

Unity – Data Save & Pause

NHN NEXT
서형석

Unity

- ScoreManager

점수관리만을 위한 Manager

**게임내에서 점수를 획득하는 방법은 너무 많을 수 있다.
그래서 어떠한 클래스에서도 간단히 접근할 수 있는 방법이 필요.**

```
public class ScoreManager : MonoBehaviour
{
    static ScoreManager _instance = null;
    public static ScoreManager Instance()
    {
        return _instance;
    }

    void Start()
    {
        if( _instance == null )
            _instance = this;
        else
            Destroy( gameObject );
    }
}
```

```
public class ScoreManager : MonoBehaviour
```

```
{
```

```
.....
```

```
int _bestScore = 0;
```

```
int _myScore = 0;
```

```
public int bestScore
```

```
{
```

```
    get
```

```
{
```

```
        return _bestScore;
```

```
    }
```

```
}
```

```
public int myScore
```

```
{
```

```
    get
```

```
{
```

```
        return _myScore;
```

```
    }
```

```
    set
```

```
{
```

```
        _myScore = value;
```

```
        if (_myScore > _bestScore)
```

```
{
```

```
            _bestScore = _myScore;
```

```
        }
```

```
    }
```

```
}
```

```
}
```

get/set 구현

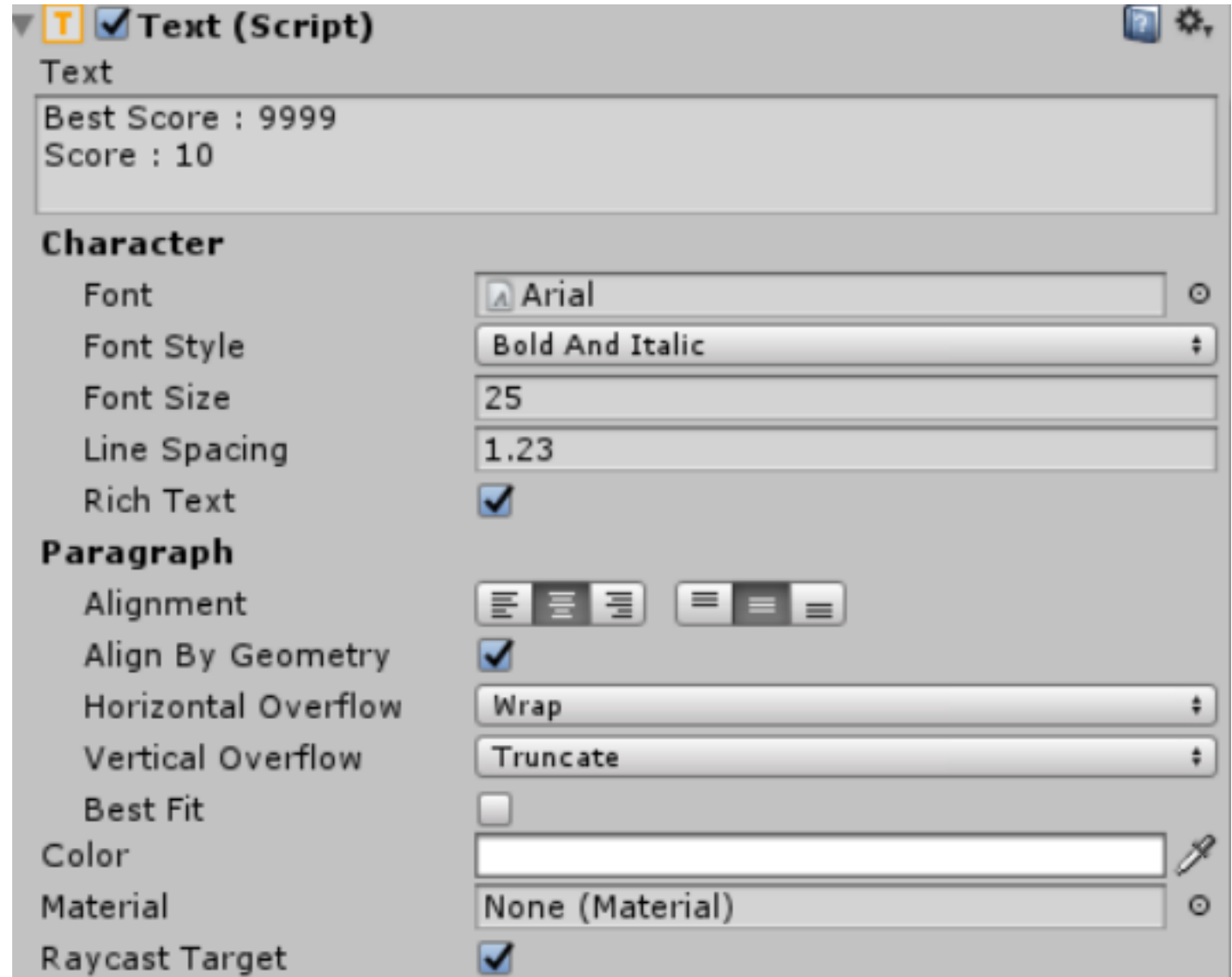
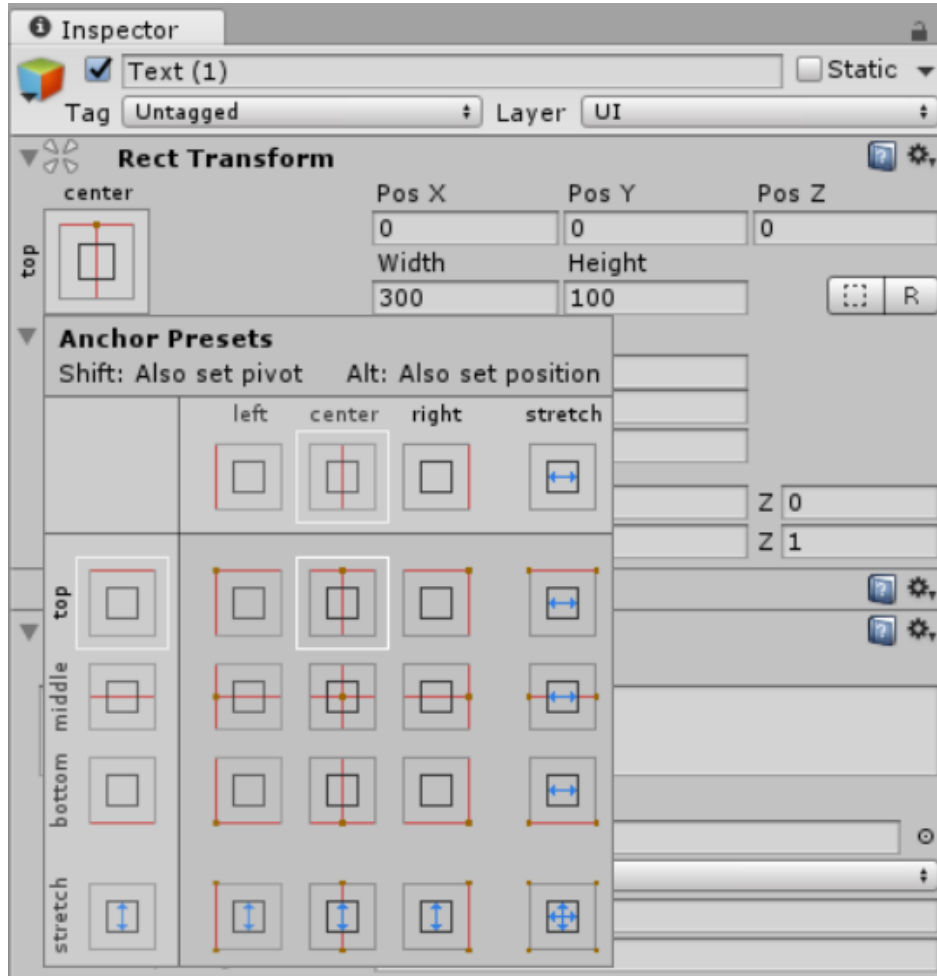
Unity

- 점수 화면에 출력



Unity

- Text UI 추가



```
using System.Text;
```

```
public class ScoreManager : MonoBehaviour  
{
```

```
    public UnityEngine.UI.Text    scoreText;
```

```
    StringBuilder    scoreTextBuilder    =    new StringBuilder();
```

```
    void    UpdateScore()  
{
```

```
        scoreTextBuilder.Length = 0;
```

```
        scoreTextBuilder.AppendFormat(    "Best Score : {0}\nScore : {1}",    bestScore,    myScore);
```

```
        scoreText.text = scoreTextBuilder.ToString();
```

```
    }
```

```
    .....
```

```
}
```

```
void    Update()  
{  
    UpdateScore();  
}
```

Unity

- 점수 화면에 출력

```
using System.Text;
```

```
public class ScoreManager : MonoBehaviour  
{  
    void Update()  
    {  
        UpdateScore();  
    }  
    .....  
}
```


Unity

- 점수 처리 : 적이 죽을 때마다 일정 점수 추가

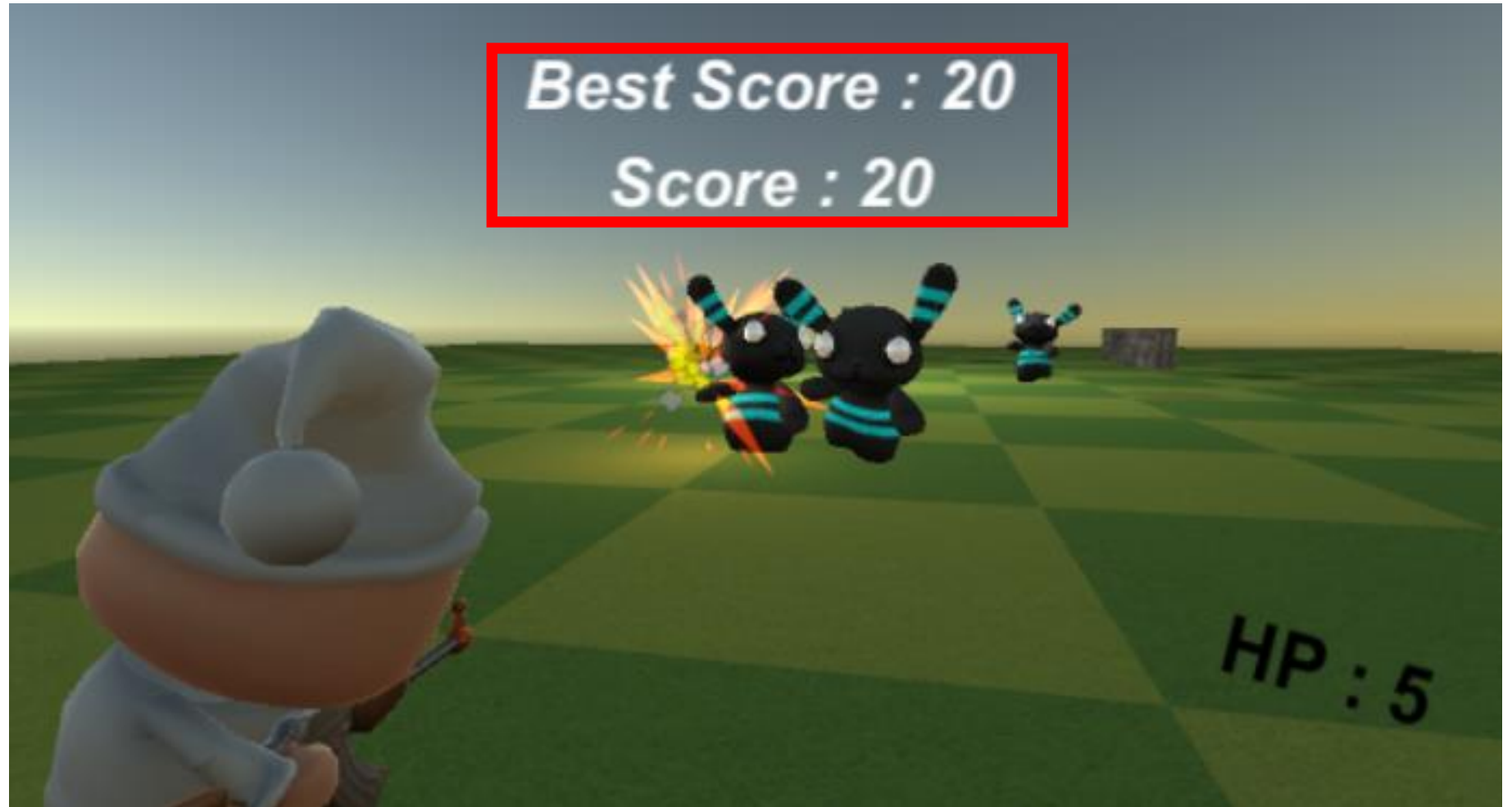
```
public class Zombie : MonoBehaviour
{
    public int score = 10;

    .....
    void Update()
    {
        case ENEMYSTATE.DEAD:
        {
            // Destroy( gameObject );
            StartCoroutine( "DeadProcess" );
            enemyState = ENEMYSTATE.NONE;

            ScoreManager.Instance().myScore += score;
        }
        break;
    }
}
```

Unity

- 점수 확인



Unity

- 문제점 파악 : 최고 점수가 남아있지 않는다.

일반적인 게임에서 최고점수는
게임이 다시 실행되어도 남아 있어야 한다.

Unity

- PlayerPrefs : Data 저장용 클래스

PlayerPrefs

Namespace: UnityEngine

Description

Stores and accesses player preferences between game sessions.

```
public class ScoreManager : MonoBehaviour
{
    .....
    void SaveBestScore()
    {
        PlayerPrefs.SetInt( "Best Score", _bestScore );
    }

    void LoadBestScore()
    {
        _bestScore = PlayerPrefs.GetInt( "Best Score", 0 );
    }
}
```

```
void Start()
{
    .....
    LoadBestScore();
}

public int myScore
{
    .....
    set
    {
        _myScore = value;
        if (_myScore > _bestScore)
        {
            _bestScore = _myScore;
            SaveBestScore();
        }
    }
}
```

```

public class ScoreManager : MonoBehaviour
{
    .....
    void Start()
    {
        if( _instance == null )
            _instance = this;
        else
            Destroy( gameObject );

        LoadBestScore();
    }

    public int myScore
    {
        get
        {
            return _myScore;
        }
        set
        {
            _myScore = value;
            if ( _myScore > _bestScore )
            {
                _bestScore = _myScore;
                SaveBestScore();
            }
        }
    }
}

```

PlayerPrefs

Save/Load

Unity

- 생각해 볼 점 : PlayerPrefs 어디에 저장되는가?

Unity

- 생각해 볼 점 : 보안상의 문제점 대응 방안.

Anti-Cheat Toolkit
스크립팅
focus
★★★★★ (1240)
\$30
불러오기

여러분의 유니티 게임에서 해킹 같은 부정 행위를 시도하는 사람들은 항상 있습니다. Anti-Cheat Toolkit(ACT)은 이와 같은 고약한 부정 행위자들의 버릇을 톡톡히 고쳐놓기 위해 제작되었습니다.

몇 가지 부정 행위 방법 및 그에 대한 ACT의 방어 대책 동영상:
<http://youtu.be/6-2JECbpSE>



Unity

- 일시 정지 구현

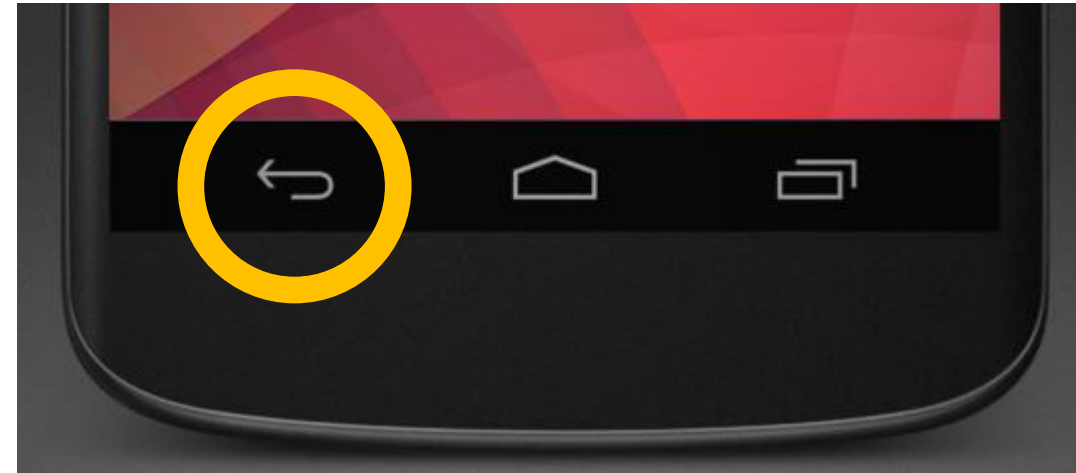


```
public class Pause : MonoBehaviour
{
    bool gamePause = false;

    void Update ()
    {
        if( Input.GetKeyDown(KeyCode.Escape))
        {
            gamePause = !gamePause;

            if (gamePause == true)
                Time.timeScale = 0.0f;
            else
                Time.timeScale = 1.0f;
        }
    }
}
```

Pause 구현



Unity

- 포탄 생성 방지

```
public class Pause : MonoBehaviour
{
    public static bool gamePause = false;
}
```

Unity

- 포탄 생성 방지

```
public class FireBall : MonoBehaviour
{
    .....
    void Update()
    {
        if( Pause.gamePause )
            return;

        .....
    }
}
```