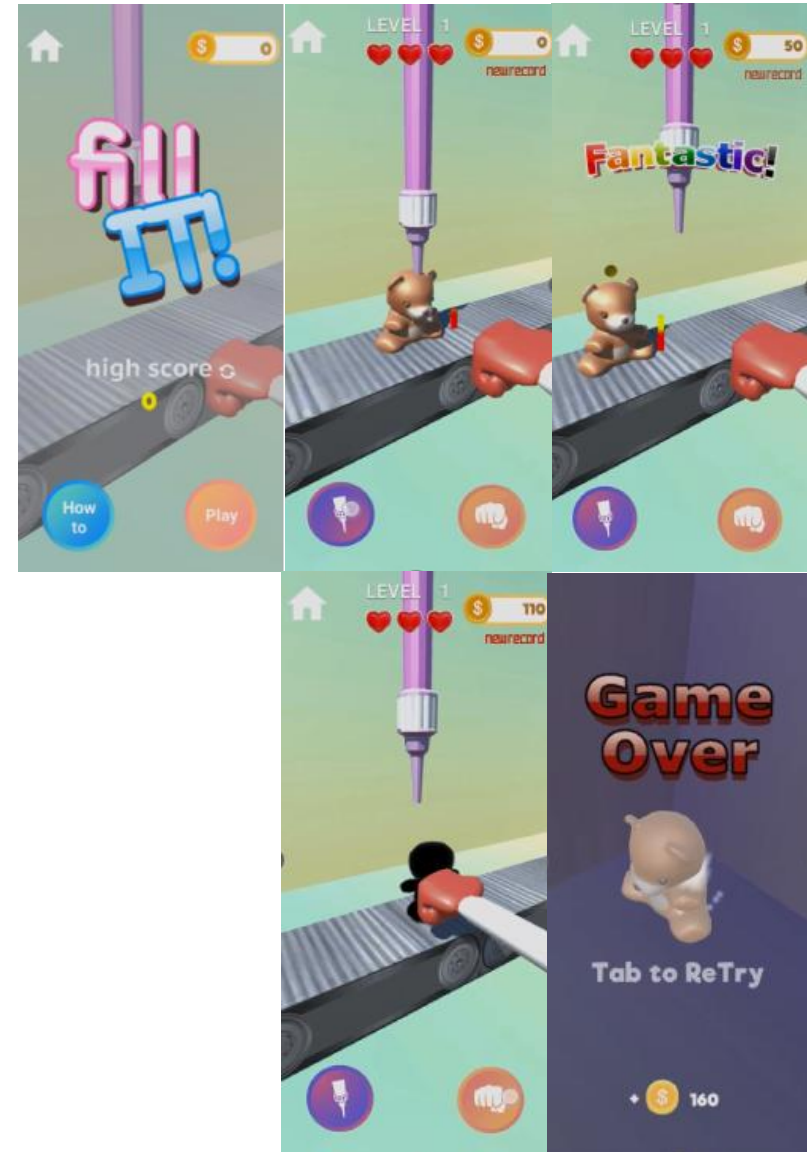
2주라는 짧은 기간동안 진행했던 프로젝트입니다.

역대 최다 인원과 협업을 해본 경험이었습니다.

실제로 apk 파일로 빌드해 주변인들에게 배포 후 피드백을 들으며 완성하였습니다.

<[Github 주소](#)>

<[시연 영상 보러가기](#)>



3. 나만의 코엑스 꾸미기 : MAXST 회사 주관 VPS 기반 코엑스 꾸미기 앱

2021 메타버스 개발자 경진대회 출품작
코엑스 내부를 돌아다니며 스티커와 메모를 추가하거나
화면 상에 선을 그려 나만의 코엑스 지도를 만들 수 있는 서비스이다.

- 개발 기간: 2021.8 ~ 2021. 10 (2달)
- 개발 도구: Unity Engine
- 개발 언어: C#, JSON

기여도

- 기획 50
- 개발 70

팀원 구성

디자인 1, 기획 & 개발 3

말은 부분

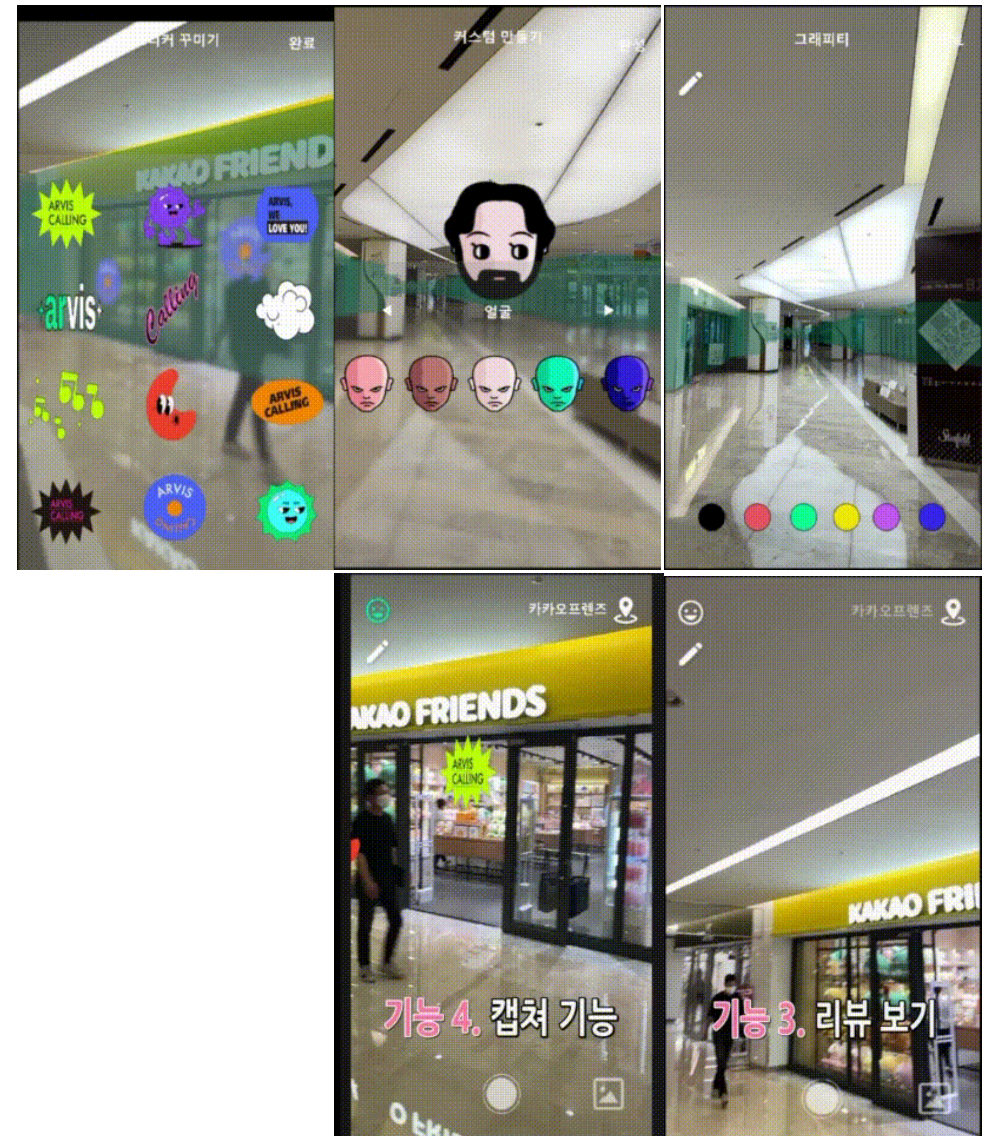
- 코엑스 매장 벽에 스티커 추가, 삭제할 수 있는 기능 구현
- 사용자가 원하는 대로 스티커를 커스터마이징하는 기능 구현
- 꾸며진 코엑스 스크린샷을 갤러리에 저장, 공유하는 기능 구현

ETC

학기중에 진행했던 프로젝트입니다. 처음으로 Unity를 사용해서 공모전에 출품한 작품입니다.

[<Github 주소>](#)

[<시연 영상 보러가기>](#)



- FileStream을 이용하여 스티커가 꾸며진 상태가 기기 안에 저장되도록 하였습니다.

- 코루틴 함수와 ScreenCapture 클래스를 사용하여 스크린샷 기능을 구현하였습니다.

```
public class TakeScreenshot : MonoBehaviour
{
    [SerializeField]
    GameObject blink;

    public void TakeAShot()
    {
        StartCoroutine("CaptureIt");
    }

    IEnumerator CaptureIt()
    {
        string timeStamp = System.DateTime.Now.ToString("dd-MM-yyyy-HH-mm-ss");
        string fileName = "Screenshot" + timeStamp + ".png";
        string pathToSave = fileName;

        GameObject.Find("Canvas").GetComponent<Canvas>().enabled = false;

        ScreenCapture.CaptureScreenshot(pathToSave);
        yield return new WaitForEndOfFrame();
        Instantiate(blink, new Vector3(0f, 0f, -43f), blink.transform.rotation);
        Destroy(GameObject.Find(blink.name + "(Clone)", 0.3f);

        GameObject.Find("Canvas").GetComponent<Canvas>().enabled = true;
    }
}
```

```
public void Save()
{
    // Getting the objects from the randomplacer
    savableObjects = maxstSceneManager.savableObjects;

    // Turns the data from the objects class into binary data
    FileStream fs = File.Create(Application.persistentDataPath + "/SavableObjects.dat");
    Debug.Log("!!" + Application.persistentDataPath);
    BinaryFormatter bf = new BinaryFormatter();
    bf.Serialize(fs, savableObjects);
    fs.Close(); // You MUST close the filestream otherwise it will cause errors!!
}

public bool Load()
{
    string path = Application.persistentDataPath + "/SavableObjects.dat";
    // Checking if the file exists
    if (File.Exists(path))
    {
        FileStream fs = File.Open(path, FileMode.Open);
        BinaryFormatter bf = new BinaryFormatter();
        // Making sure the file is not empty
        if (fs.Length > 0)
        {
            // Turns the data back from the binary into strings and floats
            savableObjects = (List<SavableObjects>)bf.Deserialize(fs);
            maxstSceneManager.savableObjects = savableObjects;
            maxstSceneManager.Reinstantiate();
            fs.Close(); // You MUST close the filestream otherwise it will cause errors!!
            return true;
        }
    }
    // This will happen if the file is non existant or empty
    return false;
}
```

4. 나바타 :Mediapipe 활용한 AR 기반 가상 옷 입히기 서비스

2022 증강현실 (AR)기술을 바탕으로 사용자에게 가상 옷 입기 체험을 제공합니다. 웹캠이나 스마트폰 카메라로 사용자의 몸과 움직임을 인식하고 원하는 옷을 선택하면 3D로 구현된 옷이 사용자의 몸에 맞춰서 실제로 착용한 것처럼 화면에 나타납니다.

- 개발 기간: 2022.2 ~ 2022. 10 (8달)
- 개발 도구: Unity Engine, 3DS MAX, IntelliJ, MySQL
- 개발 언어: C#, SpringBoot, AWS EC2 RDS

기여도

- 기획 90
- 개발 60

팀원 구성

개발 3(유니티 1(본인), 웹 프론트 1, 웹 서버 1)

맡은 부분

- 웹캠으로 실시간으로 사용자 포즈를 인식해 가상 옷 입히는 핵심 기술 개발
- 팀에서 구현한 쇼핑몰 홈페이지와 unity서비스를 연동해 exe파일로 빌드
- 구현한 서비스에서 유니티 내부의 모든 동작 구현, 리깅, 텍스처 매핑으로 옷 다양성 구현

ETC

졸업 프로젝트로 진행했으며 1년동안 개발, 발표, 시연 등을 하였습니다.

저의 아이디어로 시작되어 팀장으로써 PM 역할을 하였습니다.

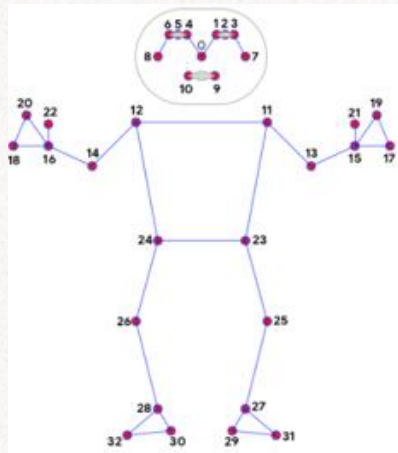
그 결과, ICT멘토링 이브와 공모전에서 동상을 수상하였습니다.

<[Github 주소](#)>

<[시연 영상 보러가기](#)>



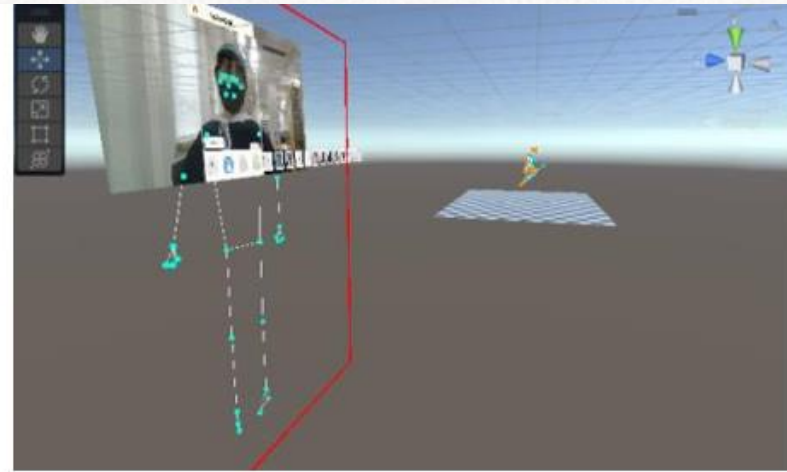
서울 창동 허브에서 100명 이상의 사용자들에게 시연 및 설명
[제10회 졸업전시회 영상 2탄\(길게 감상, 10분\) – YouTube](#) (7:37~ 8:09)



- 0. nose
- 1. left_eye_inner
- 2. left_eye
- 3. left_eye_outer
- 4. right_eye_inner
- 5. right_eye
- 6. right_eye_outer
- 7. left_ear
- 8. right_ear
- 9. mouth_left
- 10. mouth_right
- 11. left_shoulder
- 12. right_shoulder
- 13. left_elbow
- 14. right_elbow
- 15. left_wrist
- 16. right_wrist
- 17. left_pinky
- 18. right_pinky
- 19. left_index
- 20. right_index
- 21. left_thumb
- 22. right_thumb
- 23. left_hip
- 24. right_hip
- 25. left_knee
- 26. right_knee
- 27. left_ankle
- 28. right_ankle
- 29. left_heel
- 30. right_heel
- 31. left_foot_index
- 32. right_foot_index

```
public Vector3[] settingPosition()
{
    findPoints();
    for (int i = 0; i < 33; i++)
    {
        new_point[i] = new Vector3((0.5f - point[i].x) * annotation.rect.width * 0.04f, (0.5f - point[i].y) * annotation.rect.height * 0.04f, zPoint);
    }
    settingVector();
    settingZaxis();
    return new_point;
}
```

Main canvas

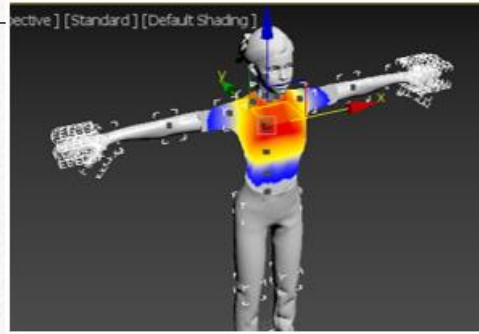


● Mediapipe에서 제공하는 좌표계에서 x, y, z값을 받아 새로운 좌표계를 구축하여 unity화면 내에서 사용자의 위치와 일치하도록 하였습니다.

```
void UpdateController()
{
    _point = pointScript.settingPosition();
    Vector3 newPosition = new Vector3((_point[11].x + _point[12].x) / 2 + man_neck, (_point[23].y + _point[24].y) / 2, _zPoint);
    avatar.transform.position = newPosition;
    settingScale(_point, newPosition);
}
```

● 좌표의 중간 지점을 의미하는 포지션을 세팅하고 3D objec의 위치를 사용자의 위치와 동일하게 하였으며, SettingScale을 이용하여 자동으로 옷의 크기를 조절하도록 하였습니다.

- 3DS MAX로 옷 material을 만들고 사용자가 입을 옷에 해당하는 부분을 Rigging하여 입은 옷이 자연스럽게 움직이도록 만들었습니다.



```
void UpdateIK()
{
    animator.SetIKPositionWeight(AvatarIKGoal.RightHand, 1);
    animator.SetIKPositionWeight(AvatarIKGoal.LeftHand, 1);
    animator.SetIKHintPositionWeight(AvatarIKHint.LeftElbow, 1);
    animator.SetIKHintPositionWeight(AvatarIKHint.RightElbow, 1);
    animator.SetIKPositionWeight(AvatarIKGoal.RightFoot, 1);
    animator.SetIKPositionWeight(AvatarIKGoal.LeftFoot, 1);

    sethandVector3();
    animator.SetIKPosition(AvatarIKGoal.RightHand, rhandVec);
    animator.SetIKPosition(AvatarIKGoal.LeftHand, lhandVec);
    animator.SetIKHintPosition(AvatarIKHint.RightElbow, new Vector3(point[13].x + 3.0f, point[13].y - 10.0f, 105.0f));
    animator.SetIKHintPosition(AvatarIKHint.LeftElbow, new Vector3(point[14].x - 3.0f, point[14].y - 10.0f, 105.0f));
    animator.SetIKPosition(AvatarIKGoal.RightFoot, point[27] - new Vector3(0.0f, 0.0f, 5.0f));
    animator.SetIKPosition(AvatarIKGoal.LeftFoot, point[28] - new Vector3(0.0f, 0.0f, 5.0f));

    animator.bodyRotation = BodyRotation(); //몸통 돌리기
}
```



- Unity내의 함수 IKControl(역운동학)을 이용해서 팔 다리의 꺾임을 구현하였습니다.