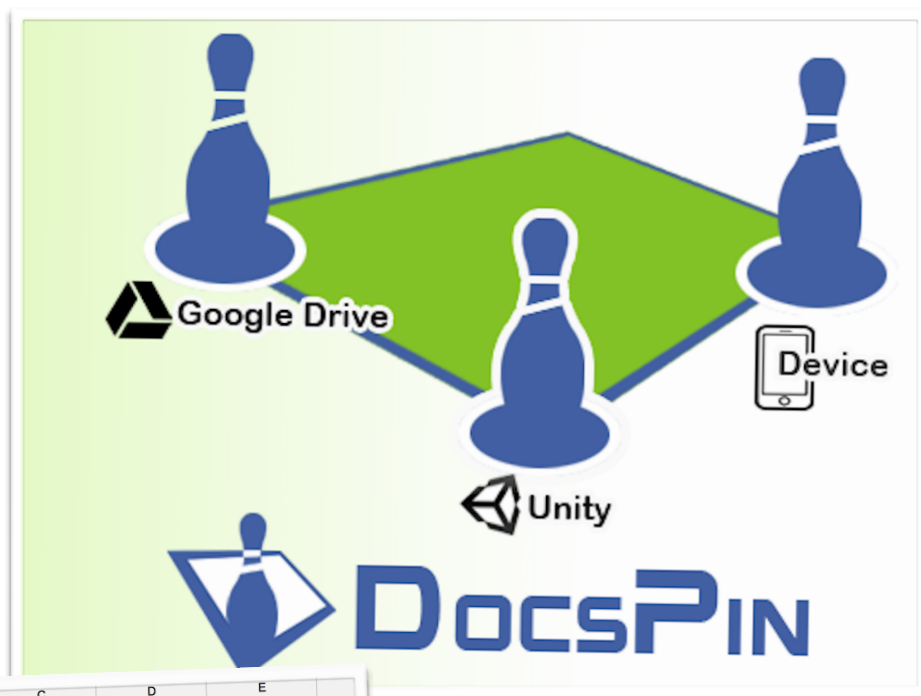
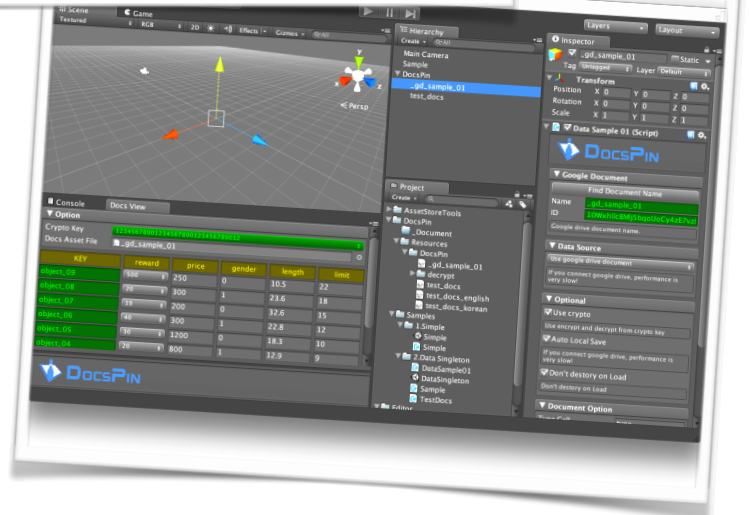


# DocsPin

## Unity Localize Script Service



	A	B	C	D	E	
		int	int	int	list	double
		gender	price	limit	reward	length
act_01		0	100	3	10,100,200	
act_02		1	200	4	10,100,200	
act_03		1	200	4	10,100,200	
act_04		1	200	4	10,100,200	
act_05		1	200	4	10,100,200	
act_06		1	200	4	10,100,200	
act_07		1	200	4	10,100,200	
act_08		1	200	4	10,100,200	
act_09		1	200	4	10,100,200	
act_10		1	200	4	10,100,200	
act_11		1	200	4	10,100,200	



# DocsPin

## Unity Localize Script Service

### Introduction

Application이나 Game에서는 많은 텍스트 및 밸런싱을 위한 데이터를 사용한다. DocsPin은 이러한 텍스트 데이터 및 밸런싱 데이터를 쉽게 Unity에서 적용할 수 있도록 만들어졌다. 텍스트 및 밸런싱 데이터는 Google Drive의 스프레드 시트에 작성하고 해당 데이터를 Unity에서 몇가지 설정으로 데이터를 쉽게 이용할 수 있다. 텍스트 또는 밸런싱 데이터를 개발 부분과 분리하여 좀더 효율적으로 개발을 진행할 수 있다.

또한 DocsPin은 기본적으로 c#의 기본 데이터 타입을 지원하며, 게임 밸런싱 데이터를 효율적으로 사용할 수 있다. 그리고 메모리 해킹 툴(Cheat Engine, Game CIH등)로 부터 데이터를 안전하게 보호하기 위한 데이터 타입도 지원합니다. 메모리 해킹도 간단히 설정하여 여러분의 밸런싱 데이터를 보호할 수 있습니다.

보통 여러분들은 Application이나 Game을 개발 할 때에 다음과 같은 고민을 할 것이다.

- 어떻게 하면 하드코딩을 줄일수 있을까? (텍스트 및 밸런싱 데이터)
- 다국어 서비스를 할 때 좀 더 쉽게 할수 있는 방법은 없을까?
- 기획자와 개발자 간에 업무를 효율적으로 할 수 없을까?

DocsPin은 위와 같은 고민을 한번에 해결할 수 있는 좋은 솔루션이다. 여러분이 DocsPin을 사용한다면 많은 리소스 낭비를 막아 줄것이다.

### 준비사항

DocsPin은 Google Drive의 스프레드 시트를 활용한다. Google Drive는 인터넷 상으로 문서를 편집하고 관리할 수 있는 도구로 많이 활용하고 있다. 특히 공동으로 개발하는 프로젝트의 경우 많은 부분 활용하고 있으며, 여러분의 텍스트 데이터 및 밸런싱 데이터를 보관하고 관리하는데에 많은 이점을 제공한다.

DocsPin을 이용하기 위해서는 다음과 같은 부분을 먼저 준비해야 한다.

- Google 계정 (기존에 있다면 해당 계정이용)
- Google Drive 사용법 (참고 : <https://www.google.co.kr/intl/en/drive/index.html>)
- Google Drive 스프레드 시트 (MS Excel과 사용법 흡사)

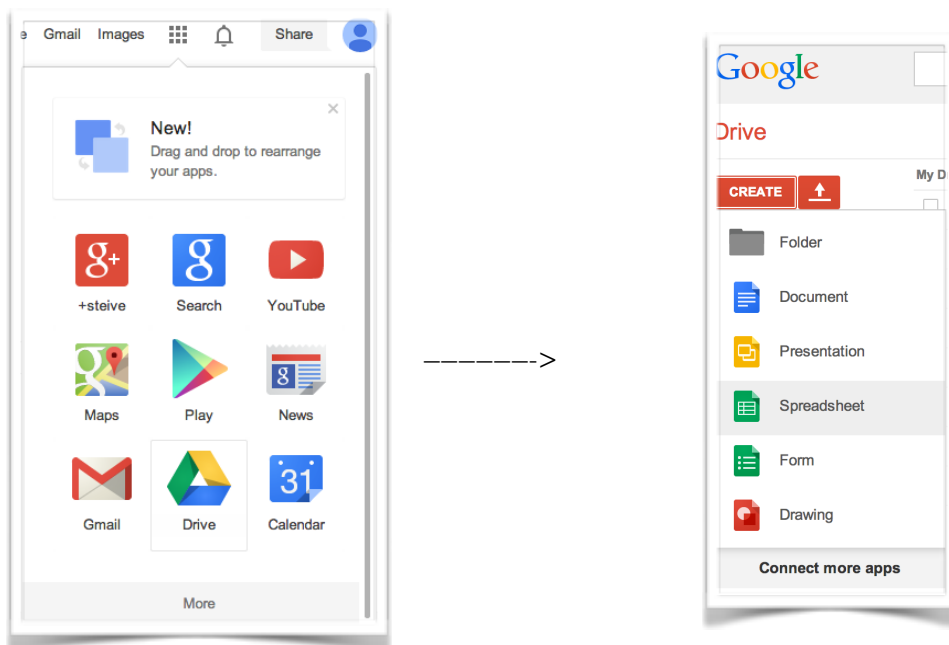
## DocsPin의 기능

DocsPin은 크게 다음과 같은 기능을 제공한다.

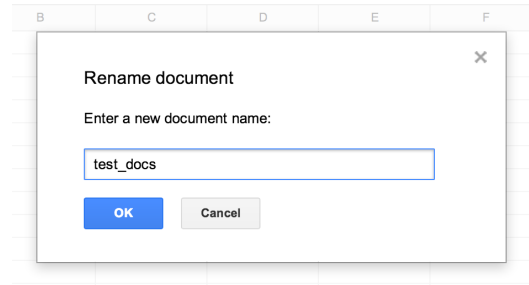
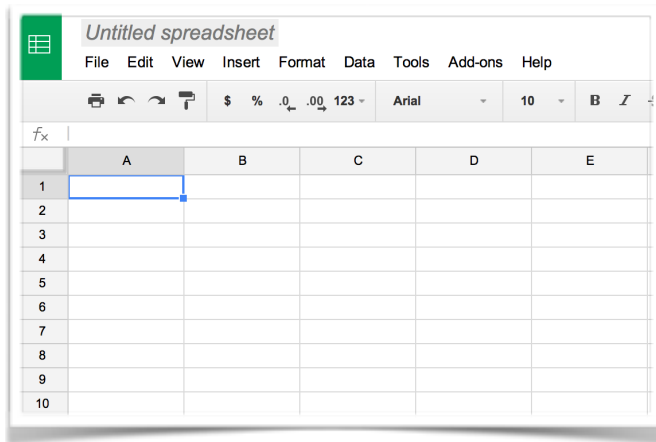
- Google Drive의 데이터를 가져오는 기능
- Google Drive의 데이터를 로컬에 파일로 저장 및 로드하는 기능
- 데이터를 암호화 및 복호화 하는 기능
- 다국어 텍스트 데이터 관리 기능
- uGUI Text, NGUI Text 값 설정 Inspector 제공
- C# 데이터 타입 지원 (sbyte, byte, short, ushort, int, uint, long, ulong, float, double, decimal, string, bool)
- 메모리 해킹에서 보호할 수 있는 Detect 타입 제공(m-sbyte, m-byte, m-short, m-ushort, m-int, m-uint, m-long, m-ulong, m-float, m-double, m-decimal, m-string, m-bool)

## Google Drive

1. Google 계정을 생성하거나 로그인을 수행한다.
2. Google Drive에서 스프레드 시트를 만든다.



### 3. 이름을 설정한다.



### 4.type을 설정한다.

	A	B	C	D	E
1	type	int	string	double	list
2					

- 첫번째 열은 type을 지정하는 열이다. 프로그램에서 사용될 데이터 타입이다.
- DocsPin은 C# 기본 타입을 지원한다. (v2.3.0)
  - sbyte , byte, short, ushort, int, uint, long, ulong, float, double, decimal, string, bool
  - list : Unity에서 ArrayList 데이터 형, list 형태는 "," 단위로 데이터를 입력해야 한다.
- 메모리 해킹을 방지하기 위한 Detect Type을 지원한다. (v2.4.0)
  - m-sbyte, m-byte, m-short, m-ushort, m-int, m-uint, m-long, m-ulong, m-float, m-double, m-decimal, m-string, m-bool, m-list

### 5. 컬럼 이름을 설정한다.

	A	B	C	D	E
1	type	int	string	double	list
2	id	price	name	limit	reward
3					

- 두번째 열은 컬럼 이름을 지정하는 열이다.
- 컬럼 이름은 프로그램에서 데이터를 가져올 때 사용된다.

- 하나의 시트에서 동일한 컬럼 이름을 설정하지 말아야 한다.

## 6. 데이터를 입력한다.

	A	B	C	D	E
1	type	int	string	double	list
2	id	price	name	limit	reward
3	object_01	10	knife	2	10,20,30
4	object_02	20	gun	1	20,50,100
5					

- 데이터는 type에 정의된 자료형에 맞게 입력해야 한다.
- id는 동일한 값으로 설정하면 안된다.(object\_01, object\_02...)
- 프로그램에서 데이터를 가져올 때에는 id 와 컬럼명을 이용한다.

## 7. 다국어 서비스 (필요없는 경우 사용하지 않아도 된다.)

다국어로 데이터 관리가 필요하면 다음과 같이 작업을 진행한다.

- 시트를 새로 만든다.

	A	B	C	D	E
1	type	int	string	double	list
2	id	price	name	limit	reward
3	object_01	10	knife	2	10,20,30
4	object_02	20	gun	1	20,50,100
5					
6					

+

	A	B	C	D	E
1					
2					
3					
4					
5					
6					

+

\* 아래의 "+" 버튼을 클릭한다.

- 시트의 이름을 지정한다.

	A	B	C	D
1				
2				
3				
4				
5				
6				

\* Unity에서 제공하는 언어 이름을 설정

\* **Application**.systemLanguage

\* <http://docs.unity3d.com/ScriptReference/SystemLanguage.html> 참조

\* English, Chinese, Korean, German, Japanese 등

- 4~6번의 과정으로 데이터를 입력한다.

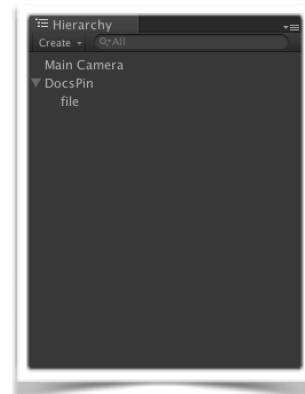
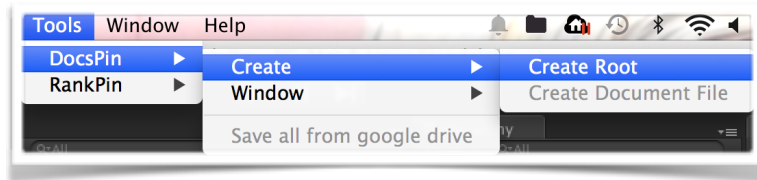
	A	B	C	D	E
1	type	string	string		
2	id	title	content		
3	object_01	KNIFE	A knife (plural knives) is a cutting tool with a cutting edge or blade, hand-held or otherwise, with		
4	object_02	GUN	A gun is a normally tubular weapon or other device designed to discharge projectiles or other m		
5					
6					

- 필요한 언어가 있는 경우 위와 같은 방법으로 시트를 추가한다.

	A	B	C
1	type	string	string
2	id	title	content
3	object_01	칼	칼은 날카로운 면을 가진 도구로 어떤 것을 자를 때 쓴다
4	object_02	총	총(銃)은 일반적으로 개인이 휴대할 수 있는 구경이 작은 화기를 말한다.
5			
6			

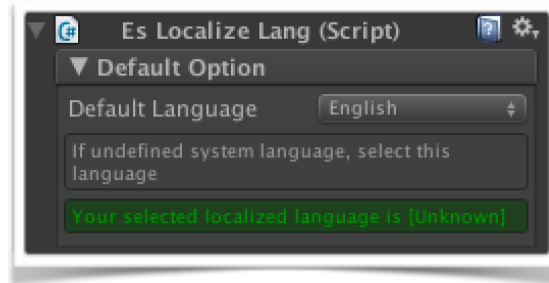
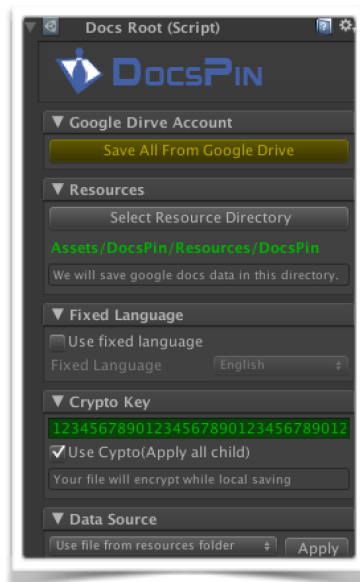
## Unity

1. 상단 메뉴 DocsPin에서 CreateRoot를 선택한다.



- Create Root를 선택하면 Hierarchy 창에 DocsPin과 file에 해당하는 GameObject가 생성된다.

2. Hierarchy에서 DocsPin GameObject를 선택한다.



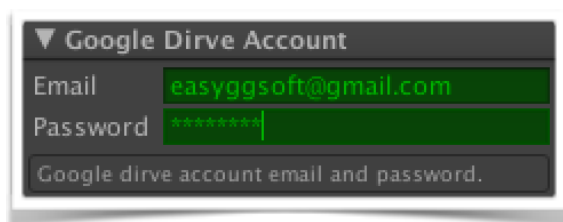
- DocsPin Root는 총 2개의 Script를 가지고 있다.

- Localize Lang Script는 기본 언어를 설정하는 Script이다. DocsPin은 기본 언어를 디바이스의 언어로 설정한다. 만약 디바이스의 언어가 한국어로 설정되어 있다면 Korean으로 설정되나 만약 Google Drive의 스프레드 시트에 Korean이 없는 경우 사용되는 옵션이다. 위에서 기본으로 설정된 언어는 Google Drive에 기본적으로 있어야 하는 시트이다.

- Docs Root Script는 구글 계정 정보 설정 및 Google Drive 스프레드 시트 데이터 관리 방법에 대한 설정이다.

- [v2.1.0] Fixed Language option 추가

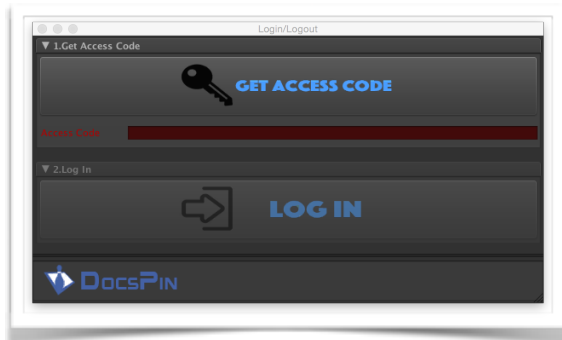
3. Inspector 창에서 Google Drive Account 정보를 입력한다.



- Google 계정의 Email 및 Password를 입력한다.

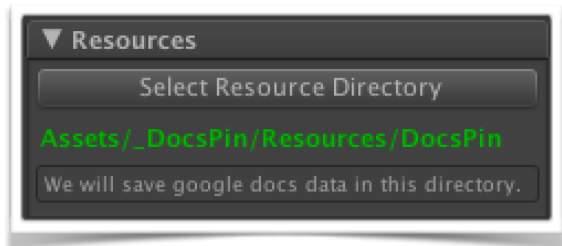
- 해당 계정을 통하여 스프레드 시트를 가져온다.

#### 4. OAuth Login방식으로 로그인을 수행한다.(v2.0)



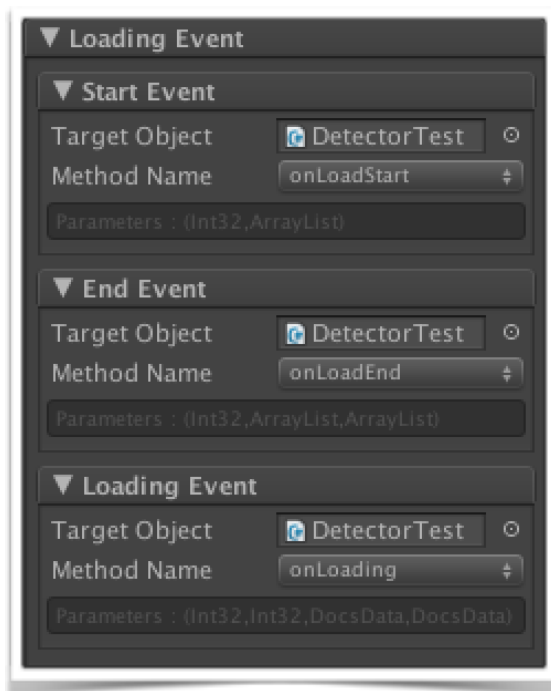
- Get Access Code를 클릭한다.
- Google 인증창이 Web페이지로 실행되며 승인을 하면 Access Code를 얻을 수 있다.
- 해당 Access Code를 Copy하여 Paste후 로그인을 버튼을 클릭하면 로그인 처리가 완료된다.

#### 5. Resources 폴더를 설정한다.



- Unity의 Resources 폴더를 설정한다.
- DocsPin은 Google Drive에 있는 데이터를 로컬에 저장하고 저장된 데이터를 프로그램에서 사용한다.
- 물론 저장하지 않고 바로 Google Drive에 데이터를 가져올 수 있으나 매번 실행시 때마다 Google Drive를 접속하여 데이터를 가져오는데 시간이 소요되므로 Resources 폴더에 저장하고 사용하는 방법이 좋다.
- IOS / Android 등 Export시에도 해당 데이터 파일을 Export 해야 하므로 꼭 Resources 폴더를 설정해야 한다.

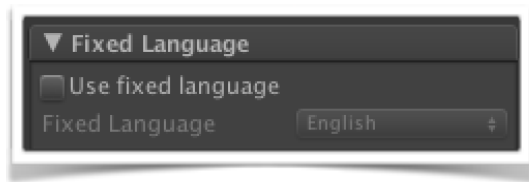
#### 6. [Loading Event. \(v2.4.0\)](#)



- Start Event : 모든 파일을 읽기 전에 설정된 타겟에 이벤트를 전달해준다. 총 읽을 파일 갯수 및 파일 이름을 알려준다.
- End Event : 모든 파일을 읽으면 설정된 타겟에 이벤트를 전달해준다. 읽기에 성공한 파일들과 읽기에 실패한 파일들을 알려준다.
- Loading Event : 하나의 파일을 읽기 직전에 보내주는 이벤트이다. 현재 읽을 파일의 인덱스 및 이름, 그리고 이전에 읽은 파일의 정보를 알려준다.

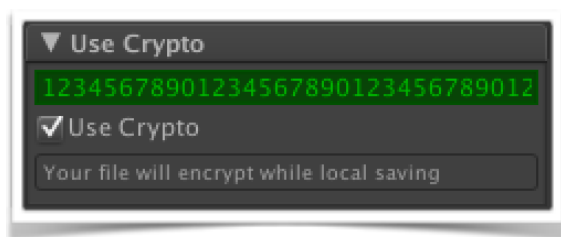


## 7. Fixed Language. (v2.1.0)



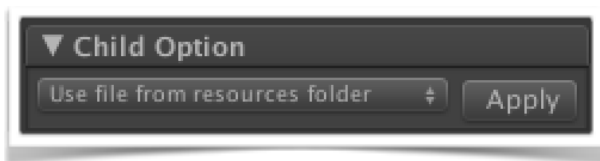
- 로컬라이즈 데이터를 로드할 때에 해당 언어로 로드하게 한다.
- 이 옵션을 사용하면 Application.SystemLanguage를 사용하지 않고 설정된 옵션값으로 데이터를 로드한다.

## 8. 암호화 키를 설정한다.(필요없다면 설정하지 않아도 된다.)



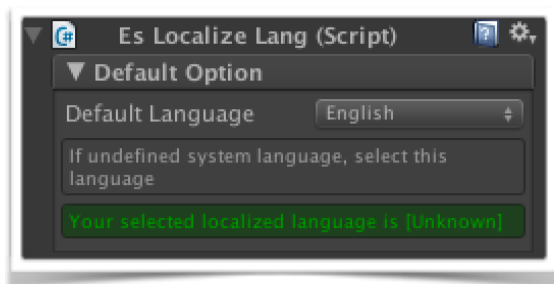
- DocsPin은 로컬에 파일을 저장하는 방식을 이용하기 때문에 필요하다면 파일을 암호화 할수 있다.
- 암호화를 사용하기 위해서는 32byte의 암호화 키를 설정해야 한다.
- 암호화가 필요없는 경우 해당 부분은 설정하지 않아도 된다.
- Use Crypto 옵션은 하위 GameObject에 모두 적용하는 옵션이다. 각 GameObject 별 암호화를 설정할 수 있으며 해당 옵션은 하위 GameObject의 사용 여부를 On/Off 할수 있는 기능이다.
- 암호화 키는 Inspector에 저장되어지며 Unity오류로 인한 분실 가능성이 있으니 따로 메모를 해두는것이 좋다.
- 예제 프로그램의 암호 : 12345678901234567890123456789012

## 9. 기타 옵션을 설정한다.



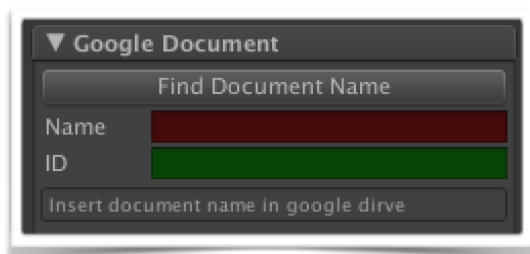
- 해당 옵션은 하위 GameObject에 모두 반영된다.
- Use file from resources folder는 로컬에 저장된 파일을 이용하는 방식이다.
- Use google drive document는 Google Drive을 바로 연결해서 데이터를 사용하는 방식이다. 이 방식은 Google Drive에 매번 접속하는 방식으로 데이터 변경 사항을 실시간으로 가져온다는 장점은 있지만 속도는 느리다.

## 10. Localize 정보를 설정한다.

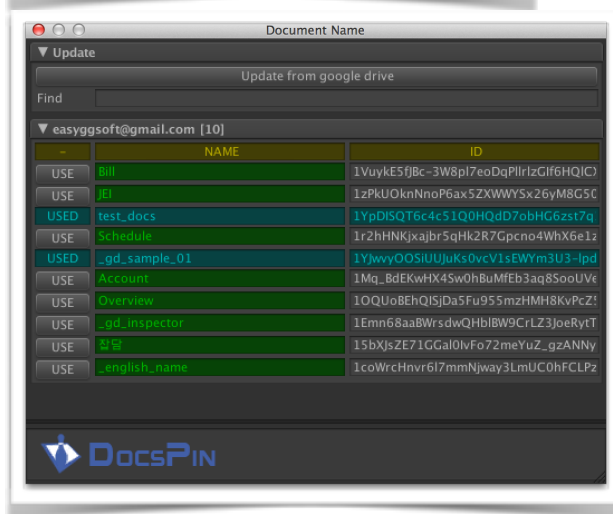


- 기본으로 지원하는 언어를 설정한다.
- Google Drive에 스프레드 시트 이름은 해당 옵션의 이름과 동일하게 설정해야 한다.

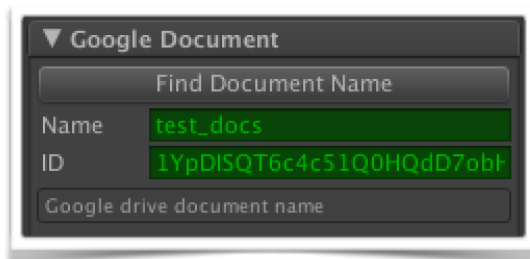
## 11. DocsPin GameObject의 하위 GameObject를 선택한다.



- Document Name과 ID를 설정하는 옵션이다.
- Find Document Name을 선택하면 아래와 같은 창이 보여진다.

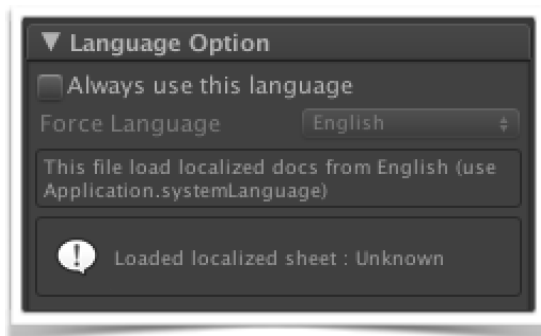


- Update from google drive를 클릭하면 Google Drive의 스프레드 시트 목록을 가져온다.
- USE 버튼을 통해서 사용할 스프레드 시트를 선택한다.
- 선택하면 Name과 ID가 설정된다.



- 프로그램에서는 Name 과 ID로 해당 GameObject를 찾을수 있다.
- DocsPin.DocsRoot.findData("test\_docs");

## 12. [Language option.\(v2.1.0\)](#)



-개별적인 파일에서 특정한 언어만 데이터를 가져오고 싶을때 해당 옵션을 사용한다.

-해당 옵션을 사용하면 선택한 언어의 데이터만 가져온다.

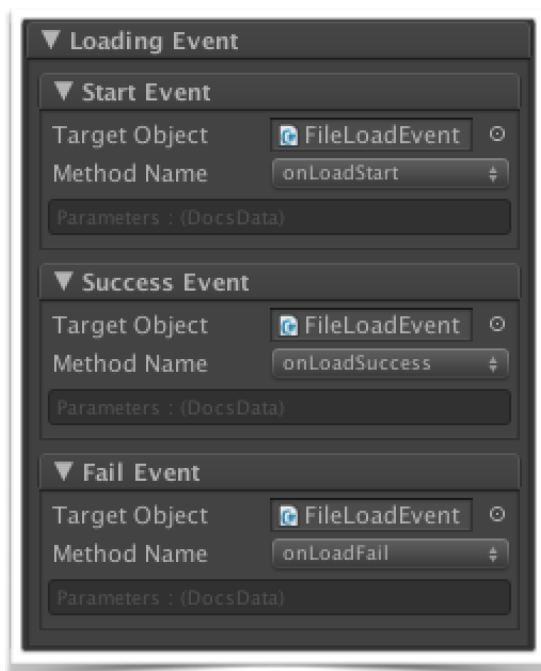
## 13. [Reference Object.\(v2.4.0\)](#)



- uGUIText/NGUText를 이용해서 해당 게임오브젝트를 참조하고 있는 Object를 보여준다.

-해당 옵션은 읽기 전용이며, 해당 파일이 어느 오브젝트가 참조하고 있는지 확인하기 위해 사용한다.

## 14. [File Loading Event.\(v2.4.0\)](#)

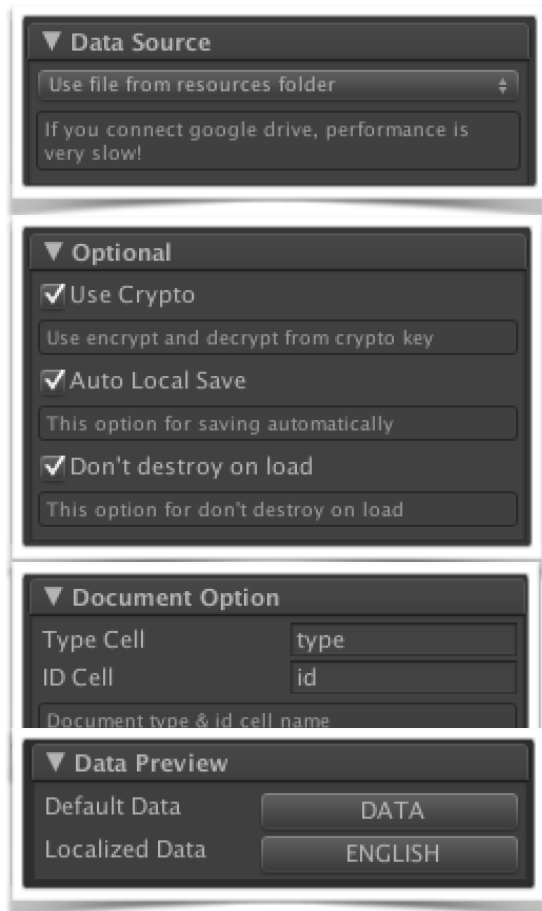


- Start Event : 해당 파일이 읽기 시작하기전에 타겟에 이벤트를 전달한다.

- Success Event : 해당 파일을 읽고 읽기가 성공한 경우 이벤트를 전달한다.

- Fail Event : 해당 파일을 읽고 읽기가 실패한 경우 이벤트를 전달한다.

## 15. 그외 옵션을 설정한다.



- 각각의 GameObject에서 로컬파일을 이용할지 Google Drive의 데이터를 실시간으로 이용할지를 선택할 수 있다.

- Use Crypto : 암호를 사용할지를 선택한다.

- Auto Local Save : Google Drive를 통해서 데이터를 가져온 경우에 자동으로 로컬에 저장할수 있다.

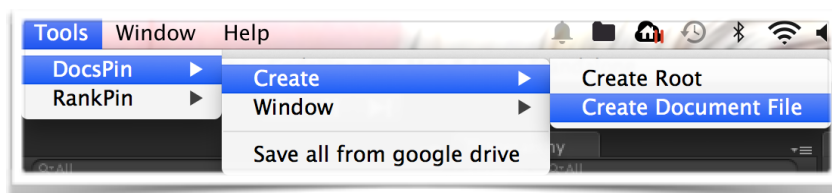
- Don't destroy on load : Scene 전환 시 해당 Object를 삭제하지 않고 사용할 수 있다.

- 스프레드 시트의 id와 type 셀의 이름을 변경하여 처리하는 경우 사용한다.

- Unity 실행시에만 확인할 수 있다.

- 현재 사용하고 있는 데이터를 확인할 수 있다.

## 16. 다른 스프레드 시트 파일을 추가하고 싶다면 상단 메뉴의 Create File을 선택한다.



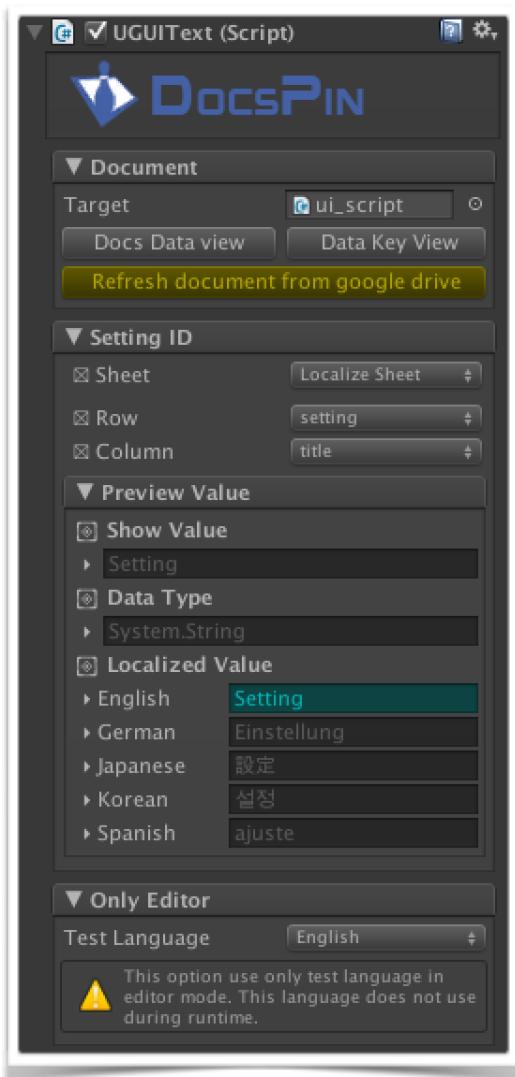
- 여러개의 스프레드 시트 데이터를 사용할 수 있다.

## Unity GUI Text (v2.2.0)

1. Unity GUI Text 게임 오브젝트를 만든다.
2. UnityEngine.UI.Text 게임 오브젝트에서 DocsPin.UI.uGUIText 컴포넌트를 Add한다.



3. Document Target을 설정하고 Row, Col ID를 설정한다.



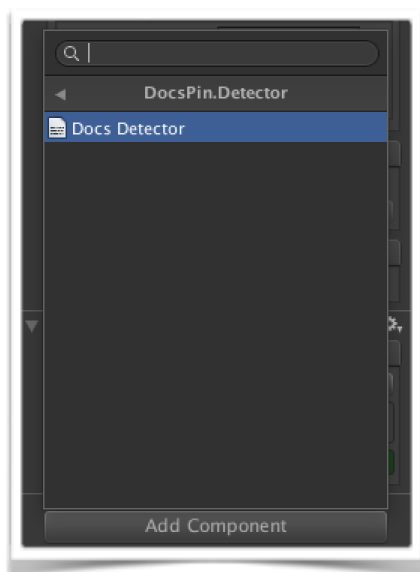
- Target : UIRoot 하위에 있는 File Object를 설정한다.
- Docs Data View : Docs 데이터를 볼수 있는 View
- Data Key View : Docs 데이터를 Key를 기준으로 볼 수 있는 View
- Refresh document from google drive : 해당 부분은 구글 드라이브 정보를 가져와서 로컬에 파일로 저장하고 해당 데이터를 Refresh한다.
- Sheet : Data Sheet / Localize Sheet 선택
- Row : Row ID
- Column : Column ID
- Test Language : 해당 옵션은 Editor상에서만 테스트가 가능하도록 언어 설정을 바꾸는 기능이다. 이 옵션값은 프로그램에 실행될 때에는 사용되지 않는다.

## Unity NGUI Label (v2.0.0)

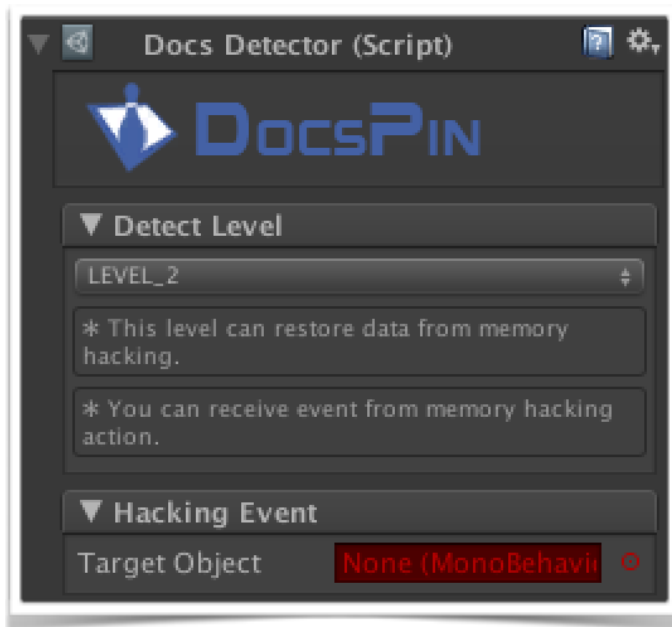
1. NGUI Label Text 게임 오브젝트를 만든다.
2. DocsPin/Scripts/UI/NGUIText.cs 의 주석을 제거한다.
3. UILabel 게임 오브젝트에서 DocsPin.UI.NGUIText 컴포넌트를 Add한다

## Memory Detect Type (v2.4.0)

1. DocsPin 게임 오브젝트에 DocsPin.Detector.DocsDetector Script를 추가한다.



## 2. DocsDetector 옵션을 설정한다.



- Detect Level : Detect Level을 설정한다.

- Hacking Event : 메모리 해킹을 감지 한 경우 이벤트를 받을수 있는 Target을 설정한다.

## 3. Detect Level

- Level\_0 : 데이터를 내부적으로 변형시켜 메모리 해킹 툴에서 메모리를 찾기 어렵게 한다. 해당 옵션을 선택했을 경우에 데이터가 메모리 해킹이 이루어질때 이벤트를 받지 못한다.
- Level\_1 : 데이터를 내부적으로 변형시키는 것은 Level\_0와 동일하나, 메모리 데이터가 변형이 일어난 경우 Hacking Event로 이벤트를 전달 받을 수 있다.
- Level\_2 : Level\_2와 동일하게 작동하나, 데이터가 복구 가능하면 복구해주는 기능을 가지고 있다. 즉 메모리 해킹이 일어나면 원본 데이터로 데이터를 복구해주고 Hacking Event로 이벤트를 전달해준다.
- \* 메모리 해킹을 방지하기 위해 별도의 메모리를 할당하게 되는데, Level\_0<Level\_1<Level\_2 순으로 메모리를 많이 사용한다. 즉 Level\_2가 가장 많은 별도의 메모리를 할당한다.
- \* 메모리 해킹은 100% 방지할 수 없습니다. 하지만 메모리 위치를 찾기 어렵게 만들며, 만약 해킹이 일어나면 기존 데이터로 복구하는것이 최선의 방법입니다. 하지만 이러한 처리로 대부분의 메모리 해킹은 포기하게 되므로 데이터를 안전하게 보호할 수 있습니다.



## Samples

### 1. Simple

스프레드 시트의 이름으로 데이터를 가져오는 예제

#### TestDocs

```
DocSPin.DocsData data = DocSPin.DocsRoot.findData("test_docs");
if(data != null)
{
    int price = data.get<int>("object_01", "price");
    int limit = data.get<int>("object_01", "limit");
    string name = data.get<string>("object_01", "name");
    string title = data.get<string>("object_01", "title");
    string content = data.get<string>("object_01", "content");
    Debug.Log(string.Format("Price:{0}, Limit:{1}, Name:{2}, Title:{3}, Content:{4}",
        price, limit, name, title, content));
}
```

#### \_gd\_sampe\_01

```
DocSPin.DocsData data = DocSPin.DocsRoot.findData("_gd_sample_01");
if(data != null)
{
    int gender = data.get<int>("object_01", "gender");
    int price = data.get<int>("object_01", "price");
    int limit = data.get<int>("object_01", "limit");
    ArrayList reward = data.get<ArrayList>("object_01", "reward");
    double length = data.get<int>("object_01", "length");
    string rewardString = "";
    foreach(object r in reward)
    {
        rewardString += string.Format("{0} ", r);
    }
    Debug.Log(string.Format("Gender:{0},Price:{1},Limit:{2},Reward:[{3}],Length:{4}",
        gender, price, limit, rewardString, length));
}
```

## 2. Data Singleton

스프레드 시트에 해당하는 GameObject를 Singleton으로 이용하는 예제

### TestDocs

```
int price = TestDocs.price("price");
int limit = TestDocs.limit("limit");
string name = TestDocs.names("name");
string title = TestDocs.title("title");
string content = TestDocs.content("content");
Debug.Log(string.Format("Price:{0}, Limit:{1}, Name:{2}, Title:{3}, Content:{4}",
    price, limit, name, title, content));
```

### \_gd\_sampe\_01

```
int gender = DataSample01.gender("object_01");
int price = DataSample01.price("object_01");
int limit = DataSample01.limit("object_01");
ArrayList reward = DataSample01.reward("object_01");
double length = DataSample01.length("object_01");
string rewardString = "";
foreach(object r in reward)
{
    rewardString += string.Format("{0} ", r);
}
Debug.Log(string.Format("Gender:{0},Price:{1},Limit:{2},Reward:[{3}],Length:{4}",
    gender, price, limit, rewardString, length));
```

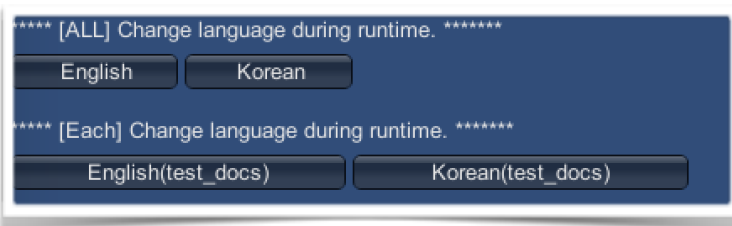
## 3. 실행 중 언어를 바꾸는 기능.(v2.1.0)

모든 언어를 한번에 변경할 때.

```
DocsPin.DocsRoot.reloadAll(SystemLanguage.Korean);
```

각각의 파일만 언어를 변경하고 싶을때.

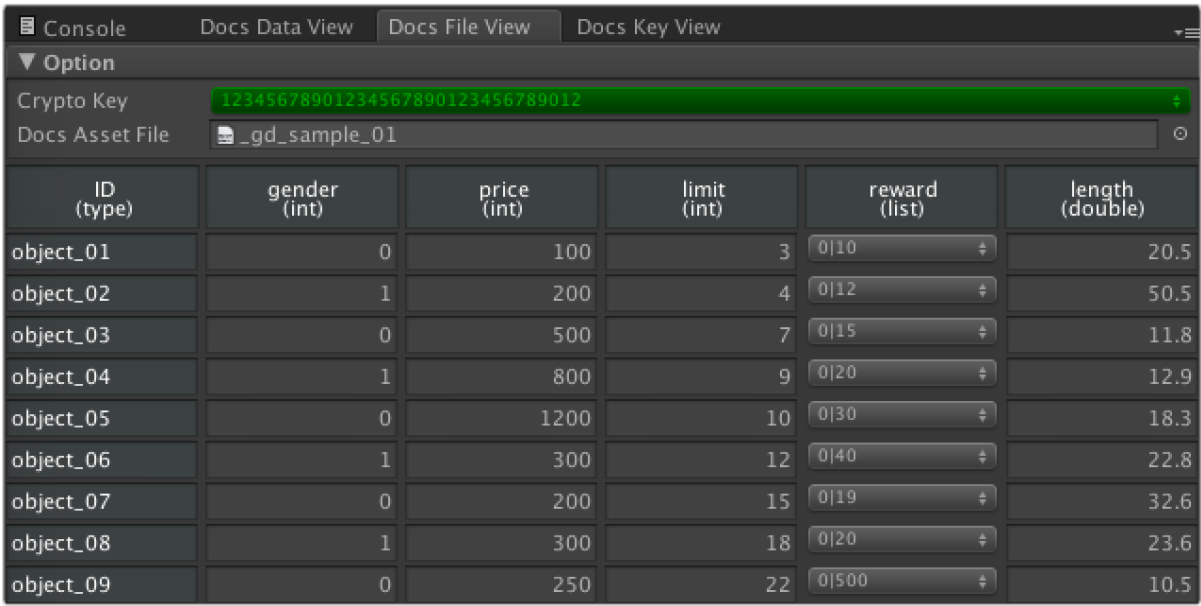
```
// By docs name.
DocsPin.DocsRoot.reloadData(SystemLanguage.Korean, "test_docs");
// By docs ID.
DocsPin.DocsRoot.reloadDataById(SystemLanguage.Korean, "1-T6HKz-GiHfv6lCcGv0nvCshPD0CVkHNRYlQUAedJio");
```



## Other Window

### 1. Docs File Window

로컬에 저장된 스프레드 시트 파일을 볼수 있는 윈도우이다.

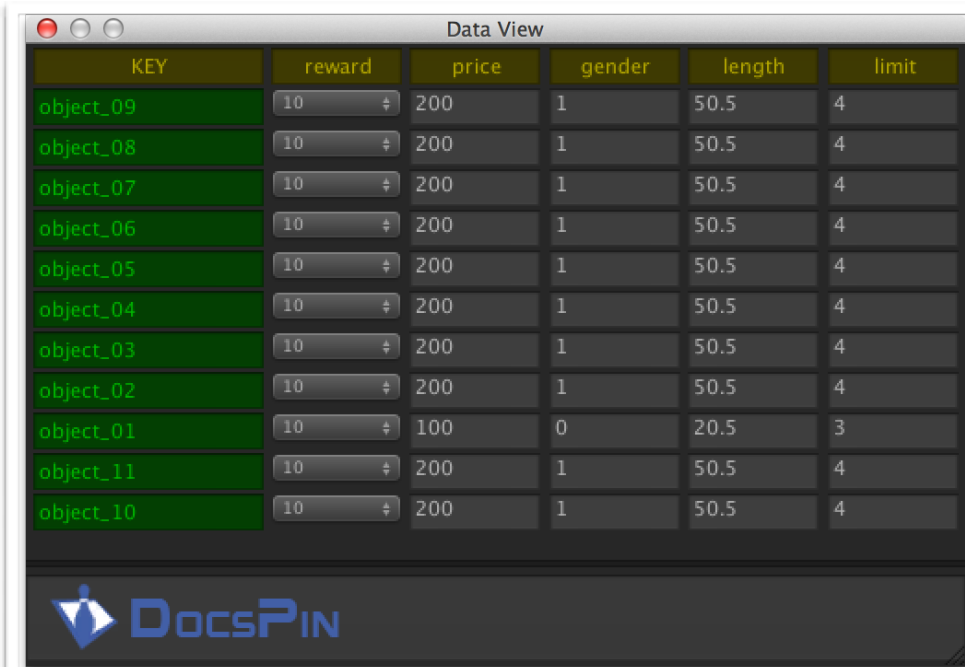


The screenshot shows the 'Docs File View' window. At the top, there are tabs for 'Console', 'Docs Data View', 'Docs File View' (selected), and 'Docs Key View'. Below the tabs, there's an 'Option' section with 'Crypto Key' set to '12345678901234567890123456789012' and 'Docs Asset File' set to '\_gd\_sample\_01'. The main area displays a table with the following data:

ID (type)	gender (int)	price (int)	limit (int)	reward (list)	length (double)
object_01	0	100	3	0 10	20.5
object_02	1	200	4	0 12	50.5
object_03	0	500	7	0 15	11.8
object_04	1	800	9	0 20	12.9
object_05	0	1200	10	0 30	18.3
object_06	1	300	12	0 40	22.8
object_07	0	200	15	0 19	32.6
object_08	1	300	18	0 20	23.6
object_09	0	250	22	0 500	10.5

### 2. Docs Data View Window

Unity 실행시 사용되는 스프레드 시트 파일을 볼수 있는 윈도우이다.



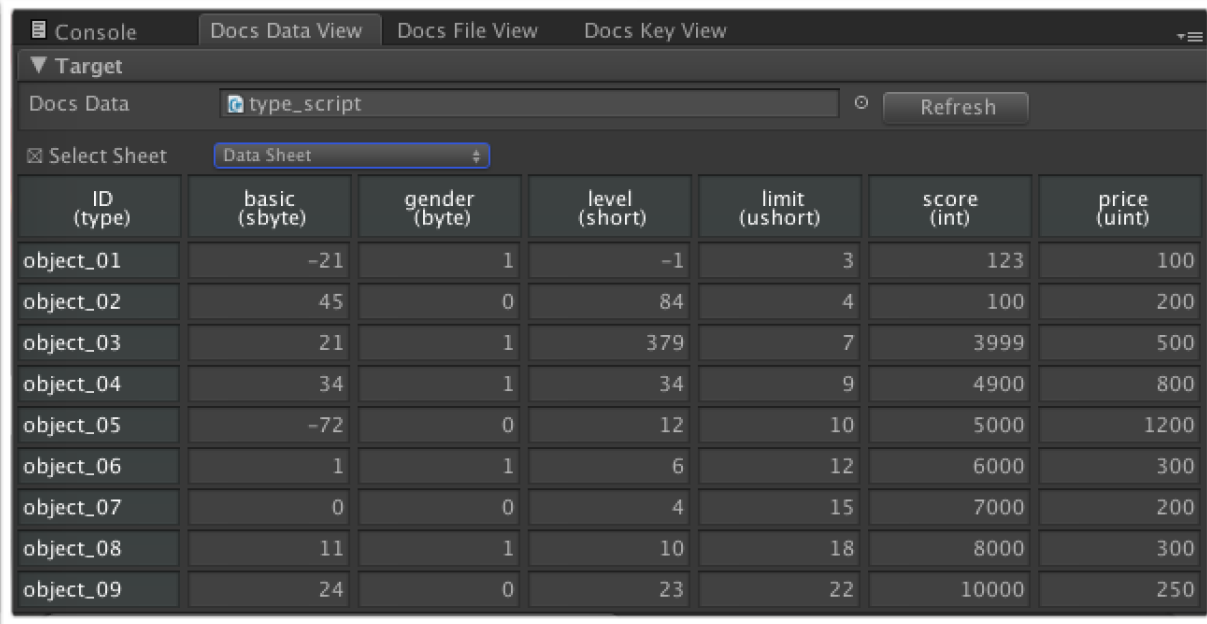
The screenshot shows the 'Data View' window. It displays a table with the following data:

KEY	reward	price	gender	length	limit
object_09	10	200	1	50.5	4
object_08	10	200	1	50.5	4
object_07	10	200	1	50.5	4
object_06	10	200	1	50.5	4
object_05	10	200	1	50.5	4
object_04	10	200	1	50.5	4
object_03	10	200	1	50.5	4
object_02	10	200	1	50.5	4
object_01	10	100	0	20.5	3
object_11	10	200	1	50.5	4
object_10	10	200	1	50.5	4

The DocsPin logo is visible at the bottom left of the window.

### 3. Docs Data View Window (v2.2.0)

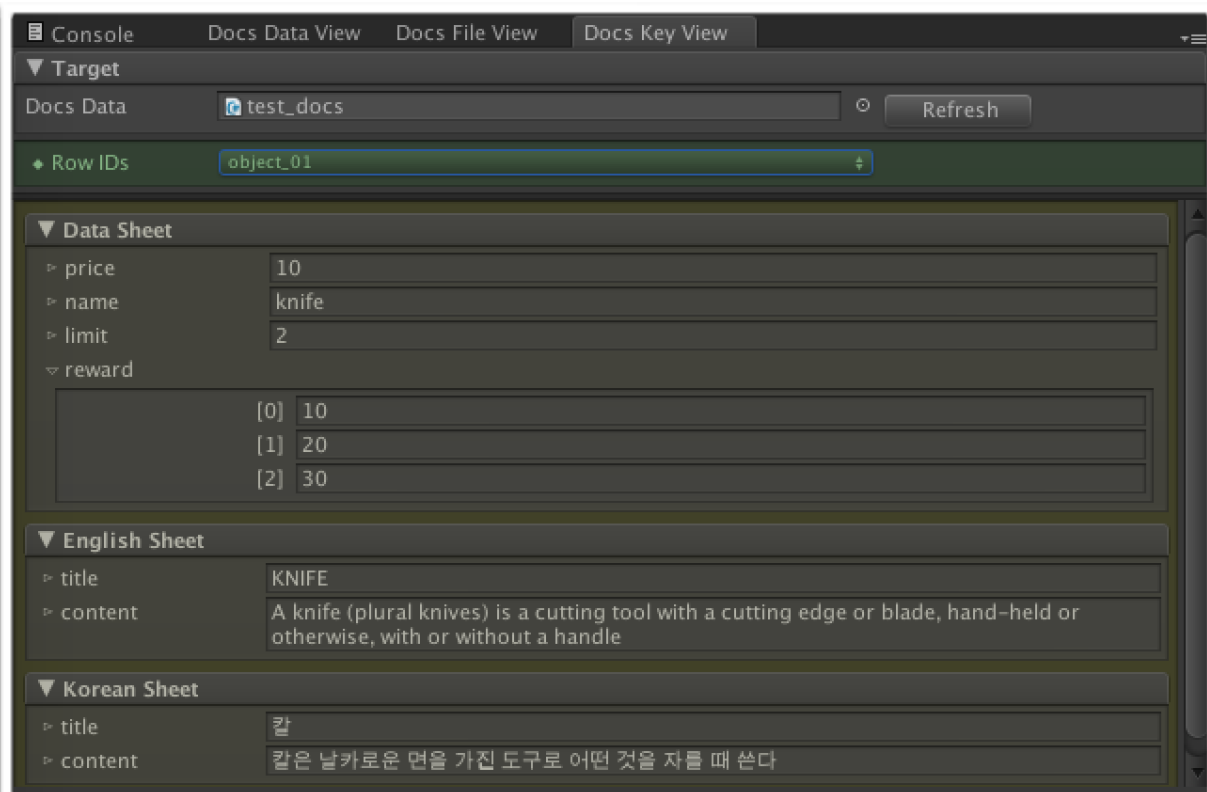
uGUIText, NGUIText에 설정된 Docs 데이터를 볼 수 있는 화면이다.



The screenshot shows the 'Docs Data View' window with the 'Target' set to 'type\_script'. The 'Data Sheet' is selected, displaying a table with 7 columns: ID (type), basic (sbyte), gender (byte), level (short), limit (ushort), score (int), and price (uint). The table contains 9 rows of data for objects object\_01 through object\_09.

ID (type)	basic (sbyte)	gender (byte)	level (short)	limit (ushort)	score (int)	price (uint)
object_01	-21	1	-1	3	123	100
object_02	45	0	84	4	100	200
object_03	21	1	379	7	3999	500
object_04	34	1	34	9	4900	800
object_05	-72	0	12	10	5000	1200
object_06	1	1	6	12	6000	300
object_07	0	0	4	15	7000	200
object_08	11	1	10	18	8000	300
object_09	24	0	23	22	10000	250

### 4. Docs Key View Window (v2.4.0)



The screenshot shows the 'Docs Key View' window with the 'Target' set to 'test\_docs'. The 'Row IDs' are set to 'object\_01'. The 'Data Sheet' is expanded, showing fields like price, name, limit, and reward. The 'English Sheet' and 'Korean Sheet' are also expanded, showing title and content for 'KNIFE' and '칼' respectively.

Data Sheet	
price	10
name	knife
limit	2
reward	[0] 10 [1] 20 [2] 30

English Sheet	
title	KNIFE
content	A knife (plural knives) is a cutting tool with a cutting edge or blade, hand-held or otherwise, with or without a handle

Korean Sheet	
title	칼
content	칼은 날카로운 면을 가진 도구로 어떤 것을 자를 때 쓴다

## Sample Google Drive 스프레드 시트

### 1. Test Docs

	A	B	C	D	E
1	type	int	string	double	list
2	id	price	name	limit	reward
3	object_01	10	knife	2	10,20,30
4	object_02	20	gun	1	20,50,100
5					
6					

+ 
≡ 
data 
English 
Korean 
✓

	A	B	C
1	type	string	string
2	id	title	content
3	object_01	KNIFE	A knife (plural knives) is a cutting tool with a cutting edge or blade, hand-held or otherwise, with or without a handle
4	object_02	GUN	A gun is a normally tubular weapon or other device designed to discharge projectiles or other material
5			

+ 
≡ 
data 
English 
Korean 
✓

	A	B	C
1	type	string	string
2	id	title	content
3	object_01	칼	칼은 날카로운 면을 가진 도구로 어떤 것을 자를 때 쓴다
4	object_02	총	총(銃)은 일반적으로 개인이 휴대할 수 있는 구경이 작은 화기를 말한다.
5			
6			
7			

+ 
≡ 
data 
English 
Korean 
✓

2. `_gd_sample_01`

	A	B	C	D	E	F
1	type	int	int	int	list	double
2	id	gender	price	limit	reward	length
3	object_01	0	100	3	10,100,200	20.5
4	object_02	1	200	4	10,100,200	50.5
5	object_03	1	200	4	10,100,200	50.5
6	object_04	1	200	4	10,100,200	50.5
7	object_05	1	200	4	10,100,200	50.5
8	object_06	1	200	4	10,100,200	50.5
9	object_07	1	200	4	10,100,200	50.5
10	object_08	1	200	4	10,100,200	50.5
11	object_09	1	200	4	10,100,200	50.5
12	object_10	1	200	4	10,100,200	50.5
13	object_11	1	200	4	10,100,200	50.5
14						
15						

+

≡

시트1 ▾

✓

### 3. ui\_script (v2.2.0)

The image displays three overlapping screenshots of a Google Sheets application window titled 'ui\_script'. The interface includes a menu bar (File, Edit, View, Insert, Format, Data, Tools, Add-ons, Help) and a toolbar with icons for undo, redo, and text formatting. The main area shows a spreadsheet with columns A, B, C, D, and E.

**Top Screenshot:** Shows the initial state with a formula bar containing 'fx'. The spreadsheet has a header row with columns A, B, C, D, and E. The data rows are:

	A	B	C	D	E
1	type	string			
2	id	view			
3	name		option		
4	age		option		
5	gender		option		
6	setting		option		
7					

**Middle Screenshot:** Shows the spreadsheet with the 'Setting' column (C) highlighted. The data rows are:

	A	B	C	D	E
1	type	string			
2	id	title			
3	name	Name			
4	age	Age			
5	gender	Gender			
6	setting	Setting			
7					

**Bottom Screenshot:** Shows the spreadsheet with the 'type' column (A) highlighted. The data rows are:

	A	B	C	D	E
1	type	string			
2	id	title			
3	name	이름			
4	age	나이			
5	gender	성별			
6	setting	설정			
7					

The bottom screenshot also shows a language selection bar at the bottom with options: data, English, Korean, German, Japanese, and Spanish.

#### 4. type\_script (v2.3.0)

type_script															
File Edit View Insert Format Data Tools Add-ons Help All changes saved in Drive															
Comments Share															
fx 849373															
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	type	sbyte	byte	short	ushort	int	uint	long	ulong	float	double	decimal	bool	string	
2	id	basic	gender	level	limit	score	price	max	reward	length	distance	far	success	name	
3	object_01	-21	1	-1	3	123	100	900000	12345567098	20.5	34324.5656		4345345	TRUE	John
4	object_02	45	0	84	4	100	200	1000000	18585037405	50.5	23423.78568		345345435	FALSE	Steive
5	object_03	21	1	379	7	3999	500	21000000	847493954847	11.8	3234.5756		345345345345	TRUE	David
6	object_04	34	1	34	9	4900	800	22000000	8473266492	12.9	866.464546		53443534534534	FALSE	Tomas
7	object_05	-72	0	12	10	5000	1200	23000000	848373	18.3	5653.786857		345346436	TRUE	Tom
8	object_06	1	1	6	12	6000	300	40000000	8694838383749	22.8	2323.232535		43646346	FALSE	Henry
9	object_07	0	0	4	15	7000	200	50000000	849373	32.6	56456.45646		43534634	TRUE	Andrew
10	object_08	11	1	10	18	8000	300	100000000	61263	23.6	3434.35345		64363463	FALSE	Daniel
11	object_09	24	0	23	22	10000	250	200000000	747464739393	10.5	45345.35		35356	TRUE	Oliver
12															

#### 5. memory\_detector (v2.4.0)

	A	B	C	D	E	F	G	H	I
type	int	long	m-int	m-long	list	m-list	list	m-list	
id	price1	reward1	price2	reward2	items1	items2	status1	status2	
object_01	12300000	1000000000000000	11000000	1000000000000000	item0,item4,item	item10,item11,ite	true,false	true,true	
object_02	22300000	2200000000000000	42600000	67675453436575	item4,item3,item	item10,item11,ite	true,false	true,true	
object_03	32300000	5646000000000000	54643000	56456654634	item6,item8,item	item10,item11,ite	true,false	true,true	
object_04	12301200	4676572000000000	34345400	345345252352357	item7,item4,item	item10,item11,ite	true,false	true,true	
object_05	14600000	425464265443366	11000000	34545454	item4,item4,item	item10,item11,ite	true,false	true,true	
object_06	32235345	4678976000000000	45600000	35645527	item0,item4,item	item10,item11,ite	true,false	true,true	
object_07	54375345	9683700000000000	11000000	23557456453523	item0,item4,item	item10,item11,ite	true,false	true,true	
object_08	56543455	6000000000000000	32423432	4545456	item0,item4,item	item10,item11,ite	true,false	true,true	
object_09	12300000	7000000000000000	34535743	363346346	item0 item4 item	item10 item11 ite	true false	true true	