

# Mysterious Forest



숲으로 들어가기

20160241 김미소  
20160247 오은주

# Mysterious Forest\_

# Contents

- 01 ◆ 'Mysterious Forest' 기획
  - 게임 기획 의도
  - 게임 시나리오
  - 게임 컨셉
  - 주요 콘텐츠
- 02 ◆ 'Mysterious Forest' 개발과정
  - 게임 기획
  - 관련 에셋 수집
  - 코딩 작업
  - 실제 구현
- 03 ◆ 'Mysterious Forest' 코딩 및 순서도
  - 코딩
  - 순서도
- 04 ◆ 최종 결과물 및 실제 구현 모습
  - 게임 조작 방법
  - 실제 플레이 화면

# Mysterious Forest\_



게임 기획 의도



게임 시나리오



게임 컨셉



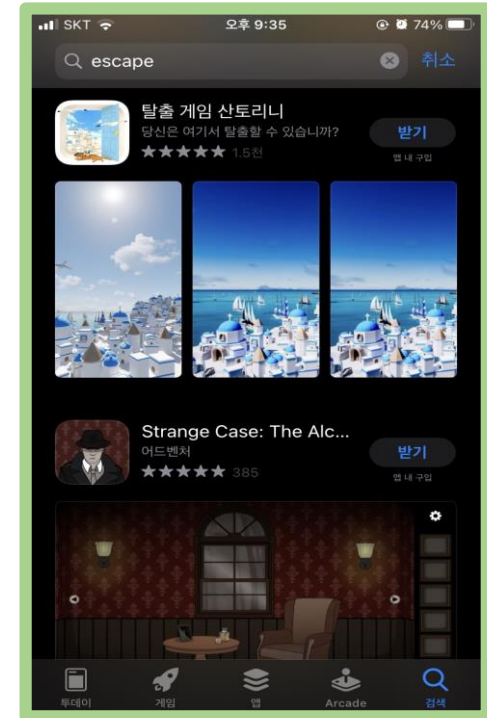
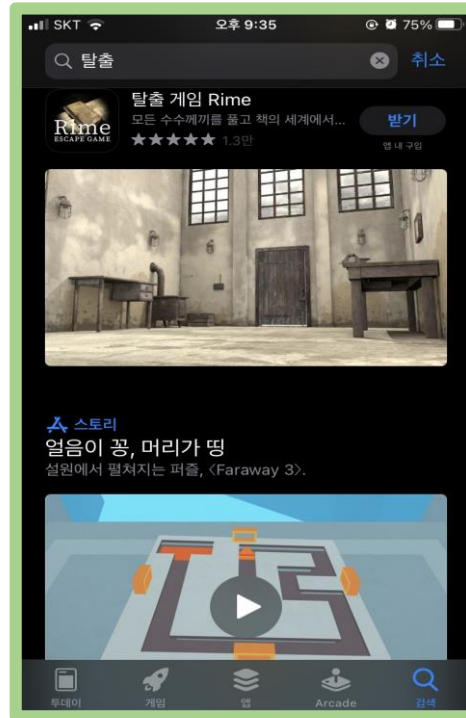
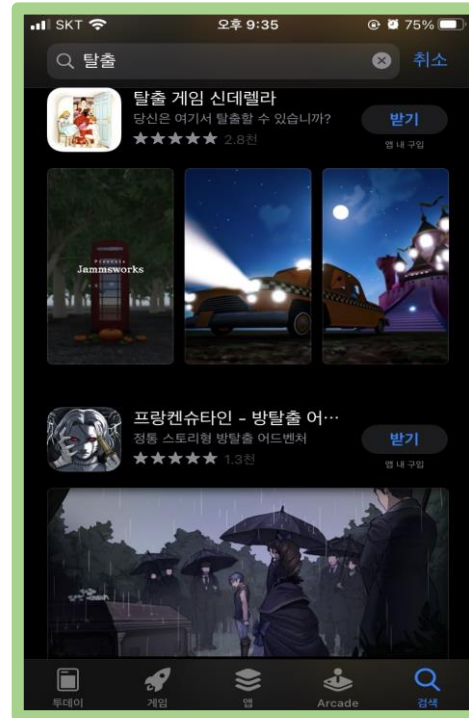
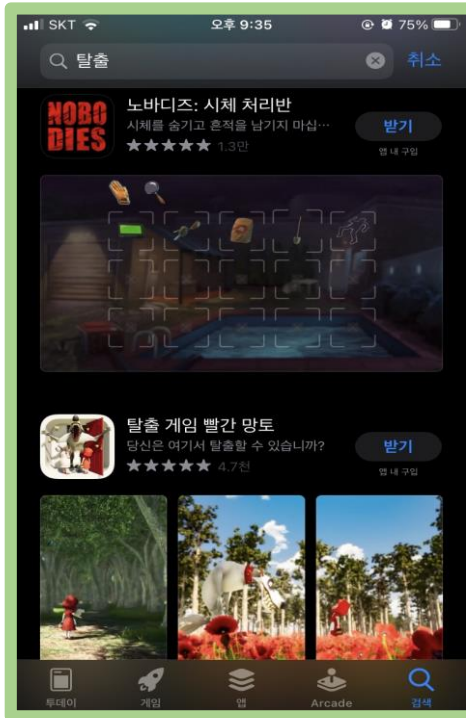
주요 콘텐츠



# Mysterious Forest\_

## 01. 'Mysterious Forest' 기획

- 게임 기획 의도



⌘ 제한된 공간에 갇히고, 문제를 풀어 공간을 탈출하는 일명 '방탈출 게임' 컨셉의 게임이 다수

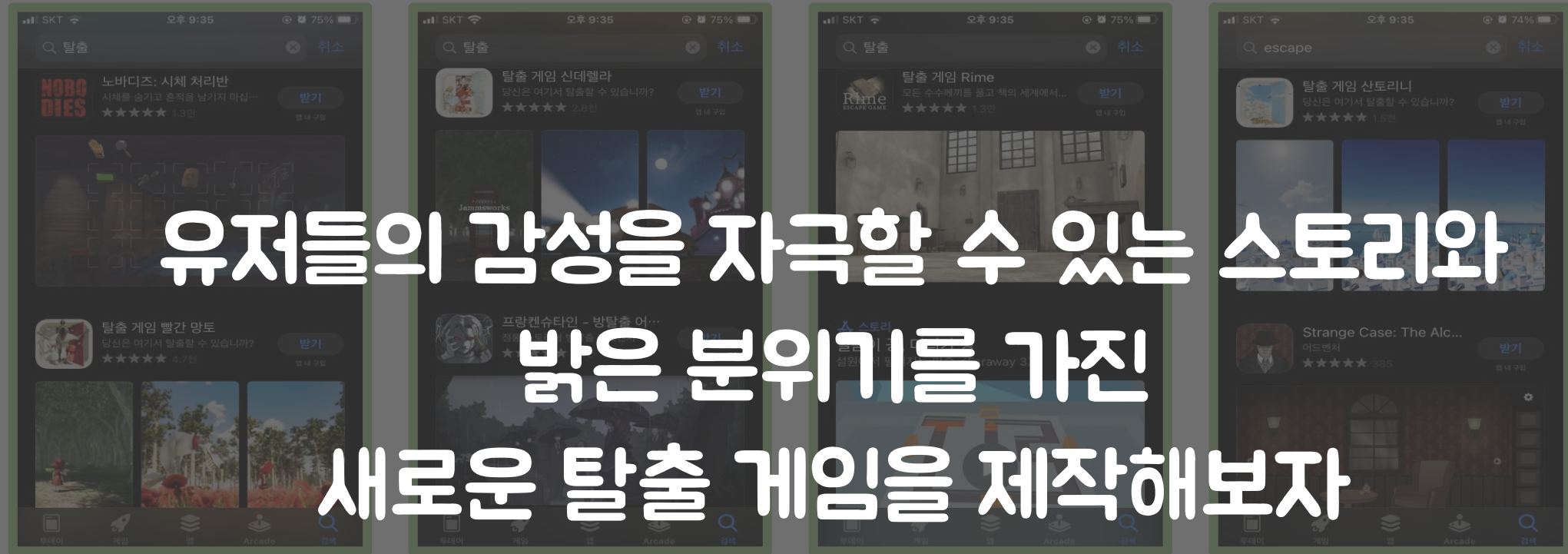
⌘ 전체적인 스토리보다 탈출이라는 목표에 집중

⌘ 외판 섬, 밀폐된 방 등 어두운 공간을 배경 사용 => 생존

# Mysterious Forest\_

## 01. 'Mysterious Forest' 기획

- 게임 기획 의도



& 제한된 공간에 갇히고, 문제를 풀어 공간을 탈출하는 일명 '방탈출 게임' 컨셉의 게임이 다수

& 전체적인 스토리보다 탈출이라는 목표에 집중

& 외판 섬, 밀폐된 방 등 어두운 공간을 배경 사용 => 생존



# Mysterious Forest\_

## 01. 'Mysterious Forest' 기획

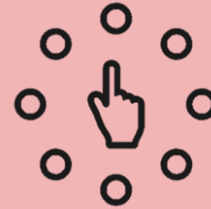
- 게임 기획 의도



유저들의 감성을  
자극하는  
스토리를 가진  
생존 + 탈출 게임



귀여운 에셋을 활용해  
10~30대 여성들을  
겨냥한 게임



다양한 콘텐츠들로  
유저들에게  
다채로운 즐거움을  
주는 게임



쉬운 조작법으로  
접근성이  
높은 게임

# Mysterious Forest\_

## 01. 'Mysterious Forest' 기획

- 게임 시나리오




단 둘뿐이지만 행복하게 살아가던 부녀가 있었다.  
하지만 딸이 불치병에 걸리고,  
이 병을 치료하기 위해서는  
신비의 숲에서 신비의 약물 찾아내야 한다.  
하지만 신비의 숲은 한번 들어가면  
나올 수 없는 숲으로 악명 높다.  
과연 아버지는 소중한 딸을 위해  
신비의 약물 을 획득할 수 있을까...?


# Mysterious Forest\_

## 01. 'Mysterious Forest' 기획

- 게임 컨셉



신비에 숲에서의 생존하고, 탈출하는 컨셉을 가진  
Mysterious Forest는 캐주얼풍의 3D 모바일 게임이다.  
아름다운 강과 나무, 꽃이 있는 신비의 숲을 배경으로  
아빠는 신비의 약물을 얻기 위해 **OVER**  
사냥과 채집 등 다양한 미션을 수행해야만 한다.



숲 입구로 돌아가기



# Mysterious Forest\_

## 01. 'Mysterious Forest' 기획

- 주요 콘텐츠

### Lake

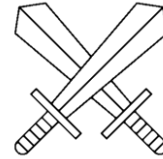
미션을 깨기 위해서는  
강을 건너야 한다.  
강에 들어가면 강의 색이 바뀌고  
Player의 HP가 깎인다.



### Weapon

각 상황별 알맞은 무기를  
사용할 수 있다.

~:맨손, 1:칼 2:방망이  
3:총 4:곡괭이



### Fruit

강 건너의 과일나무를  
베어 과일을 획득하면  
다음 단계로 넘어갈 수 있다.



### Rock

곡괭이로 바위를 부숴  
신비의 물약 제조에 필요한  
재료를 얻을 수 있다.



SURVIVE

# Mysterious Forest\_

## 01. 'Mysterious Forest' 기획

- 주요 콘텐츠

### Flower

일정 개수 이상의 꽃을  
획득해 신비의 물약 제조에  
필요한 재료를 얻을 수 있다.



### Fox

꽃을 가로막는 여우를  
방망이로 죽여야 한다.  
당으면 HP가 깎인다.



SURVIVE

### Zombie

밤이 되면 좀비가 나타난다.  
총을 이용해 좀비를  
죽여야 한다



### Potion

미션을 다 완수한 후  
신비의 물약을 찾으면  
최종 탈출을 할 수 있다.



# Mysterious Forest\_

## 01. 'Mysterious Forest' 기획

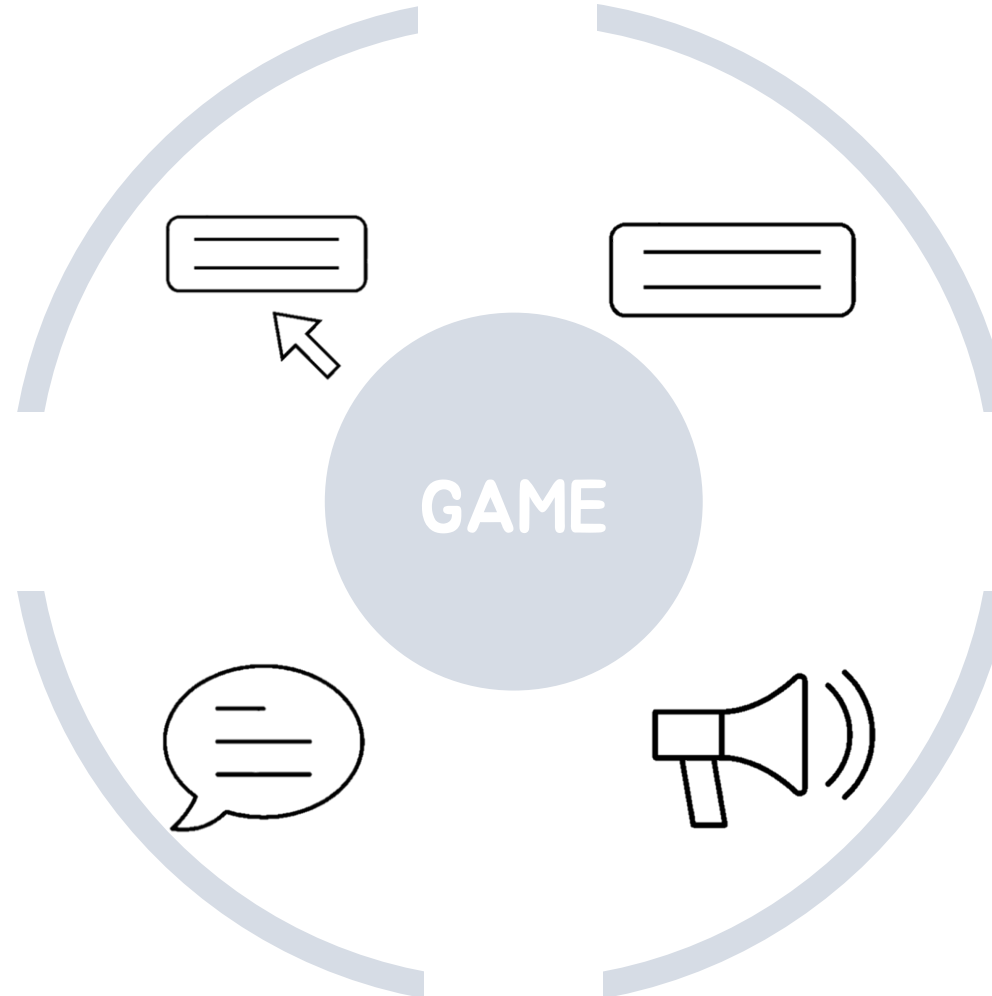
- 주요 콘텐츠

### GUI Button

Scene이 전환될 때,  
버튼을 눌러, 자연스럽게  
게임을 시작할 수 있도록 한다.

### Lable Text

레이블 텍스트를 통해  
유저가 게임을 원활하게  
진행하도록 돕는다.



### GUI Box

왼쪽 상단에 위치하여  
HP와 획득한 꽃, 과일,  
바위의 수를 나타내준다.

### Sound Effect

플레이어가 사냥 및 채집을  
할 때 효과음이 나온다.

# Mysterious Forest\_

## 01. 'Mysterious Forest' 기획

- 주요 콘텐츠

### Inventory

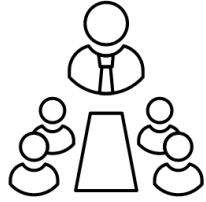
과일, 물약, 꽃을 얻으면  
인벤토리 안에 저장되고,  
획득 개수를 알려준다.



### Day and Night

낮에 신비의 물약 제조에  
필요한 꽃과 물약을 모두  
모으면 밤으로 변한다.

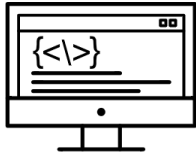
# Mysterious Forest\_



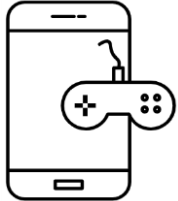
게임 기획

unity

관련 에셋 수집



코딩 작업



실제 구현



Chapter 02

**Mysterious  
Forest**  
개발과정



# Mysterious Forest\_

## 02. 'Mysterious Forest' 개발 과정

- Process

Step 01



**게임 기획**

Keidy studio 소스코드들을  
기반으로 게임 내 콘텐츠와  
스토리를 구상한다.

Step 02



**관련 에셋 수집**

기획한 게임 컨셉에  
맞는 배경, 캐릭터 등의  
에셋을 구한다.

Step 03



**코딩 작업**

게임 시나리오를  
바탕으로 준비된 에셋에  
코딩 작업을 한다.

Step 04



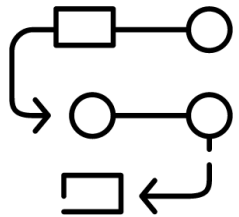
**실제 구현**

완성된 게임을 플레이  
해보고, 보완할 부분 찾아  
수정하는 과정을 거친다.

# Mysterious Forest\_



코딩



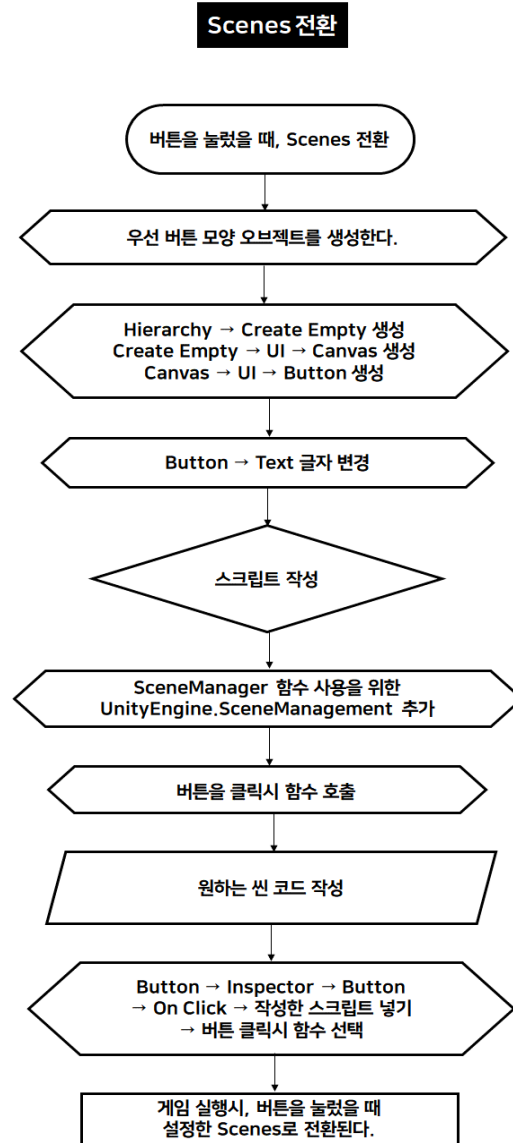
순서도

Chapter 03  
**Mysterious**  
코딩 및 순서도

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

### - Scene 전환



- 버튼을 눌렀을 때, Scenes 전환
  - StartSystem.cs (시작씬으로 전환)
  - OverSystem.cs(메인 게임 씬으로 전환)

## Mysterious Forest\_

### 03. 'Mysterious Forest' 코딩 및 순서도

#### - Scene 전환

- ▶ 버튼을 눌렀을 때, 원하는 Scenes으로 전환되도록 구현하려 한다.
- ▶ 먼저 버튼을 만들어야 한다.
- ▶ 버튼 모양 오브젝트를 생성한다.
- ▶ Hierarchy창에서 Create Empty를 생성한다.
- ▶ Create Empty 생성 후, Create Empty에서 UI를 통해 Canvas를 생성한다.
- ▶ 그리고, Canvas에서 UI를 통해 Button를 생성한다.
- ▶ Button에서 Text 글자를 변경한다.
- ▶ 이제 씬 전환을 적용시킬 스크립트를 작성한다.
- ▶ 씬 전환할 때 필요한 SceneManager 함수 사용을 위한  
UnityEngine.SceneManagement을 추가한다.

## Mysterious Forest\_

### 03. 'Mysterious Forest' 코딩 및 순서도

- Scene 전환

- ▶ 버튼을 클릭할 수 있도록 하는 함수를 호출한다.
- ▶ 버튼을 클릭할 수 있도록 하는 함수에 원하는 씬 입력한다.
- ▶ 작성한 스크립트를 Hierarchy창에서 생성한 Create Empty에 넣어준다.
- ▶ 그리고, Canvas에서 만든 Button의 Inspector창에서 Button을 찾아 버튼을 클릭시 함수를 선택한다.
- ▶ 게임 실행시, 버튼을 눌렀을 때 원하는 Scenes으로 전환된다.



# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

### - Scene 전환

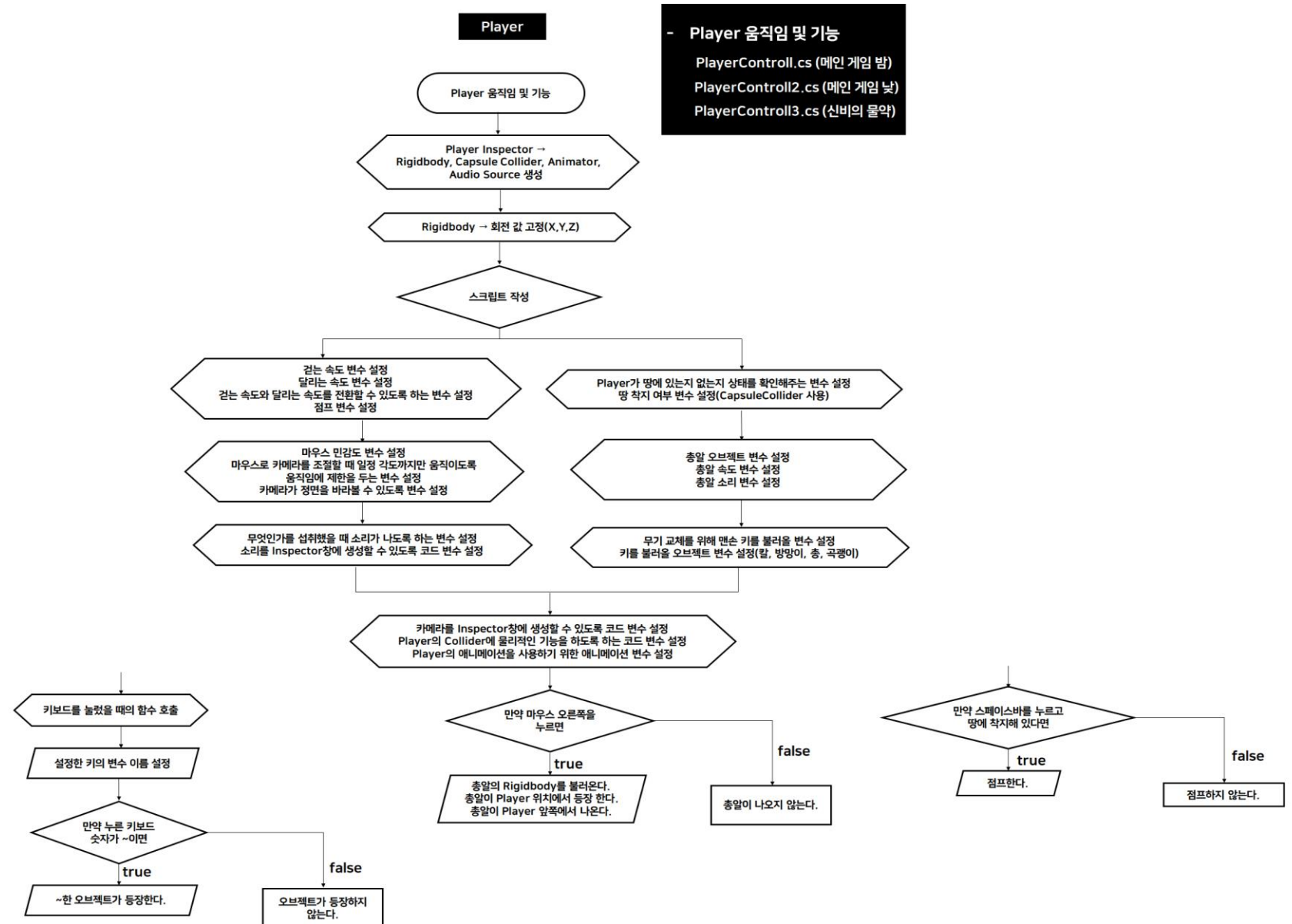
```
StartSystem.cs  X
C# 기타 파일 StartSystem
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class StartSystem : MonoBehaviour
7 {
8     public void OnClickGame()
9     {
10         SceneManager.LoadScene("DemoDay"); //시작 버튼을 누를시 메인 게임씬으로 간다.
11     }
12 }
```

```
OverSystem.cs  X
C# 기타 파일 OverSystem
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class OverSystem : MonoBehaviour
7 {
8     public void OnClickGame()
9     {
10         SceneManager.LoadScene("StartScenes"); //버튼을 누를시 게임 시작 씬으로 간다.
11     }
12 }
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

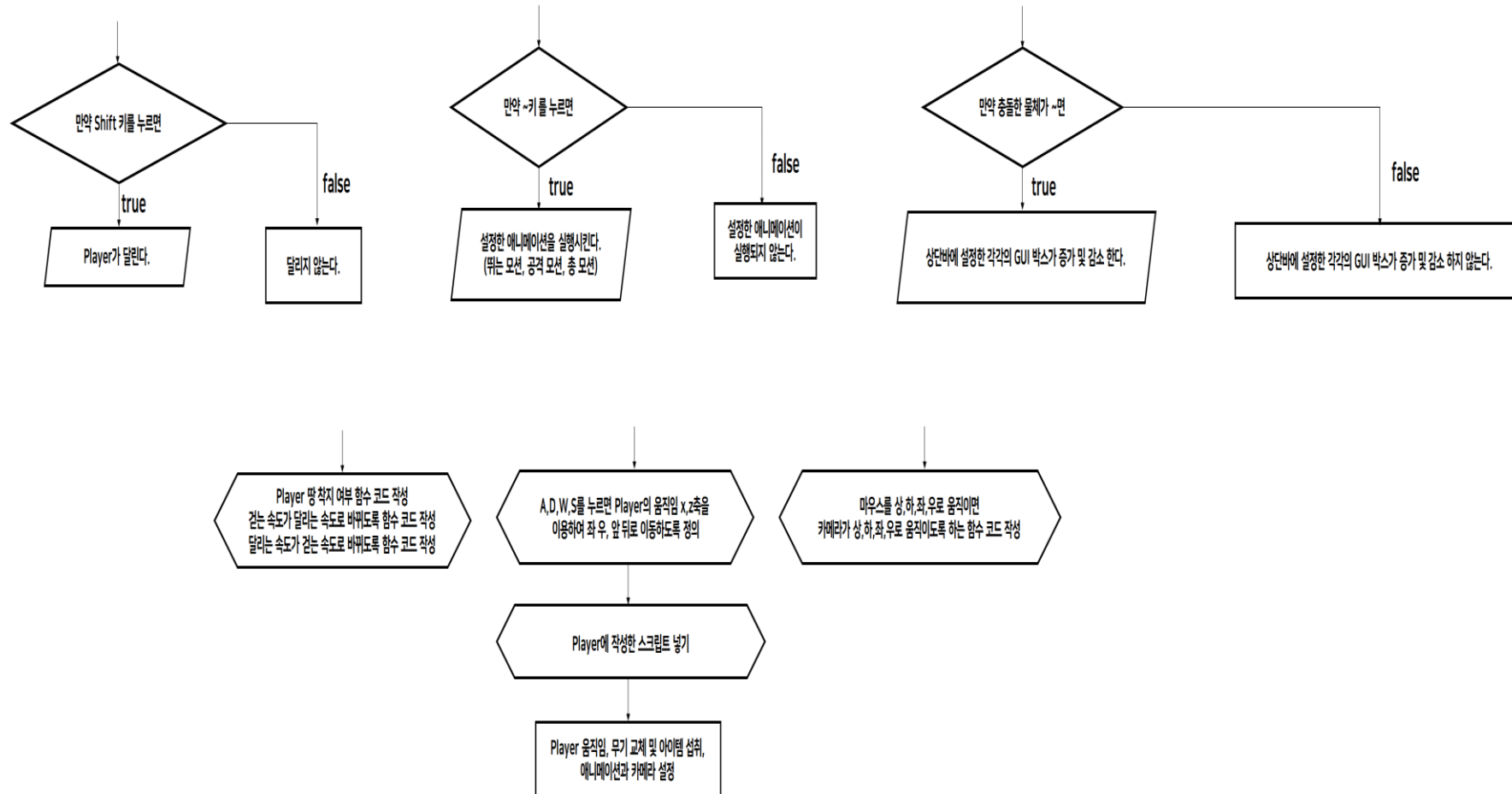
### - Player 움직임 및 기능



# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

### - Player 움직임 및 기능



# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

### - Player 움직임 및 기능

- ▶ Player의 움직임과 전체적인 기능을 첨가한다.
- ▶ 기본적으로 Player 오브젝트 Inspector 창에서 Rigidbody, Capsule Collider, Animator, Audio Source를 넣어준다.
- ▶ 여기서 Rigidbody의 회전 값은 전부 고정시켜준다.(X,Y,Z)
- ▶ 이제 Player의 움직임과 다양한 기능들을 적용시킬 스크립트를 만들어준다.
  - ▶ 먼저 Player가 걷거나 달리는 것을 구현하기 위해 걷는 속도와 달리는 속도 변수를 설정하고 걷는 속도와 달리는 속도 서로 전환될 수 있도록 하는 변수도 설정한다.
  - ▶ 점프를 위해 Player가 땅에 있는지 없는지 상태를 확인해주는 변수도 설정한다.
  - ▶ 마우스를 이용하여 Player의 시선을 처리할 수 있도록 카메라를 조절하는 변수를 설정한다.

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

### - Player 움직임 및 기능

- ▶ Player에서 총알이 나갈 수 있도록 총알 오브젝트 변수와 총알 속도 및 소리 변수를 설정한다.
- ▶ 무엇인가를 섭취했을 때 소리가 나도록 하는 변수를 설정한다.
- ▶ 소리를 Player 오브젝트 Inspector 창에서 생성할 수 있도록 코드 변수를 작성한다.
- ▶ 무기 교체를 위해 키를 불러올 변수를 설정한다. 그리고 키에 넣어줄 무기 오브젝트 변수를 설정한다.
- ▶ 카메라를 Inspector 창에서 생성할 수 있도록 코드 변수를 작성한다.
- ▶ Player의 Collider에 물리적인 기능을 하도록 하는 코드 변수를 작성한다.
- ▶ Player의 애니메이션을 사용하기 위한 애니메이션 변수를 작성한다.
- ▶ 만약 마우스 오른쪽을 누르면,(ture) ▶만약 누르지 않으면(false)



# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- Player 움직임 및 기능

- ▶ 총알의 Rigidbody를 불러온다. ▶ 총알이 나오지 않는다.
- ▶ 총알이 Player 위치에서 등장 한다.
- ▶ 총알이 Player 앞쪽에서 나온다.
- ▶ 키보드를 눌렀을 때 함수를 호출한다.
- ▶ 설정한 키의 변수 이름을 설정한다.
- ▶ 만약 누른 키보드 숫자가 ( )이면,(ture) ▶만약 키보드 숫자를 누르지 않으면(false)
- ▶ ( )로 설정한 오브젝트가 등장한다. ▶오브젝트가 등장하지 않는다.
- ▶ 만약 스페이스바를 누르고 땅에 착지해 있다면(ture)
- ▶ 점프한다.
- ▶ 스페이스바를 누르지 않고 땅에 착지하지 않으면(false)

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- Player 움직임 및 기능

- ▶ 점프하지 않는다.
- ▶ 만약 Shift 키를 누르면(true)
- ▶ 달린다.
- ▶ 만약 Shift 키를 누르지 않으면(false)
- ▶ 달리지 않는다.
- ▶ 만약 ( )키를 누르면(true)
- ▶ 설정한 애니메이션을 실행시킨다.(뛰는 모션, 공격모션, 총 모션)
- ▶ 만약 ( )키를 누르지 않으면(false)
- ▶ 설정한 애니메이션이 실행되지 않는다.
- ▶ 만약 충돌한 물체가 ( )라면(true)
- ▶ 상단바에 설정한 각각의 GUI박스가 증가 및 감소한다.

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

### - Player 움직임 및 기능

- ▶ Player 땅 착지 여부를 알 수 있는 함수 코드를 작성한다.
- ▶ 걷는 속도가 달리는 속도로 바뀌도록 함수 코드를 작성한다.
- ▶ 달리는 속도가 걷는 속도로 바뀌도록 함수 코드를 작성한다.
- ▶ A,D,W,S를 누르면 Player의 움직임 X,Z축을 이용하여 상,하,좌,우로 이동하도록 정의한다.
- ▶ 마우스를 상,하,좌,우로 움직이면 카메라가 상,하,좌,우로 움직이도록 하는 함수 코드를 작성한다.
- ▶ 작성한 스크립트를 Player에 넣는다.
- ▶ 게임 실행시, Player가 움직이고 무기 교체 및 아이템을 섭취하고 설정한 애니메이션이 실행된다.

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- Player 움직임 및 기능: **PlayerControll.cs**

```
PlayerControll.cs
[기타 파일] PlayerControll

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class PlayerControll : MonoBehaviour
6 {
7     //스피드 조정 변수
8     [SerializeField] //walkSpeed를 Inspector창에서 수정 가능하도록 하는 기능
9     private float walkSpeed; //걷는 속도 변수 설정
10
11     [SerializeField] //runSpeed를 Inspector창에서 수정 가능하도록 하는 기능
12     private float runSpeed; //달리는 속도 변수 설정
13
14     private float applySpeed; //걷는 속도와 달리는 속도를 전환할 수 있도록 하는 변수 설정
15
16     [SerializeField] //jumpForce를 Inspector창에서 수정 가능하도록 하는 기능
17     private float jumpForce; //점프 변수 설정
18
19     private bool isGround = true; //땅에 있는지 없는지 상태를 확인해주는 변수 코드
20
21
22     private CapsuleCollider capsuleCollider; //땅 착지 여부
23
24     //민감도
25     [SerializeField] //lookSensitivity를 Inspector창에서 수정 가능하도록 하는 기능
26     private float lookSensitivity; //카메라의 민감도 변수를 설정한다.
27
28     //카메라 한계
29     [SerializeField]
30     private float cameraRotationLimit; //마우스로 카메라를 조절할 때 일정 각도까지만 움직이도록 제한을 준다.
31     private float currentCameraRotationX = 0; //카메라가 정면을 바라볼 수 있도록 설정
32
33     public GameObject bullet; //총알 오브젝트 변수
34     public float bulletSpeed = 10.0f; //총알 속도
35     private AudioSource audioBullet; //총알 소리 변수 설정
36     private GameManager gameManager; //체력을 깎기 위해 gameManager 변수를 설정한다.
37
38     [SerializeField]
39     private Camera theCamera; //Camera의 GetComponent를 불러온다.
40
41     private Rigidbody myRigid; //Player의 Collider에 물리적인 기능을 하도록 하는 기능 //Inspector창에 Rigidbody추가 후 작성
42
43     private Animator playerAnim; //Player의 애니메이션을 사용하기 위해 애니메이션 변수를 설정한다.
44
45 }
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- Player 움직임 및 기능: **PlayerControll.cs**

```
PlayerControll.cs
[기타 파일] PlayerControll

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class PlayerControll : MonoBehaviour
6 {
7     //스피드 조정 변수
8     [SerializeField] //walkSpeed를 Inspector창에서 수정 가능하도록 하는 기능
9     private float walkSpeed; //걷는 속도 변수 설정
10
11     [SerializeField] //runSpeed를 Inspector창에서 수정 가능하도록 하는 기능
12     private float runSpeed; //달리는 속도 변수 설정
13
14     private float applySpeed; //걷는 속도와 달리는 속도를 전환할 수 있도록 하는 변수 설정
15
16     [SerializeField] //jumpForce를 Inspector창에서 수정 가능하도록 하는 기능
17     private float jumpForce; //점프 변수 설정
18
19     private bool isGround = true; //땅에 있는지 없는지 상태를 확인해주는 변수 코드
20
21
22     private CapsuleCollider capsuleCollider; //땅 착지 여부
23
24     //민감도
25     [SerializeField] //lookSensitivity를 Inspector창에서 수정 가능하도록 하는 기능
26     private float lookSensitivity; //카메라의 민감도 변수를 설정한다.
27
28     //카메라 한계
29     [SerializeField]
30     private float cameraRotationLimit; //마우스로 카메라를 조절할 때 일정 각도까지만 움직이도록 제한을 준다.
31     private float currentCameraRotationX = 0; //카메라가 정면을 바라볼 수 있도록 설정
32
33     public GameObject bullet; //총알 오브젝트 변수
34     public float bulletSpeed = 10.0f; //총알 속도
35     private AudioSource audioBullet; //총알 소리 변수 설정
36     private GameManager gameManager; //체력을 깎기 위해 gameManager 변수를 설정한다.
37
38     [SerializeField]
39     private Camera theCamera; //Camera의 GetComponent를 불러온다.
40
41     private Rigidbody myRigid; //Player의 Collider에 물리적인 기능을 하도록 하는 기능 //Inspector창에 Rigidbody추가 후 작성
42
43     private Animator playerAnim; //Player의 애니메이션을 사용하기 위해 애니메이션 변수를 설정한다.
44
45 }
```



# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- Player 움직임 및 기능: **PlayerControll.cs**

```
46 // Use this for initialization
47 void Start()
48 {
49     capsuleCollider = GetComponent<CapsuleCollider>(); //Inspector창에서 CapsuleCollider를 불러온다.
50     myRigid = GetComponent<Rigidbody>(); //Inspector창에서 Rigidbody를 불러온다.
51     audioBullet = GetComponent<AudioSource>(); //총알 소리를 불러온다.
52     applySpeed = walkSpeed; //달리기 전까지 걷는 상태
53     playerAnim = GameObject.Find("Player").GetComponent<Animator>(); //Inspector창에서 Animator를 불러온다.
54     gameManager = GameObject.Find("GameManager").GetComponent<GameManager>(); //GameManager를 불러온다.
55 }
56
57
58
59
60 // Update is called once per frame
61 void Update()
62 {
63     IsGround(); //땅에 착지하고 있는지 하지 않는지 제어하기 위한 코드 작성 전 함수 정의
64     TryJump(); //점프를 실행 및 제어하기 위한 코드 작성 전 함수 정의
65     TryRun(); //뛰는지 걷는지 제어하기 위한 코드 작성 전 함수 정의(움직임 제어)
66     Move(); //클릭을 하면 실시간으로 움직이도록 하는 코드를 작성하기 위한 함수 정의
67     CameraRotation(); //카메라 움직임 함수 정의
68     CharacterRotation(); //캐릭터 움직임 함수 정의
69
70     if (Input.GetMouseButtonDown(1))
71     {
72         GameObject newBullet = Instantiate(bullet, transform.position + transform.forward, transform.rotation) as GameObject; //총알이 Player에서 나오도록하기
73         Rigidbody rbBullet = newBullet.GetComponent<Rigidbody>(); //총알의 Rigidbody불러오기
74         rbBullet.velocity = transform.forward * bulletspeed; //총알이 Player 앞에 나오도록 하기
75     }
76
77 }
78
79 // 지면 체크.
80 private void IsGround()
81 {
82     isGround = Physics.Raycast(transform.position, Vector3.down, capsuleCollider.bounds.extents.y + 0.1f); //Player의 Y축 방향으로 0.1 거리 만큼 레이저를 쏘아 Player가 땅에 닿는지 닿지 않는지 여부 인식(땅에 착지한 순간)
83 }
84
85 // 점프 시도
86 private void TryJump()
87 {
88     if (Input.GetKeyDown(KeyCode.Space) && isGround) //Space키가 눌러지고 Player가 땅에 착지하고 있으면
89     {
90         Jump();
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- Player 움직임 및 기능: **PlayerControll.cs**

```
91     }
92 }
93
94 // 점프
95 private void Jump()
96 {
97     myRigid.velocity = transform.up * jumpForce; //Player의 움직임에서 Y축으로 설정한 점프 힘만큼 증가하도록 한다.
98 }
99
100 //달리기 시도
101 private void TryRun()
102 {
103     if (Input.GetKey(KeyCode.LeftShift)) //왼쪽 Shift 키를 누르면
104     {
105         Running(); //달리기가 실행됨
106     }
107     if (Input.GetKeyUp(KeyCode.LeftShift)) //왼쪽 Shift키가 올라올라왔을 때
108     {
109         RunningCancel(); //달리지 않는다.
110     }
111 }
112
113 // 달리기 실행
114 private void Running()
115 {
116     applySpeed = runSpeed; //걸는 속도가 달리는 속도로 바뀐다.
117 }
118 // 달리기 취소
119 private void RunningCancel()
120 {
121     applySpeed = walkSpeed; //달리는 속도가 걷는 속도로 바뀐다.
122 }
123
124 private void Move()
125 {
126
127     float _moveDirX = Input.GetAxisRaw("Horizontal"); //A,D를 눌렀을 때
128     float _moveDirZ = Input.GetAxisRaw("Vertical"); //W,S를 눌렀을 때
129
130     Vector3 _moveHorizontal = transform.right * _moveDirX; //Player의 transform X축을 이용하여 좌, 우 이동하도록 정의
131     Vector3 _moveVertical = transform.forward * _moveDirZ; //Player의 transform Z축을 이용하여 앞, 뒤 이동하도록 정의
132
133     Vector3 _velocity = (_moveHorizontal + _moveVertical).normalized * applySpeed; //속도는 방향이 움직이면 속도값을 곱하여 얼마나 빨리 이동할 것인지 설정한다.
134
135     myRigid.MovePosition(transform.position + _velocity * Time.deltaTime); //Rigidbody에 움직임 함수를 정의하여 Player의 위치에 속도를 더해주고, 1초동안 속도만큼 움직이게 하겠다.
136 }
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- Player 움직임 및 기능: **PlayerControll.cs**

```
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178

if (_moveDirZ > 0.05f || _moveDirZ < -0.05f) //만약 A,D를 누르고 있으면
{
    playerAnim.SetBool("Run", true); //뛰어라
}
else
{
    playerAnim.SetBool("Run", false); //멈춰라
}

if (Input.GetMouseButtonDown(0)) //왼쪽 마우스 버튼을 누르면
{
    playerAnim.SetBool("Hit", true); //때리는 애니메이션을 실행
}

if (Input.GetMouseButtonUp(0)) //왼쪽 마우스 버튼이 올라오면
{
    playerAnim.SetBool("Hit", false); //때리는 애니메이션을 멈춰라
}

if (Input.GetMouseButtonDown(1)) //오른쪽 마우스 버튼을 누르면
{
    playerAnim.SetBool("Dig", true); //총쏘는 애니메이션을 실행
    audioBullet.Play();
}

if (Input.GetMouseButtonUp(1)) //오른쪽 마우스 버튼이 올라오면
{
    playerAnim.SetBool("Dig", false); //총쏘는 애니메이션을 멈춰라
}

}

private void CharacterRotation()
{
    // 좌우 캐릭터 회전
    float _yRotation = Input.GetAxisRaw("Mouse X"); //마우스를 좌 우로 움직였을 때, 카메라가 좌 우(Y축)으로 움직인다.
    Vector3 _characterRotationY = new Vector3(0f, _yRotation, 0f) * lookSensitivity; //카메라가 천천히 움직이도록 설정한다.
    myRigid.MoveRotation(myRigid.rotation * Quaternion.Euler(_characterRotationY)); //Player의 회전값에 천천히 움직이도록 설정한 카메라 값을 곱하여 회전하도록 한다.
}
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- Player 움직임 및 기능: **PlayerControll.cs**

```
178
179 private void CameraRotation()
180 {
181     //상 하 카메라 회전
182     float _xRotation = Input.GetAxisRaw("Mouse Y"); //마우스를 앞 뒤로 움직였을 때, 카메라가 상 하(X축)으로 움직인다.
183     float _cameraRotationX = _xRotation * lookSensitivity; //카메라가 천천히 움직이도록 설정한다.
184     currentCameraRotationX += _cameraRotationX; //현재 카메라 움직임은 마우스를 앞 뒤로 움직이는 값에 따라 움직인다.
185     currentCameraRotationX = Mathf.Clamp(currentCameraRotationX, -cameraRotationLimit, cameraRotationLimit); //cameraRotationLimit의 최대, 최소값을 결정한다.(설정한 각도만큼 이상 움직이지 못하도록 제한을 둔다.)
186
187     theCamera.transform.localEulerAngles = new Vector3(currentCameraRotationX, 0f, 0f); //실제 카메라에 적용시키기 위한 설정을 한다. 마우스를 앞 뒤로 움직였을 때 카메라가 상 하(X축)로만 움직이도록 설정한다.
188 }
189 void OnCollisionEnter(Collision collision)
190 {
191     if (collision.gameObject.name == "zombie(Clone)") //충돌한 대상이 좀비라면
192     {
193         gameManager.HP -= 1; //체력이 깎인다.
194     }
195 }
196
197
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- Player 움직임 및 기능: **PlayerControll2.cs**

```
PlayerControll2.cs
[기타 파일]
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class PlayerControll2 : MonoBehaviour
6 {
7     //스피드 조정 변수
8     [SerializeField] //walkSpeed를 Inspector창에서 수정 가능하도록 하는 기능
9     private float walkSpeed; //걸는 속도 변수 설정
10
11     [SerializeField] //runSpeed를 Inspector창에서 수정 가능하도록 하는 기능
12     private float runSpeed; //달리는 속도 변수 설정
13
14     private float applySpeed; //걸는 속도와 달리는 속도를 대입할 수 있도록 하는 변수 설정
15
16     [SerializeField] //jumpForce를 Inspector창에서 수정 가능하도록 하는 기능
17     private float jumpForce; //점프 변수 설정
18
19     //상태 변수
20     private bool isGround = true; //땅에 있는지 없는지 상태를 확인해주는 변수 코드
21
22
23     private CapsuleCollider capsuleCollider; //땅 착지 여부
24
25     //민감도
26     [SerializeField] //lookSensitivity를 Inspector창에서 수정 가능하도록 하는 기능
27     private float lookSensitivity; //카메라의 민감도 변수를 설정한다.
28
29     //카메라 한계
30     [SerializeField]
31     private float cameraRotationLimit; //마우스로 카메라를 조절할 때 일정 각도까지만 움직이도록 제한을 두는 변수 설정
32     private float currentCameraRotationX = 0; //카메라가 정면을 바라볼 수 있도록 설정
33
34     public GameObject bullet; //총알 오브젝트 변수
35     public float bulletspeed = 10.0f; //총알 속도
36     private AudioSource audioBullet; //총알 소리 변수 설정
37     private GameManager2 gameManager; //체력을 깎기위해 gameManager 변수를 설정한다.
38
39     public AudioClip FlowerSource; //무엇인가를 섭취했을 때 소리가 나도록 하는 변수 설정
40     AudioSource FlowerSource01; //소리를 Inspector창에 생성할 수 있도록 코드 변수 설정
41
42
43     public GameObject Kal; //칼 오브젝트 설정
44     public GameObject Mang; //방망이 오브젝트 설정
45     public GameObject Gun; //총 오브젝트 설정
46     public GameObject G01; //곡괭이 오브젝트 설정
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- Player 움직임 및 기능: **PlayerControl2.cs**

```
48 private bool hand; //맨손 키를 불러올 변수 설정
49 private bool pon1; //키를 불러올 오브젝트 변수 설정(칼)
50 private bool pon2; //키를 불러올 오브젝트 변수 설정(방망이)
51 private bool pon3; //키를 불러올 오브젝트 변수 설정(총)
52 private bool pon4; //키를 불러올 오브젝트 변수 설정(곡괭이)
53
54 [SerializeField]
55 private Camera theCamera; //Camera의 GetComponent를 불러온다.
56
57 private Rigidbody myRigid; //Player의 Collider에 물리적인 기능을 하도록 하는 기능 //Inspector창에 Rigidbody추가 후 작성
58
59 private Animator playerAnim; //Player의 애니메이션을 사용하기 위한 애니메이션 변수를 설정한다.
60 private void GetInput() //키보드를 눌렀을 때
61 {
62     hand = Input.GetButtonDown("Hand");
63     pon1 = Input.GetButtonDown("Pon1");
64     pon2 = Input.GetButtonDown("Pon2");
65     pon3 = Input.GetButtonDown("Pon3");
66     pon4 = Input.GetButtonDown("Pon4");
67
68     if (hand) //키보드 숫자 1 누르면
69     {
70         Kal.SetActive(false);
71         Mang.SetActive(false);
72         Gun.SetActive(false);
73         GOI.SetActive(false);
74     }
75
76     if (pon1) //키보드 숫자 1 누르면
77     {
78         Kal.SetActive(true); //칼 오브젝트 등장
79         Mang.SetActive(false);
80         Gun.SetActive(false);
81         GOI.SetActive(false);
82     }
83
84     if (pon2) //키보드 숫자 2 누르면
85     {
86         Mang.SetActive(true); //방망이 오브젝트 등장
87         Kal.SetActive(false);
88         Gun.SetActive(false);
89         GOI.SetActive(false);
90     }
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- Player 움직임 및 기능: **PlayerControll2.cs**

```
92     if (pon3) //키보드 숫자 3 누르면
93     {
94         Mang.SetActive(false);
95         Kal.SetActive(false);
96         Gun.SetActive(true); //총 오브젝트 등장
97         GO1.SetActive(false);
98     }
99     if (pon4) //키보드 숫자 4 누르면
100    {
101        Mang.SetActive(false);
102        Kal.SetActive(false);
103        Gun.SetActive(false);
104        GO1.SetActive(true); //곡괭이 오브젝트 등장
105    }
106 }
107 // Use this for initialization
108 void Start()
109 {
110     capsuleCollider = GetComponent<CapsuleCollider>(); //Inspector창에서 CapsuleCollider를 불러온다.
111     myRigid = GetComponent<Rigidbody>(); //Inspector창에서 Rigidbody를 불러온다.
112     audioBullet = GetComponent<AudioSource>(); //총알 소리를 불러온다.
113     applySpeed = walkSpeed; //달리기 전까지 걷는 상태
114     playerAnim = GameObject.Find("Player").GetComponent<Animator>(); //Inspector창에서 Animator를 불러온다.
115     gameManager = GameObject.Find("GameManager2").GetComponent<GameManager2>(); //GameManager2를 불러온다.
116     FlowerSource01 = GetComponent<AudioSource>(); //소리를 불러온다.
117     Kal.SetActive(false); //시작할 때 무기를 안보이게 설정
118     Mang.SetActive(false); //시작할 때 무기를 안보이게 설정
119     Gun.SetActive(false); //시작할 때 무기를 안보이게 설정
120     GO1.SetActive(false); //시작할 때 무기를 안보이게 설정
121 }
122
123
124
125
126 // Update is called once per frame
127 void Update()
128 {
129     IsGround(); //땅에 착지하고 있는지 하지 않는지 제어하기 위한 코드 작성 전 함수 정의
130     TryJump(); //점프를 실행 및 제어하기 위한 코드 작성 전 함수 정의
131     TryRun(); //뛰는지 걷는지 제어하기 위한 코드 작성 전 함수 정의(움직임 제어)
132     Move(); //클릭을 하면 실시간으로 움직이도록 하는 코드를 작성하기 위한 함수 정의
133     CameraRotation(); //카메라 움직임 함수 정의
134     CharacterRotation(); //캐릭터 움직임 함수 정의
135     GetInput(); //키를 눌렀을 때 함수 정의
136 }
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- Player 움직임 및 기능: **PlayerControll2.cs**

```
137 if (Input.GetMouseButtonDown(1)) // 만약 마우스 오른쪽을 누르면
138 {
139     GameObject newBullet = Instantiate(bullet, transform.position + transform.forward, transform.rotation) as GameObject; // 총알이 Player에서 나오도록 한다.
140     Rigidbody rbBullet = newBullet.GetComponent<Rigidbody>(); // 총알의 Rigidbody 불러오기
141     rbBullet.velocity = transform.forward * bulletspeed; // 총알이 Player 앞에 나오도록 하기
142 }
143
144
145 }
146
147 private void IsGround() // 지면을 체크하는 함수 코드
148 {
149     isGround = Physics.Raycast(transform.position, Vector3.down, capsuleCollider.bounds.extents.y + 0.1f); // Player의 Y축 방향으로 0.1 거리 만큼 레이저를 쏘아 Player가 땅에 닿는지 닿지 않는지 여부 인식(땅에 착지한 순간)
150 }
151
152 // 점프 시도
153 private void TryJump()
154 {
155     if (Input.GetKeyDown(KeyCode.Space) && isGround) // Space키가 눌러지고 Player가 땅에 착지하고 있으면
156     {
157         Jump();
158     }
159 }
160
161 // 점프
162 private void Jump()
163 {
164     myRigid.velocity = transform.up * jumpForce; // Player의 움직임에서 Y축으로 설정한 점프 힘만큼 증가하도록 한다.
165 }
166
167 // 달리기 시도
168 private void TryRun()
169 {
170     if (Input.GetKey(KeyCode.LeftShift)) // 왼쪽 Shift 키를 누르면
171     {
172         Running(); // 달리가 실행됨
173     }
174     if (Input.GetKeyUp(KeyCode.LeftShift)) // 왼쪽 Shift키가 올라올라왔을 때
175     {
176         RunningCancel(); // 달리지 않는다.
177     }
178 }
179
```



# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- Player 움직임 및 기능: **PlayerControl2.cs**

```
179
180 // 달리기 실행
181 private void Running()
182 {
183     applySpeed = runSpeed; //걷는 속도가 달리는 속도로 바뀐다.
184 }
185 // 달리기 취소
186 private void RunningCancel()
187 {
188     applySpeed = walkSpeed; //달리는 속도가 걷는 속도로 바뀐다.
189 }
190
191 private void Move()
192 {
193
194     float _moveDirX = Input.GetAxisRaw("Horizontal"); //A,D를 눌렀을 때
195     float _moveDirZ = Input.GetAxisRaw("Vertical"); //W,S를 눌렀을 때
196
197     Vector3 _moveHorizontal = transform.right * _moveDirX; //Player의 transform X축을 이용하여 좌, 우 이동하도록 정의
198     Vector3 _moveVertical = transform.forward * _moveDirZ; //Player의 transform Z축을 이용하여 앞, 뒤 이동하도록 정의
199
200     Vector3 _velocity = (_moveHorizontal + _moveVertical).normalized * applySpeed; //속도는 방향이 움직이면 속도값을 곱하여 얼마나 빨리 이동할 것인지 설정한다.
201
202     myRigid.MovePosition(transform.position + _velocity * Time.deltaTime); //Rigidbody에 움직임 함수를 정의하여 Player의 위치에 속도를 더해주고, 1초동안 속도만큼 움직이게 하겠다.
203
204     if (_moveDirZ > 0.05f || _moveDirZ < -0.05f) //만약 A,D를 누르고 있으면
205     {
206         playerAnim.SetBool("Run", true); //뛰어라
207     }
208     else
209     {
210         playerAnim.SetBool("Run", false); //멈춰라
211     }
212
213
214     if (Input.GetMouseButtonDown(0)) //왼쪽 마우스 버튼을 누르면
215     {
216         playerAnim.SetBool("Hit", true); //때리는 애니메이션을 실행
217         audioBullet.Play();
218     }
219
220     if (Input.GetMouseButtonUp(0)) //왼쪽 마우스 버튼이 올라오면
221     {
222         playerAnim.SetBool("Hit", false); //때리는 애니메이션을 멈춰라
223     }
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- Player 움직임 및 기능: **PlayerControl2.cs**

```
224
225     if (Input.GetMouseButtonDown(1)) //오른쪽 마우스 버튼을 누르면
226     {
227         playerAnim.SetBool("Dig", true); //총쏘는 애니메이션을 실행
228     }
229
230
231     if (Input.GetMouseButtonUp(1)) //오른쪽 마우스 버튼이 올라오면
232     {
233         playerAnim.SetBool("Dig", false); //총쏘는 애니메이션을 멈춰라
234     }
235
236 }
237
238
239 private void CharacterRotation()
240 {
241     // 좌우 캐릭터 회전
242     float _yRotation = Input.GetAxisRaw("Mouse X"); //마우스를 좌 우로 움직였을 때, 카메라가 좌 우(Y축)으로 움직인다.
243     Vector3 _characterRotationY = new Vector3(0f, _yRotation, 0f) * lookSensitivity; //카메라가 천천히 움직이도록 설정한다.
244     myRigid.MoveRotation(myRigid.rotation * Quaternion.Euler(_characterRotationY)); //Player의 회전값에 천천히 움직이도록 설정한 카메라 값을 곱하여 회전하도록 한다.
245 }
246
247 private void CameraRotation()
248 {
249     //상 하 카메라 회전
250     float _xRotation = Input.GetAxisRaw("Mouse Y"); //마우스를 앞 뒤로 움직였을 때, 카메라가 상 하(X축)으로 움직인다.
251     float _cameraRotationX = _xRotation * lookSensitivity; //카메라가 천천히 움직이도록 설정한다.
252     currentCameraRotationX -= _cameraRotationX; //현재 카메라 움직임은 마우스를 앞 뒤로 움직이는 값에 따라 움직인다.
253     currentCameraRotationX = Mathf.Clamp(currentCameraRotationX, -cameraRotationLimit, cameraRotationLimit); //cameraRotationLimit의 최대, 최소값을 결정한다.(설정한 각도만큼 이상 움직이지 못하도록 제한을 둔다.)
254
255     theCamera.transform.localEulerAngles = new Vector3(currentCameraRotationX, 0f, 0f); //실제 카메라에 적용시키기 위한 설정을 한다. 마우스를 앞 뒤로 움직였을 때 카메라가 상 하(X축)로만 움직이도록 설정한다.
256 }
257
258 void OnCollisionEnter(Collision collision)
259 {
260     if (collision.gameObject.tag == "Fox") //만약 충돌한 오브젝트가 여우면
261     {
262         gameManager.HP -= 1; //체력이 감소한다.
263     }
264
265     if (collision.gameObject.tag == "blue") //만약 충돌한 오브젝트가 꽃이면
266     {
267         gameManager.Flower += 1; //꽃 갯수가 증가한다.
268         FlowerSource01.PlayOneShot(FlowerSource); //설정한 소리가 재생된다.
269     }
270 }
```

## Mysterious Forest\_

### 03. 'Mysterious Forest' 코딩 및 순서도

- Player 움직임 및 기능: **PlayerControll2.cs**

```
269
270 if (collision.gameObject.tag == "food") //만약 충돌한 오브젝트가 과일이면
271 {
272     gameManager.HP += 1; //체력이 증가한다.
273     gameManager.food += 1; //음식이 증가한다.
274     FlowerSource01.PlayOneShot(FlowerSource); //설정 한 소리가 재생된다.
275 }
276
277 if (collision.gameObject.tag == "Potion") //만약 충돌한 오브젝트가 물약이면
278 {
279     gameManager.rock += 1; //물약이 증가한다.
280     FlowerSource01.PlayOneShot(FlowerSource); //설정 한 소리가 재생된다.
281 }
282 }
283
284
285 }
286
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- Player 움직임 및 기능: **PlayerControll3.cs**

```
PlayerControll3.cs → X
C# 기타 파일 PlayerControll3

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class PlayerControll3 : MonoBehaviour
6 {
7     //스피드 조정 변수
8     [SerializeField] //walkSpeed를 Inspector창에서 수정 가능하도록 하는 기능
9     private float walkSpeed; //걷는 속도 변수 설정
10
11     [SerializeField] //runSpeed를 Inspector창에서 수정 가능하도록 하는 기능
12     private float runSpeed; //달리는 속도 변수 설정
13
14     private float applySpeed; //걷는 속도와 달리는 속도를 대입할 수 있도록 하는 변수 설정
15
16     [SerializeField] //jumpForce를 Inspector창에서 수정 가능하도록 하는 기능
17     private float jumpForce; //점프 변수 설정
18
19     private bool isGround = true; //땅에 있는지 없는지 상태를 확인해주는 변수 코드
20
21
22     private CapsuleCollider capsuleCollider; //땅 착지 여부
23
24     //민감도
25     [SerializeField] //lookSensitivity를 Inspector창에서 수정 가능하도록 하는 기능
26     private float lookSensitivity; //카메라의 민감도 변수를 설정한다.
27
28     //카메라 한계
29     [SerializeField]
30     private float cameraRotationLimit; //마우스로 카메라를 조절할 때 일정 각도까지만 움직이도록 제한을 준다.
31     private float currentCameraRotationX = 0; //카메라가 경면을 바라볼 수 있도록 설정
32
33     public GameObject bullet; //총알 오브젝트 변수
34     public float bulletspeed = 10.0f; //총알 속도
35     private AudioSource audioBullet; //총알 소리 변수 설정
36     private GameManager3 gameManager; //체력을 각기위해 gameManager 변수를 설정한다.
37
38
39     public AudioClip PotionSource;
40     AudioSource PotionSource01;
41
42     public GameObject Kal;
43     public GameObject Mang;
44     public GameObject Gun;
45 }
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- Player 움직임 및 기능: **PlayerControl3.cs**

```
46 private bool hand;
47 private bool pon1;
48 private bool pon2;
49 private bool pon3;
50
51
52 //필요한 컴포넌트
53 [SerializeField]
54 private Camera theCamera; //Camera의 GetComponent를 불러온다.
55
56 private Rigidbody myRigid; //Player의 Collider에 물리적인 기능을 하도록 하는 기능 //Inspector창에 Rigidbody추가 후 작성
57
58 private Animator playerAnim; //Player의 애니메이션을 사용하기 위해 애니메이션 변수를 설정한다.
59
60 private void GetInput()
61 {
62     hand = Input.GetButtonDown("Hand");
63     pon1 = Input.GetButtonDown("Pon1");
64     pon2 = Input.GetButtonDown("Pon2");
65     pon3 = Input.GetButtonDown("Pon3");
66
67     if (hand) //키보드 숫자 1 누르면
68     {
69         Kal.SetActive(false);
70         Mang.SetActive(false);
71         Gun.SetActive(false);
72     }
73
74     if (pon1) //키보드 숫자 1 누르면
75     {
76         Kal.SetActive(true);
77         Mang.SetActive(false);
78         Gun.SetActive(false);
79     }
80
81     if (pon2) //키보드 숫자 2 누르면
82     {
83         Mang.SetActive(true);
84         Kal.SetActive(false);
85         Gun.SetActive(false);
86     }
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- Player 움직임 및 기능: **PlayerControll3.cs**

```
88     if (pon3) //키보드 숫자 2 누르면
89     {
90         Mang.SetActive(false);
91         Kal.SetActive(false);
92         Gun.SetActive(true);
93     }
94 }
95
96 // Use this for initialization
97 void Start()
98 {
99     capsuleCollider = GetComponent<CapsuleCollider>(); //Inspector창에서 CapsuleCollider를 불러온다.
100    myRigid = GetComponent<Rigidbody>(); //Inspector창에서 Rigidbody를 불러온다.
101    audioBullet = GetComponent<AudioSource>(); //총알 소리를 불러온다.
102    applySpeed = walkSpeed; //달리기 전까지 걷는 상태
103    playerAnim = GameObject.Find("Player").GetComponent<Animator>(); //Inspector창에서 Animator를 불러온다.
104    gameManager = GameObject.Find("GameManager3").GetComponent<GameManager3>(); //GameManager를 불러온다.
105    PotionSource01 = GetComponent<AudioSource>(); //소리를 불러온다.
106 }
107
108
109
110
111 // Update is called once per frame
112 void Update()
113 {
114     IsGround(); //땅에 착지하고 있는지 하지 않는지 제어하기 위한 코드 작성 전 함수 정의
115     TryJump(); //점프를 실행 및 제어하기 위한 코드 작성 전 함수 정의
116     TryRun(); //뛰는지 걷는지 제어하기 위한 코드 작성 전 함수 정의(움직임 제어)
117     Move(); //클릭을 하면 실시간으로 움직이도록 하는 코드를 작성하기 위한 함수 정의
118     CameraRotation(); //카메라 움직임 함수 정의
119     CharacterRotation(); //캐릭터 움직임 함수 정의
120
121     if (Input.GetMouseButtonDown(1)) //만약 마우스 오른쪽 키를 누르면
122     {
123         GameObject newBullet = Instantiate(bullet, transform.position + transform.forward, transform.rotation) as GameObject; //총알이 Player에서 나오도록 한다.
124         Rigidbody rbBullet = newBullet.GetComponent<Rigidbody>(); //총알의 Rigidbody불러오기
125         rbBullet.velocity = transform.forward * bulletspeed; //총알이 Player 앞에 나오도록 하기
126     }
127
128
129 }
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- Player 움직임 및 기능: **PlayerControl3.cs**

```
130
131 // 지면 체크.
132 private void IsGround()
133 {
134     isGround = Physics.Raycast(transform.position, Vector3.down, capsuleCollider.bounds.extents.y + 0.1f); //Player의 Y축 방향으로 0.1 거리 만큼 레이저를 쏘아 Player가 땅에 닿는지 닿지 않는지 여부 인식(땅에 착지한 순간)
135 }
136
137 // 점프 시도
138 private void TryJump()
139 {
140     if (Input.GetKeyDown(KeyCode.Space) && isGround) //Space키가 눌러지고 Player가 땅에 착지하고 있으면
141     {
142         Jump();
143     }
144 }
145
146 // 점프
147 private void Jump()
148 {
149     myRigid.velocity = transform.up * jumpForce; //Player의 움직임에서 Y축으로 설정한 점프 힘만큼 증가하도록 한다.
150 }
151
152 //달리기 시도
153 private void TryRun()
154 {
155     if (Input.GetKey(KeyCode.LeftShift)) //왼쪽 Shift 키를 누르면
156     {
157         Running(); //달리기가 실행됨
158     }
159     if (Input.GetKeyUp(KeyCode.LeftShift)) //왼쪽 Shift키가 올라올라왔을 때
160     {
161         RunningCancel(); //달리지 않는다.
162     }
163 }
164
165 // 달리기 실행
166 private void Running()
167 {
168     applySpeed = runSpeed; //걸는 속도가 달리는 속도로 바뀐다.
169 }
170 // 달리기 취소
171 private void RunningCancel()
172 {
173     applySpeed = walkSpeed; //달리는 속도가 걷는 속도로 바뀐다.
174 }
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- Player 움직임 및 기능: **PlayerControl3.cs**

```
175
176 private void Move()
177 {
178
179     float _moveDirX = Input.GetAxisRaw("Horizontal"); //A,D를 눌렀을 때
180     float _moveDirZ = Input.GetAxisRaw("Vertical"); //W,S를 눌렀을 때
181
182     Vector3 _moveHorizontal = transform.right * _moveDirX; //Player의 transform X축을 이용하여 좌, 우 이동하도록 정의
183     Vector3 _moveVertical = transform.forward * _moveDirZ; //Player의 transform Z축을 이용하여 앞, 뒤 이동하도록 정의
184
185     Vector3 _velocity = (_moveHorizontal + _moveVertical).normalized * applySpeed; //속도는 방향이 움직이면 속도값을 곱하여 얼마나 빨리 이동할 것인지 설정한다.
186
187     myRigid.MovePosition(transform.position + _velocity * Time.deltaTime); //Rigidbody에 움직임 함수를 정의하여 Player의 위치에 속도를 더해주고, 1초동안 속도만큼 움직이게 하겠다.
188
189     if (_moveDirZ > 0.05f || _moveDirZ < -0.05f) //만약 A,D를 누르고 있으면
190     {
191         playerAnim.SetBool("Run", true); //뛰어라
192     }
193     else
194     {
195         playerAnim.SetBool("Run", false); //멈춰라
196     }
197
198
199     if (Input.GetMouseButtonDown(0)) //왼쪽 마우스 버튼을 누르면
200     {
201         playerAnim.SetBool("Hit", true); //때리는 애니메이션을 실행
202     }
203
204     if (Input.GetMouseButtonUp(0)) //왼쪽 마우스 버튼이 올라오면
205     {
206         playerAnim.SetBool("Hit", false); //때리는 애니메이션을 멈춰라
207     }
208
209     if (Input.GetMouseButtonDown(1)) //오른쪽 마우스 버튼을 누르면
210     {
211         playerAnim.SetBool("Dig", true); //총쏘는 애니메이션을 실행
212         audioBullet.Play();
213     }
214
215     if (Input.GetMouseButtonUp(1)) //오른쪽 마우스 버튼이 올라오면
216     {
217         playerAnim.SetBool("Dig", false); //총쏘는 애니메이션을 멈춰라
218     }
219 }
```



# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- Player 움직임 및 기능: **PlayerControl3.cs**

```
223 private void CharacterRotation()
224 {
225     // 좌우 캐릭터 회전
226     float _yRotation = Input.GetAxisRaw("Mouse X"); //마우스를 좌 우로 움직였을 때, 카메라가 좌 우(Y축)으로 움직인다.
227     Vector3 _characterRotationY = new Vector3(0f, _yRotation, 0f) * lookSensitivity; //카메라가 천천히 움직이도록 설정한다.
228     myRigid.MoveRotation(myRigid.rotation * Quaternion.Euler(_characterRotationY)); //Player의 회전값에 천천히 움직이도록 설정한 카메라 값을 곱하여 회전하도록 한다.
229 }
230
231 private void CameraRotation()
232 {
233     //상 하 카메라 회전
234     float _xRotation = Input.GetAxisRaw("Mouse Y"); //마우스를 앞 뒤로 움직였을 때, 카메라가 상 하(X축)으로 움직인다.
235     float _cameraRotationX = _xRotation * lookSensitivity; //카메라가 천천히 움직이도록 설정한다.
236     currentCameraRotationX += _cameraRotationX; //현재 카메라 움직임은 마우스를 앞 뒤로 움직이는 값에 따라 움직인다.
237     currentCameraRotationX = Mathf.Clamp(currentCameraRotationX, -cameraRotationLimit, cameraRotationLimit); //cameraRotationLimit의 최대, 최소값을 결정한다.(설정한 각도만큼 이상 움직이지 못하도록 제한을 둔다.)
238
239     theCamera.transform.localEulerAngles = new Vector3(currentCameraRotationX, 0f, 0f); //실제 카메라에 적용시키기 위한 설정을 한다. 마우스를 앞 뒤로 움직였을 때 카메라가 상 하(X축)로만 움직이도록 설정한다.
240 }
241
242 void OnCollisionEnter(Collision collision)
243 {
244
245     if (collision.gameObject.name == "potion_yellow") //충돌한 물체가 물약이라면
246     {
247         gameManager._Mysteriouspotion += 1; //신비의 물약이 증가한다.
248         PotionSource01.PlayOneShot(PotionSource); //소리가 재생된다.
249     }
250 }
251
252
```

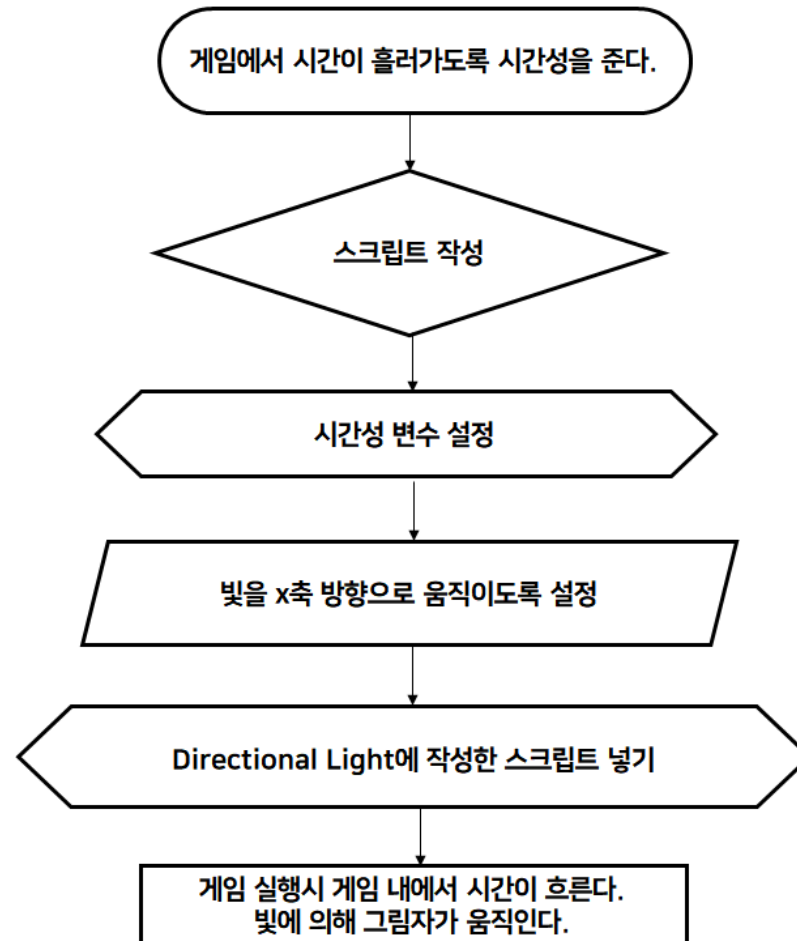
# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- 게임 시간성(낮과 밤)

게임 시간성  
(낮과 밤)

- 게임에서 시간의 흐름을 준다.  
DayandNight.cs



## Mysterious Forest\_

### 03. 'Mysterious Forest' 코딩 및 순서도

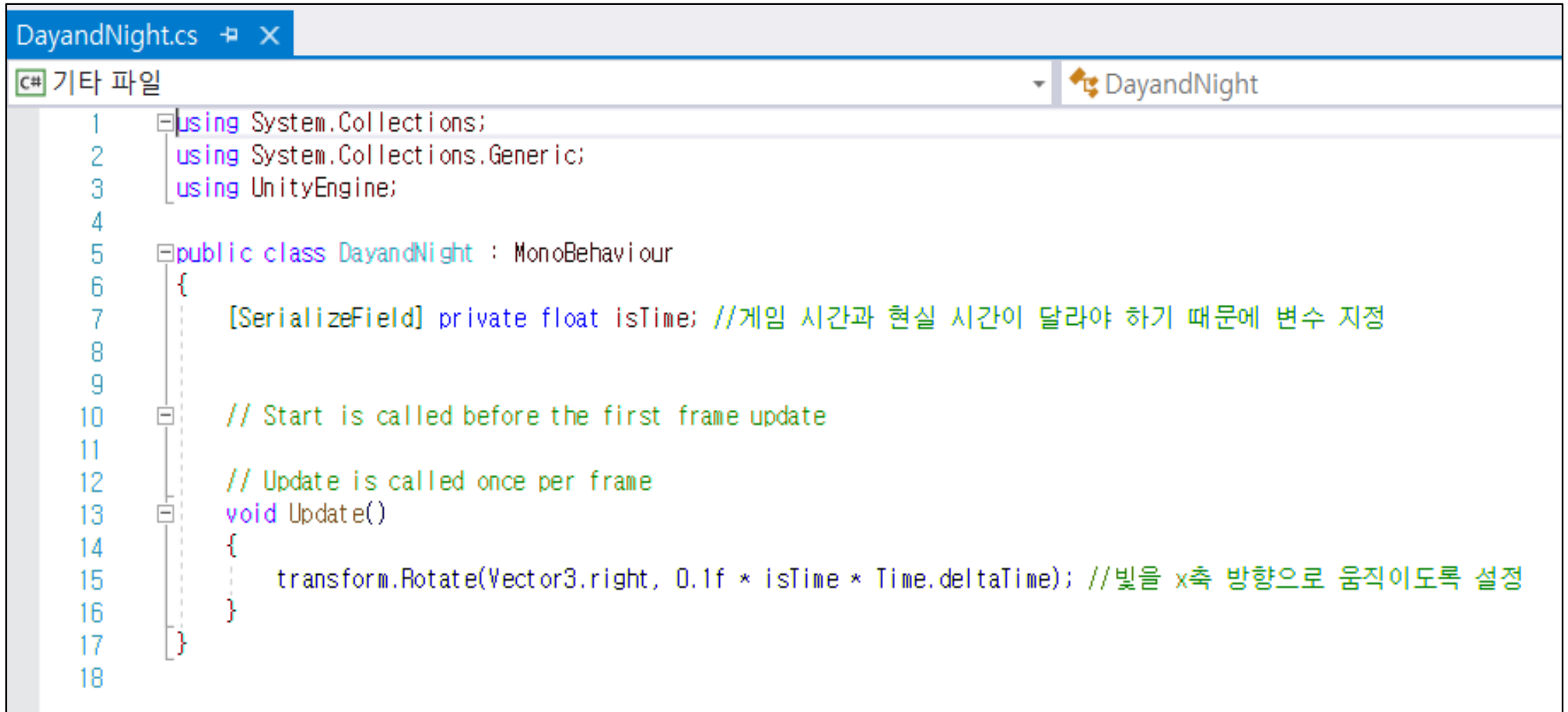
- 게임 시간성(낮과 밤)

- ▶ 게임에서 시간이 흘러가도록 시간성을 준다.
- ▶ Directional Light에 넣을 스크립트 작성한다.
  - ▶ 시간성 변수를 설정한다.
  - ▶ 빛을 x축 방향으로 움직이도록 설정한다.
- ▶ Directional Light에 작성한 스크립트를 넣어준다.
- ▶ 게임 실행시 게임 내에서 시간이 흐른다.  
(빛에 의해 그림자가 움직이는 것으로 알 수 있다.)

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- 게임 시간성(낮과 밤): **DayandNight.cs**

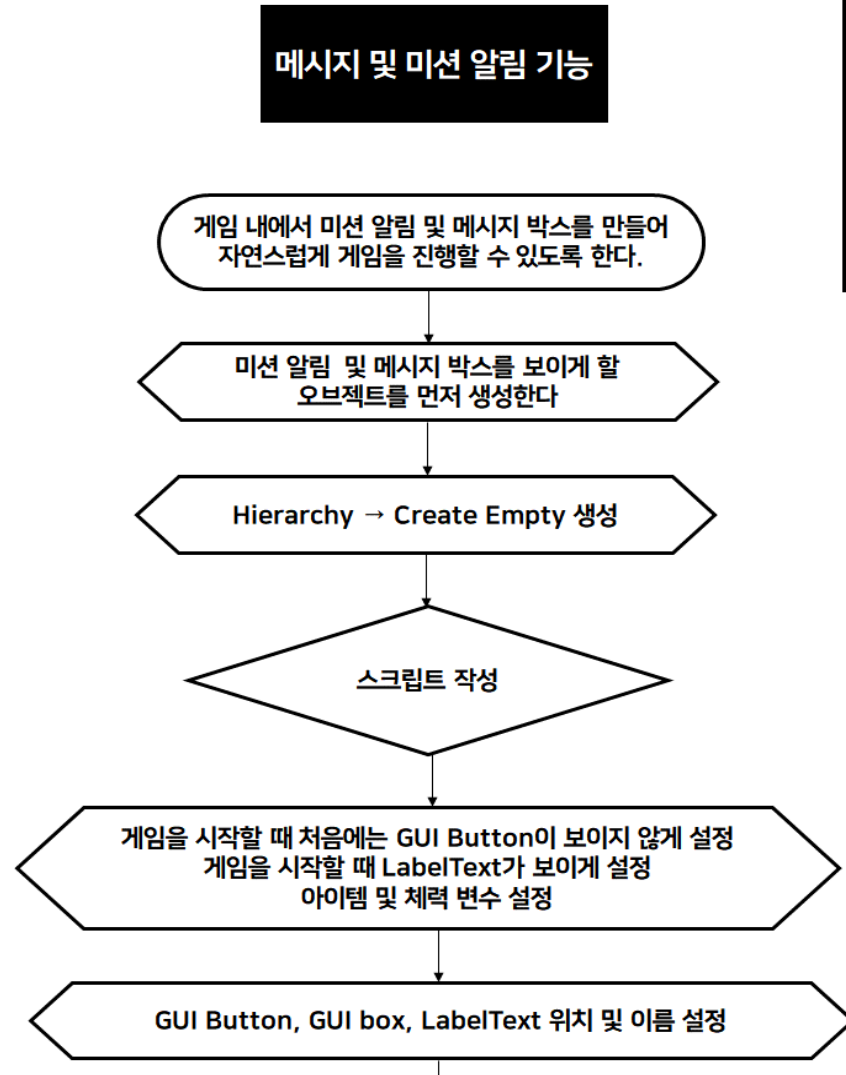


```
DayandNight.cs
C# 기타 파일
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class DayandNight : MonoBehaviour
6 {
7     [SerializeField] private float isTime; //게임 시간과 현실 시간이 달라야 하기 때문에 변수 지정
8
9
10    // Start is called before the first frame update
11
12    // Update is called once per frame
13    void Update()
14    {
15        transform.Rotate(Vector3.right, 0.1f * isTime * Time.deltaTime); //빛을 x축 방향으로 움직이도록 설정
16    }
17 }
18
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- GUI 설정



- GUI 설정

GameManger.cs (메인 게임 밤)

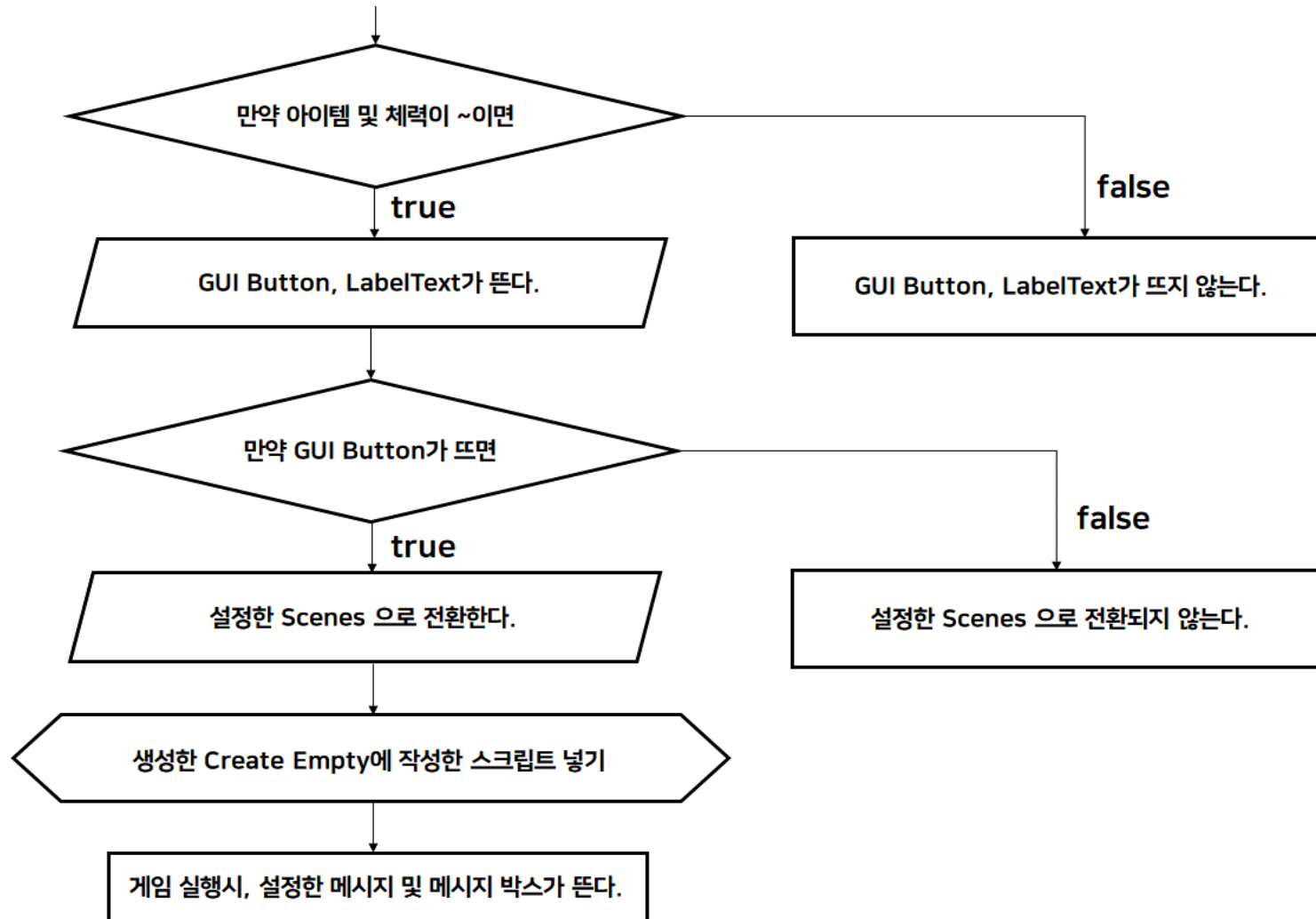
GameManger2.cs (메인 게임 낮)

GameManger3.cs (신비의 물약)

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

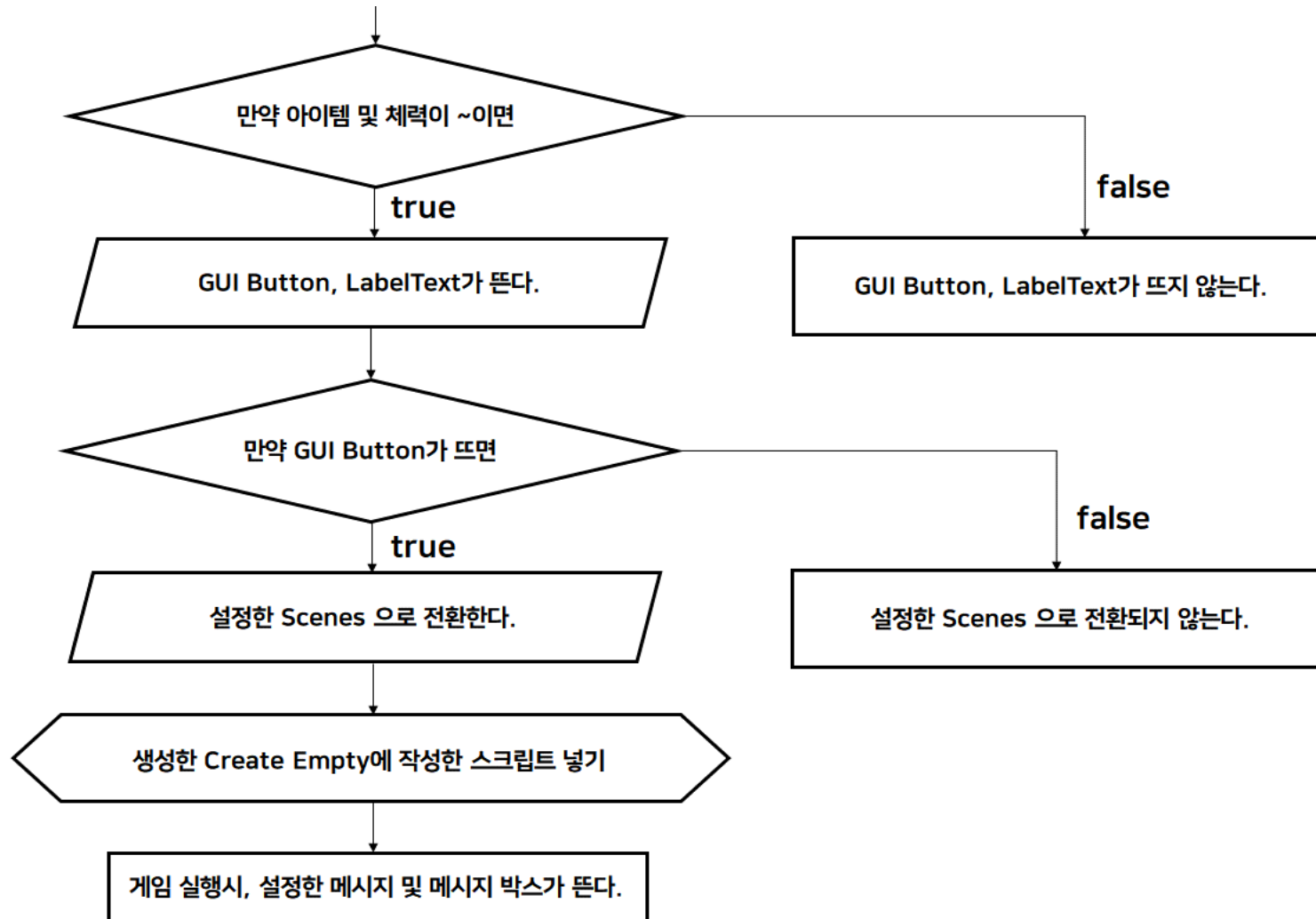
- GUI 설정



# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- GUI 설정



# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

### - GUI 설정

- ▶ 게임 내에서 미션 알림 및 메시지 박스를 만들어 자연스럽게 게임을 진행할 수 있도록 한다.
- ▶ 먼저 미션 알림 및 메시지 박스를 보이게 할 오브젝트를 먼저 생성한다.
- ▶ 게임 내 Hierarchy창에서 Create Empty를 생성한다.
- ▶ 생성한 Create Empty에 넣어줄 스크립트를 작성한다.
  - ▶ 게임을 시작할 때 처음에는 GUI Button이 보이지 않게 설정한다.
  - ▶ 게임을 시작할 때 LabelText가 보이게 설정한다.
  - ▶ 아이템 및 체력 변수를 설정한다.
  - ▶ GUI Button, GUI Box, LabelText 위치 및 이름을 설정한다.
  - ▶ 만약 아이템 및 체력이 ( )이면(true)
  - ▶ GUI Button과 LabelText가 뜬다.



# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

### - GUI 설정

- ▶ 만약 아이템 및 체력이 ( )이 아니면 (false)
- ▶ GUI Button과 LabelText가 뜨지 않는다.
- ▶ 만약 GUI Button이 뜨면 (true)
- ▶ 설정한 Scenes로 전환된다.
- ▶ 만약 GUI Button이 뜨지 않으면 (false)
- ▶ 설정한 Scenes로 전환되지 않는다.
- ▶ 생성한 Create Empty에 작성한 스크립트를 넣어준다.
- ▶ 게임 실행시, 설정한 메시지 및 메시지 박스가 뜬다.

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- GUI 설정: **GameManager.cs**

```
GameManager.cs ➤ X
C# 기타 파일
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  public class GameManager : MonoBehaviour
7  {
8      public string labelText = "낮이 되기 전까지 좀비를 10마리 죽이자!"; //labelText가 보이도록 설정
9      private int playerHP = 10; //Player 체력 설정
10     public bool dieWindow = false; //처음에는 보이지 않게 설정
11     public bool winWindow = false; //처음에는 보이지 않게 설정
12
13     public int HP //체력 함수
14     {
15         get { return playerHP; } //Player 체력값 다시 반복
16         set
17         {
18             playerHP = value; //player체력 값 저장
19
20             if (playerHP <= 0) //player 체력이 0이면
21             {
22                 dieWindow = true; //설정한 GUI 창이 나온다.
23                 labelText = ""; //labelText가 보이지 않게 한다.
24             }
25         }
26     }
27
28     private int playerScore = 0; //player가 좀비를 죽이면 스코어가 올라가는 함수
29     public int Score //스코어 함수
30     {
31         get { return playerScore; } //스코어 값 다시 반복
32         set
33         {
34             playerScore = value; //스코어 함수 값 저장
35
36             if (playerScore >= 7) //스코어가 10이상이면
37             {
38                 winWindow = true; //설정한 GUI 창이 나온다.
39                 labelText = ""; //처음에 설정한 labelText 안보이게 설정
40             }
41         }
42     }
43 }
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

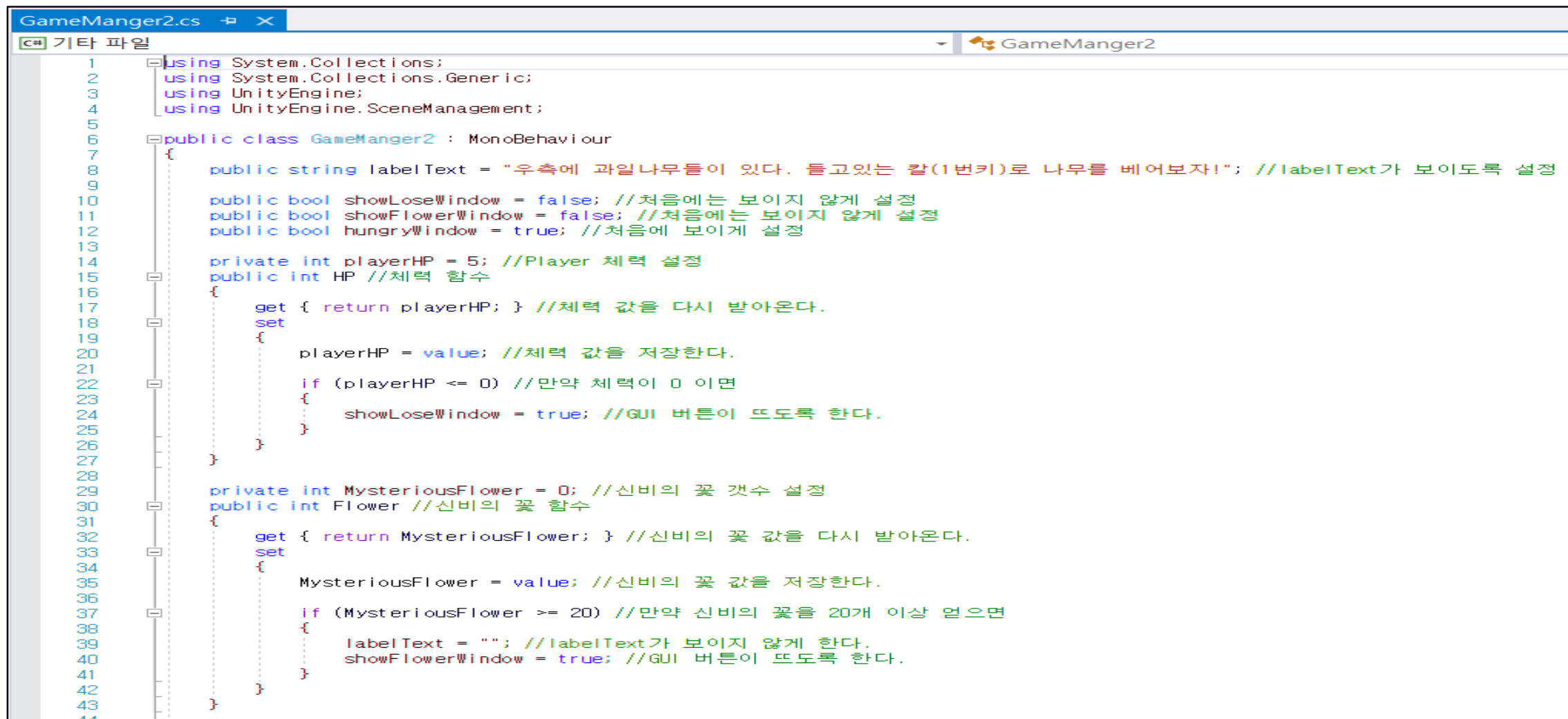
- GUI 설정: **GameManager.cs**

```
43
44
45 // Update is called once per frame
46
47 void OnGUI()
48 {
49     GUI.Box(new Rect(20, 20, 150, 25), "Player HP: " + playerHP); //GUI 박스 위치 및 이름 설정
50     GUI.Box(new Rect(750, 20, 150, 25), "Score: " + playerScore); //GUI 박스 위치 및 이름 설정
51     GUI.Label(new Rect(Screen.width / 2 - 100, Screen.height - 50, 350, 50), labelText); //GUI 박스 위치 및 이름 설정
52
53     if (dieWindow) //만약 dieWindow가 true면
54     {
55         if (GUI.Button(new Rect(Screen.width / 2 - 100, Screen.height / 2 - 50, 200, 100), "DIE")) //GUI Button 위치 및 이름 설정
56         {
57             SceneManager.LoadScene("OverNightScene"); //밤을 배경으로 하는 게임오버씬으로 전환
58         }
59     }
60
61     if (winWindow) //만약 winWindow가 true면
62     {
63         if (GUI.Button(new Rect(Screen.width / 2 - 100, Screen.height / 2 - 50, 200, 100), "곧 날이 밝아옵니다!")) //GUI Button 위치 및 이름 설정
64         {
65             SceneManager.LoadScene("PortionScene"); //물약씬으로 전환
66         }
67     }
68 }
69
70
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- GUI 설정: **GameManger2.cs**



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class GameManger2 : MonoBehaviour
7 {
8     public string labelText = "우측에 과일나무들이 있다. 들고있는 칼(1번키)로 나무를 베어보자!"; //labelText가 보이도록 설정
9
10    public bool showLoseWindow = false; //처음에는 보이지 않게 설정
11    public bool showFlowerWindow = false; //처음에는 보이지 않게 설정
12    public bool hungryWindow = true; //처음에 보이게 설정
13
14    private int playerHP = 5; //Player 체력 설정
15    public int HP //체력 함수
16    {
17        get { return playerHP; } //체력 값을 다시 받아온다.
18        set
19        {
20            playerHP = value; //체력 값을 저장한다.
21
22            if (playerHP <= 0) //만약 체력이 0 이면
23            {
24                showLoseWindow = true; //GUI 버튼이 뜨도록 한다.
25            }
26        }
27    }
28
29    private int MysteriousFlower = 0; //신비의 꽃 갯수 설정
30    public int Flower //신비의 꽃 함수
31    {
32        get { return MysteriousFlower; } //신비의 꽃 값을 다시 받아온다.
33        set
34        {
35            MysteriousFlower = value; //신비의 꽃 값을 저장한다.
36
37            if (MysteriousFlower >= 20) //만약 신비의 꽃을 20개 이상 얻으면
38            {
39                labelText = ""; //labelText가 보이지 않게 한다.
40                showFlowerWindow = true; //GUI 버튼이 뜨도록 한다.
41            }
42        }
43    }
44 }
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- GUI 설정: **GameManager2.cs**

```
45 private int Food = 0; //과일 체력 설정
46 public int food //과일 체력 함수
47 {
48     get { return Food; } //과일 체력 값을 다시 받아온다.
49     set
50     {
51         Food = value; //과일 값을 저장한다.
52
53         if (Food >= 5) //만약 과일을 5개 이상 얻으면
54         {
55             labelText = "이제 신비의 물약 재료에 필요한 물약 2개를 모으자! 곡괭이(4번키)로 바위를 깨면 물약이 나온다."; //labelText가 보이도록 한다.
56         }
57     }
58 }
59
60 private int Rock = 0; //바위 체력 설정
61 public int rock //바위 함수
62 {
63     get { return Rock; } //바위 체력 값을 다시 받아 온다.
64     set
65     {
66         Rock = value; //바위 값을 저장한다.
67
68         if (Rock >= 2) //바위에서 나온 아이템(물약)이 2개 이상이면
69         {
70             labelText = "마지막으로 신비의 물약을 얻기 위한 재료인 꽃을 20송이 모으자!"; //labelText가 보이도록 한다.
71         }
72     }
73 }
74
75 // Update is called once per frame
76 void OnGUI ()
77 {
78     GUI.Box(new Rect(20, 20, 150, 25), "Player HP: " + playerHP); //GUI 박스 위치 및 이름 설정
79     GUI.Box(new Rect(20, 50, 150, 25), "Food: " + Food); //GUI 박스 위치 및 이름 설정
80     GUI.Box(new Rect(20, 80, 150, 25), "Rock: " + Rock); //GUI 박스 위치 및 이름 설정
81     GUI.Box(new Rect(20, 110, 150, 25), "Mysterious Flower: " + MysteriousFlower); //GUI 박스 위치 및 이름 설정
82     GUI.Label(new Rect(Screen.width / 3, Screen.height - 40, 600, 100), labelText); //GUI Label 위치 설정
83
84     if (showLoseWindow) //만약 showLoseWindow가 true면
85     {
86         if (GUI.Button(new Rect(Screen.width / 2 - 100, Screen.height / 2 - 50, 200, 100), "DIE")) //GUI Button 위치 및 이름 설정
87         {
88             SceneManager.LoadScene("OverScene"); //나트 게임 오버 신으로 전환
89         }
90     }
```

## Mysterious Forest\_

### 03. 'Mysterious Forest' 코딩 및 순서도

- GUI 설정: **GameManager2.cs**

```
90     }
91
92     if (showFlowerWindow) //만약 showFlowerWindow가 true면
93     {
94         if (GUI.Button(new Rect(Screen.width / 2 - 150, Screen.height / 2 - 50, 495, 50), "신비의 꽃을 다모았군요! 신비의 물약은 다음날 날이 밝으면 완성됩니다!")) //GUI Button 위치 및 이름 설정
95         {
96             showFlowerWindow = false;
97             SceneManager.LoadScene("DemoNight"); //밤 씬으로 전환
98         }
99     }
100
101     if (hungryWindow) //만약 hungryWindow가 true면
102     {
103         if (GUI.Button(new Rect(Screen.width / 2 - 150, Screen.height / 2 - 50, 300, 50), "우선 과일을 먹어 배를 채우자!")) //GUI Button 위치 및 이름 설정
104         {
105             hungryWindow = false; //GUI 버튼이 뜨지 않도록 한다.
106         }
107     }
108 }
109
110
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

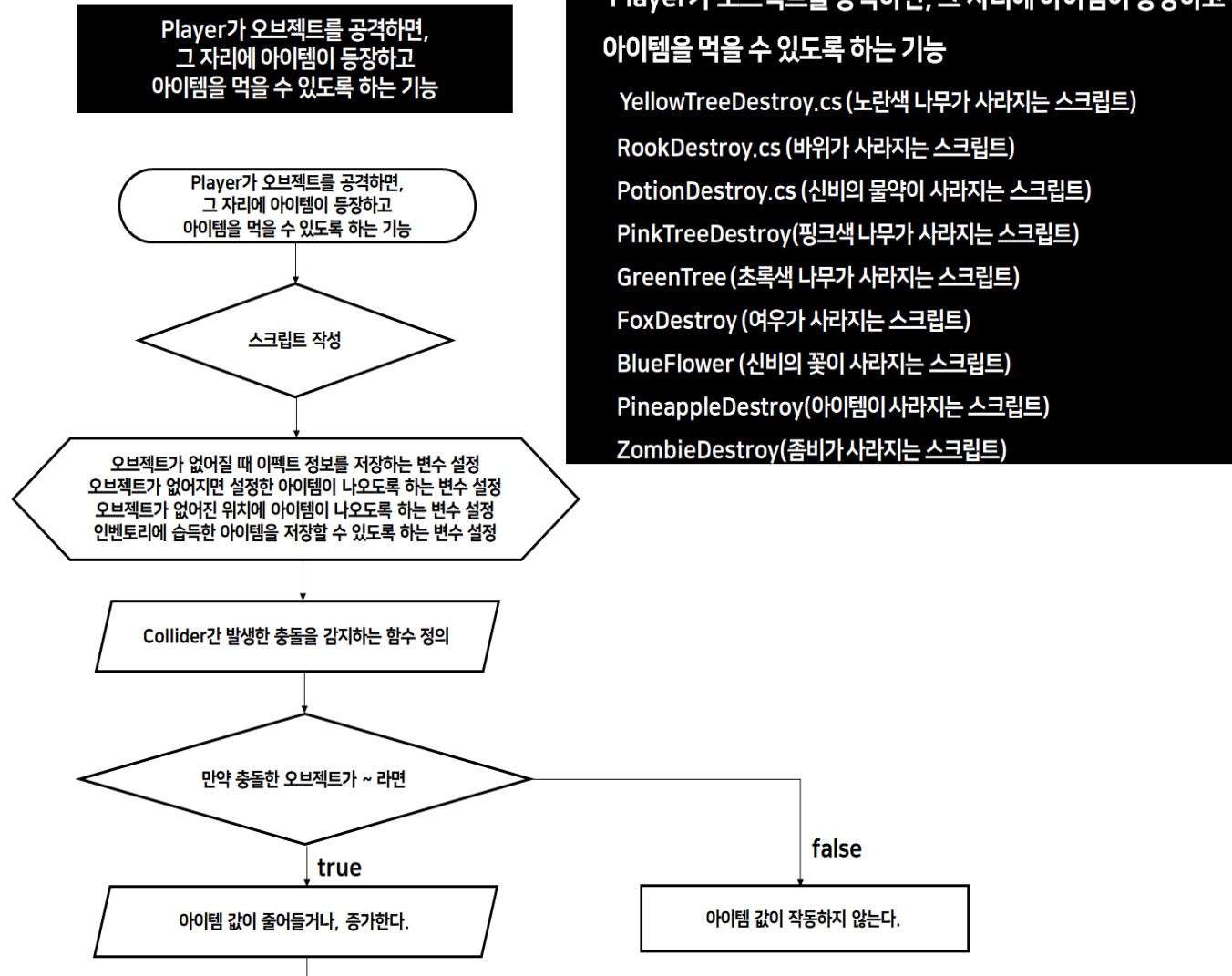
- GUI 설정: **GameManger3.cs**

```
GameManger3.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class GameManger3 : MonoBehaviour
7 {
8     public string labelText = "신비의 물약이 완성되었다!"; //labelText가 보이도록 설정
9
10    public bool showWinWindow = false; //처음에는 보이지 않게 설정
11
12    private int Mysteriouspotion = 0; //신비의 물약 값 설정
13    public int _Mysteriouspotion //신비의 물약 값 함수
14    {
15        get { return Mysteriouspotion; } //신비의 물약 값을 다시 불러온다.
16        set
17        {
18            Mysteriouspotion = value; //신비의 물약 값을 저장한다.
19
20            if (Mysteriouspotion >= 1) //만약 신비의 물약을 획득했을 경우
21            {
22                labelText = ""; //labelText가 보이지 않게 한다.
23                showWinWindow = true; //GUI 버튼이 뜨도록 한다.
24            }
25        }
26    }
27
28    void OnGUI()
29    {
30        GUI.Label(new Rect(Screen.width / 2 - 100, Screen.height - 50, 350, 50), labelText); //GUI Label 위치 설정
31
32        if (showWinWindow)
33        {
34            GUI.Button(new Rect(Screen.width / 2 - 100, Screen.height / 2 - 50, 380, 100), "신비한 물약을 획득했습니다. 서둘러 딸에게 돌아가세요!"); //GUI Button 위치 및 이름 설정
35            {
36                SceneManager.LoadScene("Win Scene"); //게임 성공 씬으로 전환
37            }
38        }
39    }
40 }
41
42
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

### - 오브젝트 Destroy

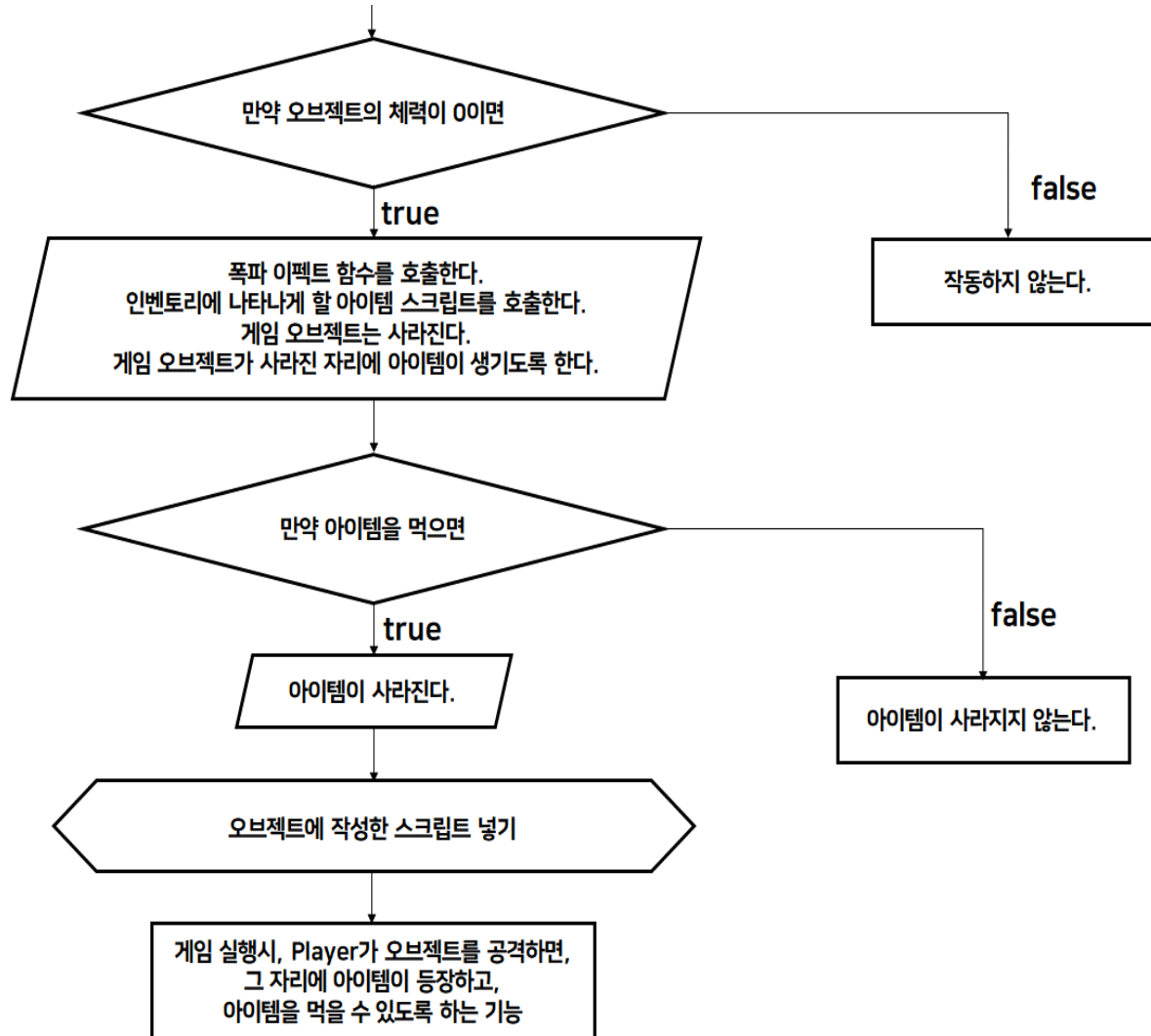




# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- 오브젝트 Destroy



# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- 오브젝트 Destroy

- ▶ Player가 오브젝트를 공격하면, 그 자리에 아이템이 등장하고, 아이템을 먹을 수 있도록 하는 기능을 추가한다.
- ▶ 다양한 오브젝트에 적용할 스크립트를 작성한다.
  - ▶ 오브젝트가 없어질 때 이펙트 정보를 저장하는 변수를 설정한다.
  - ▶ 오브젝트가 없어지면 설정한 아이템이 나오도록 하는 변수를 설정한다.
  - ▶ 오브젝트가 없어진 위치에 아이템이 나오도록 하는 변수를 설정한다.
  - ▶ 인벤토리에 습득한 아이템을 저장할 수 있도록 하는 변수를 설정한다.
  - ▶ Collider간 발생한 충돌을 감지하는 함수를 정의한다.
  - ▶ 만약 충돌한 오브젝트가 ( )라면
    - ▶ 아이템 값이 줄어들거나, 증가한다(true)
    - ▶ 아이템 값이 작동하지 않는다(false)

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- 오브젝트 Destroy

- ▶ 만약 오브젝트의 체력이 0이면
- ▶ 폭파 이펙트 함수를 호출한다.(true)
- ▶ 인벤토리에 나타나게 할 아이템 스크립트를 호출한다.(true)
- ▶ 게임 오브젝트는 사라진다.(true)
- ▶ 게임 오브젝트가 사라진 자리에 아이템이 생기도록 한다.(true)
- ▶ 게임 오브젝트가 사라지지 않는다. 아이템이 생기지 않고 이펙트도 생기지 않는다.(false)
- ▶ 만약 아이템을 먹으면
- ▶ 아이템이 사라진다.(true)
- ▶ 아이템이 사라지지 않는다.(false)
- ▶ 다양한 오브젝트에 작성한 스크립트를 넣는다.
- ▶ 게임 실행시 Player가 오브젝트를 공격하면, 그 자리에 아이템이 등장하고, 아이템을 먹을 수 있게 된다.

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- 오브젝트 Destroy: **YellowTreeDestroy.cs**

```
YellowTreeDestroy.cs
[기타 파일] YellowTreeDestroy

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class YellowTreeDestroy : MonoBehaviour
6  {
7      public ParticleSystem yellowtree; //나무가 없어질 때, 이펙트 정보를 저장하는 변수를 설정한다.
8      public GameObject PineApple; //나무가 없어지면 설정한 파인애플이 나오도록 하는 변수 설정
9      public Transform TreePost; //나무가 없어진 위치에 파인애플이 나오도록하는 변수 설정
10
11     int YellowTreeHp = 3; //나무의 체력변수 설정한다.
12
13     [SerializeField]
14     private Inventory theInventory; //인벤토리에 습득한 아이템을 저장할 수 있도록 하는 변수 설정
15
16     void OnCollisionEnter(Collision col) //Collider간 발생한 충돌을 감지하는 함수 설정
17     {
18         if (col.gameObject.name == "w_sword_B") //만약 충돌하는 물체가 칼이라면
19         {
20             YellowTreeHp -= 1; //나무의 체력이 1 줄어든다.
21         }
22     }
23
24     // Update is called once per frame
25     void Update()
26     {
27         if (YellowTreeHp == 0) //만약 나무의 체력이 0이라면
28         {
29             ParticleSystem fire = Instantiate(yellowtree, transform.position, Quaternion.identity); //폭파 이펙트 함수 호출한다.
30             fire.Play();
31             theInventory.AcquireItem(GetComponent<ItemPickUp>().item); // ItemPickUp안에 item를 인벤토리에 저장한다.
32             Destroy(gameObject); //나무는 사라진다.
33             Instantiate(PineApple, TreePost.position, TreePost.rotation); //나무가 없어진 자리에 파인애플이 뜨는 함수 코드
34         }
35     }
36 }
37
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- 오브젝트 Destroy: **RockDestroy.cs**

```
RookDestroy.cs  + X
[C#] 기타 파일  RookDestroy

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class RookDestroy : MonoBehaviour
6  {
7      public GameObject Potion; //바위가 없어지면 설정한 물약이 나오도록 하는 변수 설정
8      public Transform RockPost; //바위가 없어진 위치에 물약이 나오도록하는 변수 설정
9
10     int Rock = 5; //바위의 체력변수 설정한다.
11
12     [SerializeField]
13     private Inventory theInventory; //인벤토리에 습득한 아이템을 저장할 수 있도록 하는 변수 설정
14
15     void OnCollisionEnter(Collision col) //Collider간 발생한 충돌을 감지하는 함수 설정
16     {
17         if (col.gameObject.name == "Pickaxe") //만약 충돌하는 물체가 곡괭이라면
18         {
19             Rock -= 1; //바위의 체력이 -1 작인다.
20         }
21     }
22
23     // Update is called once per frame
24     void Update()
25     {
26         if (Rock == 0) //만약 바위의 체력이 0이라면
27         {
28             theInventory.AcquireItem(GetComponent<ItemPickUp>().item); //ItemPickUp안에 item를 인벤토리에 저장한다.
29             Destroy(gameObject); //바위는 사라진다.
30             Instantiate(Potion, RockPost.position, RockPost.rotation); //바위가 없어진 자리에 물약이 뜨는 함수 코드
31         }
32     }
33 }
34
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- 오브젝트 Destroy: **PotionDestroy.cs**

```
PotionDestroy.cs  [icon] [X]
C# 기타 파일
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class PotionDestroy : MonoBehaviour
6  {
7
8      void OnCollisionEnter(Collision collision)
9      {
10         if (collision.gameObject.name == "Player") // 만약 충돌대상이 Player라면
11         {
12             Destroy(gameObject); // 물약이 사라진다.
13         }
14     }
15 }
16
17
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- 오브젝트 Destroy: **PinkTreeDestroy.cs**

```
PinkTreeDestroy.cs
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PinkTreeDestroy : MonoBehaviour
{
    public ParticleSystem Pinktree; //나무가 없어질 때, 이펙트 정보를 저장하는 변수를 설정한다.
    public GameObject tomato; //나무가 없어지면 설정한 토마토가 나오도록 하는 변수 설정
    public Transform PinkTreePost; //나무가 없어진 위치에 토마토가 나오도록하는 변수 설정

    int PinkTreeHp = 3; //나무의 체력변수 설정한다.

    [SerializeField]
    private Inventory theInventory;

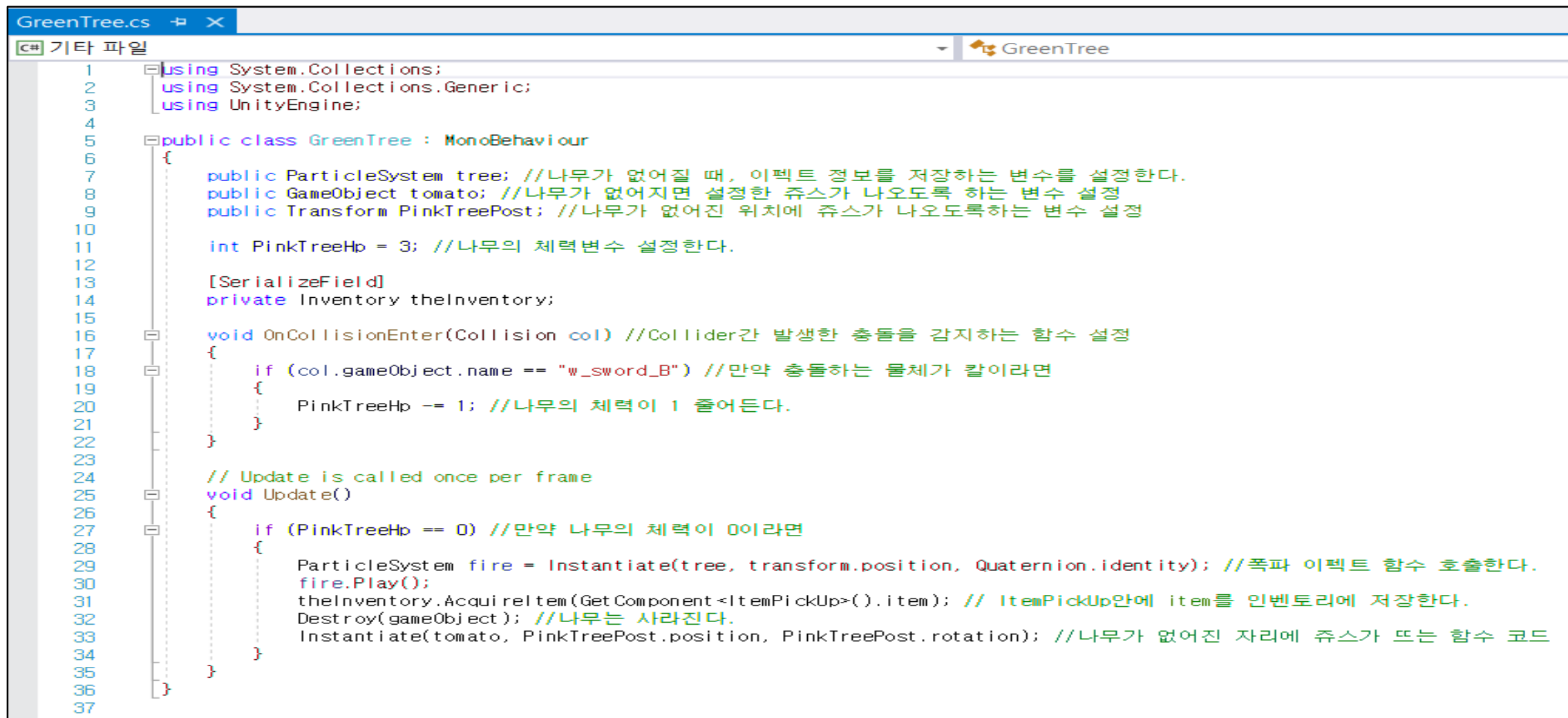
    void OnCollisionEnter(Collision col) //Collider간 발생한 충돌을 감지하는 함수 설정
    {
        if (col.gameObject.name == "w_sword_B") //만약 충돌하는 물체가 칼이라면
        {
            PinkTreeHp -= 1; //나무의 체력이 1 줄어든다.
        }
    }

    // Update is called once per frame
    void Update()
    {
        if (PinkTreeHp == 0) //만약 나무의 체력이 0이라면
        {
            ParticleSystem fire = Instantiate(Pinktree, transform.position, Quaternion.identity); //폭파 이펙트 함수 호출한다.
            fire.Play();
            theInventory.AcquireItem(GetComponent<ItemPickUp>().item); // ItemPickUp안에 item을 인벤토리에 저장한다.
            Destroy(gameObject); //나무는 사라진다.
            Instantiate(tomato, PinkTreePost.position, PinkTreePost.rotation); //나무가 없어진 자리에 파인애플이 뜨는 함수 코드
        }
    }
}
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- 오브젝트 Destroy: **GreenTree.cs**



```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class GreenTree : MonoBehaviour
6  {
7      public ParticleSystem tree; //나무가 없어질 때, 이펙트 정보를 저장하는 변수를 설정한다.
8      public GameObject tomato; //나무가 없어지면 설정한 주스가 나오도록 하는 변수 설정
9      public Transform PinkTreePost; //나무가 없어진 위치에 주스가 나오도록하는 변수 설정
10
11     int PinkTreeHp = 3; //나무의 체력변수 설정한다.
12
13     [SerializeField]
14     private Inventory theInventory;
15
16     void OnCollisionEnter(Collision col) //Collider간 발생한 충돌을 감지하는 함수 설정
17     {
18         if (col.gameObject.name == "w_sword_B") //만약 충돌하는 물체가 칼이라면
19         {
20             PinkTreeHp -= 1; //나무의 체력이 1 줄어든다.
21         }
22     }
23
24     // Update is called once per frame
25     void Update()
26     {
27         if (PinkTreeHp == 0) //만약 나무의 체력이 0이라면
28         {
29             ParticleSystem fire = Instantiate(tree, transform.position, Quaternion.identity); //폭파 이펙트 함수 호출한다.
30             fire.Play();
31             theInventory.AcquireItem(GetComponent<ItemPickUp>().item); // ItemPickUp안에 item를 인벤토리에 저장한다.
32             Destroy(gameObject); //나무는 사라진다.
33             Instantiate(tomato, PinkTreePost.position, PinkTreePost.rotation); //나무가 없어진 자리에 주스가 뜨는 함수 코드
34         }
35     }
36 }
37
```



# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- 오브젝트 Destroy: **FoxDestroy.cs**

```
FoxDestroy.cs  + X
C# 기타 파일 FoxDestroy
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class FoxDestroy : MonoBehaviour
6  {
7      // Start is called before the first frame update
8
9      private void OnCollisionEnter(Collision collision)
10     {
11         if (collision.gameObject.name == "_baseballbat 1 1") //만약 충돌대상이 방망이라면
12         {
13             Destroy(gameObject); //해당 오브젝트는 사라진다.
14         }
15     }
16 }
17
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- 오브젝트 Destroy: **BlueFlower.cs**

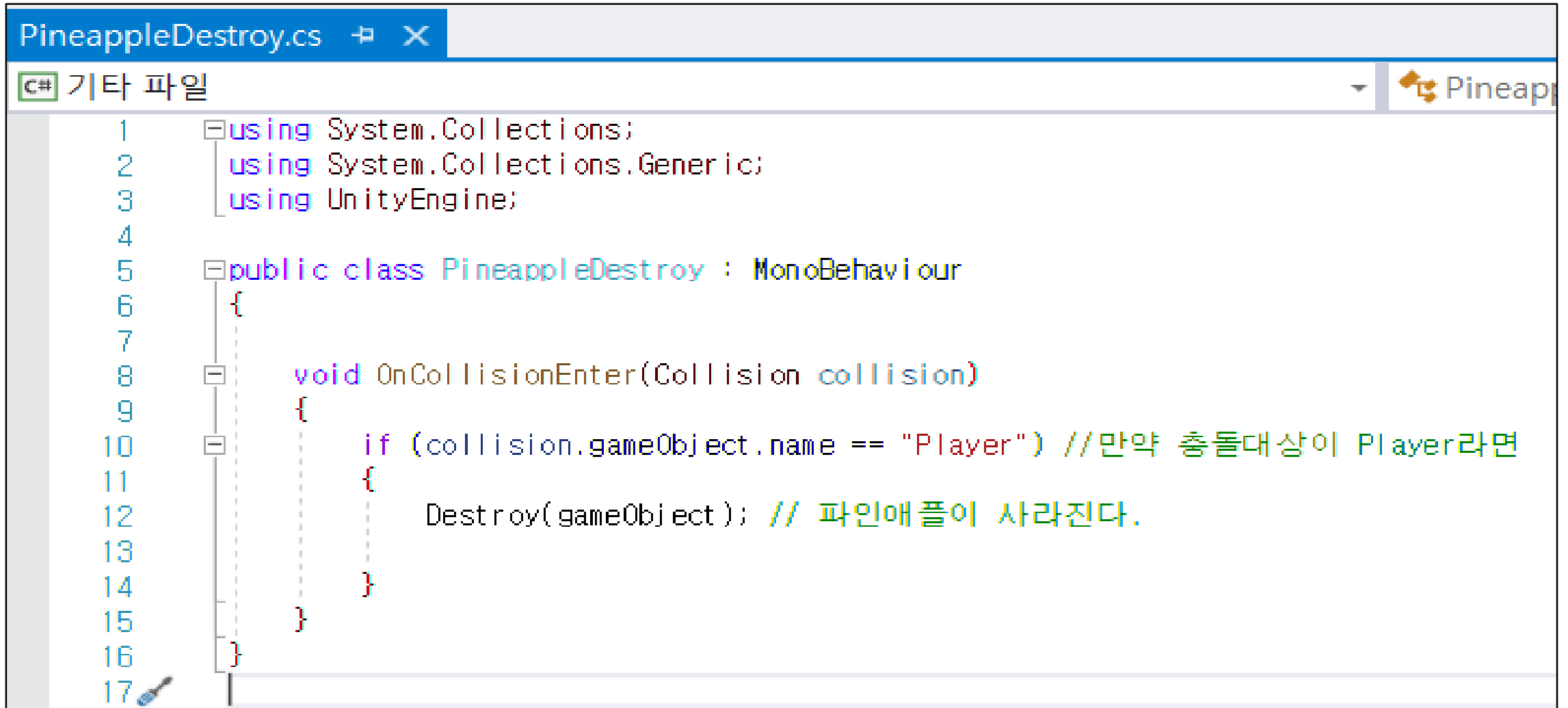
```
BlueFlower.cs  + X
C# 기타 파일  BlueFlower  theInventory

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5  using UnityEngine.UI;
6
7  public class BlueFlower : MonoBehaviour
8  {
9      [SerializeField]
10     private Inventory theInventory;
11
12     void OnCollisionEnter(Collision collision)
13     {
14         if (collision.gameObject.name == "Player") //만약 충돌대상이 Player라면
15         {
16             theInventory.AcquireItem(GetComponent<ItemPickUp>().item); // ItemPickUp안에 item를 인벤토리에 저장한다.
17             Destroy(gameObject); //꽃은 사라진다.
18         }
19     }
20 }
21
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- 오브젝트 Destroy: **PineappleDestroy.cs**



```
PineappleDestroy.cs
C# 기타 파일
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class PineappleDestroy : MonoBehaviour
6 {
7
8     void OnCollisionEnter(Collision collision)
9     {
10         if (collision.gameObject.name == "Player") // 만약 충돌대상이 Player라면
11         {
12             Destroy(gameObject); // 파인애플이 사라진다.
13         }
14     }
15 }
16
17
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

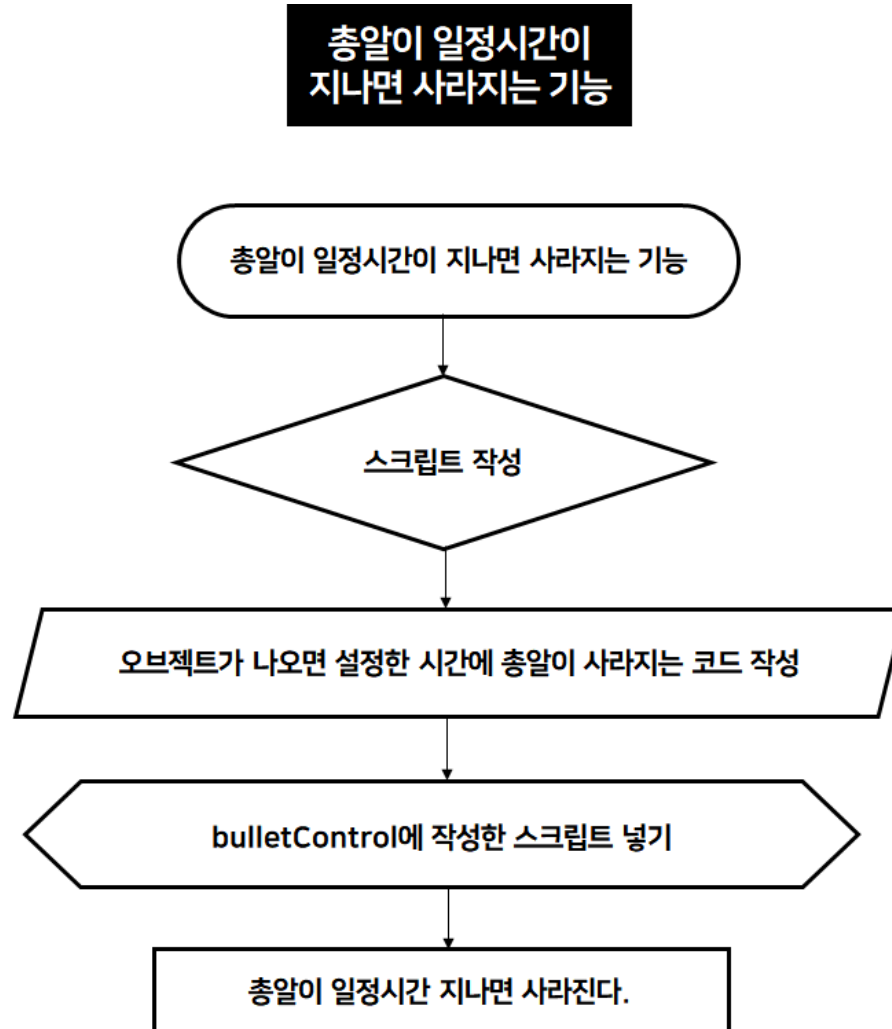
- 오브젝트 Destroy: **ZombieDestroy.cs**

```
ZombieDestroy.cs  X
C# 기타 파일
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class ZombieDestroy : MonoBehaviour
6  {
7      void OnCollisionEnter(Collision collision)
8      {
9          if (collision.gameObject.name == "Player") //충돌한 대상이 player라면
10         {
11             Destroy(gameObject); //좀비는 사라진다.
12         }
13     }
14
15 }
16
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- 총알



- 총알이 일정시간이 지나면 사라지는 기능.  
`bulletControl.cs`

## Mysterious Forest\_

### 03. 'Mysterious Forest' 코딩 및 순서도

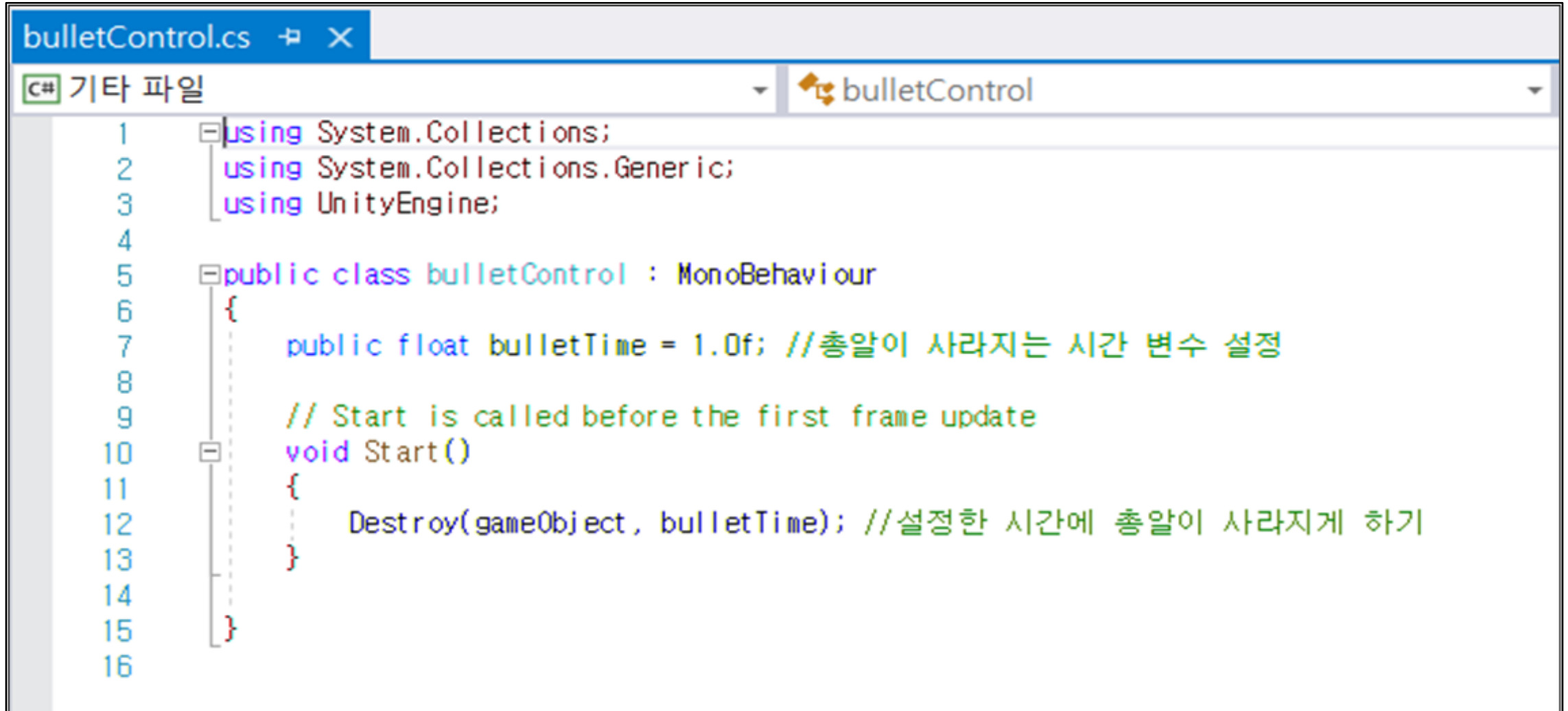
- 총알

- ▶ 총알이 일정시간이 지나면 사라지는 기능을 넣는다.
- ▶ 스크립트를 작성한다.
  - ▶ 오브젝트가 나오면 설정한 시간에 총알이 사라지는 코드 작성
- ▶ 작성한 스크립트를 총알에 넣는다.
- ▶ 총알을 발사했을 때, 총알이 일정시간이 지나면 사라진다.

## Mysterious Forest\_

### 03. 'Mysterious Forest' 코딩 및 순서도

- 총알: **bulletControl**



```
bulletControl.cs  X
C# 기타 파일 bulletControl
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class bulletControl : MonoBehaviour
6  {
7      public float bulletTime = 1.0f; //총알이 사라지는 시간 변수 설정
8
9      // Start is called before the first frame update
10     void Start()
11     {
12         Destroy(gameObject, bulletTime); //설정 한 시간에 총알이 사라지게 하기
13     }
14
15 }
16
```

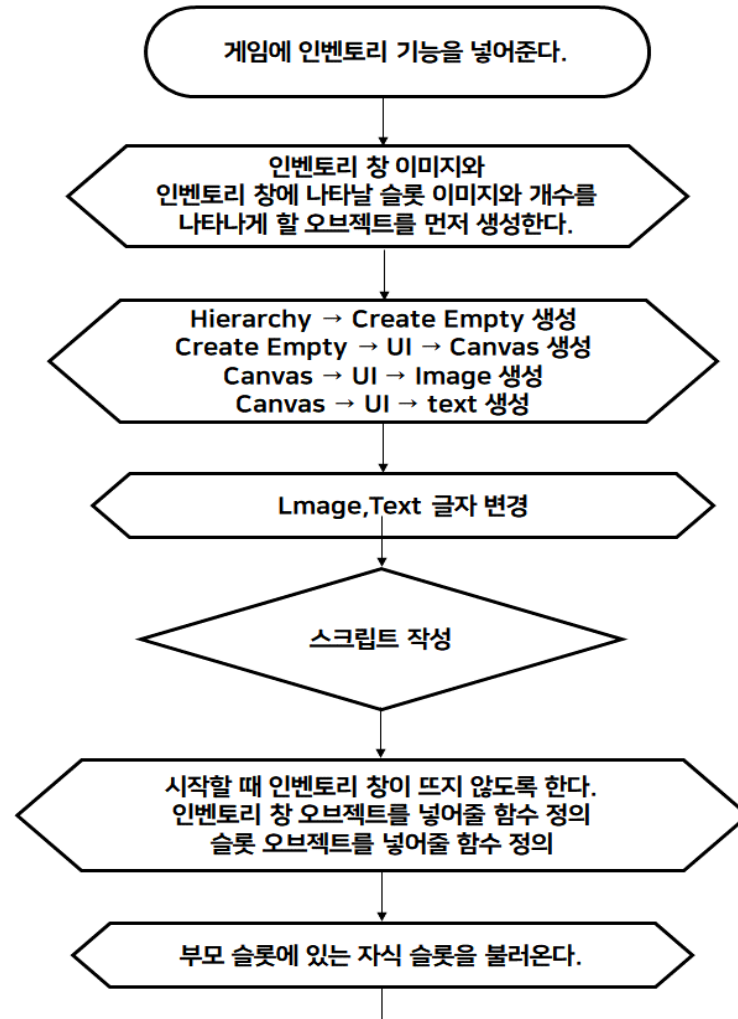
# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- 인벤토리

인벤토리

- 인벤토리 기능  
Inventory.cs

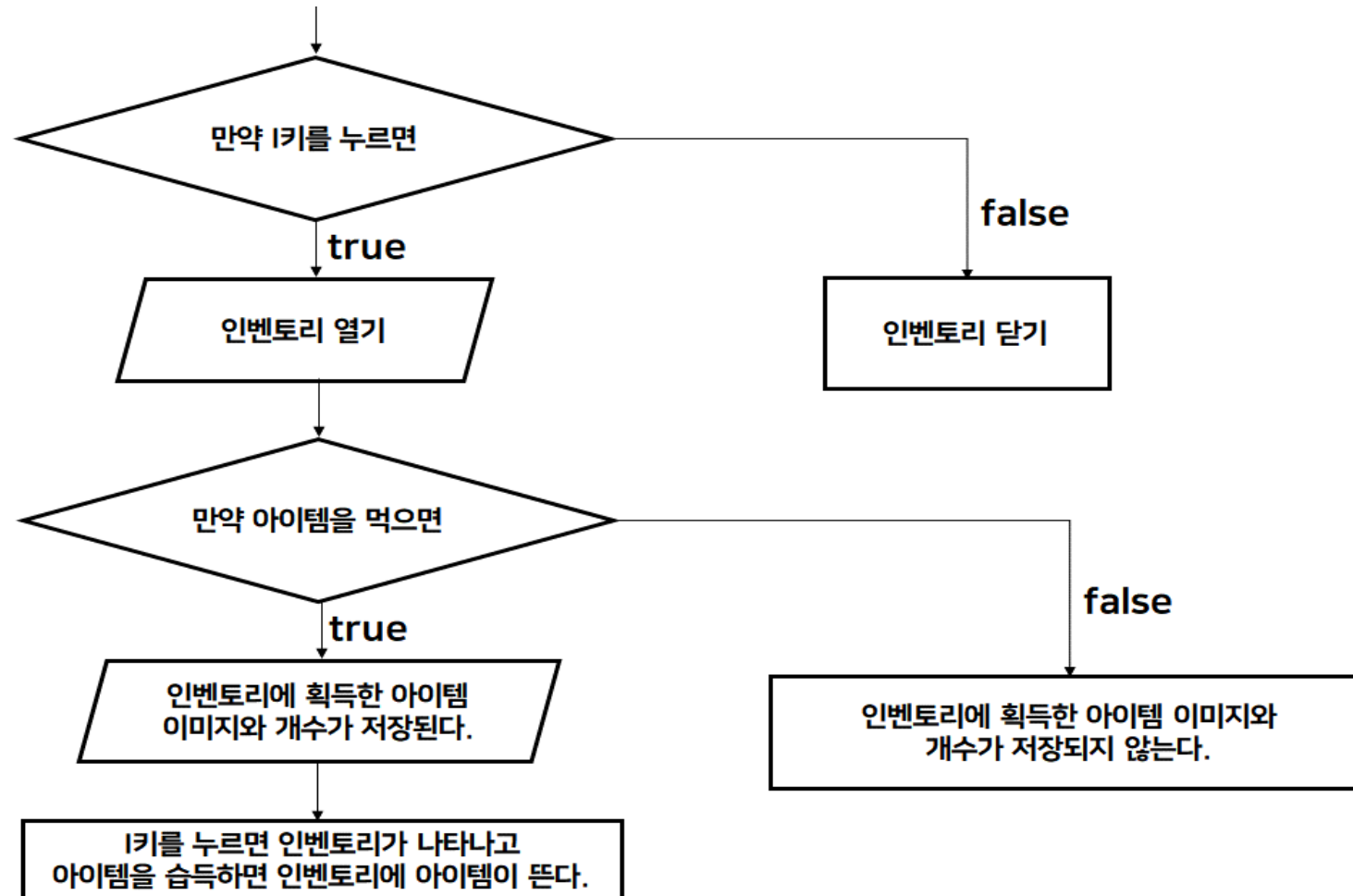




# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- 인벤토리



# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

### - 인벤토리

- ▶ 게임에 인벤토리 기능을 넣어준다.
- ▶ 우선 인벤토리 창 이미지와 인벤토리 창에 나타날 슬롯 이미지와 개수를 나타나게 할 오브젝트를 먼저 생성한다.
- ▶ Hierarchy창에 Create Empty을 생성한다.
- ▶ Create Empty에서 UI를 통해 Canvas 생성한다.
- ▶ Canvas에서 UI를 통해 Image 생성한다.
- ▶ Canvas에서 UI를 통해 text 생성한다.
- ▶ Lmage,Text 글자 변경한다.
- ▶ 스크립트를 작성한다.
  - ▶ 시작할 EO 인벤토리 창이 뜨지 않도록 한다.
  - ▶ 인벤토리 창 오브젝트를 넣어줄 함수를 정의한다.
  - ▶ 슬롯 오브젝트를 넣어줄 함수를 정의한다.

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

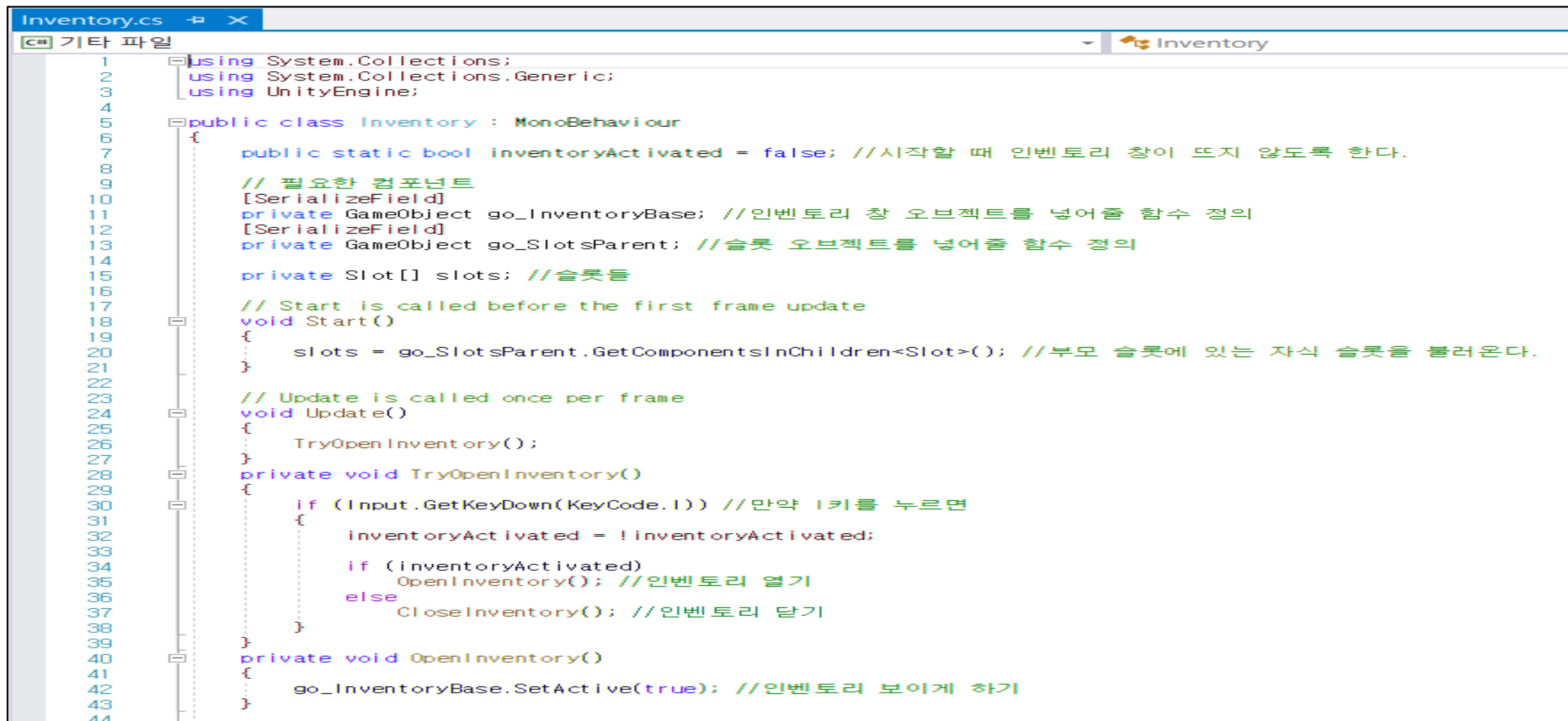
### - 인벤토리

- ▶ 부모 슬롯에 있는 자식 슬롯을 불러온다.
- ▶ 만약 I키를 누르면
- ▶ 인벤토리를 연다(true)
- ▶ 인벤토리가 열리지 않는다(false)
- ▶ 만약 아이템을 먹지 않으면
- ▶ 인벤토리에 획득한 아이템 이미지와 개수가 저장된다.(true)
- ▶ 인벤토리에 획득한 아이템 이미지와 개수가 저장되지 않는다.(false)
- ▶ 게임을 실행하면 인벤토리가 나타나고 아이템을 습득하면 인벤토리에 아이템이 뜬다.

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- 인벤토리: **Inventory.cs**



```
Inventory.cs # X
[C#] 기타 파일
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Inventory : MonoBehaviour
6 {
7     public static bool inventoryActivated = false; //시작할 때 인벤토리 창이 뜨지 않도록 한다.
8
9     // 필요한 컴포넌트
10    [SerializeField]
11    private GameObject go_InventoryBase; //인벤토리 창 오브젝트를 넣어줄 함수 정의
12    [SerializeField]
13    private GameObject go_SlotsParent; //슬롯 오브젝트를 넣어줄 함수 정의
14
15    private Slot[] slots; //슬롯들
16
17    // Start is called before the first frame update
18    void Start()
19    {
20        slots = go_SlotsParent.GetComponentInChildren<Slot>(); //부모 슬롯에 있는 자식 슬롯을 불러온다.
21    }
22
23    // Update is called once per frame
24    void Update()
25    {
26        TryOpenInventory();
27    }
28    private void TryOpenInventory()
29    {
30        if (Input.GetKeyDown(KeyCode.I)) //만약 I키를 누르면
31        {
32            inventoryActivated = !inventoryActivated;
33
34            if (inventoryActivated)
35                OpenInventory(); //인벤토리 열기
36            else
37                CloseInventory(); //인벤토리 닫기
38        }
39    }
40    private void OpenInventory()
41    {
42        go_InventoryBase.SetActive(true); //인벤토리 보이게 하기
43    }
44
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

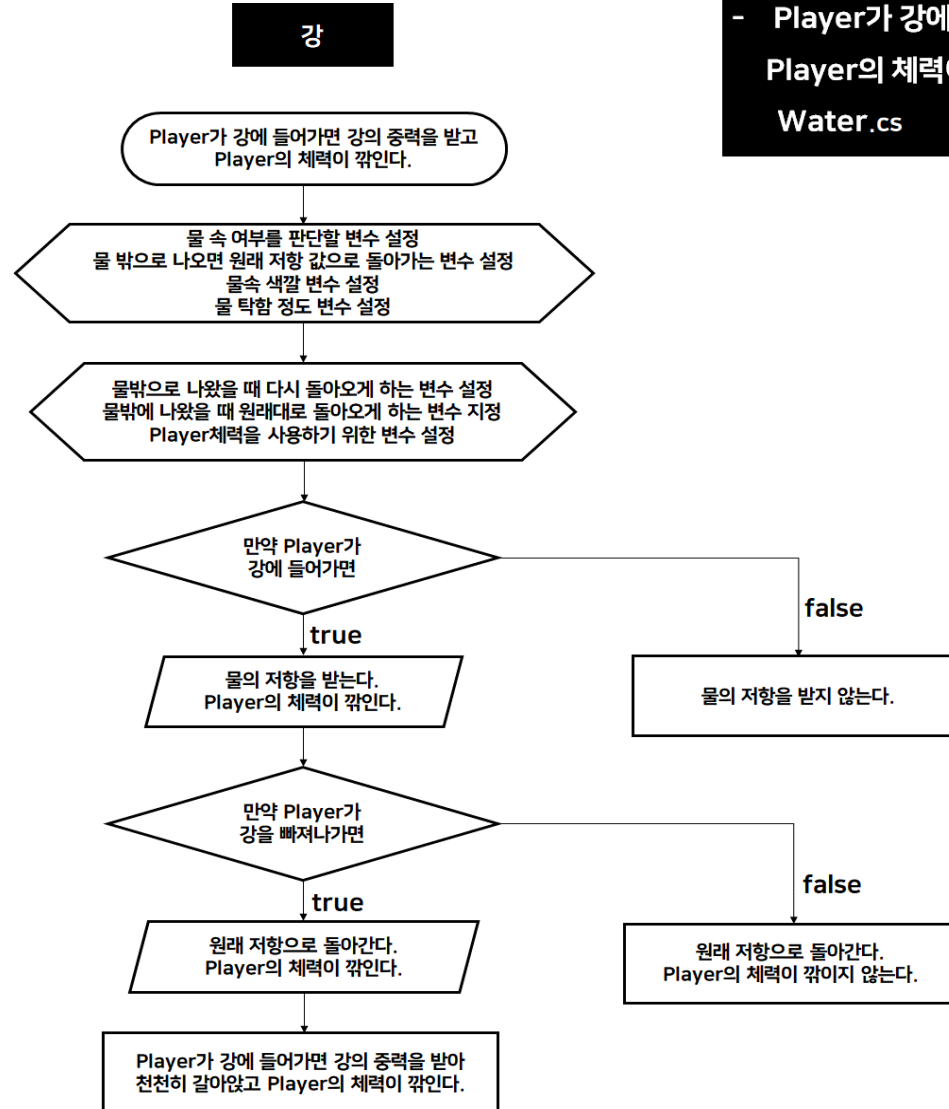
- 인벤토리: **Inventory.cs**

```
45 private void CloseInventory()  
46 {  
47     go_InventoryBase.SetActive(false); //인벤토리 보이지 않게 하기  
48 }  
49 public void AcquireItem(Item _item, int _count = 1) //아이템을 불러온다.  
50 {  
51     for (int i = 0; i < slots.Length; i++) //아이템을 먹으면 슬롯에 계속해서 추가한다.  
52     {  
53         if (slots[i].item != null) //만약 아이템의 슬롯이 비었으면  
54         {  
55             if (slots[i].item.itemName == _item.itemName) //만약 아이템의 이름 슬롯이 비었으면  
56             {  
57                 slots[i].SetSlotCount(_count); //슬롯에 나타나는 아이템 갯수를 증가하거나 감소시켜준다.  
58                 return;  
59             }  
60         }  
61     }  
62  
63     for (int i = 0; i < slots.Length; i++) //아이템을 먹으면 슬롯에 계속해서 추가한다.  
64     {  
65         if (slots[i].item == null) //만약 아이템의 슬롯이 비었으면  
66         {  
67             slots[i].AddItem(_item, _count); //슬롯에 나타나는 아이템 이미지를 증가하거나 감소시켜준다.  
68             return; //값을 되돌린다.  
69         }  
70     }  
71 }  
72  
73  
74  
75  
76 }  
77
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- 강



- Player가 강에 들어가면 강의 중력을 받고  
Player의 체력이 깎인다.

Water.cs

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- 강

- ▶ Player가 강에 들어가면 강의 중력을 받고 Player의 체력이 깎이도록 한다.
- ▶ 강 오브젝트에 넣어줄 스크립트를 작성한다.
  - ▶ 우선 물속 여부를 판단할 변수를 설정한다.
  - ▶ 물 밖으로 나오면 원래 저항 값으로 돌아가는 변수를 설정한다.
  - ▶ 물속 색 변수를 설정한다.
  - ▶ 물 탁함 정도 변수를 설정한다.
  - ▶ 물 밖으로 나왔을 때 다시 돌아오게 하는 변수를 설정한다.
  - ▶ 물 밖으로 나왔을 때 원래대로 돌아오게 하는 변수를 지정한다.
  - ▶ 만약 Player가 강에 들어가면
    - ▶ 물의 저항을 받는다.(true)
    - ▶ Player의 체력이 깎인다.(true)
    - ▶ 물의 저항을 받지 않는다.(false)

## Mysterious Forest\_

### 03. 'Mysterious Forest' 코딩 및 순서도

- 강

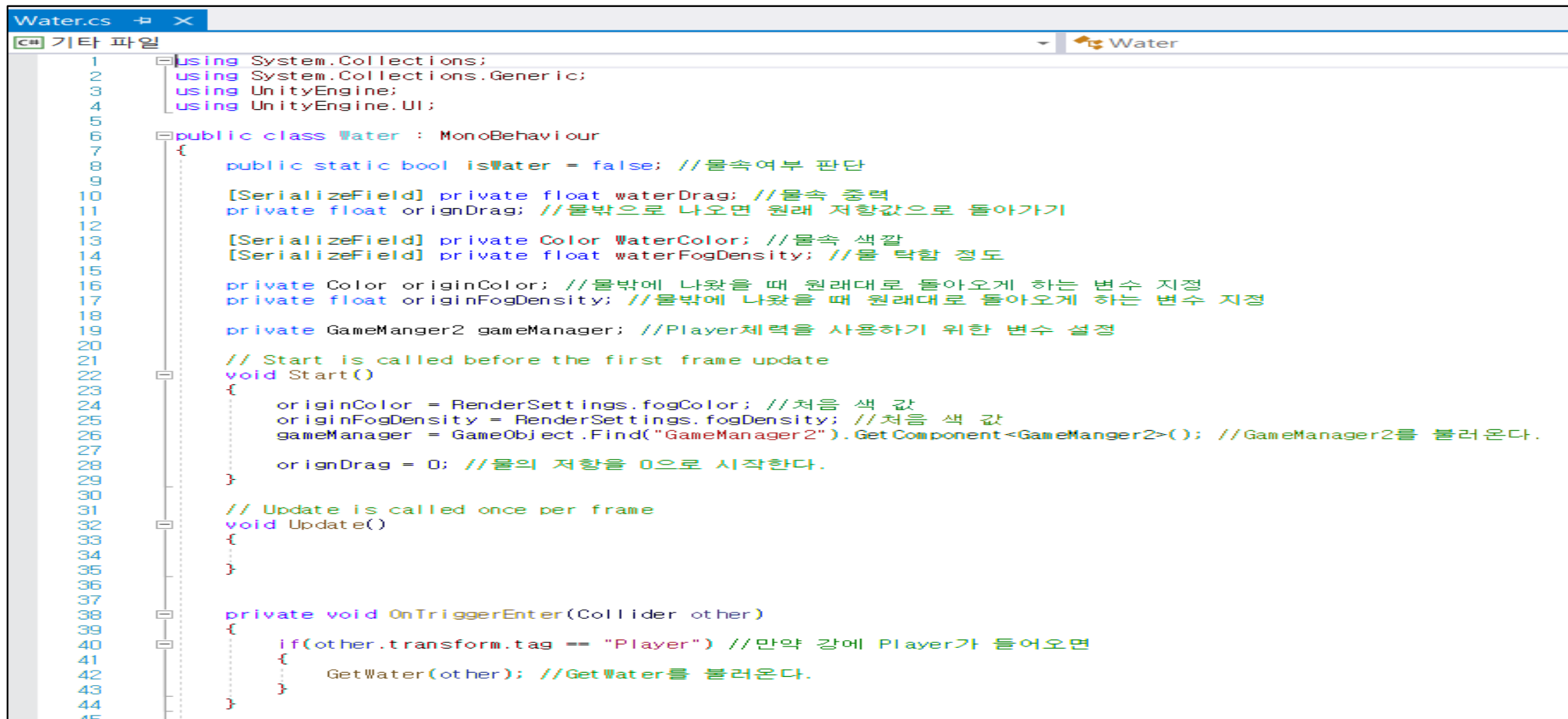
- ▶ 만약 Player가 강을 빠져나가면
- ▶ 원래 저항으로 돌아간다.
- ▶ Player의 체력이 깎인다.(true)
- ▶ 원래 저항으로 돌아간다.(true)
- ▶ Player의 체력이 깎이지 않는다.(false)
- ▶ 작성한 스크립트를 강 오브젝트에 넣어준다.
- ▶ 결과적으로 Player가 강에 들어가면 강의 중력을 받아 천천히 갈아 앉고  
Player의 체력이 깎인다.



# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- 강: **Water.cs**

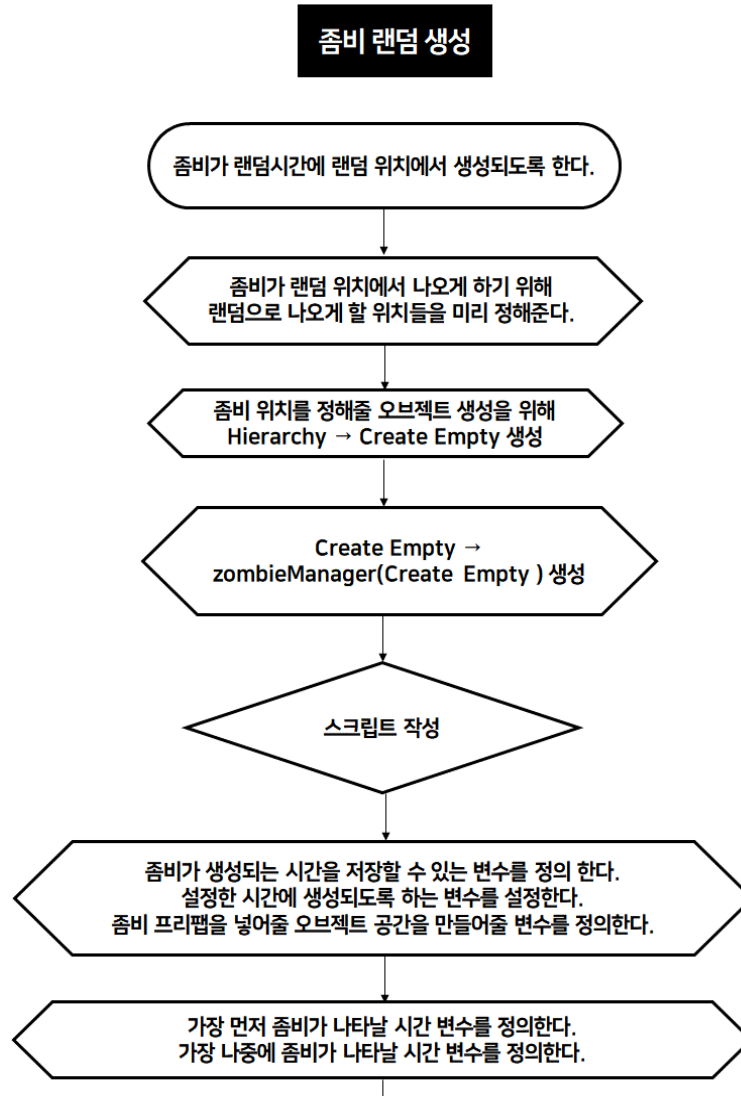


```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
6 public class Water : MonoBehaviour
7 {
8     public static bool isWater = false; //물속여부 판단
9
10    [SerializeField] private float waterDrag; //물속 중력
11    private float originDrag; //물밖으로 나오면 원래 저항값으로 돌아가기
12
13    [SerializeField] private Color WaterColor; //물속 색깔
14    [SerializeField] private float waterFogDensity; //물 탁함 정도
15
16    private Color originColor; //물밖에 나왔을 때 원래대로 돌아오게 하는 변수 지정
17    private float originFogDensity; //물밖에 나왔을 때 원래대로 돌아오게 하는 변수 지정
18
19    private GameManager2 gameManager; //Player체력을 사용하기 위한 변수 설정
20
21    // Start is called before the first frame update
22    void Start()
23    {
24        originColor = RenderSettings.fogColor; //처음 색 값
25        originFogDensity = RenderSettings.fogDensity; //처음 색 값
26        gameManager = GameObject.Find("GameManager2").GetComponent<GameManager2>(); //GameManager2를 불러온다.
27
28        originDrag = 0; //물의 저항을 0으로 시작한다.
29    }
30
31    // Update is called once per frame
32    void Update()
33    {
34    }
35
36
37
38    private void OnTriggerEnter(Collider other)
39    {
40        if(other.transform.tag == "Player") //만약 강에 Player가 들어오면
41        {
42            GetWater(other); //GetWater를 불러온다.
43        }
44    }
45
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

### - 좀비 랜덤 생성



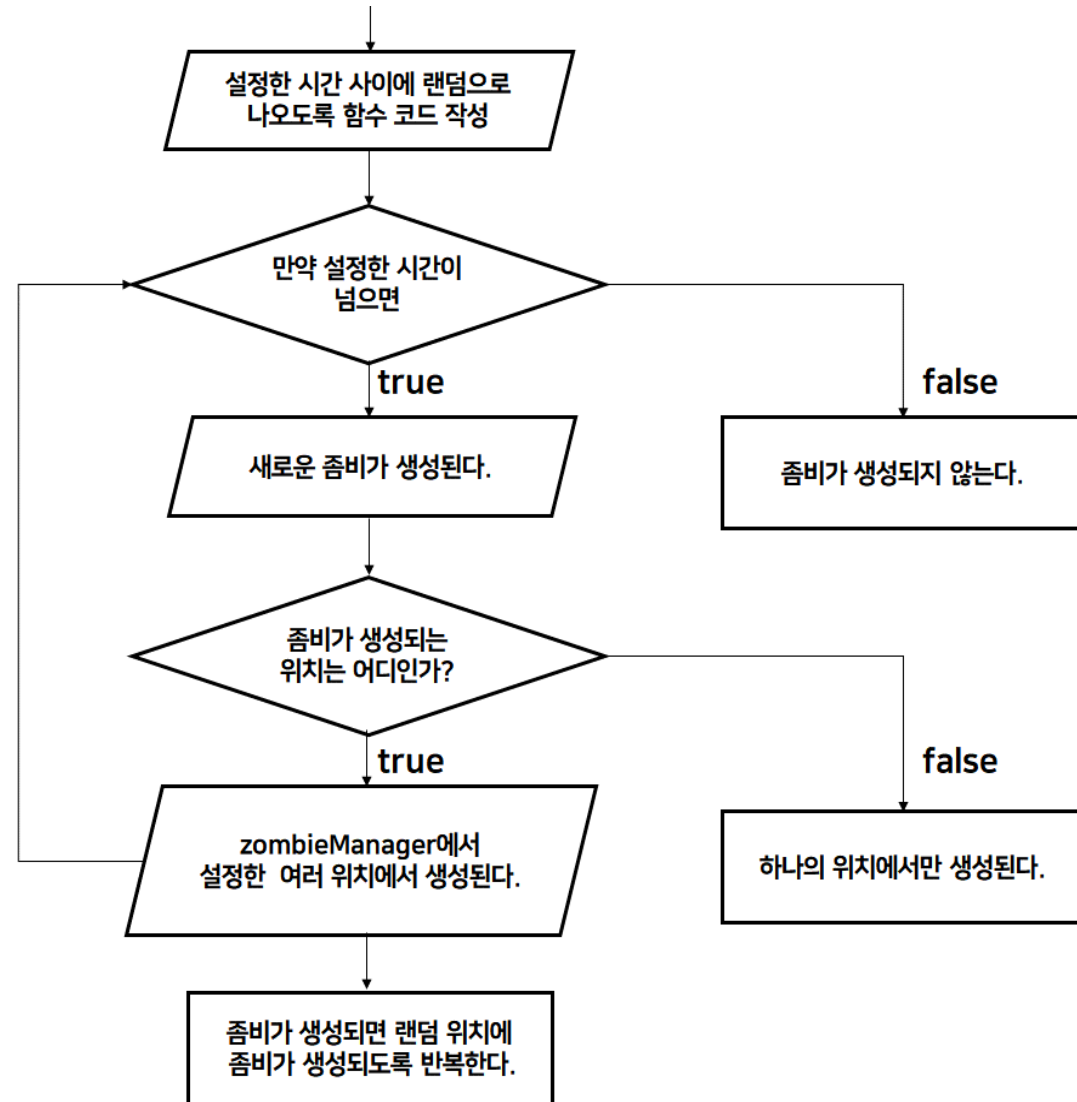
- 좀비가 랜덤시간 사이에 랜덤 위치에서 반복적으로 생성된다.

`zombieManager.cs`

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- 좀비 랜덤 생성



# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

### - 좀비 랜덤 생성

- ▶ 좀비가 랜덤 시간에 랜덤 위치에서 반복적으로 생성되도록 한다.
- ▶ 좀비가 랜덤 위치에서 나오게 하기 위해 랜덤으로 나오게 할 위치들을 미리 정해준다.
- ▶ 좀비 위치를 정해줄 오브젝트 생성을 위해 Hierarchy창에서 Create Empty 생성한다.
- ▶ Create Empty에서 Create Empty를 하나 더 생성하여 이름을 zombieManager으로 변경한다.
- ▶ 이제 zombieManager에 넣어줄 스크립트를 작성한다.
  - ▶ 우선 좀비가 생성되는 시간을 저장할 수 있는 변수를 정의한다.
  - ▶ 설정한 시간에 생성되도록 하는 변수를 설정한다.
  - ▶ 좀비 프리팹을 넣어줄 오브젝트 공간을 만들어줄 변수를 정의한다.

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

### - 좀비 랜덤 생성

- ▶ 가장 먼저 좀비가 나타날 시간 변수를 정의한다.
- ▶ 가장 나중에 좀비가 나타날 시간 변수를 정의한다.
- ▶ 설정한 시간 사이에 랜덤으로 나오도록 함수 코드를 작성한다.
- ▶ 만약 설정한 시간이 넘으면
- ▶ 새로운 좀비가 생성된다.(true)
- ▶ 만약 설정한 시간이 넘지 않으면
- ▶ 새로운 좀비가 생성되지 않는다.(false)
- ▶ 좀비가 생성되는 위치는 zombieManager에서 설정한 여러 위치에서 생성된다.(true)
- ▶ 그렇지 않으면 하나의 위치에서만 좀비가 생성된다.(false)
- ▶ 작성한 스크립트를 zombieManager에 넣어준다.
- ▶ 좀비가 생성되면 랜덤 위치에 좀비가 생성되도록 반복한다.

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- 좀비 랜덤 생성: **zombiManager.cs**

```
zombieManager.cs
C# 기타 파일
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class zombieManager : MonoBehaviour
6  {
7      float currentTime; //시간을 저장할 수 있는 변수 정의
8      public float createTime = 1.0f; //설정한 시간에 생성되도록 설정
9      public GameObject zombie; // 대상은 좀비 게임 오브젝트 생성
10     public float minTime = 5.0f; //가장 먼저 나타나는 시간
11     public float maxTime = 15.0f; //가장 늦게 나타나는 시간
12
13     // Start is called before the first frame update
14     void Start()
15     {
16         currentTime = 0.0f; //처음에는 0부터 시작
17         createTime = UnityEngine.Random.Range(minTime, maxTime); //설정한 시간 사이에 랜덤으로 나오도록 설정
18     }
19
20     // Update is called once per frame
21     void Update()
22     {
23         currentTime += Time.deltaTime; //한번 생성되고, 걸리는 시간
24
25         if (currentTime > createTime) //설정한 시간이 넘었을 때,
26         {
27             GameObject newzombie = Instantiate(zombie); //게임오브젝트의 새로운 좀비가 생성된다.
28             newzombie.transform.position = transform.position; //위치는 zombieManager에서 설정한 위치로 한다.
29
30             currentTime = 0.0f; //업데이트할 때마다 1초를 세는 방법으로 생성된다.
31             createTime = UnityEngine.Random.Range(minTime, maxTime); //다시 생성하면 랜덤으로 좀비를 생성하도록 반복
32         }
33     }
34 }
35
```

# Mysterious Forest\_

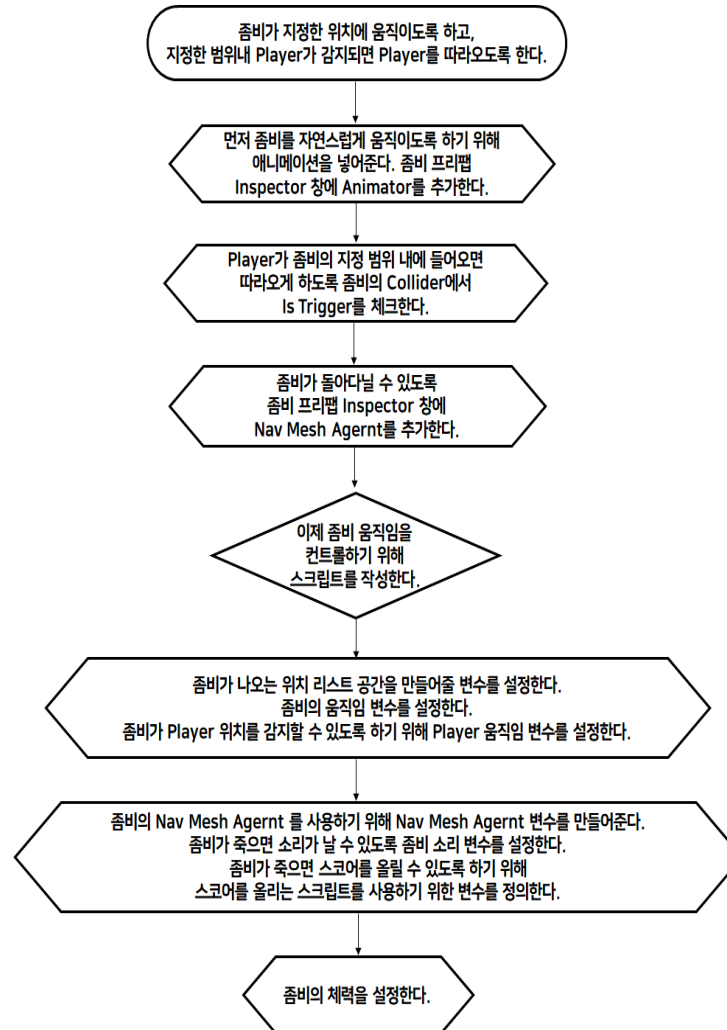
## 03. 'Mysterious Forest' 코딩 및 순서도

### - 좀비 움직임

#### 좀비 움직임

#### - 좀비의 움직임

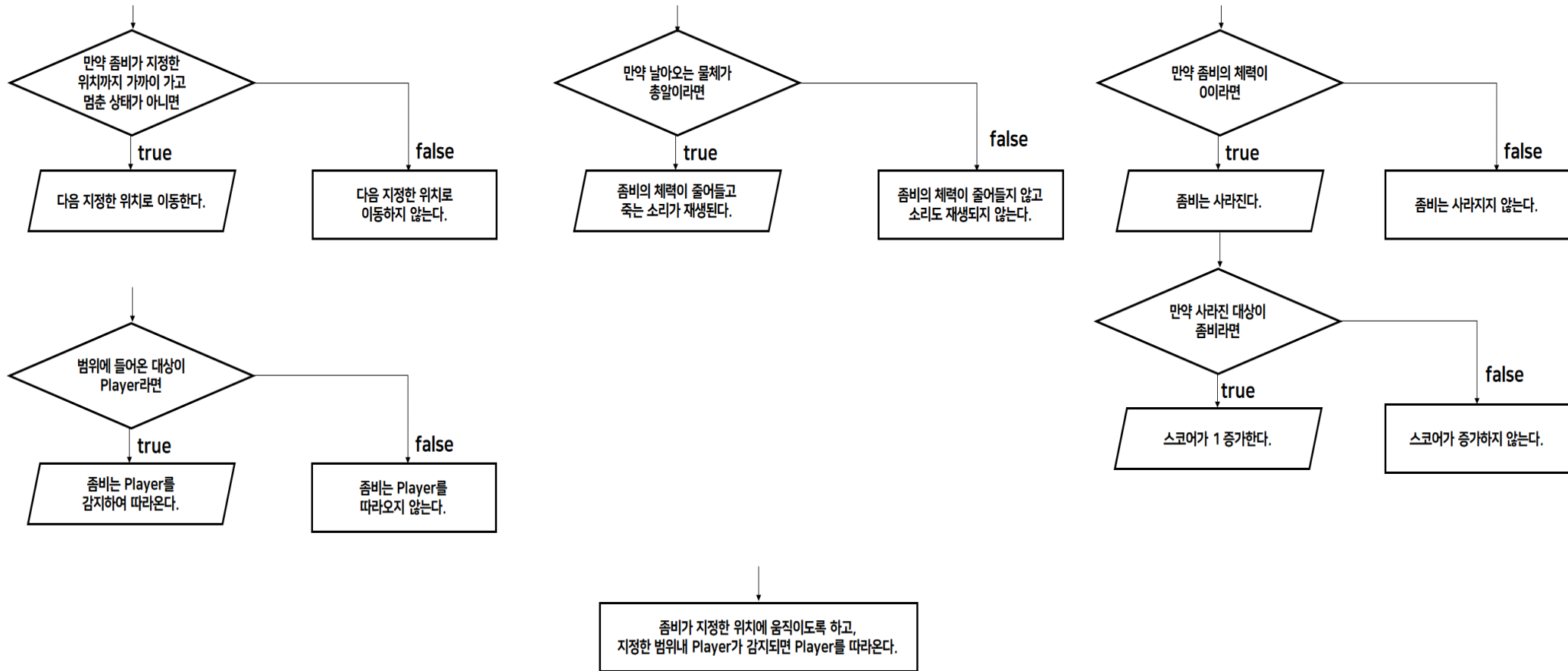
zombieMove.cs



# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

### - 좀비 움직임





## Mysterious Forest\_

### 03. 'Mysterious Forest' 코딩 및 순서도

#### - 좀비 움직임

- ▶ 좀비가 랜덤으로 생성되고 위치도 랜덤으로 계속해서 생성되도록 한다.
- ▶ 먼저 좀비를 자연스럽게 움직이도록 하기 위해 애니메이션을 넣어준다.
- ▶ 좀비 프리팹 Inspector창에서 Animator를 추가한다.
- ▶ Player가 좀비의 지정 범위 내에 들어오면 따라오게 하도록 좀비의 Collider에서 Is Trigger를 체크한다.
- ▶ 다음 좀비가 돌아다닐 수 있도록 좀비 프리팹 Inspector창에서 Nav Mesh Agent를 추가한다.
- ▶ 이제 좀비 움직임을 컨트롤 하기 위해 스크립트를 작성한다.
  - ▶ 좀비가 나오는 위치 리스트 공간을 만들어줄 변수를 설정한다.
  - ▶ 좀비의 움직임 변수를 설정한다.
  - ▶ 좀비가 Player 위치를 감지할 수 있도록 하기 위해 Player 움직임 변수를 설정한다.

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

### - 좀비 움직임

- ▶ 좀비의 Nav Mesh Agent를 사용하기 위해 Nav Mesh Agent 변수를 만들어준다.
- ▶ 좀비가 죽으면 소리가 날 수 있도록 좀비 소리 변수를 설정한다.
- ▶ 스코어를 올리는 스크립트를 사용하기 위한 변수를 정의한다.
- ▶ 좀비의 체력을 설정한다.
- ▶ 만약 날아오는 물체가 총알이라면
- ▶ 좀비의 체력이 줄어들고 죽는 소리가 재생된다.(true)
- ▶ 좀비의 체력이 줄어들지 않고 소리도 재생되지 않는다.(false)
- ▶ 만약 좀비의 체력이 0이라면
- ▶ 좀비는 사라진다.(true)
- ▶ 좀비는 사라지지 않는다.(false)
- ▶ 만약 사라진 대상이 좀비라면
- ▶ 스코어가 1 증가한다.(true)
- ▶ 스코어가 증가하지 않는다.(true)

## Mysterious Forest\_

### 03. 'Mysterious Forest' 코딩 및 순서도

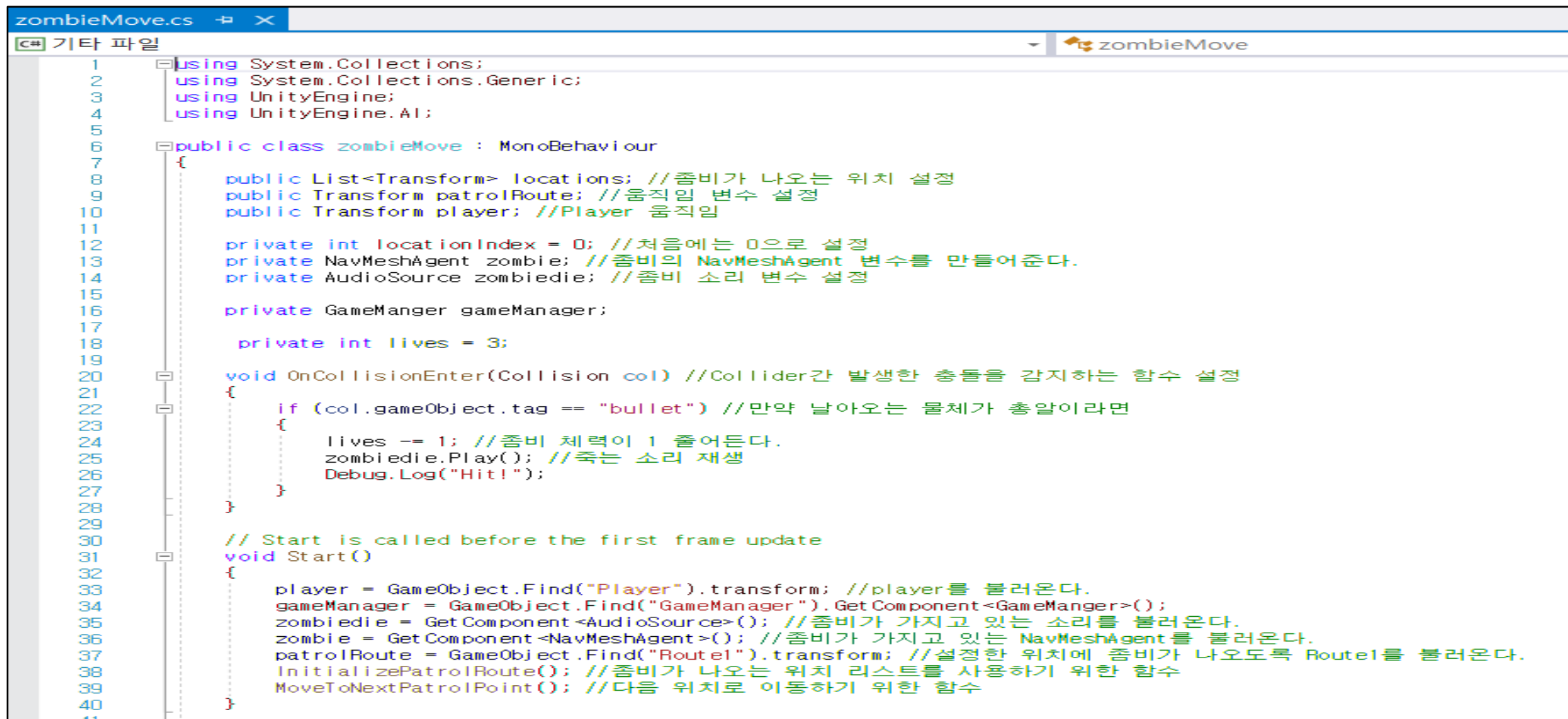
#### - 좀비 움직임

- ▶ 만약 좀비가 지정한 위치까지 가까이 가고 멈춘 상태가 아니라면
- ▶ 다음 지정한 위치로 이동한다.(true)
- ▶ 다음 지정한 위치로 이동하지 않는다.(false)
- ▶ 범위에 들어온 대상이 Player라면
- ▶ 좀비는 Player를 감지하여 따라온다.(true)
- ▶ 좀비는 Player를 따라오지 않는다.(false)
- ▶ 좀비 움직임을 위해 작성한 스크립트를 좀비 프리팹에 넣어준다.
- ▶ 좀비가 지정한 위치에서 계속 움직이고 지정한 범위 내 Plater가 감지되면  
Player를 따라온다.

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

- 좀비 움직임: **zombieMove.cs**



```
zombieMove.cs
C# 기타 파일
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.AI;
5
6 public class zombieMove : MonoBehaviour
7 {
8     public List<Transform> locations; //좀비가 나오는 위치 설정
9     public Transform patrolRoute; //움직임 변수 설정
10    public Transform player; //Player 움직임
11
12    private int locationIndex = 0; //처음에는 0으로 설정
13    private NavMeshAgent zombie; //좀비의 NavMeshAgent 변수를 만들어준다.
14    private AudioSource zombiedie; //좀비 소리 변수 설정
15
16    private GameManager gameManager;
17
18    private int lives = 3;
19
20    void OnCollisionEnter(Collision col) //Collider간 발생한 충돌을 감지하는 함수 설정
21    {
22        if (col.gameObject.tag == "bullet") //만약 날아오는 물체가 총알이라면
23        {
24            lives -= 1; //좀비 체력이 1 줄어든다.
25            zombiedie.Play(); //죽는 소리 재생
26            Debug.Log("Hit!");
27        }
28    }
29
30    // Start is called before the first frame update
31    void Start()
32    {
33        player = GameObject.Find("Player").transform; //player를 불러온다.
34        gameManager = GameObject.Find("GameManager").GetComponent<GameManager>();
35        zombiedie = GetComponent<AudioSource>(); //좀비가 가지고 있는 소리를 불러온다.
36        zombie = GetComponent<NavMeshAgent>(); //좀비가 가지고 있는 NavMeshAgent를 불러온다.
37        patrolRoute = GameObject.Find("Route1").transform; //설정된 위치에 좀비가 나오도록 Route1를 불러온다.
38        InitializePatrolRoute(); //좀비가 나오는 위치 리스트를 사용하기 위한 함수
39        MoveToNextPatrolPoint(); //다음 위치로 이동하기 위한 함수
40    }
41}
```

# Mysterious Forest\_

## 03. 'Mysterious Forest' 코딩 및 순서도

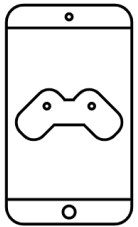
- 좀비 움직임: **zombieMove.cs**

```
41
42 // Update is called once per frame
43 void Update()
44 {
45     if (zombie.remainingDistance < 0.1f && !zombie.pathPending) //좀비가 일정위치까지 가까이 가고, 좀비가 멈춘 상태가 아닐때
46     {
47         MoveToNextPatrolPoint(); //다음 지정한 포지션으로 이동
48     }
49
50     if (lives <= 0) //만약 체력이 0이라면
51     {
52         GameObject.Find("zombie(Clone)").GetComponent<AudioSource>().Play();
53
54         Destroy(this.gameObject); //좀비는 사라진다.
55
56         if (this.gameObject.name == "zombie(Clone)")
57         {
58             gameManager.Score += 1;
59         }
60     }
61 }
62
63 void InitializePatrolRoute()
64 {
65     foreach (Transform child in patrolRoute) //위치를 설정한 patrolRoute 안에 있는 child 움직임
66     {
67         locations.Add(child); //child를 저장한다.
68     }
69 }
70 void MoveToNextPatrolPoint()
71 {
72     zombie.destination = locations[locationIndex].position; //다음 리스트로 이동하기 위한 함수 코드
73
74     locationIndex = (locationIndex + 1) % locations.Count; //반복적으로 이동할 수 있게 하는 함수 코드
75 }
76
77 private void OnTriggerEnter(Collider collider)
78 {
79     if (collider.name == "Player")
80     {
81         zombie.destination = player.position;
82     }
83 }
84 }
85
```

# Mysterious Forest\_



게임 조작 방법



실제 플레이 화면

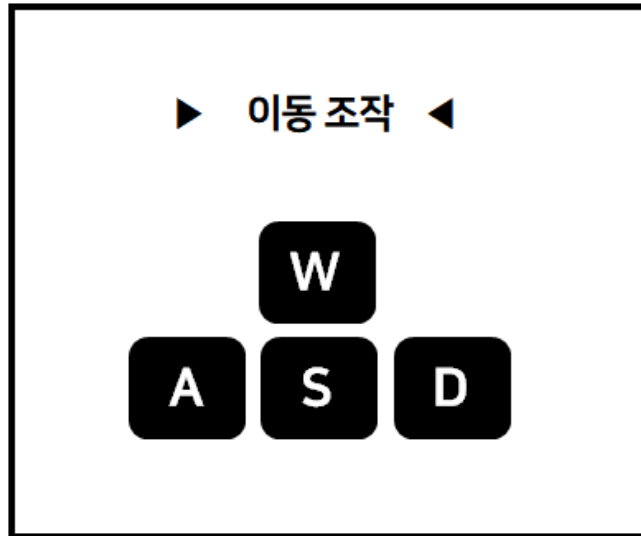
Chapter 04

**Mysterious**  
최종 결과물 및  
실제 구현 모습

# Mysterious Forest\_

## 04. 최종 결과물 및 실제 구현 모습

- 게임 조작 방법



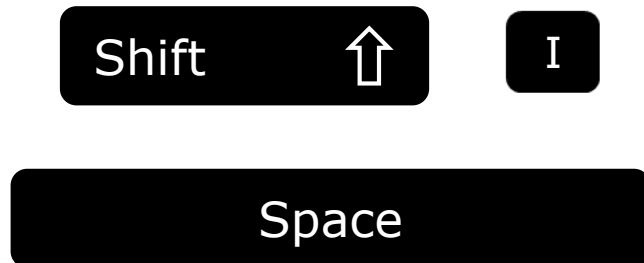
· 캐릭터 움직임 조절 : A / S / D / W

· 캐릭터 방향키 조절 : 마우스

· 달리기 : Shift

· 점프 : Space

· 인벤토리 창 : I



· 무기 교체 : ~(맨손) / 1(칼) / 2(방망이) / 3(총) / 4(곡괭이)



# Thank You

20160241 김미소  
20160247 오은주