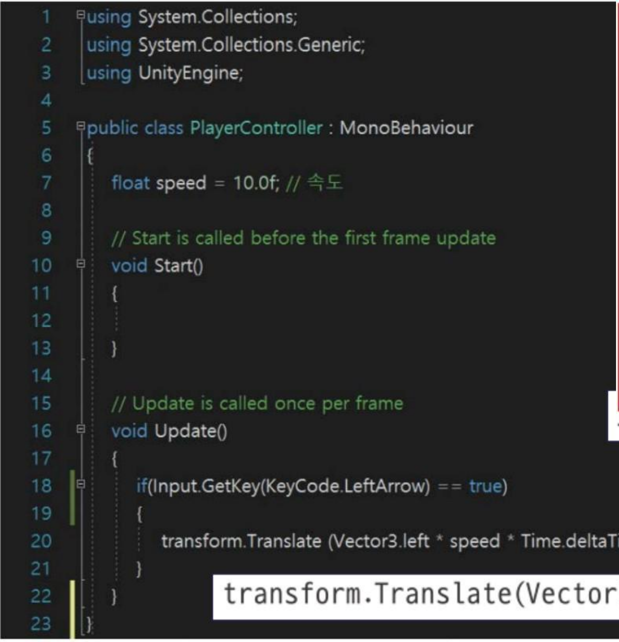
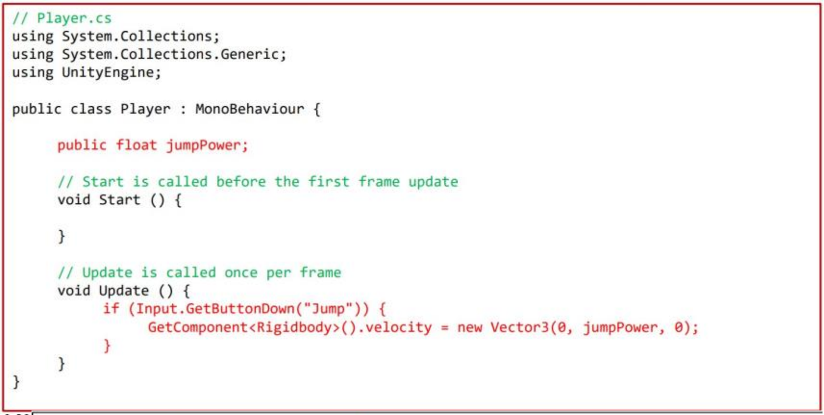
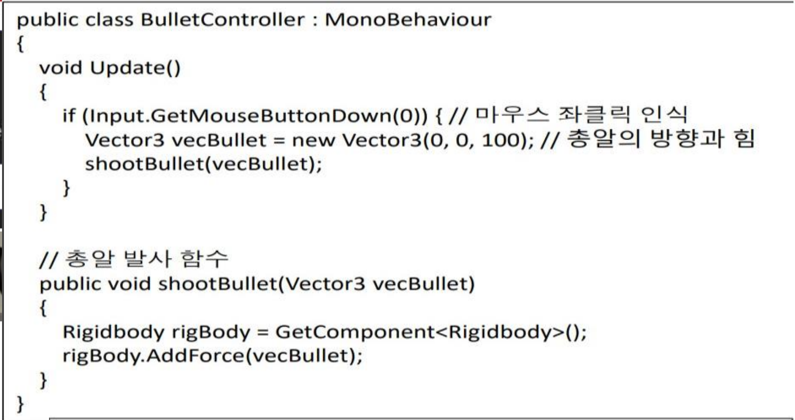


```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerController : MonoBehaviour
{
    float speed = 10.0f; // 속도

    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        if(Input.GetKey(KeyCode.LeftArrow) == true)
        {
            transform.Translate (Vector3.left * speed * Time.deltaTime
        }
    }
}
```

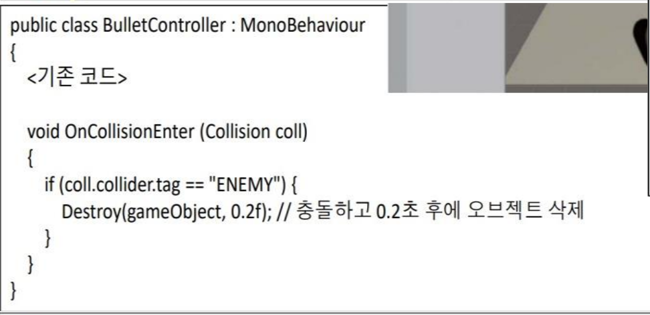
```
transform.Translate(Vector3.
```



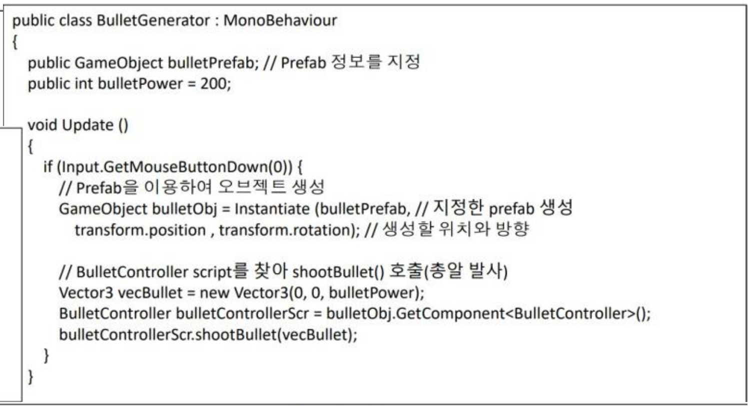
```
// Player.cs
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Player : MonoBehaviour {

      public float jumpPower;

      // Start is called before the first frame update
      void Start () {

      }

      // Update is called once per frame
      void Update () {
            if (Input.GetButtonDown("Jump")) {
                  GetComponent<Rigidbody>().velocity = new Vector3(0, jumpPower, 0);
            }
      }
}
```



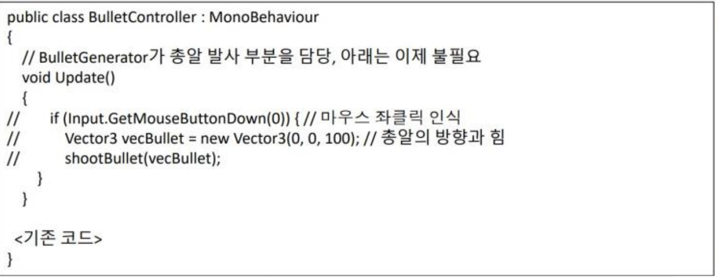
```
public class BulletController : MonoBehaviour
{
   void Update()
   {
      if (Input.GetMouseButtonDown(0)) { // 마우스 좌클릭 인식
         Vector3 vecBullet = new Vector3(0, 0, 100); // 총알의 방향과 힘
         shootBullet(vecBullet);
      }
   }

   // 총알 발사 함수
   public void shootBullet(Vector3 vecBullet)
   {
      Rigidbody rigBody = GetComponent<Rigidbody>();
      rigBody.AddForce(vecBullet);
   }
}
```



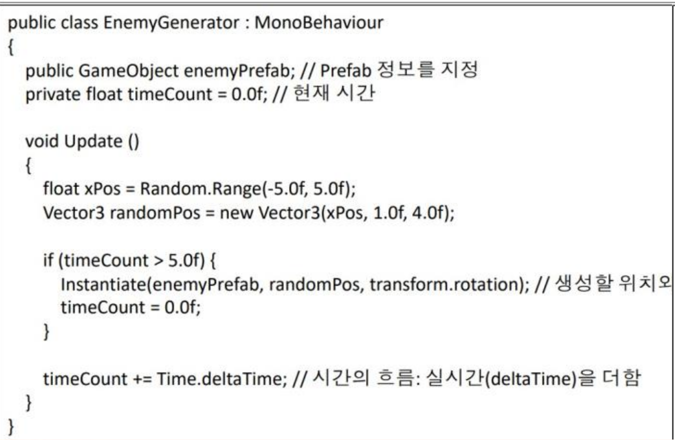
```
public class BulletController : MonoBehaviour
{
  <기존 코드>

  void OnCollisionEnter (Collision coll)
  {
    if (coll.collider.tag == "ENEMY") {
      Destroy(gameObject, 0.2f); // 충돌하고 0.2초 후에 오브젝트 삭제
    }
  }
}
```



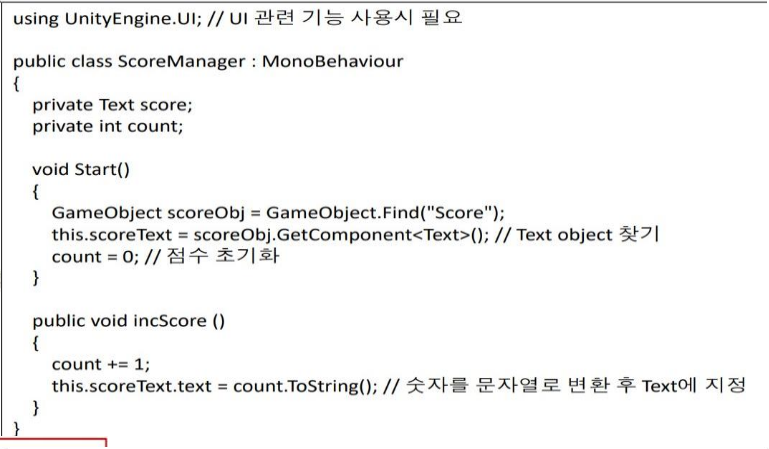
```
public class BulletGenerator : MonoBehaviour
{
   public GameObject bulletPrefab; // Prefab 정보를 지정
   public int bulletPower = 200;

   void Update ()
   {
     if (Input.GetMouseButtonDown(0)) {
       // Prefab을 이용하여 오브젝트 생성
       GameObject bulletObj = Instantiate (bulletPrefab, // 지정한 prefab 생성
          transform.position , transform.rotation); // 생성할 위치와 방향

       // BulletController script를 찾아 shootBullet() 호출(총알 발사)
       Vector3 vecBullet = new Vector3(0, 0, bulletPower);
       BulletController bulletControllerScr = bulletObj.GetComponent<BulletController>();
       bulletControllerScr.shootBullet(vecBullet);
     }
   }
```



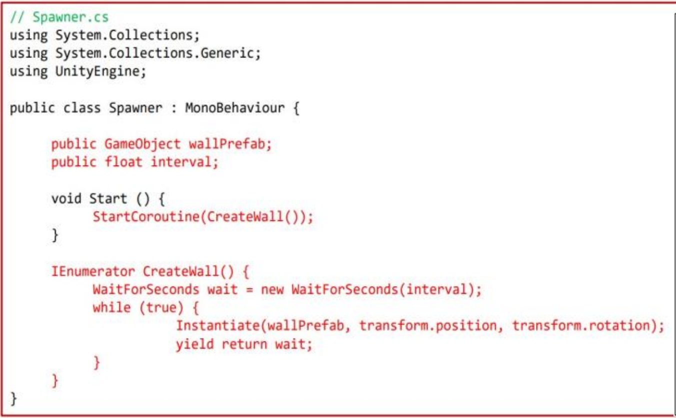
```
public class BulletController : MonoBehaviour
{
   // BulletGenerator가 총알 발사 부분을 담당, 아래는 이제 불필요
   void Update()
   {
//      if (Input.GetMouseButtonDown(0)) { // 마우스 좌클릭 인식
//         Vector3 vecBullet = new Vector3(0, 0, 100); // 총알의 방향과 힘
//         shootBullet(vecBullet);
      }
   }

 <기존 코드>
}
```



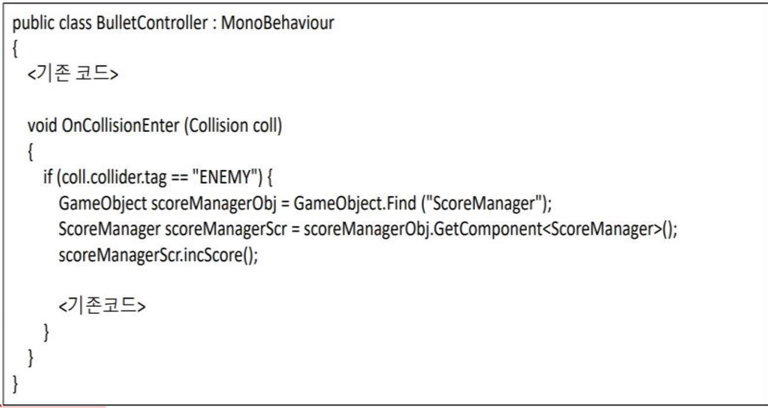
```
public class EnemyGenerator : MonoBehaviour
{
   public GameObject enemyPrefab; // Prefab 정보를 지정
   private float timeCount = 0.0f; // 현재 시간

   void Update ()
   {
     float xPos = Random.Range(-5.0f, 5.0f);
     Vector3 randomPos = new Vector3(xPos, 1.0f, 4.0f);

     if (timeCount > 5.0f) {
        Instantiate(enemyPrefab, randomPos, transform.rotation); // 생성할 위치오
        timeCount = 0.0f;
     }

     timeCount += Time.deltaTime; // 시간의 흐름: 실시간(deltaTime)을 더함
   }
}
```



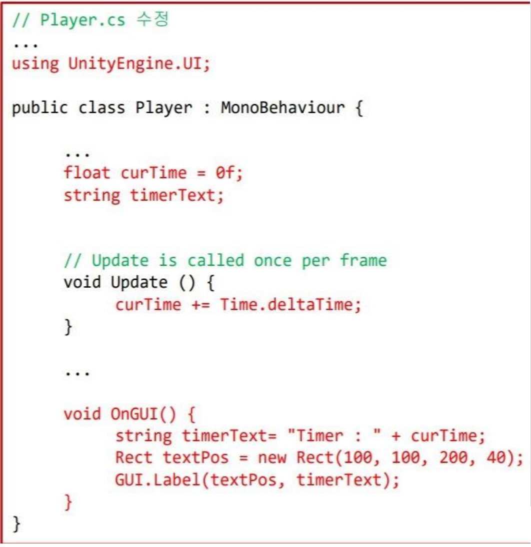
```
using UnityEngine.UI; // UI 관련 기능 사용시 필요

public class ScoreManager : MonoBehaviour
{
   private Text score;
   private int count;

   void Start()
   {
     GameObject scoreObj = GameObject.Find("Score");
     this.scoreText = scoreObj.GetComponent<Text>(); // Text object 찾기
     count = 0; // 점수 초기화
   }

   public void incScore ()
   {
     count += 1;
     this.scoreText.text = count.ToString(); // 숫자를 문자열로 변환 후 Text에 지정
   }
}
```



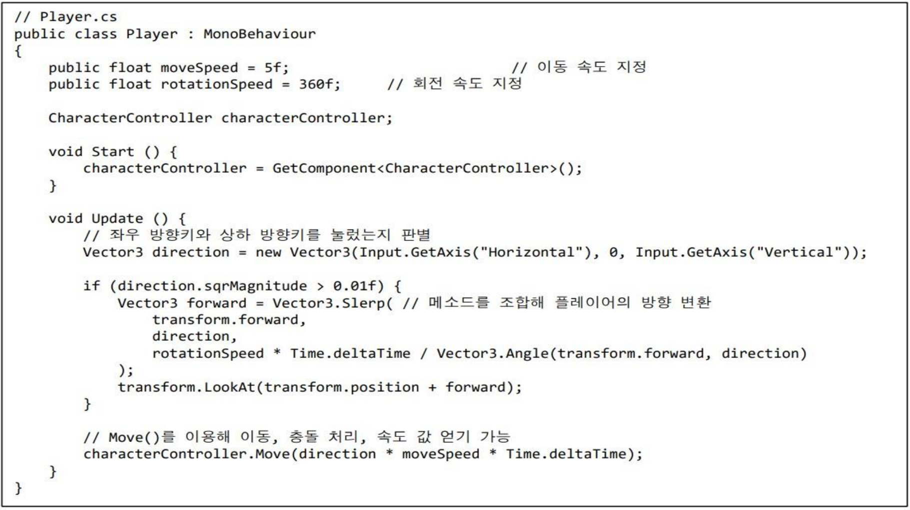
```
// Spawner.cs
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Spawner : MonoBehaviour {

      public GameObject wallPrefab;
      public float interval;

      void Start () {
            StartCoroutine(CreateWall());
      }

      IEnumerator CreateWall() {
            WaitForSeconds wait = new WaitForSeconds(interval);
            while (true) {
                        Instantiate(wallPrefab, transform.position, transform.rotation);
                        yield return wait;
            }
      }
}
```



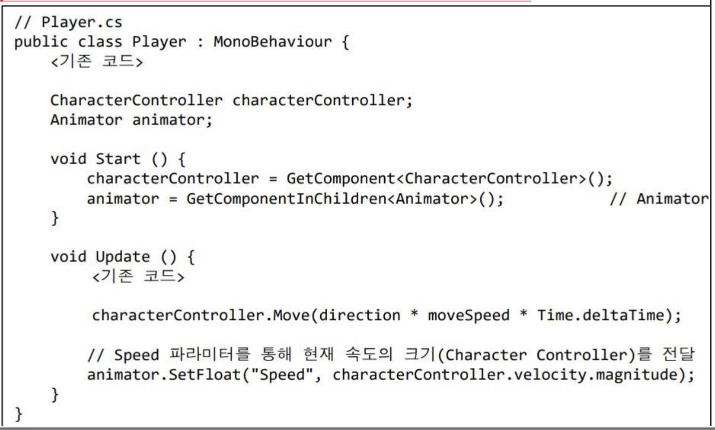
```
public class BulletController : MonoBehaviour
{
  <기존 코드>

  void OnCollisionEnter (Collision coll)
  {
    if (coll.collider.tag == "ENEMY") {
      GameObject scoreManagerObj = GameObject.Find ("ScoreManager");
      ScoreManager scoreManagerScr = scoreManagerObj.GetComponent<ScoreManager>();
      scoreManagerScr.incScore();

      <기존코드>
    }
  }
}
```



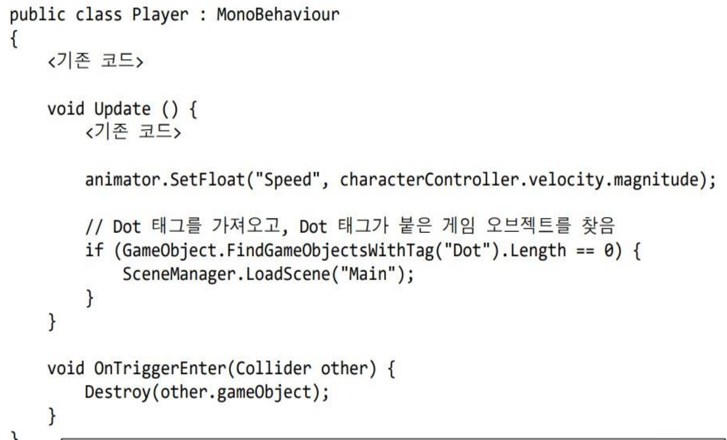
```
// Player.cs 수정
...
using UnityEngine.UI;

public class Player : MonoBehaviour {

    ...
    float curTime = 0f;
    string timerText;


    // Update is called once per frame
    void Update () {
        curTime += Time.deltaTime;
    }

    ...

    void OnGUI() {
        string timerText= "Timer : " + curTime;
        Rect textPos = new Rect(100, 100, 200, 40);
        GUI.Label(textPos, timerText);
    }
}
```



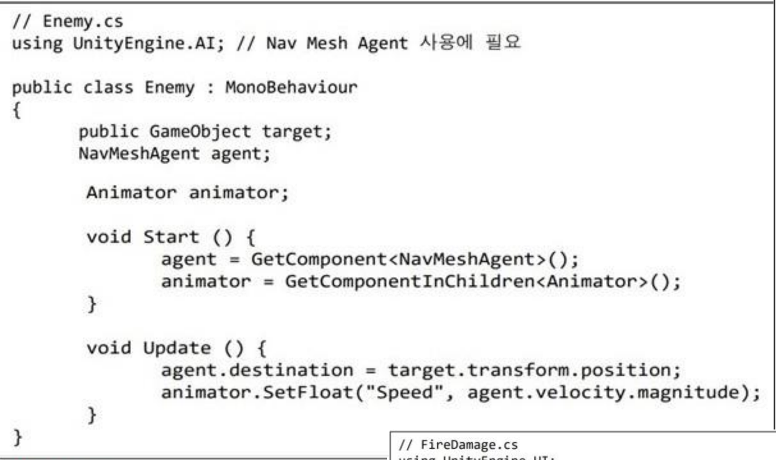
```
// Player.cs
public class Player : MonoBehaviour
{
    public float moveSpeed = 5f;                    // 이동 속도 지정
    public float rotationSpeed = 360f;      // 회전 속도 지정

    CharacterController characterController;

    void Start () {
        characterController = GetComponent<CharacterController>();
    }

    void Update () {
        // 좌우 방향키와 상하 방향키를 눌렀는지 판별
        Vector3 direction = new Vector3(Input.GetAxis("Horizontal"), 0, Input.GetAxis("Vertical"));

        if (direction.sqrMagnitude > 0.01f) {
            Vector3 forward = Vector3.Slerp( // 메소드를 조합해 플레이어의 방향 변환
                transform.forward,
                direction,
                rotationSpeed * Time.deltaTime / Vector3.Angle(transform.forward, direction)
            );
            transform.LookAt(transform.position + forward);
        }

        // Move()를 이용해 이동, 충돌 처리, 속도 값 얻기 가능
        characterController.Move(direction * moveSpeed * Time.deltaTime);
    }
}
```



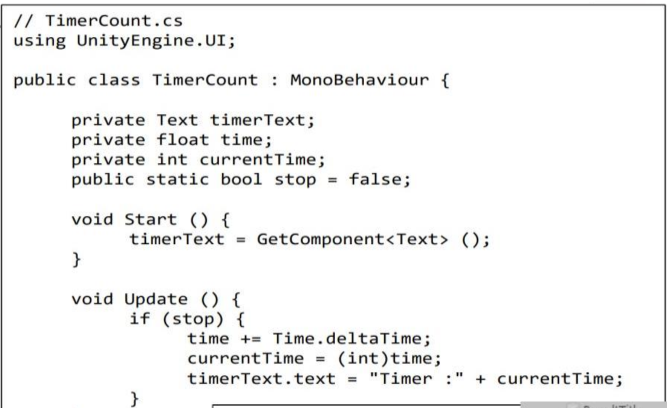
```
// Player.cs
public class Player : MonoBehaviour {
    <기존 코드>

    CharacterController characterController;
    Animator animator;

    void Start () {
        characterController = GetComponent<CharacterController>();
        animator = GetComponentInChildren<Animator>();          // Animator
    }

    void Update () {
        <기존 코드>

        characterController.Move(direction * moveSpeed * Time.deltaTime);

        // Speed 파라미터를 통해 현재 속도의 크기(Character Controller)를 전달
        animator.SetFloat("Speed", characterController.velocity.magnitude);
    }
}
```



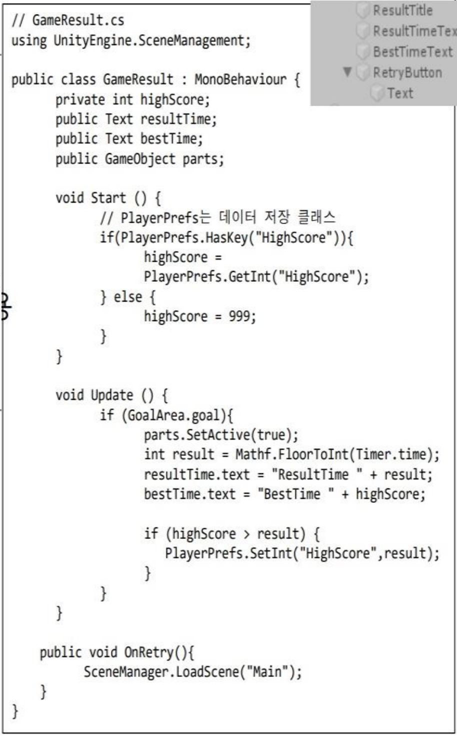
```
public class Player : MonoBehaviour
{
    <기존 코드>

    void Update () {
        <기존 코드>

        animator.SetFloat("Speed", characterController.velocity.magnitude);

        // Dot 태그를 가져오고, Dot 태그가 붙은 게임 오브젝트를 찾음
        if (GameObject.FindGameObjectsWithTag("Dot").Length == 0) {
            SceneManager.LoadScene("Main");
        }
    }

    void OnTriggerEnter(Collider other) {
        Destroy(other.gameObject);
    }
}
```



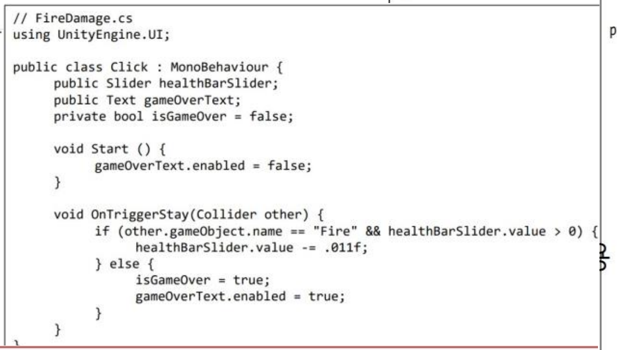
```
// Enemy.cs
using UnityEngine.AI; // Nav Mesh Agent 사용에 필요

public class Enemy : MonoBehaviour
{
      public GameObject target;
      NavMeshAgent agent;

       Animator animator;

       void Start () {
             agent = GetComponent<NavMeshAgent>();
             animator = GetComponentInChildren<Animator>();
       }

       void Update () {
             agent.destination = target.transform.position;
             animator.SetFloat("Speed", agent.velocity.magnitude);
       }
}
```



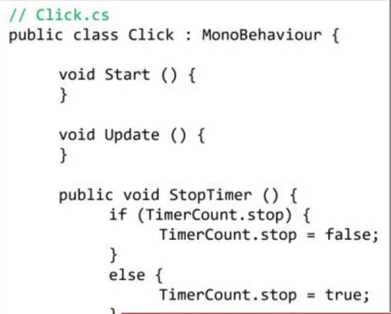
```
// TimerCount.cs
using UnityEngine.UI;

public class TimerCount : MonoBehaviour {

     private Text timerText;
     private float time;
     private int currentTime;
     public static bool stop = false;

     void Start () {
          timerText = GetComponent<Text> ();
     }

     void Update () {
          if (stop) {
               time += Time.deltaTime;
               currentTime = (int)time;
               timerText.text = "Timer :" + currentTime;
          }
     }
}
```



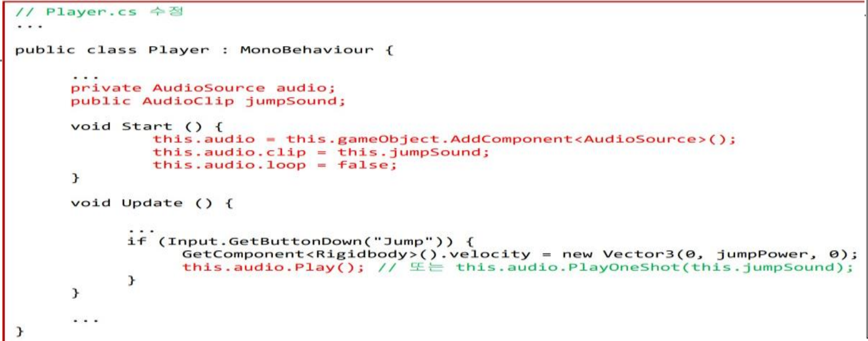
```
// GameResult.cs
using UnityEngine.SceneManagement;

public class GameResult : MonoBehaviour {
      private int highScore;
      public Text resultTime;
      public Text bestTime;
      public GameObject parts;

      void Start () {
            // PlayerPrefs는 데이터 저장 클래스
            if(PlayerPrefs.HasKey("HighScore")){
                  highScore =
                  PlayerPrefs.GetInt("HighScore");
            } else {
                  highScore = 999;
            }
      }

      void Update () {
            if (GoalArea.goal){
                  parts.SetActive(true);
                  int result = Mathf.FloorToInt(Timer.time);
                  resultTime.text = "ResultTime " + result;
                  bestTime.text = "BestTime " + highScore;

                  if (highScore > result) {
                     PlayerPrefs.SetInt("HighScore",result);
                  }
            }
      }

    public void OnRetry(){
          SceneManager.LoadScene("Main");
    }
}
```



```
// FireDamage.cs
using UnityEngine.UI;

public class Click : MonoBehaviour {
      public Slider healthBarSlider;
      public Text gameOverText;
      private bool isGameOver = false;

      void Start () {
            gameOverText.enabled = false;
      }

      void OnTriggerStay(Collider other) {
            if (other.gameObject.name == "Fire" && healthBarSlider.value > 0) {
                  healthBarSlider.value -= .011f;
            } else {
                  isGameOver = true;
                  gameOverText.enabled = true;
            }
      }
```



```
// Click.cs
public class Click : MonoBehaviour {

    void Start () {
    }

    void Update () {
    }

    public void StopTimer () {
        if (TimerCount.stop) {
             TimerCount.stop = false;
        }
        else {
             TimerCount.stop = true;
        }
    }
}
```



```
// Player.cs 수정
...
public class Player : MonoBehaviour {

    ...
    private AudioSource audio;
    public AudioClip jumpSound;

    void Start () {
        this.audio = this.gameObject.AddComponent<AudioSource>();
        this.audio.clip = this.jumpSound;
        this.audio.loop = false;
    }

    void Update () {
        ...
        if (Input.GetButtonDown("Jump")) {
            GetComponent<Rigidbody>().velocity = new Vector3(0, jumpPower, 0);
            this.audio.Play(); // 또는 this.audio.PlayOneShot(this.jumpSound);
        }
    }

    ...
}
```