

기 업 의 요 구 사 항 을 확 실 히 인 지 하 고 업 무 에 임 하 겠 습 니 다 .

기업의 성공적인 비즈니스를 위해
꾸준하게 덤비겠습니다.



- 정보처리기사
- 컴퓨터그래픽스운용기능사
- 1종 보통 / 2종 소형
- Limited working Proficiency

[illegible]

1

GAME

- 모바일게임 프로젝트
- VR / AR 프로젝트

2

UI / UX

- KDB스타트업_UX디자인

3

WEB

- Jsp, Spring 웹사이트



개발 기간 / 환경

- 2018.02 ~ 2018.03
- Unity 2018.1.6f1
- MagicaVoxel Graphic Tool

팀원 내 역할

- 1인 개발

성과

- 기초적인 Unity의 기술들을 복습하였음.
- MagicaVoxel를 통해 3D Object에 대한 지식을 습득하였음.



개발 기간 / 환경

- 2018.04 ~ 2018.06 3학년 1학기 캡스톤디자인
- Unity 2018.1.6f1

팀원 내 역할

- 4인 중 기획, 클라이언트 프로그래머, FX 담당

성과

- 기술들을 구현하는 과정에서 프로그래밍 디자인 패턴을 적용할 수 있었음.
- Shader, FX에 대한 기초적인 지식을 학습할 수 있었음.

MOBILE GAME PROJECT



“ 미니멀한 아이디어와
빠져드는 재미를 추구하자. ”

- 프로토타입 기획 당시 -



“ 너무 단순하다.
개발이 너무 어려워질 것 같다. ”

- 1학기 개발 후 -



“ 좀 더 대중성있고
아케이드의 특성을 살리자. ”

- 2학기 개발 방향 -

```

public class KnightController : NormalController {
    private SetKnight m_chessState;
    private GameObject m_blockAuraParticle = null;

    int cost = 2;

    public override void Start () {
        base.Start ();
        m_chessState = transform.parent.GetComponent<SetKnight> ();
    }

    public override void AnimGetHit (float hitAngle = 0) {
        if (animator.GetBool ("IsActiveSkillBlock")) {
            if (Utility.IsAngleFaceFront (hitAngle, 270)) {
                animator.Play ("Shield-Block-GetHit1", -1, 0);
                FindObjectOfType<GenerateParticleManage> ().PlayKnightBlockHit (transform);
            } else {
                animator.Play ("Armed-GetHit-B1", -1, 0);
                m_chessState.ActiveSkillBlock (false);
            }
        } else {
            animator.Play ("Armed-GetHit-B1", -1, 0);
            m_chessState.ActiveSkillBlock (false);
        }
    }

    public override void AnimeShooting () {
        animator.Play ("Shield-Run-Forward-Charge", -1, 0);
    }

    public override void AnimeKnight_Attack () {
        animator.Play ("Knight@Sword_Attack", -1, 0);
    }
}

```

- 상속 및 다형성을 통해 캐릭터의 특징을 세분화하였습니다.

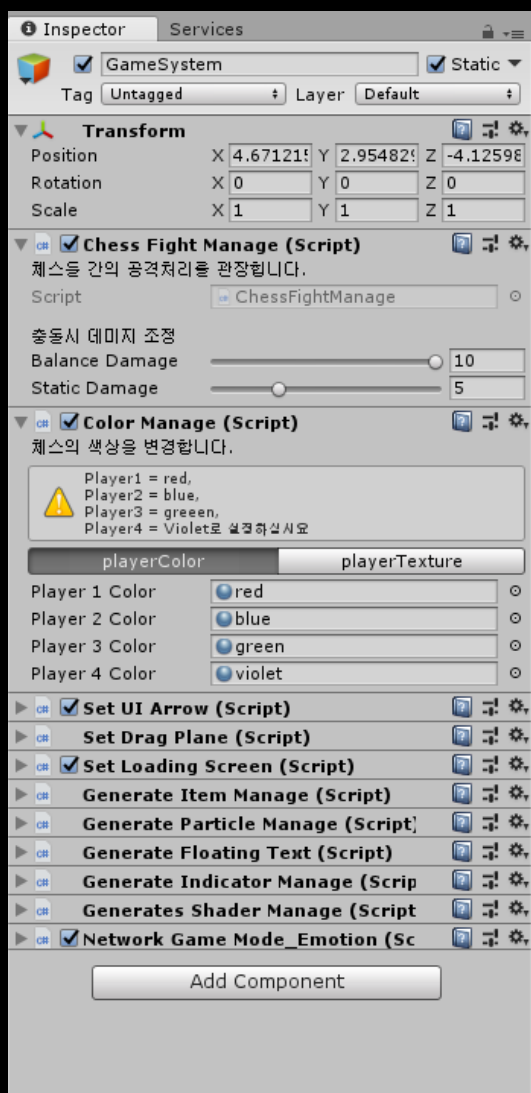
```

#region 유저별 체스 생성
float playerAngle = 6.4f / visitors.Count;
foreach (var index in visitors)
{
    GameObject makechess1 = Instantiate (Resources.Load ("Prefabs/Obj_" + Utility.setMainChess[index.Key]) as GameObject);
    makechess1.GetComponent<ChessState>().isKing = true; //킹 체스
    if (!makechess1.GetComponent<ChessState>().chessStates.ContainsKey ((int) Utility.Slot.player)) makechess1.GetComponent<ChessState>().chessStates[(int)Utility.Slot.player] = index.Key;
    else makechess1.GetComponent<ChessState>().chessStates[(int)Utility.Slot.player] = index.Key;
    namingChess = " " + index.Key + " King 1";
    makechess1.name = makechess1.name.Replace("(Clone)", namingChess);
    makechess1.transform.position = Vector3.up * 2;
    makechess1.transform.position += new Vector3(Mathf.Cos(angle) * 14, 0, Mathf.Sin(angle) * 14);
    makechess1.transform.Find("model").LookAt(Vector3.zero + Vector3.up * makechess1.transform.Find("model").transform.position.y);

    GameObject makechess2 = Instantiate(Resources.Load("Prefabs/Obj_normalChessC") as GameObject);
    if (!makechess2.GetComponent<ChessState>().chessStates.ContainsKey ((int) Utility.Slot.player)) makechess2.GetComponent<ChessState>().chessStates[(int)Utility.Slot.player] = index.Key;
    else makechess2.GetComponent<ChessState>().chessStates[(int)Utility.Slot.player] = index.Key;
    namingChess = " " + index.Key + " 2";
    makechess2.name = makechess2.name.Replace("(Clone)", namingChess);
    makechess2.transform.position = Vector3.up * 2;
    makechess2.transform.position += new Vector3(Mathf.Cos(angle - 0.2f) * 12, 0, Mathf.Sin(angle - 0.2f) * 12);
    makechess2.transform.Find("model").Rotate(makechess1.transform.Find("model").rotation.eulerAngles);
}

```

- 동일한 추상화 메소드, 오버라이딩을 통해 간결한 코드 내에서 모든 클래스를 호출하도록 작업하였습니다.



```
public class VSGameMode_TurnManage : MonoBehaviour {
    [Tooltip ("Obj_UICompliation 내 My Turn에 연결시킵니다.")] public Text m_myTurn;
    [Tooltip ("Obj_UICompliation 내 ( count )에 연결시킵니다.")] public Text m_count;
    [Tooltip ("Obj_UICompliation 내 ( mana )에 연결시킵니다.")] public Text m_mana;
    [Tooltip ("Obj_UICompliation 내 End Turn에 연결시킵니다.")] public Button m_endTurn;
    [Tooltip ("UICanvas 내 LogBar에 연결시킵니다.")] public GameObject m_logBar;
    [Tooltip ("LogBar 내 Log에 연결시킵니다.")] public Text m_log;
    [HideInInspector] public static string playerTurn;
    [HideInInspector] public static int stackTurn;
    [HideInInspector] public Dictionary<string, int> playerGauge = new Dictionary<string, int> ();
    [HideInInspector] public int Level;
    private float m_deltaSec;
    private int m_maxMana;

    void Start () {
        Assert.IsNotNull (FindObjectOfType<VSGameMode_FlowManager> (), "게임 내 VSGameMode_FlowManager를 찾을 수 없습니다.");
        Assert.IsNotNull (FindObjectOfType<VSGameMode_UIManage> (), "게임 내 VSGameMode_UIManage를 찾을 수 없습니다.");
        Assert.IsNotNull (FindObjectOfType<Obj_Segment> (), "게임 내 Obj_Segment를 찾을 수 없습니다.");
        Assert.IsNotNull (FindObjectOfType<VSGameMode_UIManage> (), "게임 내 VSGameMode_UIManage를 찾을 수 없습니다.");
        stackTurn = 0;
        m_deltaSec = 0.0f;
        playerTurn = Utility.player1;
        m_myTurn.text = Utility.player1;
        m_count.text = Convert.ToString (15);
        foreach (var index in Utility.visitors) {
            playerGauge.Add (index.Key, 0);
            if (index.Key == playerTurn)
                m_mana.text = playerGauge[index.Key].ToString ();
        }
        Level = 1;
        SetManaFull ();
    }

    public IEnumerator Timer () //타이머
    {
        while (FindObjectOfType<VSGameMode_FlowManager> ().IsGamePlaying) //게임 중일 때만
        {
            switch (stackTurn) {
                case 5:
                    Level = 2;
                    break;
                case 12:

```

- Event(), IEnumerator () 등을 활용하여 코드를 모듈화하였습니다.
- 커스텀 에디터 시스템과 주석을 통하여 팀원과의 효율적인 작업을 도모하였습니다.

개발 기간 / 환경

- 2018.08 ~ 2018.10 3학년 2학기 캡스톤디자인
- Unity 2018.1.6f1
- MagicaVoxel, maya

팀원 내 역할

- 4인 중 클라이언트 프로그래머, 모델링 담당

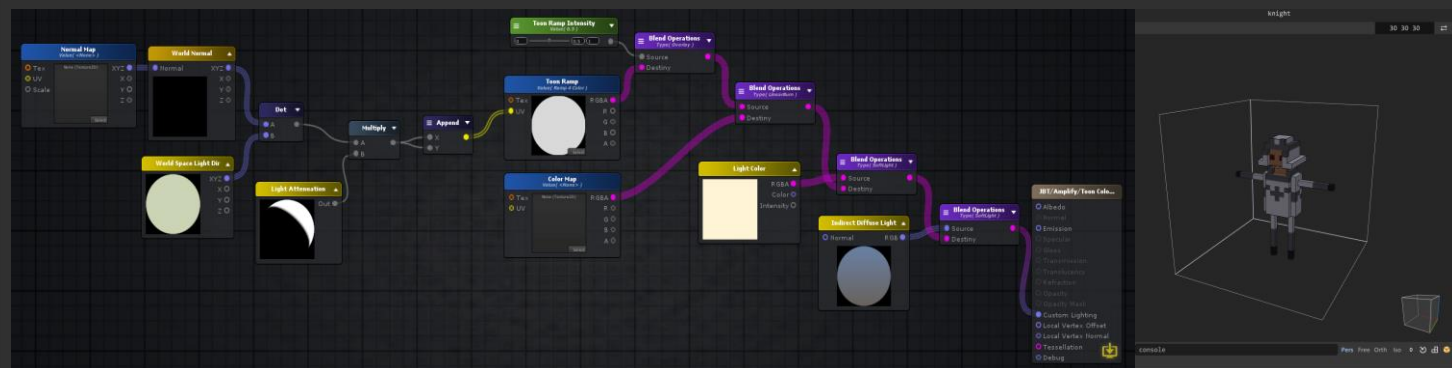
성과

- 안드로이드 Store에 배포하였음.
- MagicaVoxel -> maya 모델링과정을 학습하였음.
- Amplify Shader Assets를 활용하여 그래픽 파이프라인에 대한 전반적인 지식을 습득하였음.

- AS-IS
- ① 게임이 너무 단순하다.
 - ② 개발 과정이 어려울 것 같다.
 - ③ 좀 더 확실한 대중성을 띄었으면 좋겠다.



- To-Be
- ① 아케이드 특성을 띄자.
 - ② 성취 욕구를 자극하도록 퍼즐 요소를 넣자.
 - ③ 개발은 간단하지만 다양한 조합Prefab을 이용할 수 있도록 하자.

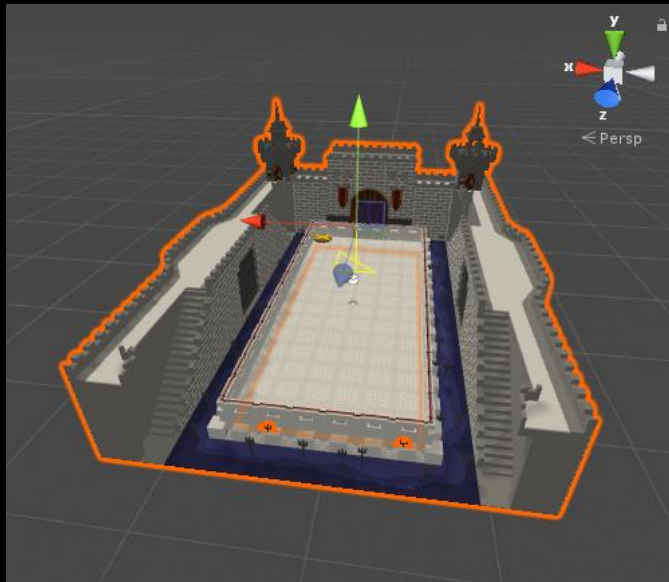
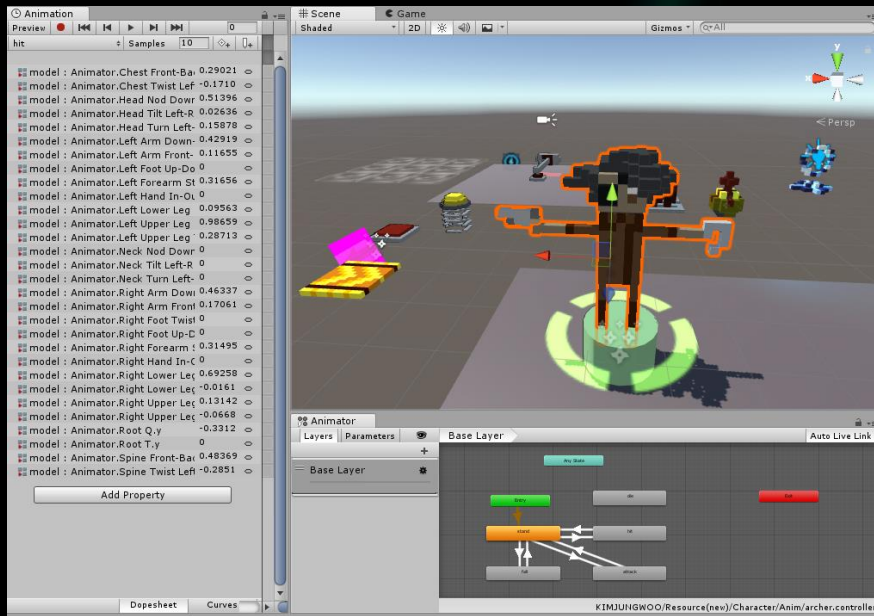


-Amplify Shader Tool을 이용한 Toon Shading작업 과정-

-MagicaVoxel로 3D 모델링-



Copyright© 2019, Team Cube.
All rights reserved.



```
#region 라인렌더러 함수
public void onDirection (float maxPower = 100) {
    if (maxPower == 100) { maxPower = DefinedDistance; }
    maxDistance = maxPower;
    isDirectionOn = true;
    lineRenderer.enabled = true;
}

public void offDirection () {
    isDirectionOn = false;
    lineRenderer.enabled = false;
}
#endregion

#region 써클 함수
public void onCircle () { m_circle.SetActive (true); }
public void offCircle () { m_circle.SetActive (false); }
#endregion

#region 웨이더 함수

public void Default () {
    //디폴트 설정
    foreach (Renderer m in GetComponentsInChildren<Renderer> ()) {
        if (m.material.name != "mtl_Direction (Instance)" && !m.material.name.Contains ("mtl_Circle")) {
            m.material.SetColor ("_RimColor", Color.white);
            m.material.SetFloat ("_RimMin", false ? 1f : 1f);
            m.material.SetFloat ("_RimMax", false ? 0.8f : 1f);
            m.material.SetColor ("_OutlineColor", Color.black);
            m.material.SetFloat ("_Outline", 1);
        }
    }
}

public void Character_SelectTrue () {
    //캐릭터 선택 시
    Default ();
    StartCoroutine (BlinkMoment ());
    foreach (Renderer m in GetComponentsInChildren<Renderer> ()) {
        if (m.material.name != "mtl_Direction (Instance)" && !m.material.name.Contains ("mtl_Circle")) {
            m.material.SetColor ("_OutlineColor", Color.green);
            m.material.SetFloat ("_Outline", 5);
        }
    }
}

private IEnumerator BlinkMoment () {
    foreach (Renderer m in GetComponentsInChildren<Renderer> ()) {

```

- 전반적인 그래픽디자인, 애니메이션, 웨이더를 담당하였습니다.
- 코드를 통해 웨이더 파이프라인에 접근할 수 있도록 하였습니다.



개발 기간 / 환경

- 2019.06 ~ 2019.08 하계방학 인턴쉽
- Unity 2018.2.9f1
- SAMSUNG VR Odyssey plus

팀원 내 역할

- 2인 중 기획, 클라이언트 프로그래머, 애니메이션 담당

성과

- 12가지 동일한 섹션을 구축하기 위해 일관된 프레임워크 개발.
- VR기기에 대한 다양한 지원 함수 및 라이브러리에 대한 학습.
- 이용자에 적합한 VR전용 UI와 매뉴얼 등을 구축하였음.
- 플러그드, 언플러그드 코딩 교육에 대한 차이를 알 수 있었음.



- 기 업 요 구 사 항 -

AS-IS

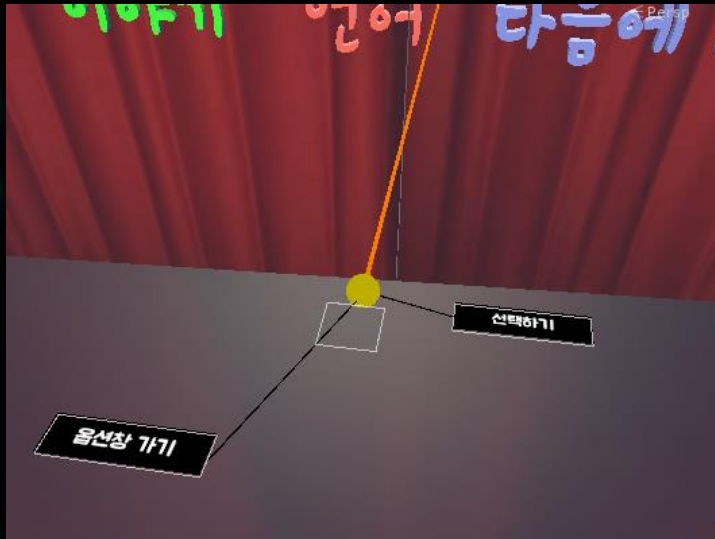
- ① 기존 게임요소가 너무 부족하다.
- ② 12가지 애니메이션에 대한 차별화된 콘텐츠를 제작하고 코딩교육 요소를 넣어달라.
- ③ 좀 더 편리한 UI를 제공해달라.



To-Be

- ① 유아에 적합한 UI / UX 기획 후 개발
- ② 12가지 미니게임 기획 후 개발
- ③ 1개의 게임에 코딩교육 요소를 삽입.





```
public IEnumerator GameScript () {
    TryAgain ();
    while (!isGameEnd) {
        {
            #region isSelectable이 활성화 되었을 때만 선택하도록 함
            if (isSelectable && FindObjectOfType<ControllerModeChanger> ().RightControllerMode != (int) ControllerModeChanger.ControllerMode.OptionMode &&
                FindObjectOfType<ControllerModeChanger> ().RightControllerMode != (int) ControllerModeChanger.ControllerMode.GamePointerTriggerMode) {
                FindObjectOfType<ControllerModeChanger> ().ChangeRightControllerMode ((int) ControllerModeChanger.ControllerMode.GamePointerTriggerMode);
                FindObjectOfType<ControllerModeChanger> ().ChageTriggerToolTipName ("선택하기");
            } else if (!isSelectable && FindObjectOfType<ControllerModeChanger> ().RightControllerMode != (int) ControllerModeChanger.ControllerMode.Optic
                FindObjectOfType<ControllerModeChanger> ().RightControllerMode != (int) ControllerModeChanger.ControllerMode.ListenMode) {
                FindObjectOfType<ControllerModeChanger> ().ChangeRightControllerMode ((int) ControllerModeChanger.ControllerMode.ListenMode);
            }
        }
        #endregion

        #region 캐릭터 스케일조정 (숨김)
        if (!isSelectable || Cow1.GetComponent<Obj_Cow> ().isMove || Cow2.GetComponent<Obj_Cow> ().isMove || Wolf.GetComponent<Obj_Cow> ().isMove) {
            if (!Cow1.GetComponent<Obj_Cow> ().isMove) { Cow1.transform.localScale = Vector3.zero; } else { Cow1.transform.localScale = Vector3.one *
            if (!Cow2.GetComponent<Obj_Cow> ().isMove) { Cow2.transform.localScale = Vector3.zero; } else { Cow2.transform.localScale = Vector3.one *
        }
    }
}
```

```
//Trigger Pressed
if (VRTK_SDK_Bridge.GetControllerButtonState(SDK_BaseController.ButtonTypes.Trigger, SDK_BaseController.ButtonPressTypes.PressDown, controllerRe
{
    #region 트리거 버튼 누름
    if(FindObjectOfType<ControllerModeChanger>().RightControllerMode == (int)ControllerModeChanger.ControllerMode.SceneEndMode)
    {
        FindObjectOfType<SceneChpaterManager>().NextScene(SceneManager.GetActiveScene().buildIndex+1);
    }
    if(FindObjectOfType<ControllerModeChanger>().RightControllerMode == (int)ControllerModeChanger.ControllerMode.GameStartMode)
    {
        FindObjectOfType<SceneChpaterManager>().SceneChapterStart();
    }
    if(FindObjectOfType<ControllerModeChanger>().RightControllerMode == (int)ControllerModeChanger.ControllerMode.GameOnlyTriggerMode
    || FindObjectOfType<ControllerModeChanger>().RightControllerMode == (int)ControllerModeChanger.ControllerMode.GamePointerTriggerMode)
    {
        FindObjectOfType<basic_NarrationScript>().GameTriggerEvent();
    }
    OnTriggerPressed(SetControllerEvent(ref triggerPressed, true, currentTriggerAxis.x));
    #endregion
}
```

- VR 콘텐츠를 쉽게 파악할 수 있도록 컨트롤 가이드 및 이야기 선택 scene을 개발하였습니다.
- 모든 VR기기에 연동되도록 VRTK_API, STEAM_VR 등을 활용하였습니다.

```

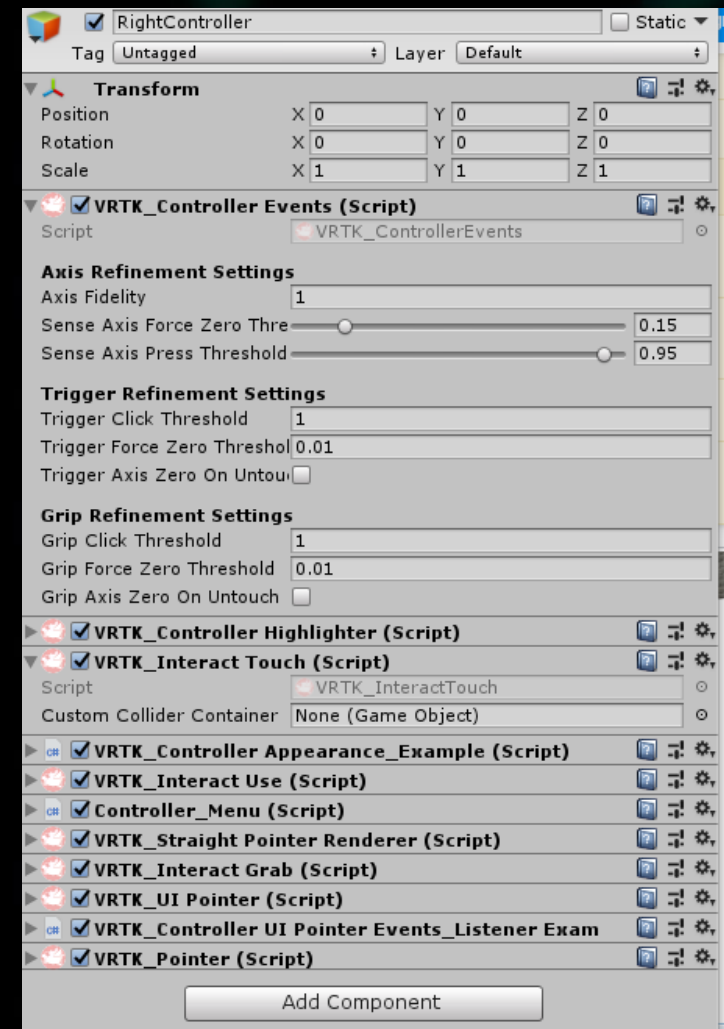
public class ControllerModeChanger : MonoBehaviour {
    #region 연결된 Controller의 Mode
    public enum ControllerMode {
        DefalutSelectionMode,
        ListenMode, //애니메이션 실행 중
        OptionMode, //옵션창으로 갔을 때
        SceneEndMode, //씬이 끝났을 때
        GameStartMode, //미니게임이 시작될 때
        GameOnlyTriggerMode, //트리거만 사용하는 모드에서 사용
        GamePointerTriggerMode, //포인터와 트리거만 사용하는 모드에서 사용
    }
    #endregion

    [HideInInspector] public GameObject rightController = null; //연결된 RightController
    [HideInInspector] public int currentRightControllerMode = 0; //연결된 RightController의 예전 Mode
    [HideInInspector] public int RightControllerMode = 0; //연결된 RightController 현재 Mode
    private bool settingControllerModeTrigger = false; //연결된 RightController의 Mode를 켜 시작 시 설정하기위한 bool

    void Update () {
        if (VRTK_DeviceFinder.GetControllerRightHand () != null) {
            if (rightController == null) {
                rightController = VRTK_DeviceFinder.GetControllerRightHand ();
            }
            #region 미니게임 켜 시작시 RightController의 Mode를 GameStartMode로 설정 : 기본 DefaultMode
            if (FindObjectOfType<basic_NarrationScript> ().isMiniGameScene && !settingControllerModeTrigger) {
                ChangeRightControllerMode ((int) ControllerMode.GameStartMode);
            }
            #endregion
        }
    }

    #region RightController의 Mode를 바꿈
    public void ChangeRightControllerMode (int modeNumber) {
        if (rightController != null)
            if (rightController.GetComponent<VRTK_ControllerEvents> () != null) {
                currentRightControllerMode = RightControllerMode;
                RightControllerMode = modeNumber;
                rightController.GetComponent<VRTK_Pointer> ().enabled = true;
                rightController.transform.Find ("ControllerToolTips").GetComponent<VRTK_ControllerToolTips> ().ResetTooltip ();
                rightController.GetComponent<VRTK_ControllerHighlighter> ().UnhighlightController ();
                switch (modeNumber) {
                    case (int) ControllerMode.DefalutSelectionMode:
                        rightController.transform.Find ("ControllerToolTips").GetComponent<VRTK_ControllerToolTips> ().UpdateText (VRTK_ControllerToolTips.DefaultText);
                        rightController.transform.Find ("ControllerToolTips").GetComponent<VRTK_ControllerToolTips> ().UpdateText (VRTK_ControllerToolTips.DefaultText);
                }
            }
    }
    #endregion
}

```



- VRTK_API를 활용하여 각 Scene마다 요구하는 컨트롤 기능을 달리하여 switch()하였습니다.



- ▼ Projects
- ▶ 1.TheAntsAndTheGrasshopper
 - ▶ 2.HideAndSeek
 - ▶ 3.TheGreedyDog
 - ▶ 4.TheVeryTimidRabbitAndTheFrogs
 - ▶ 5.FishingTigerWithTail
 - ▶ 6.TidyUpMyRoom
 - ▶ 7.IWillHelpYou
 - ▶ 8.SeesawGame
 - ▶ 9.TheSnail'sHome
 - ▶ 10.ThreeLittleCow
 - ▶ 11.TwoStubbornBillyGoats
 - ▶ 12.Frog

2019 하계인턴십 VR작업 내용

기간 : 2019.06.25 ~ 2019.08.23

유니티 버전 : Unity 2018.2.9f1 (64-bit)

테스트 기기 : Windows Mixed MR (Samsung VR Odyssey+)

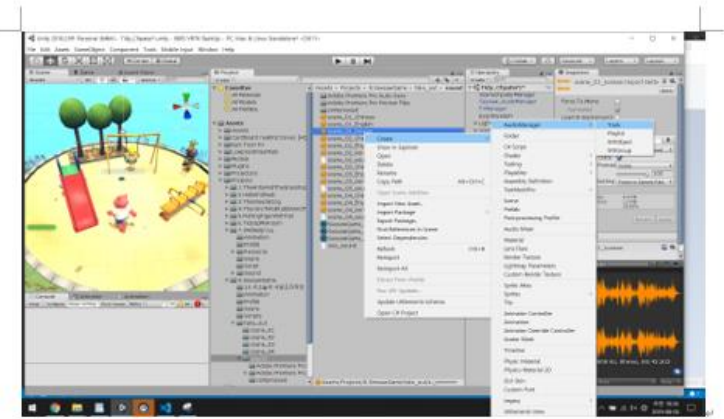
사용한 에셋 : VRTK, Master, Deformer2 (머쉬변형 툴 -> 기차게임에서 사용), Post-Processing (Vignette효과, Color-Grading효과), Audio Manager Pro (오디오소스들 코드로 재생), 이외는 웨이더, 메테리얼, 모델한 임포트

(*기존에 작업들을 이어서 작업하기가 까다로워 처음부터 다시 기획하였음)

*작업내용

- VRTK-Master 에셋 활용
- 이야기 처음(첫 씬) ~ 끝(엔드 씬)까지 프레임워크 개발 및 적용
- 모든 이야기를 한 앱에서 접근할 수 있도록 UI기획
- 애니메이션과 3개국어 나레이션 싱크 적용 후 검토
- AR전용 애니메이션을 VR컨텐츠로 전환하기위해 촬영 및 재디자인
- 마이크로캡류팅 Unity계정으로 구입한 헬리오스VR템 Assets를 통해 360도 모바일VR애니메이션 제작
- 이야기 마다 미니게임 1개씩 적용

1.개미와배짱이	<ul style="list-style-type: none"> 포도를 받아라 실종 나무 마루고 떨어지는 포도를 6개 통해 받으세요
2.꼭꼭숨어라	<ul style="list-style-type: none"> 통 속에 숨은 개를 찾아라 실종 무지개와 하나의 통해 개가 숨고, 통들은 빠르게 섞이게 됩니다 3개의 통 중 하나의 숨은 개를 찾으세요
3.목심쟁이강아지	<ul style="list-style-type: none"> 낚시하기 실종 : 잔뜩 속에 있는 물고기가 필요할 때 버릇을 빠르게 움직여라 개이자를 채우세요. 물고기가 볼 속에 있을 때 움직여라 되면 개이자가 떨어지니 조심하세요



①. '8.SeesawGame'폴더 안의 Korean, English, Chinese 나레이션 사운드를 찾아서

오른쪽클릭 > Create > AudioManager > Track을 선택합니다.



- 12가지 게임 중 8가지 게임을 기획, 제작하였습니다.
- 위 내용을 보고서 및 Document를 작성하여 정리하였습니다.

GoogleVR Project



개발 기간 / 환경

- 2019.08 ~ 2019.10 4학년 2학기 과제물
- Unity 2018.2.9f1
- Google CargoBoard

팀원 내 역할

- 3인 중 기획, 클라이언트 프로그래머, FX담당

성과

- Unity Particle System에 전반적인 사용법 학습.
- Google VR Kit 사용법 및 실습.

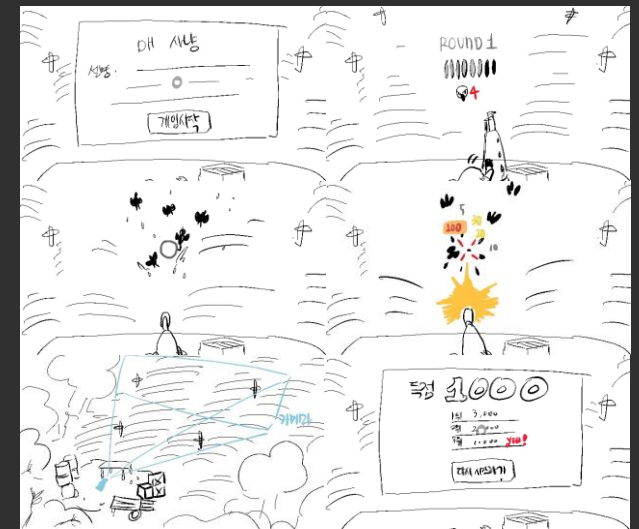


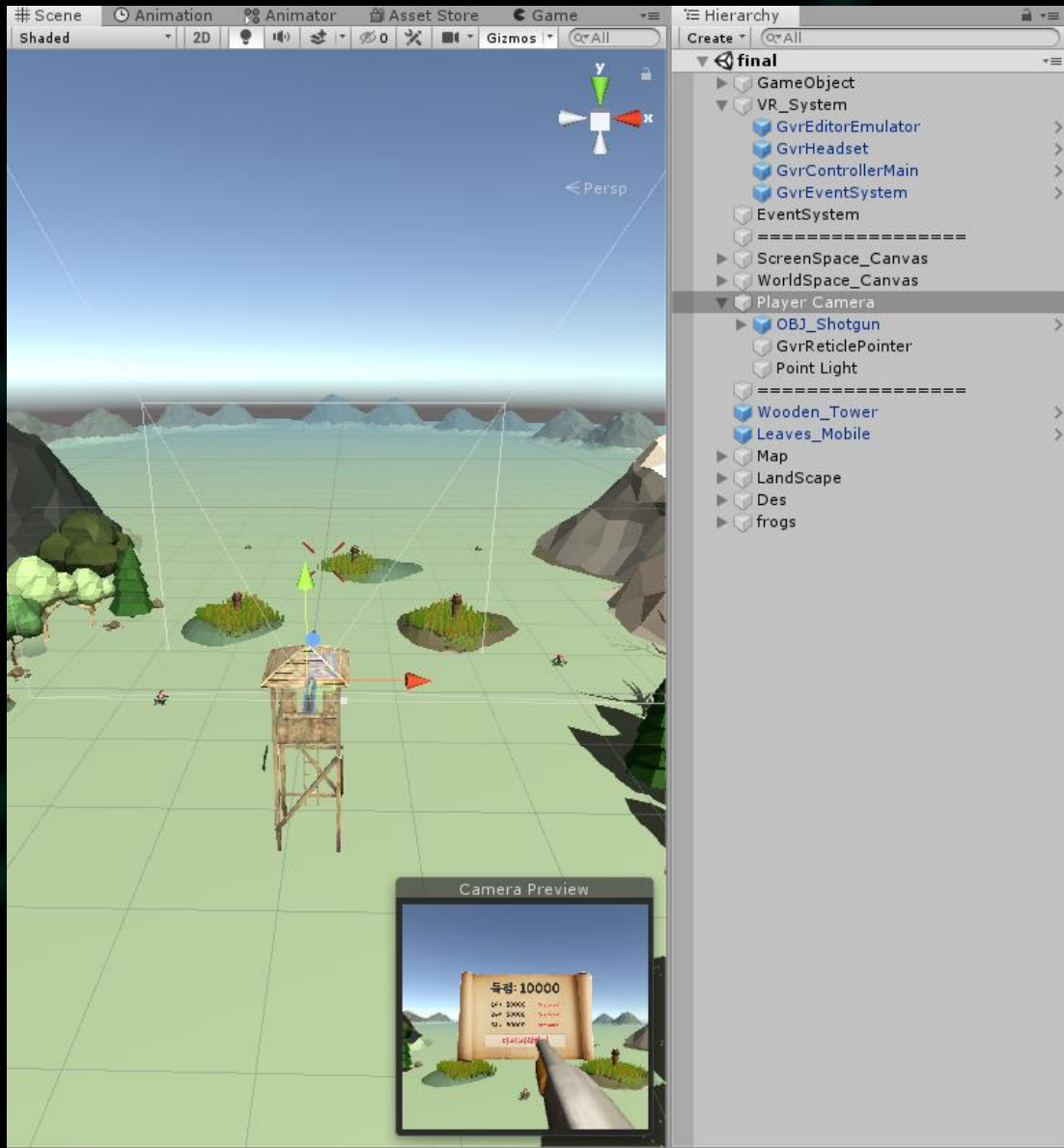
- 게임 요구 사항 -

- ① 모바일에 적합한 UI 및 리소스 사용.
- ② 게임 루프가 간단하도록 기획.
- ③ GoogleVR에 적합한 원터치 게임을 기획



- 러프 스케치 -





```
public override void OnPointerClickDown(GameObject Target = null) //화면 터치 시
{
    bool isBulletAvailable = transform.root.GetComponentInChildren<OBJ_Shotgun>().Shot();
    if (!isBulletAvailable) return;
    m_crosshair.GetComponent<UnityEngine.UI.Image>().color = Color.gray;
    List<Eagle_Prefab> hitedEagle = new List<Eagle_Prefab>();
    for (var i = 0; i < 30; i++)
    {
        var v3Offset = Camera.main.transform.up * Random.Range(10f, 60.0f);
        v3Offset = Quaternion.AngleAxis(Random.Range(0.0f, 360.0f), Camera.main.transform.forward) * v3Offset;
        var v3Hit = Camera.main.transform.forward * 1000 + v3Offset;
        DrawLine(transform.root.GetComponentInChildren<OBJ_Shotgun>().FireOut.position + v3Hit.normalized * 5, v3Hit, Color.yellow);
        RaycastHit hit;
        Ray ray = new Ray(Camera.main.transform.position, v3Hit);
        if (Physics.Raycast(ray, out hit, 1000))
        {
            if (hit.collider != null)
            {
                if (hit.collider.gameObject.CompareTag("Monster"))
                {
                    if (hit.collider.gameObject.GetComponent<Eagle_Prefab>().isAlive)
                    {
                        hit.collider.gameObject.GetComponent<Eagle_Prefab>().totalDamage += 30;
                        hitedEagle.Add(hit.collider.gameObject.GetComponent<Eagle_Prefab>());
                        showCrosshairTime = 0.3f;
                    }
                }
            }
        }
    }
    foreach (var item in hitedEagle)
    {
        if (item.totalDamage >= 40)
        {
            m_crosshair.GetComponent<UnityEngine.UI.Image>().color = Color.yellow;
        }
        int totalDamage = item.totalDamage;
        item.GetShot(totalDamage);
    }
}
```

- Google VR_API를 활용하여 OnePointClick() 컨트롤에 맞춰 개발하였습니다.
- 보다 직관적인 시각효과를 위해, 발사 시 총알궤적 및 히트스캔, 데미지 팝업 등의 기능을 구현하였습니다.

1

GAME

- 모바일게임 프로젝트
- VR / AR 프로젝트

2

UI / UX

- KDB스타트업_UX디자인

3

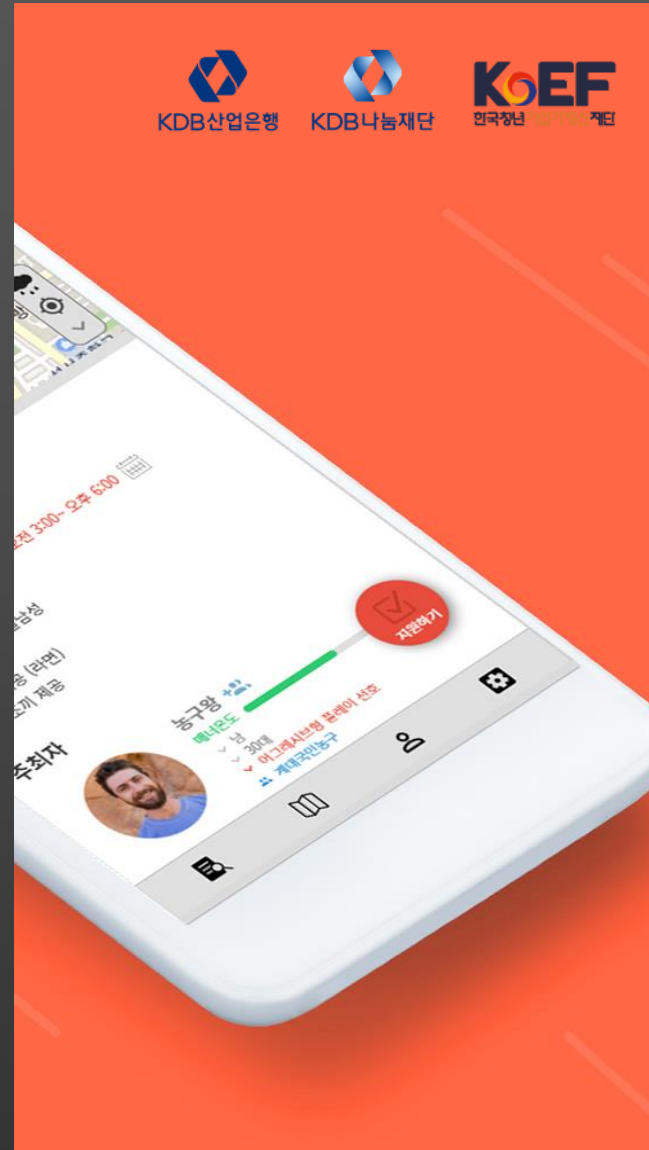
WEB

- Jsp, Spring 웹사이트



실시간으로 지역 정보를 파악하는

종합 스포츠앱



KDB START-UP

개발 기간 / 환경

- 2019.08 ~ 2019.10 4학년 2학기 과제물
- 2019.11.05 KDB STARTUP 2019 [KDB STARTUP 데모데이&창업 교육 프로그램 성과공유회] 참가
- Naver OVEN beta

팀원 내 역할

- 4인 중 기획, 목업 담당

성과

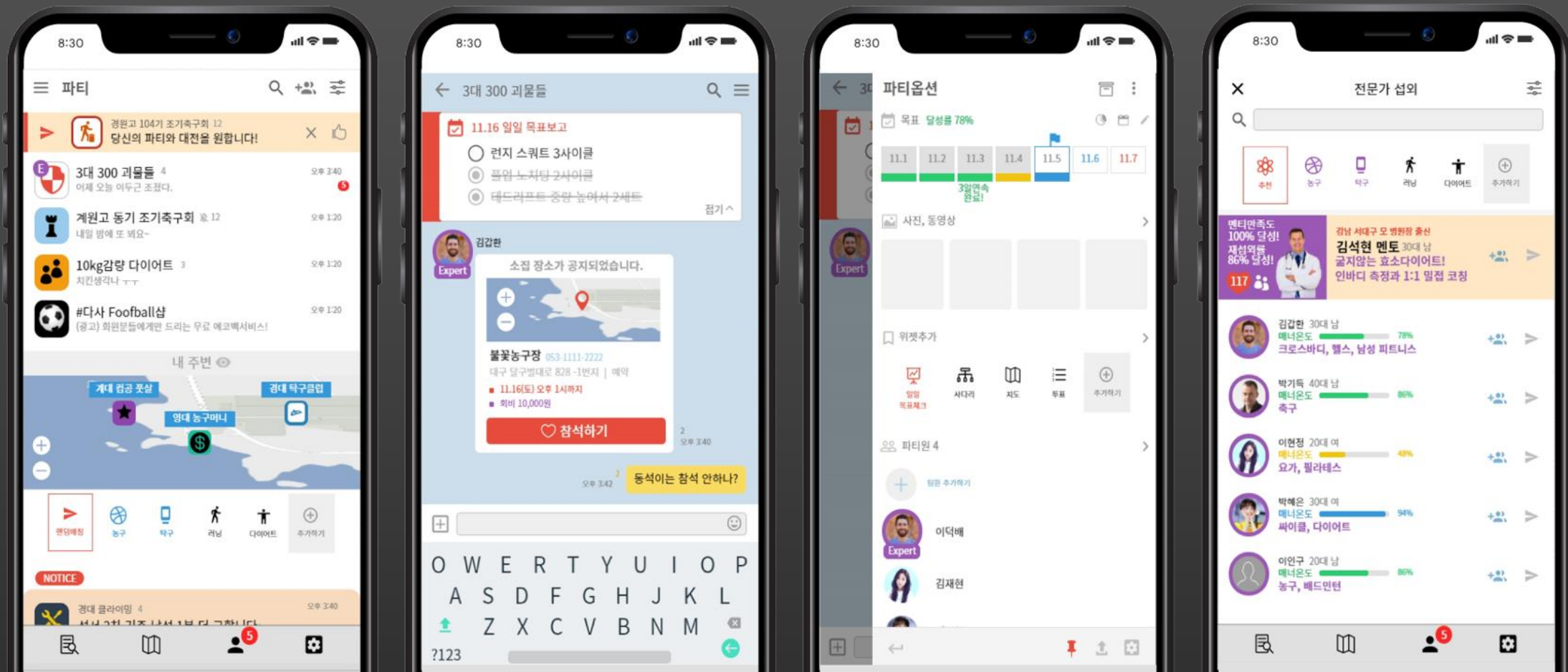
- 실질적인 고객 인터뷰를 통해 숨겨진 핵심가치 도출하였음.
- UX목업 개발법과 최근 디자인 트렌드 등을 학습하였음.



- 활동 과정 -

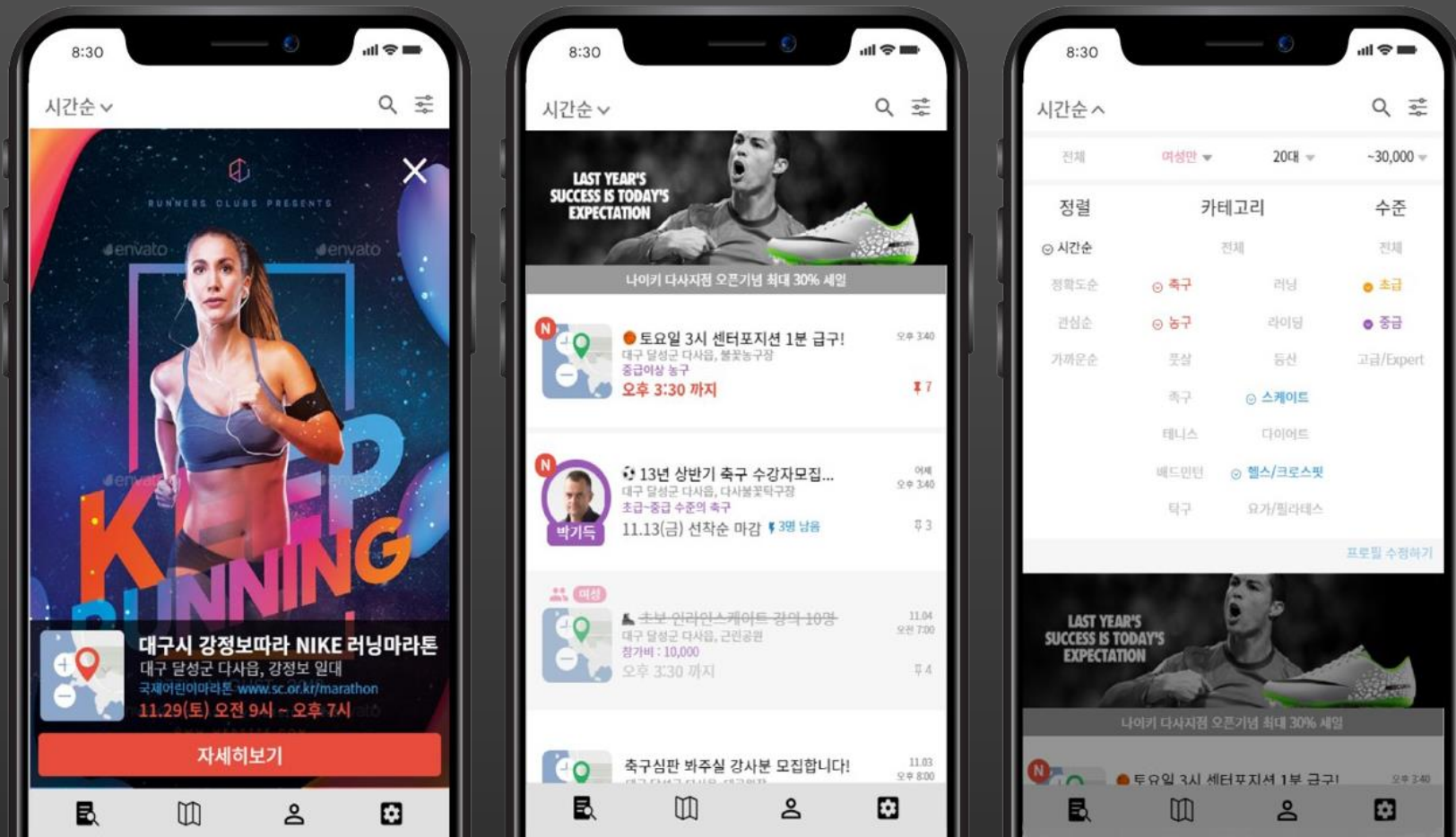
소수 인원의 파티활동

친구 또는 가족끼리, 동네 친구들을 불러 파티를 꾸려보세요!
멘토를 초청하여 함께 목표를 정하고 달성하도록 노력해요!



실시간 올라오는 지역공고

게시판을 확인하여 지역 내에서 진행되는
프로모션 및 공고를 확인하세요!



hypothesis

- ① 동네에서 같이 운동할 친구를 찾았으면 좋겠다.
- ② 이왕이면 실력이 비슷한 상대
- ③ 이왕이면 Pro-tuer선수에게 코칭받고싶다.
- ④ 체육관 대관을 App으로 했으면 좋겠다.



AS-IS

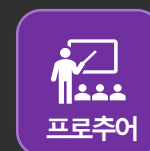
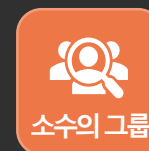
- ① 서울/경기 지역 한정
- ② 커뮤니티 비활성화, 사용성 떨어짐
- ③ 대관 시 대규모 또는 유료시설 한정



- 새로운 가치 제안 도출 -

To-Be

- ① 지역 내 공공 체육시설을 한 눈에 파악
- ② 지역 주민이 직접 게시하는 위키피디아 시스템
- ③ 동네 Pro-tuer선수들에게 저렴하게 코칭받음
- ④ 자기계발을 원하는 소수의 커뮤니티 그룹 활성화



1

GAME

- 모바일게임 프로젝트
- VR / AR 프로젝트

2

UI / UX

- KDB스타트업_UX디자인

3

WEB

- Jsp, Spring 웹사이트

VR / AR PROJECT

VR / AR PROJECT

MOBILE GAME PROJECT

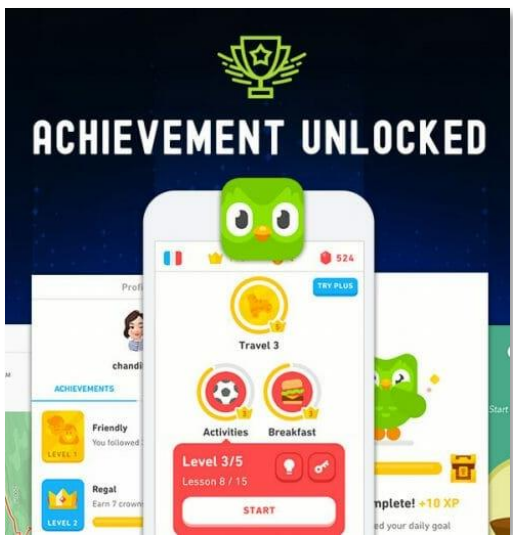
진행중 입니다...

JSP 게시판 만들기 repo : <https://github.com/kim-goro/ShowcasePhonecase>

Thank you

이 제 서 야 웹 분 야 에 뛰 어 들 었 지 만 ...

"여러 분야의 프로젝트를 진행한 덕분에
다양한 경험을 쌓을 수 있었고 이를 바탕
으로 창의적인 발상과 분석력으로 빠르게
적응할 수 있습니다.



g a m i f i c a t i o n

게 임 전 공 학

‘GAMIFICATION’을 기반으로 하여 사용자에게
적절한 보상과 동기부여를 제공하고 지속적인
구독 서비스를 이용할 수 있도록 할 수 있습니다.

T r e n d L e a d e r

‘밀레니엄 세대’

저는 현세의 트렌드를 이끄는 밀레니엄 세대로,
유행과 이슈에 민감하고 사용자의 니즈를 빠르
게 캐치해 낼 수 있으며 포토샵 및 일러스트를
통해 구체적인 시안을 제시할 수 있습니다.

