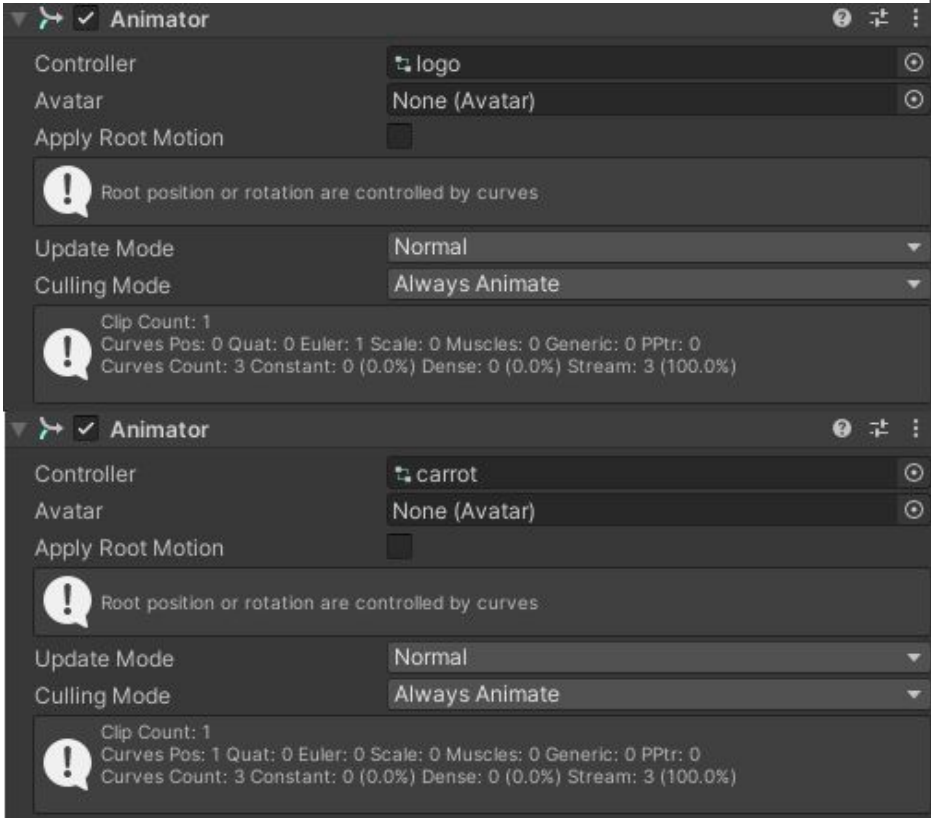
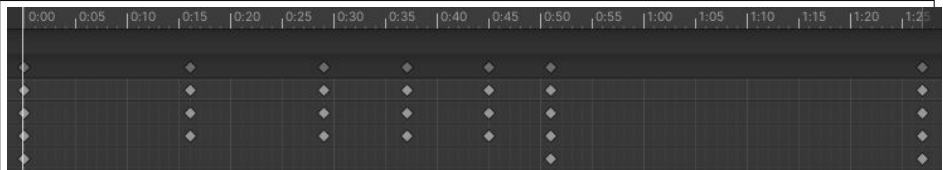


■ 서식 2 : 결과보고서 요약본

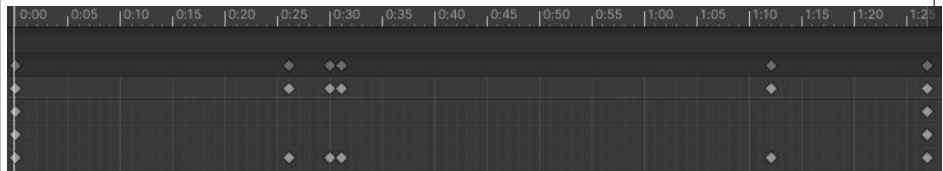
## Project Based Learning 결과보고서 요약본

교과목명	국문	게임엔진프로그래밍			
	영문	GameEngine Programming			
PBL 관련 능력단위 (능력단위코드)	PBL 관련 능력단위요소 (능력단위요소코드)				
게임엔진응용프로그래밍 (0803020527_16v3)	상용게임엔진응용프로그램 구현하기 (0803020527_16v3.2)				
학년 반	2학년 1반	조원	임하연, 전가은, 변규태		
프로젝트 주제	클리커 게임 만들기				
지도교수	이성현	산업체 참가여부	<input type="checkbox"/> 유 <input checked="" type="checkbox"/> 무		
참여학생	임하연(202127032), 전가은(202127002), 변규태(202127004)				
작품 개요 (주제 선정 이유)	아이디어 평가기준	창의성	30		
		제작용이성	40		
		협업가능성	30		
	논의된 아이디어	아이디어 주제			
		당근 농장			
		풍선 터뜨리기			
광물 캐기					
아이디어 평가 (팀원들 평가점수 평균값)	주제	창의성	제작용이성	협업가능성	
	당근 농장	30	35	30	
	풍선 터뜨리기	20	30	20	
	광물 캐기	25	25	30	
선정된 아이디어	당근 농장				
<p>달에 불시착한 토끼(플레이어)가 달나라 토끼들에게 당근의 맛을 알려주기 위해 공장을 차린다는 독창적인 세계관과 스토리로 플레이어에게 동기를 준다. 이러한 귀여운 설정이라는 점에서 시선을 사로잡을 수 있는 아기자기한 비주얼의 게임을 만들 수 있을 것 같아 선정하게 되었다.</p>					
작품 구조도 (문제점 제시 및 개선방안)	<pre> graph LR     A[스플래시 스크린] -- 2프루 --&gt; B[타이틀]     B -- 엔터 키를 누를 때 --&gt; C[메인]     C -- ↑ 키를 누를 때 --&gt; D[월드맵]     C -- ↓ 키를 누를 때 --&gt; E[토끼 집 버튼]     C -- 스페이스바를 누를 때 --&gt; F[패시브]     D -- 토끼 집 버튼을 누를 때 --&gt; G[토끼 &lt;br/&gt; 당근주기]     G -- 토끼 진영도를 만화점 --&gt; H[토끼 진영도를 만화점]     </pre>				

	<p>인트로 화면에서 로고를 보여준 뒤 2초가 지나면 타이틀 화면으로 넘어간다. 타이틀 화면에서 화면을 클릭하면 카메라가 아래로 내려가며 메인 게임의 화면이 보이게 된다.</p> <p>메인 게임 화면에서 하단에 있는 맵 버튼을 누르게 되면 맵 창이 뜬다. 만약 맵 창에 있는 각각의 집 버튼을 누르면 해당하는 토끼의 모습이 뜬다. “당근 주기” 라고 쓰여있는 버튼을 눌러 NPC에게 아이템을 줄 수 있다.</p> <p>메인 게임 화면에서 우측 상단에 있는 패시브 버튼을 누르게 되면 보유하고 있는 패시브가 나열되어있는 창이 뜬다. 각 NPC들에게 아이템을 주고, 최대치가 충족 될 때마다 패시브 창에 패시브가 추가된다.</p> <p>모든 창 화면에는 “x”버튼이 있고, 버튼을 누르게 되면 창이 닫히고 메인 게임 화면이 보이게 된다.</p>
<p>관련 이론</p>	<p>[Title]</p> <p>1. 로고 애니메이션</p>  <p>오브젝트 logo와 carrot에 애니메이터를 넣어서 구현</p>



carrot의 애니메이터에 들어간 Animation은 총 7개의 키로 구성



logo의 애니메이터에 들어간 Animation은 총 6개의 키로 구성

## 2. 씬 로드

```
using UnityEngine.SceneManagement;
```

씬 로드를 위해 UnityEngine.SceneManagement를 불러옴

```
Unity 메시지 | 참조 0개
private void Update()
{
    S_time += Time.deltaTime;
    if (S_time > 2)
        SceneManager.LoadScene(1);
}
```

2초를 기다린 후 SceneManager.LoadScene을 통하여 Main 씬을 로드

[Main]

<메인 화면>

### 1. 게임 시작

```
Unity 메시지 | 참조 0개
private void Start()
{
    StartCoroutine(StartGame());
}
```

start\_game은 씬이 로드되자마자 StartGame 코루틴을 실행

```
yield return new WaitForSeconds(2);
```

마우스를 클릭할 때까지 기다린 후

```
factory.SetCollider(false);
Destroy(Intro_game_Text);
```

공장의 콜라이더의 크기를 0으로 만든 후 인트로 텍스트 캔버스를 파괴함

```
while (transform.position.y > 0)
{
    transform.position = Vector3.Lerp(transform.position, Target, speed * Time.deltaTime);
    yield return null;
}
```

이후 Main Camera의 y가 0이 될 때까지 매 프레임마다 Main

Camera를 (0, -2, -10)을 목적으로 이동시킴

```
transform.position = Goal;
Main_game.SetActive(true);
Destroy(Intro_game);
factory.SetCollider(true);
gm.enabled = true;
Destroy(GetComponent<start_game>());
```

이후 Main Camera의 위치를 확정시키고 메인 UI를 보여주고 인트로 화면을 파괴한 뒤 공장의 콜라이더를 원상복구 시키고 게임매니저 오브젝트를 활성화하고 이걸 실행시킨 스크립트를 파괴함

## 2. 공장

```
Unity 메시지 |참조 0개
private void OnMouseDown()
{
    gm.FactoryClick();
    Ani.enabled = true;
    Ani.Reseter();
}
참조 1개
public void FactoryClick()
{
    Cool_time -= 1;
    PlaySound(Sounds.se_factory, Channels.SE);
}
```

클릭할 때마다 팩토리 애니메이션 스크립트를 활성화하고 쿨타임을 1 감소시킴

## 3. 당근 소지량

```
참조 4개
public int Carrot
{
    get => carrot;
    set
    {
        scoretext.text = value.ToString();
        carrot = value;
    }
}
```

당근을 얻거나 잃을 때마다 당근 소지량 텍스트를 갱신함

#### 4. 쿨타임

```
CoolDesc = new string[(int)Cm.Ceil(cool_time_seconds)];  
for (int i = 0; i < CoolDesc.Length; i++)  
    CoolDesc[i] = Cm.SbConcat(new string[] { "당근 재배까지 ", (i + 1).ToString(), "초" });
```

참조 5개

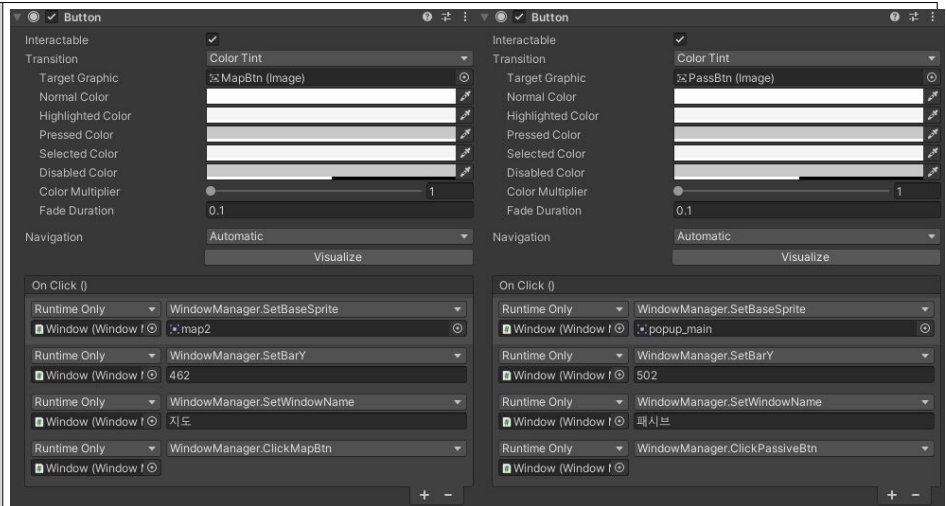
```
public float Cool_time  
{  
    get => cool_time;  
    set  
    {  
        float instant = Cm.Ceil(value);  
        if (instant != Count_Cool)  
        {  
            cooltimetext.text = CoolDesc[(int)instant - 1];  
            Count_Cool = instant;  
        }  
        cool_time = value;  
    }  
}
```

초기 쿨타임만큼 문자열을 생성한 이후 현재 출력된 텍스트의 쿨타임과 실제 쿨타임의 올림값이 다르면 텍스트를 갱신함

```
private void Update()  
{  
    Cool_time -= Time.deltaTime;  
    if (Cool_time <= 0)  
    {  
        Carrot += get_carrot;  
        Cool_time = cool_time_seconds;  
        newcarrot.SetActive(true);  
        PlaySound(Sounds.se_getCarrot, Channels.SE);  
    }  
}
```

또한 쿨타임이 0이 되면 당근 획득량만큼 당근을 획득하고 유니티 자체 이펙트 기능으로 만든 이펙트를 활성화함

#### 5. 메인 UI 버튼



Button 컴포넌트로 구현했음

onclick이 작동되면 윈도우 베이스 스프라이트와 윈도우 정보 위치와 윈도우 이름을 갱신함

참조 2개

```
private void OpenWindow( )
{
    blackScreen.SetActive(true);
    iTween.MoveTo(element, HashAdj_Y(WindowShow, 0));
    mainButtons.SetActive(false);
    factory.SetCollider(false);
    rabbitWindow.SetActive(false);
    gm.PlaySound(GameManager.Sounds.se_btnclick, GameManager.Channels.SE);
}
```

참조 0개

```
public void ClickMapBtn( )
{
    OpenWindow();
    worldWindow.SetActive(true);
    passiveWindow.SetActive(false);
}
```

참조 0개

```
public void ClickPassiveBtn( )
{
    OpenWindow();
    worldWindow.SetActive(false);
    passiveWindow.SetActive(true);
}
```

이후 블랙스크린을 켜고 메인 UI 버튼들을 숨기고 공장의 콜라이더의 크기를 0으로 만들고 창을 Anchored Position 기준 (0, 0)으로 이동 시킴

6. 주인공 지구 토끼

참조 1개

```

public void Counter()
{
    Count++;
    if (Count == cm.SizeOfRabbit)
    {
        GetComponent<Animator>().SetTrigger("Dance");
        gm.PlaySound(GameManager.Sounds.clear_bgm, GameManager.Channels.BGM);
        Destroy(GetComponent<player_dance>());
    }
}

```

Layers Parameters Base Layer

Name +

Dance

Any State

Entry

Player

Player\_Animation

Conditions

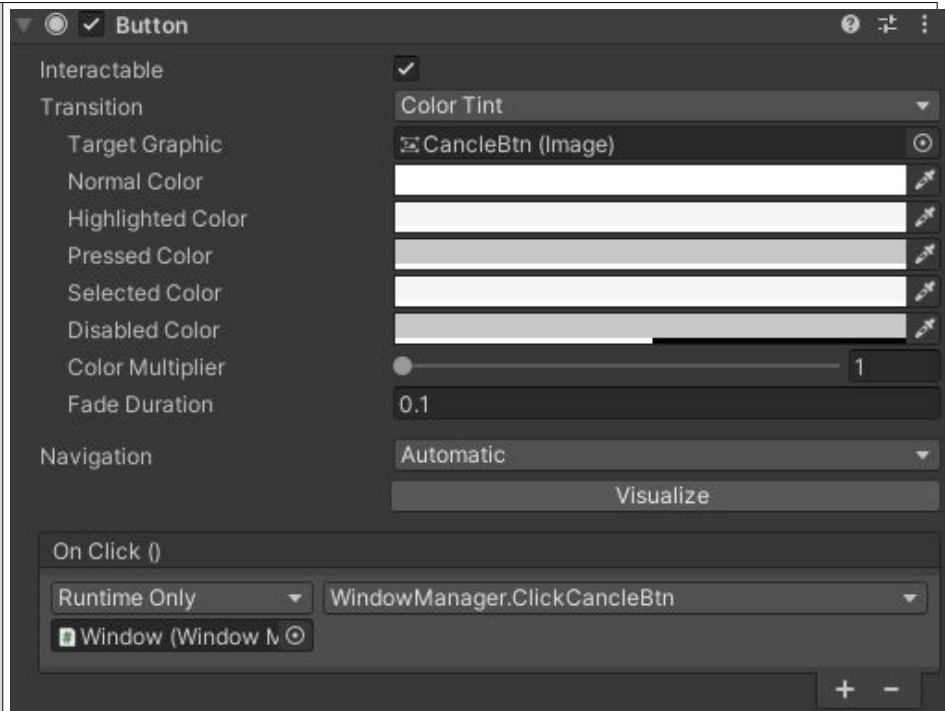
= Dance true

+ -

처음에는 아무 것도 하지 않지만 플레이어 댄스 카운터가 5가 되면 애니메이터의 Dance 트리거를 작동시켜 반복되는 댄스 애니메이션을 실행시키고 BGM을 바꾼 뒤 이를 실행시킨 스크립트를 파괴함

<윈도우>

1. X 버튼



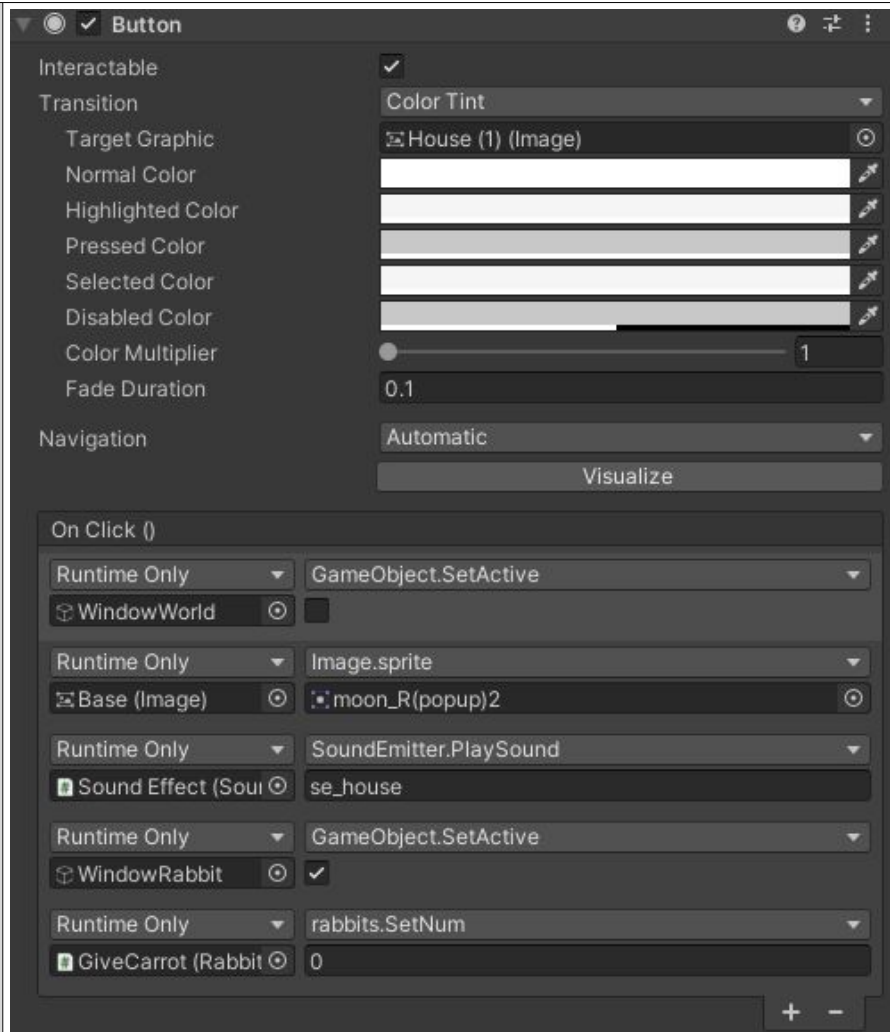
Button 컴포넌트로 구현했음

```
참조 0개
public void ClickCancleBtn()
{
    gm.PlaySound(GameManager.Sounds.se_btnclose, GameManager.Channels.SE);
    iTween.MoveTo(element, HashAdj_Y(WindowHide, 1800));
}
WindowHide = iTween.Hash("delay", 0.01f, "time", 0.2f,
    "oncomplete", nameof(ResetWindow), "oncompletetarget", gameObject);
참조 1개
private void ResetWindow()
{
    elementTf.anchoredPosition = InitPos;
    blackScreen.SetActive(false);
    factory.SetCollider(true);
    mainButtons.SetActive(true);
}
```

onclick이 작동되면 창을 Anchored Position 기준 (0, 1800)으로 이동시킨 이후 창을 원래 위치로 돌려놓고 블랙스크린을 끄고 공장의 콜라이더를 원상복구하고 메인 UI 버튼을 보여줌

2. 집 버튼





Button 컴포넌트로 구현했음

onclick이 작동되면 맵 UI를 비활성화하고 윈도우 베이스 스프라이트와 윈도우 이름을 갱신함

```
참조 0개
public void SetNum(int num)
{
    wm.SetWindowName_Rabbit(num);
    if (Num != num)
    {
        Num = num;
        wm.LoadRabbitInfo(Num);
        gm.SetWindowUI(Num);
    }
}
```

```
참조 2개
public void SetWindowUI(int num)
{
    wm.SetProgressImage(Level[num]);
    wm.SetGiveButton_Interactable(Level[num] != 10);
}
```

이후 토끼 이미지와 진행도 이미지와 당근 주기 버튼과 토끼 UI를 활성화함

3. 당근 주기 버튼

```
참조 0개
public void clicked()
{
    gm.GiveCarrot(Num);
}
```

```
참조 1개
public void GiveCarrot(int num)
{
    if (Carrot >= NeedCarrot[num])
    {
        Carrot -= NeedCarrot[num];
        Level[num]++;
        SetWindowUI(num);
        PlaySound(Sounds.se_giveCarrot, Channels.SE);
    }
}
```

취야하는 당근보다 가지고 있는 당근이 더 많다면 취야하는 당근만큼 가지고 있는 당근을 차감하고 진행도 이미지와 당근 주기 버튼을 갱신한 후 그 토끼의 진행도 레벨을 올림

```
if (gm.MaxLevel(Num))
{
    RabbitsHelp[Num].SetActive(true);
    wm.PassiveDescActive(Num);
    player.Counter();
    passBtn.onClick.Invoke();
    gm.RabbitResult(Num);
    gm.PlaySound(GameManager.Sounds.se_fullCarrot, GameManager.Channels.SE);
}
```

만약 10 레벨이라면 메인 화면의 응원하는 토끼를 활성화하고 패시브 설명을 활성화하고 플레이어 댄스 카운트를 1 상승시키고 패시브 버튼을 작동시켜 패시브 창을 열고 패시브를 적용함

결과물 제작  
(문제점 개선사항)

[인트로] - 스플래시 스크린 화면

1. 화면 출력물



▲ 게임을 시작했을 때 먼저 출력되는 인트로 화면

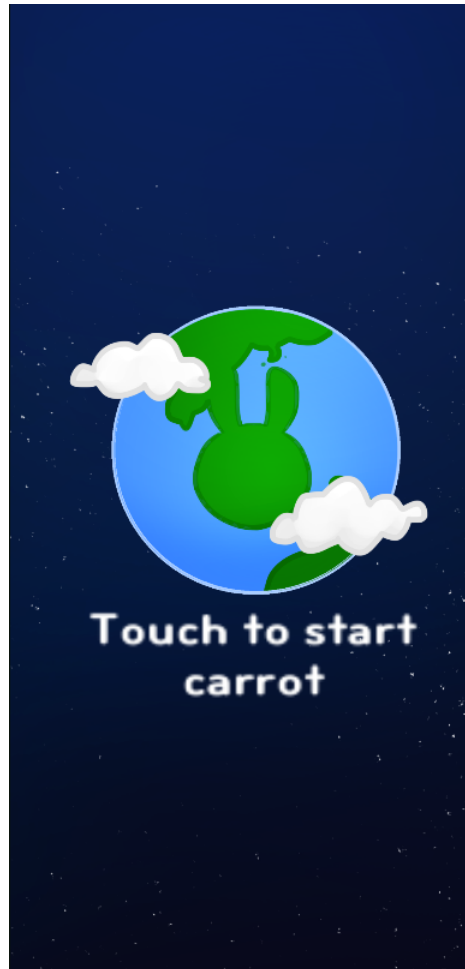
## 2. 애니메이션



▲ 일정 시간이 지나면 오브젝트가 떨어지고 게임 로고가 흔들리는 애니메이션이 실행됨

[인트로] - 메인 화면

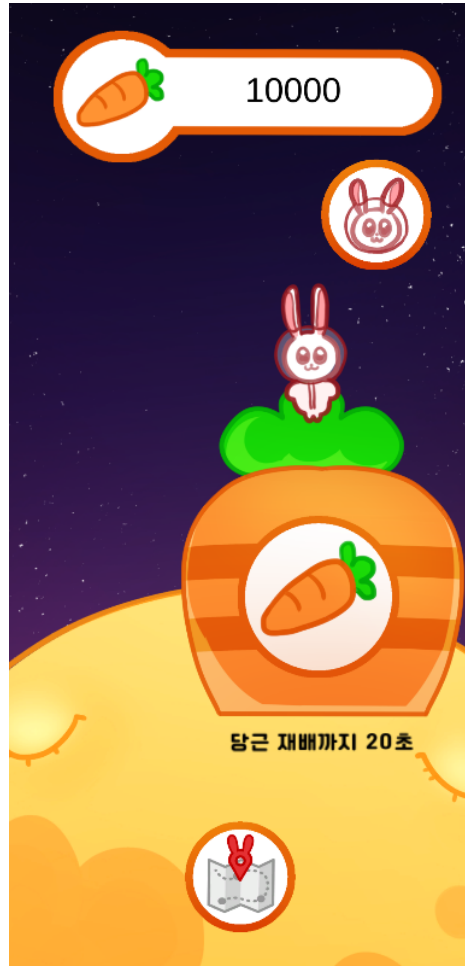
### 1. 화면 출력물



▲ 화면을 터치하면 카메라가 아래로 내려가 [게임 화면]-메인 화면으로 이동함

#### [게임 화면] - 메인 화면

##### 1. 플레이 화면



▲ 메인 게임 화면. 플레이어, 당근 공장, 당근 생성까지의 쿨타임, 획득한 당근의 수, 지도 버튼, 패시트 버튼을 보여줌

## 2. 획득한 당근의 수



▲ 획득한 당근이 수를 보여주고 당근을 얻을 때 마다 갱신됨

## 3. 공장 터치



▲ 공장 터치 시 공장이 커졌다가 작아지는 모션이 실행됨.

### 3. 지도 버튼 / 패시브 버튼

[게임 화면] - 특수 창 화면(지도)

#### 1. 지도 창 화면



▲ 떡토끼들의 집, 지도 창을 닫는 버튼을 보여줌

#### 2. 떡 토끼들의 집



▲ 클릭하면 해당 집에 사는 떡토끼들의 창으로 이동함

### 3. 닫기 버튼



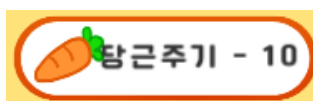
▲ 클릭 시 해당 창(지도)를 닫음

[게임 화면] - 특수 창 화면(떡토끼)



▲ 떡토끼, 당근 주기 버튼, 호감도 바, 패시브 텍스트, 창을 닫는 버튼을 보여줌

### 1. 당근 주기 버튼



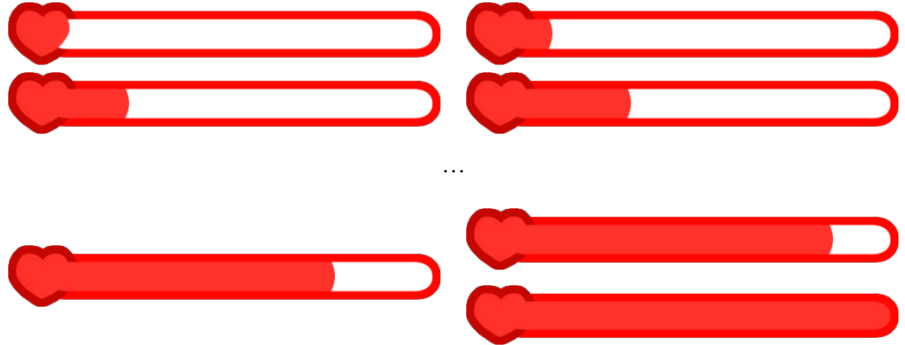
▲ 해당 떡토끼가 요구하는 수의 당근을 줄 수 있음, 클릭 시 요구하는

당근이 수 만큼 획득한 당근의 수를 감소함

## 2. 호감도 바



▲ 당근을 줄 때마다 조금씩 게이지가 상승함(총 10회 상승), 호감도 게이지를 채웠을 경우 해당 떡토끼에 해당하는 패시브를 획득하며 [게임 화면] - 특수 창 화면(패시브 화면)으로 이동함



▲ 게이지가 상승하는 모습

## 3. 닫기 버튼



▲ 클릭 시 해당 창(떡토끼)를 닫음

[게임 화면] - 특수 창 화면(패시브 화면)

### 1. 패시브 창 화면






## 2. 획득 패시브



▲ 획득한 패시브를 보여줌

## 3. 달기 버튼

	 <p>▲ 클릭 시 해당 창(패시브 화면)를 닫음</p>
기대 효과 및 활용방안	<p>이번 팀 프로젝트 과제를 진행하며 특정 주제를 가지고 팀원과 적절한 게임을 만들기 위해 상의하는 과정에서 한 주제로 다양한 게임을 구상할 수 있다는 것을 경험하였다.</p> <p>작년에 했던 팀 프로젝트를 경험 삼아 각자 할 일을 분배했고, 저번과 같은 팀원과 프로젝트를 진행하며 전에 했던 실수를 보완하기 위해 노력하는 과정을 겪으며 한 층 더 성장할 수 있던 기회가 되었다. 각 팀원이 혼자 해결하지 못한 일을 서로 도와주며 협업의 좋은 점을 느꼈다.</p> <p>수업 시간에 예제를 통해 구현했던 기능들을 직접 팀 프로젝트 게임에 구현해보며 수업 시간의 복습을 하는 시간을 갖게 되었고, Unity를 좀 더 능숙하게 다룰 수 있게 되었다. 또한, 세부적으로 일정을 계획하여 차근 차근 진행해가며 일정을 세세하게 짜는 것이 중요함을 느끼게 되었다.</p>
<p>2022년 6월 8일</p> <p style="text-align: right;">지도교수      이 성 현      (인)</p>	