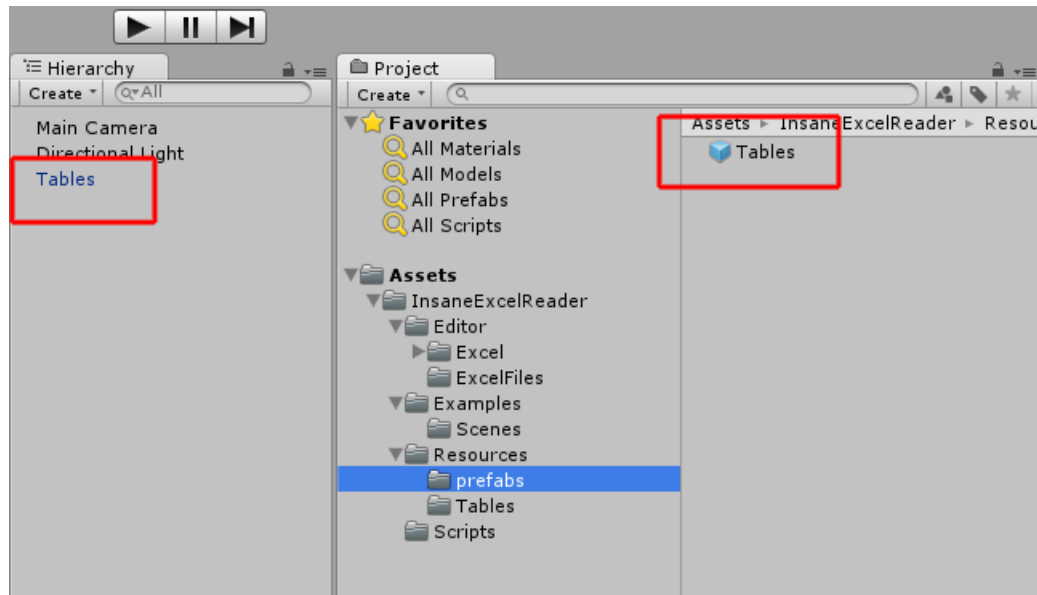


How to use InsaneExcelManager.

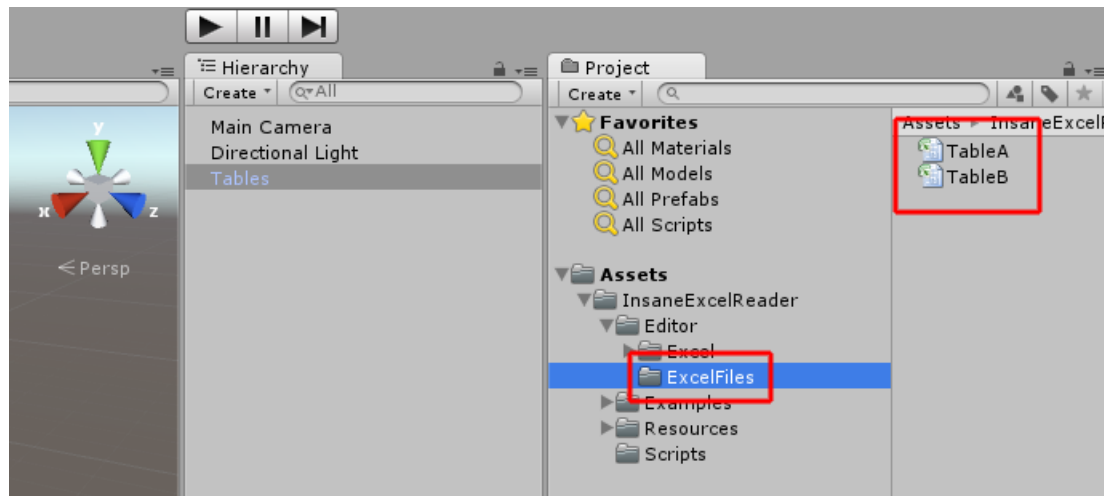
Please refer to this small video.

<https://www.youtube.com/watch?v=sloTiei0XwM>

Drag Tables prefab to Hierarchy view in the following image.



There are some Excel files for example.



There are Excel files named Player, Level, Item, Enemy.

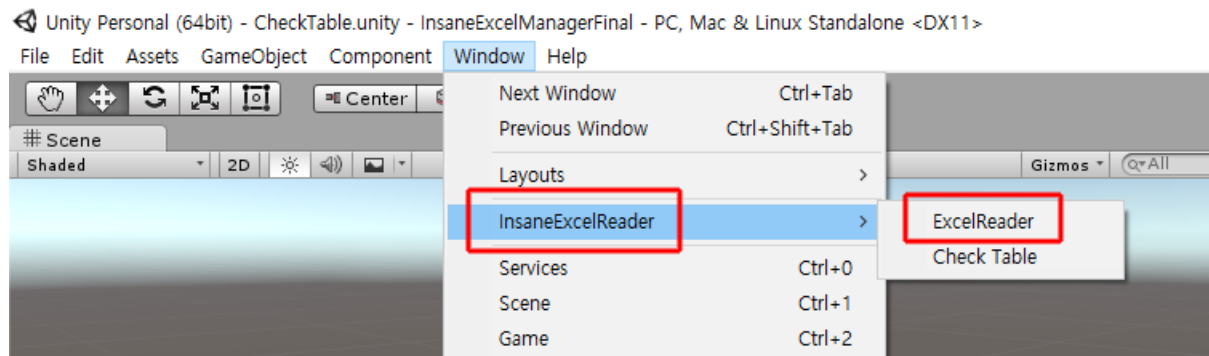
	A	B	C	D	E	F	G	H	I	J	K
1	int	int	string	int	oops	int	int				
2	key	ID	Name	BaseHealth		BaseAttack	BaseDefense				
3	100001	100001	Bulcan	1000		100	50				
4	100002	100002	Player0	1100		110	55				
5	100003	100003	Alpha	1200		120	60				
6	100004	100004	John	1300		130	65				
7	100005	100005	Omega	1400		140	70				
8											
9											
10											

At the bottom of the table, there are tabs for 'Player', 'Level', and 'Item'. The 'Player' tab is highlighted with a red box.

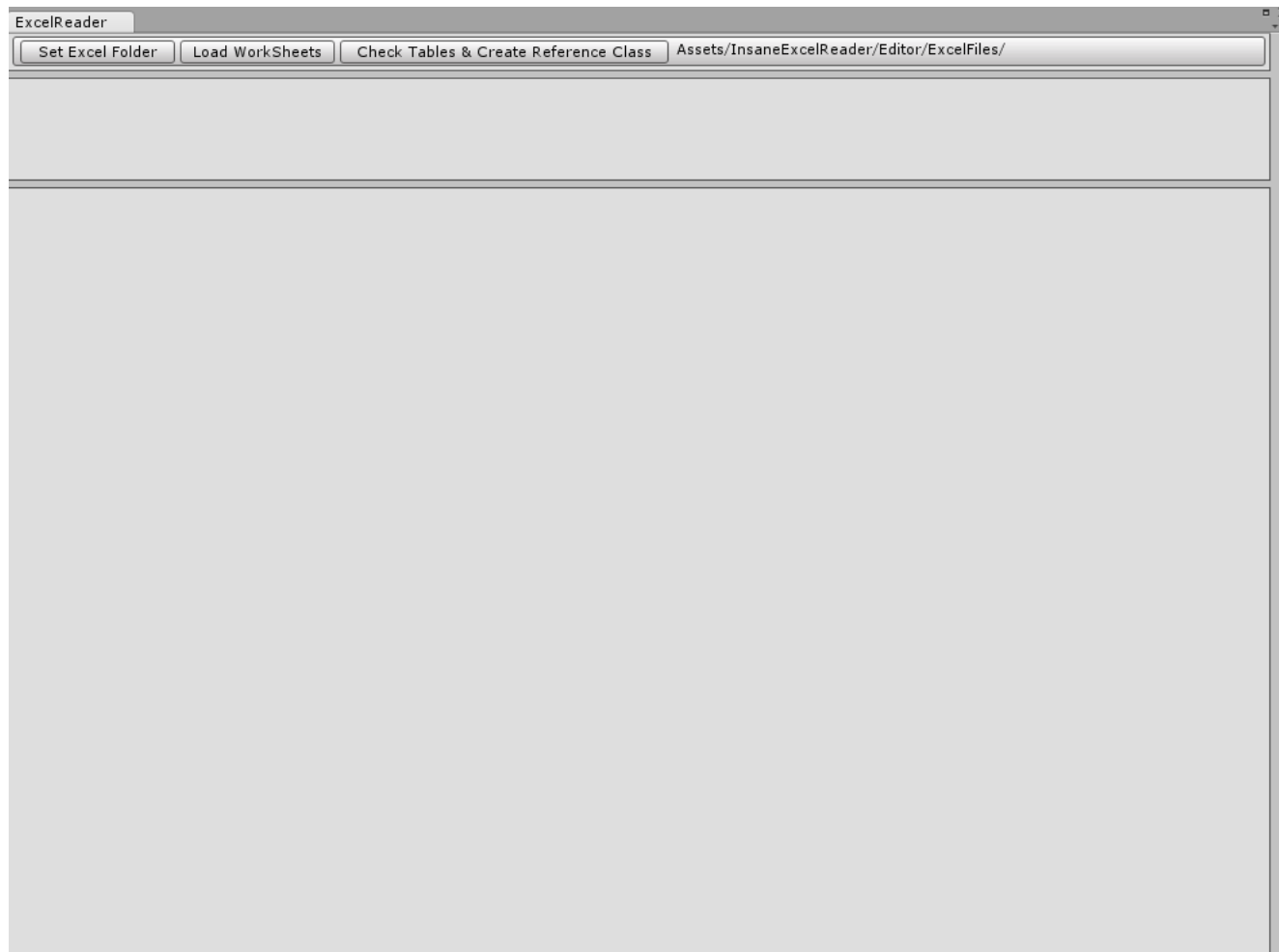
	A	B	C	D	E	F	G	H	I	J
1	string	string	string	int	int	int				
2	Key	ID	Name	BaseHealth	PlayerHealth	BaseDefense				
3	200001	200001	Enemy001	1000		50				
4	200002	200002	Enemy002	1100	110	55				
5	200003	200003	Enemy003	1200	120	60				
6	200004	200004		1300	130	65				
7	200005	200005	Enemy005	1400	140	70				
8	200006	200006	Enemy006	1500	150	75				
9	200007	200007	Enemy007		160	80				
10	200008	200008	Enemy008	1700	170	85				
11	200009	200009	Enemy009	1800	180	90				
12	200010	200010	Enemy010	1900	190	95				
13										

Now, let's read all tables using InsaneExcelManager.

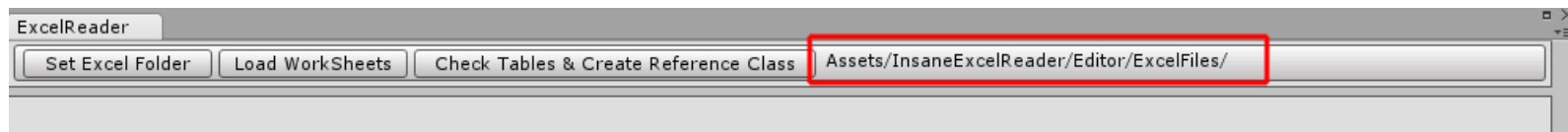
Select InsaneExcelReader in the following image.

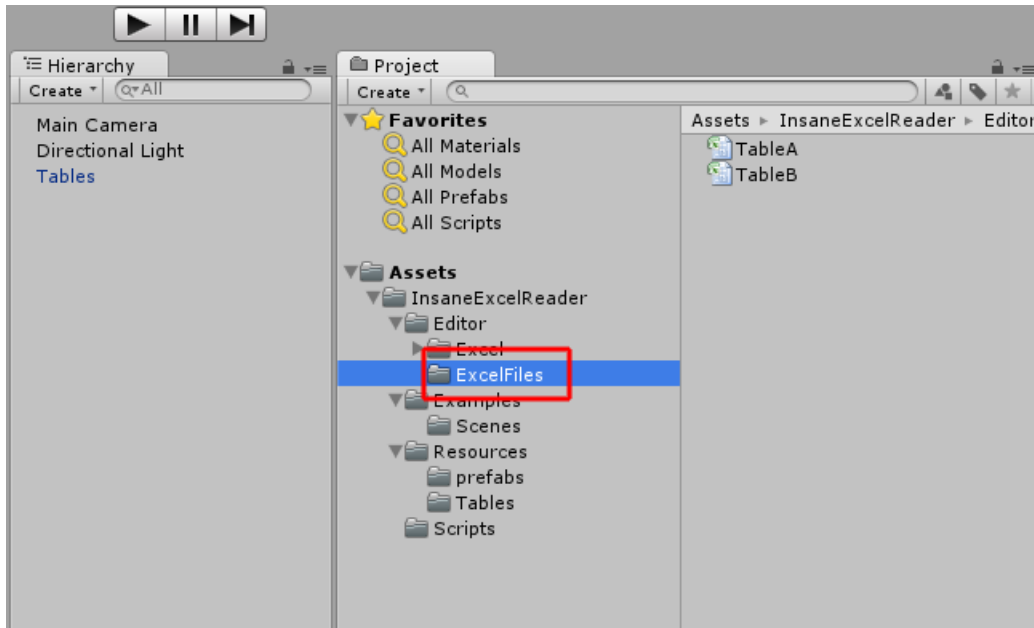


You can see the tool in the following image.

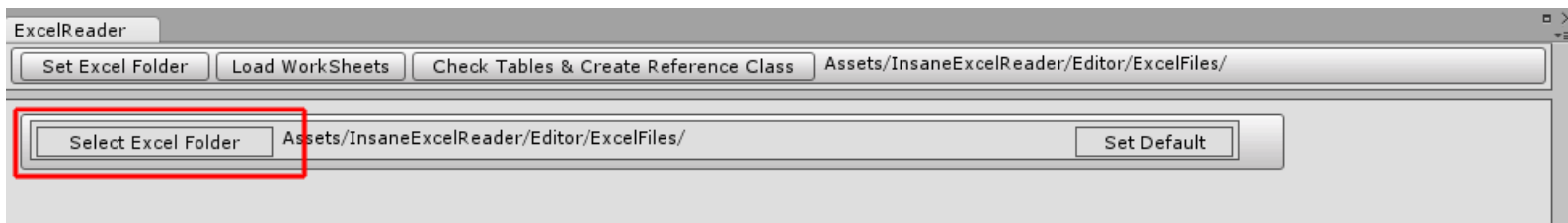
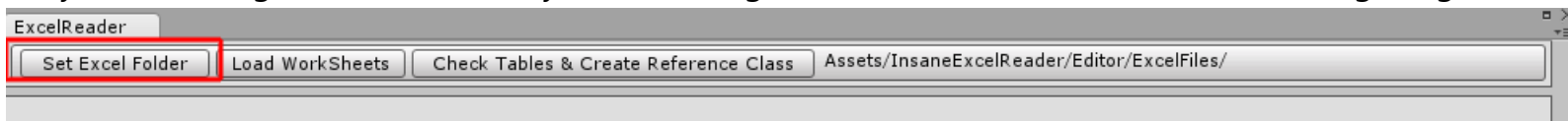


Default folder is Assets/InsaneExcelReader/Editor/ExcelFiles.

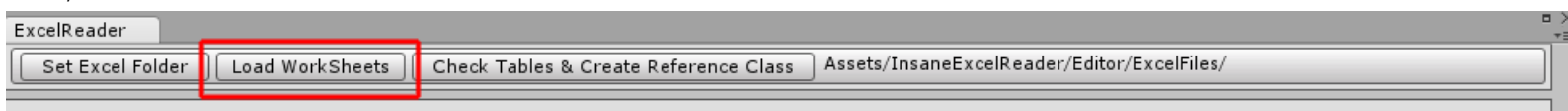




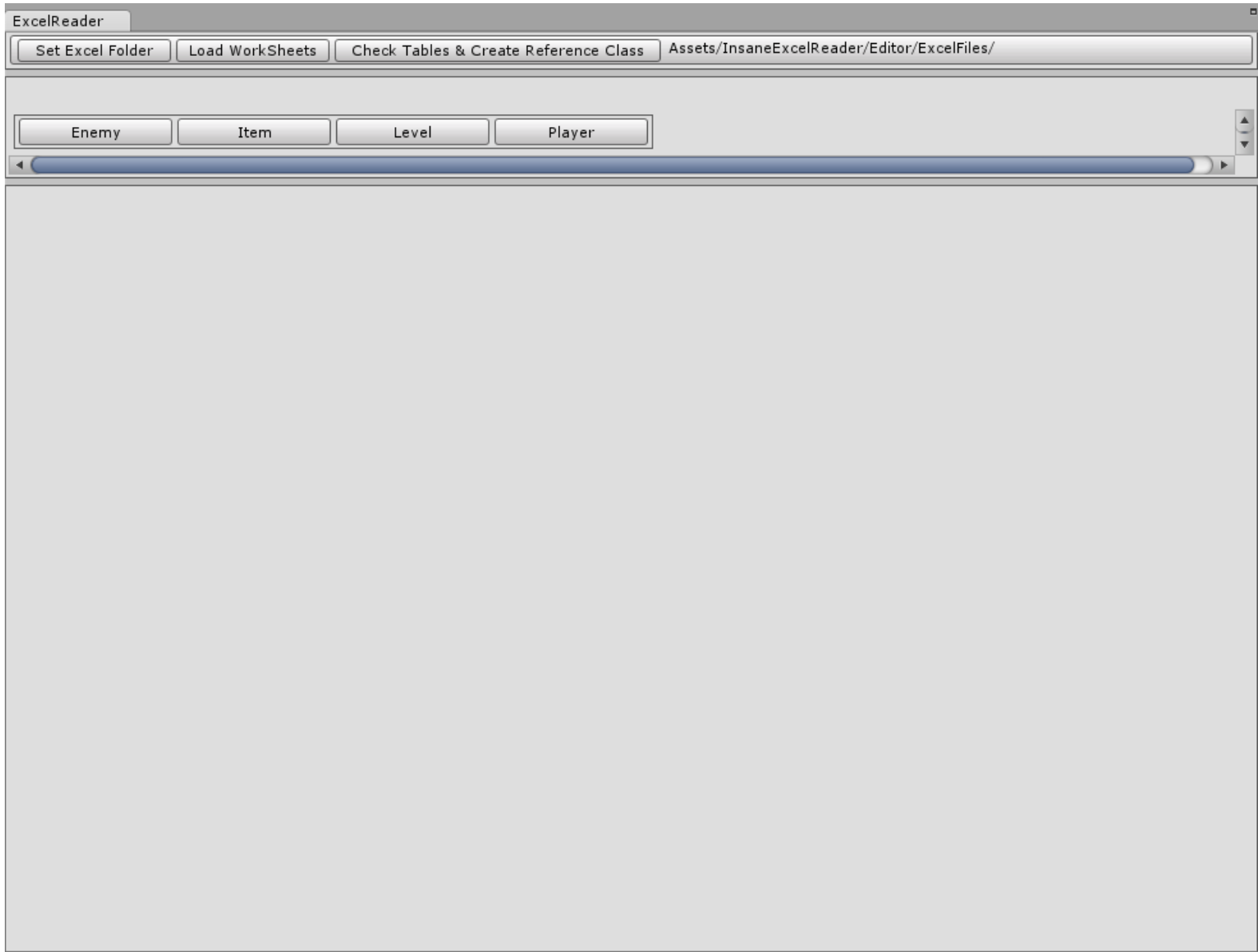
But you can change folder whenever you want using "Set Excel Folder" button in the following image.



Now, let's read Excel files. Click the button named "Load WorkSheets"



You can see the result in the following image.



Click each buttons.

ExcelReader

Set Excel FolderLoad WorkSheetsCheck Tables & Create Reference ClassAssets/InsaneExcelReader/Editor/ExcelFiles/

Enemy worksheet is selected !!

Enemy

Item

Level

Player

[Variable Type]	string	string	string	int	int	int
[Variable Name]	Key	ID	Name	BaseHealth	PlayerHealth	BaseDefense
0	200001	200001	Enemy001	1000	0	50
1	200002	200002	Enemy002	1100	110	55
2	200003	200003	Enemy003	1200	120	60
3	200004	200004	None	1300	130	65
4	200005	200005	Enemy005	1400	140	70
5	200006	200006	Enemy006	1500	150	75
6	200007	200007	Enemy007	0	160	80
7	200008	200008	Enemy008	1700	170	85
8	200009	200009	Enemy009	1800	180	90
9	200010	200010	Enemy010	1900	190	95

ExcelReader

Set Excel FolderLoad WorkSheetsCheck Tables & Create Reference ClassAssets/InsaneExcelReader/Editor/ExcelFiles/

Item worksheet is selected !!

EnemyItemLevelPlayer

[Variable Type]	int	int	float	float	float	string
[Variable Name]	Key	ID	AddHealth	AddAttack	AddDefense	Name
0	30000001	30000001	2.1	3.1	2.5	Item001
1	30000002	30000002	2.2	3.2	2.6	Item002
2	30000003	30000003	2.3	3.3	2.7	Item003
3	30000004	30000004	2.4	3.4	2.8	Item004
4	30000005	30000005	2.5	3.5	2.9	Item005
5	30000006	30000006	2.6	3.6	3	Item006
6	30000007	30000007	2.7	3.7	3.1	Item007
7	30000008	30000008	2.8	3.8	3.2	Item008
8	30000009	30000009	2.9	3.9	3.3	Item009
9	30000010	30000010	3	4	3.4	Item010
10	30000011	30000011	3.1	4.1	3.5	Item011
11	30000012	30000012	3.2	4.2	3.6	Item012
12	30000013	30000013	3.3	4.3	3.7	Item013

ExcelReader

Set Excel FolderLoad WorkSheetsCheck Tables & Create Reference ClassAssets/InsaneExcelReader/Editor/ExcelFiles/

Level worksheet is selected !!

EnemyItemLevelPlayer

[Variable Type]	int	int	float	float	float
[Variable Name]	Key	ID	AddHealth	AddAttack	AddDefense
0	1	1	0	0	0
1	2	2	0.3	0.2	0.25
2	3	3	0.6	0.3	0.3
3	4	4	0.9	0.4	0.35
4	5	5	1.2	0.5	0.4
5	6	6	1.5	0.6	0.45
6	7	7	1.8	0.7	0.5
7	8	8	2.1	0.8	0.55
8	9	9	2.4	0.9	0.6
9	10	10	2.7	1	0.65
10	11	11	3	1.1	0.7
11	12	12	3.3	1.2	0.75
12	13	13	3.6	1.3	0.8
13	14	14	3.9	1.4	0.85
14	15	15	4.2	1.5	0.9

ExcelReader

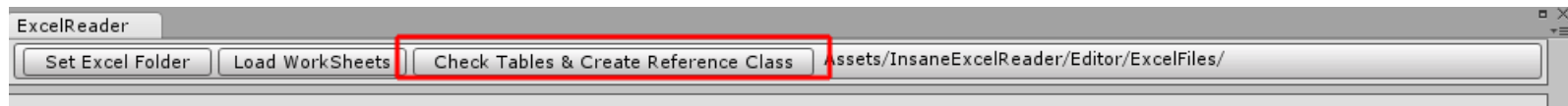
Set Excel Folder Load WorkSheets Check Tables & Create Reference Class Assets/InsaneExcelReader/Editor/ExcelFiles/

Player worksheet is selected !!

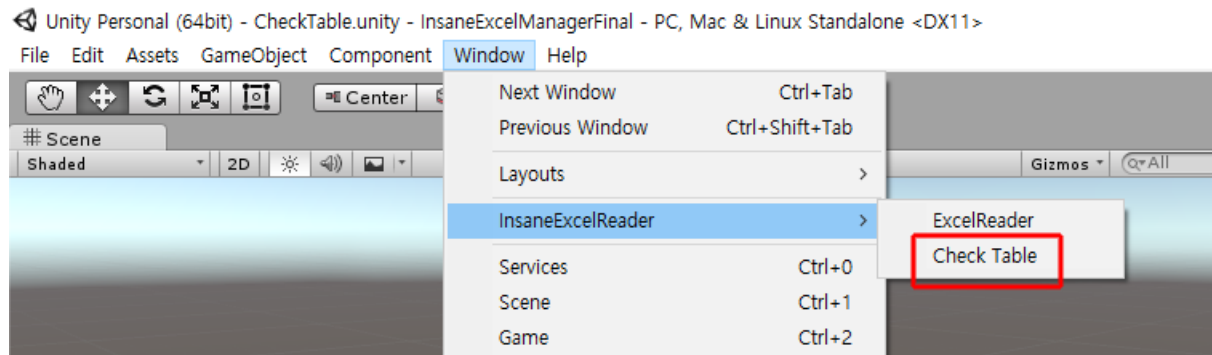
Enemy Item Level **Player**

[Variable Type]	int	int	string	int	int	int
[Variable Name]	key	ID	Name	BaseHealth	BaseAttack	BaseDefense
0	100001	100001	Bulcan	1000	100	50
1	100002	100002	Player0	1100	110	55
2	100003	100003	Alpha	1200	120	60
3	100004	100004	John	1300	130	65
4	100005	100005	Omega	1400	140	70

Now, let's export tables for game. Click the button named "Check Tables & Create Reference Class"



Finished. So simple. Let's check in the game. Click "Check Table".



You can see the result in game in the following image.

Game

Display 1

AAA (1280x720)

Maximize on

Enemy

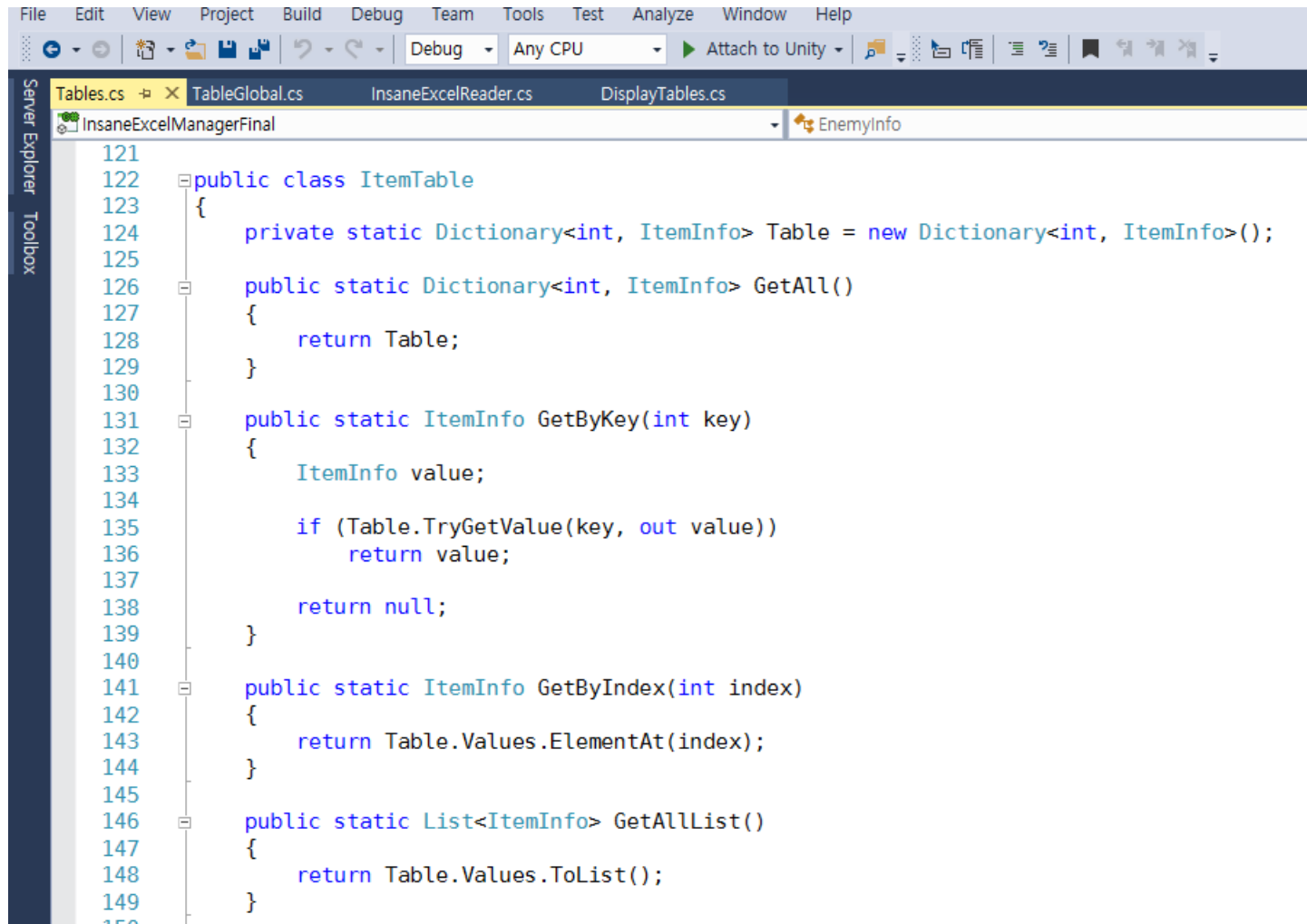
Item

Level

Player

Key: 200001		ID: 200001	Name: Enemy001	BaseHealth: 1000	PlayerHealth: 0	BaseDefen
Key: 200002		ID: 200002	Name: Enemy002	BaseHealth: 1100	PlayerHealth: 110	BaseDefen
Key: 200003		ID: 200003	Name: Enemy003	BaseHealth: 1200	PlayerHealth: 120	BaseDefen
Key: 200004		ID: 200004	Name: None	BaseHealth: 1300	PlayerHealth: 130	BaseDefen
Key: 200005		ID: 200005	Name: Enemy005	BaseHealth: 1400	PlayerHealth: 140	BaseDefen
Key: 200006		ID: 200006	Name: Enemy006	BaseHealth: 1500	PlayerHealth: 150	BaseDefen
Key: 200007		ID: 200007	Name: Enemy007	BaseHealth: 0	PlayerHealth: 160	BaseDefen
Key: 200008		ID: 200008	Name: Enemy008	BaseHealth: 1700	PlayerHealth: 170	BaseDefen
Key: 200009		ID: 200009	Name: Enemy009	BaseHealth: 1800	PlayerHealth: 180	BaseDefen
Key: 200010		ID: 200010	Name: Enemy010	BaseHealth: 1900	PlayerHealth: 190	BaseDefen

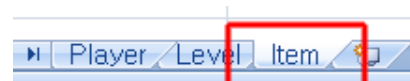
Here is a Class explanation. Four function is prepared for you. For example, Item table in the following image.
 GetAll(), GetByKey(int key), GetByIndex(int index), GetAllList function is prepared for you.



```
121
122 public class ItemTable
123 {
124     private static Dictionary<int, ItemInfo> Table = new Dictionary<int, ItemInfo>();
125
126     public static Dictionary<int, ItemInfo> GetAll()
127     {
128         return Table;
129     }
130
131     public static ItemInfo GetByKey(int key)
132     {
133         ItemInfo value;
134
135         if (Table.TryGetValue(key, out value))
136             return value;
137
138         return null;
139     }
140
141     public static ItemInfo GetByIndex(int index)
142     {
143         return Table.Values.ElementAt(index);
144     }
145
146     public static List<ItemInfo> GetAllList()
147     {
148         return Table.Values.ToList();
149     }
150 }
```

Use this function as follows.

If table name is item?



You can use Item + Table as follows.

```

var ItemAll = ItemTable.GetAll();
var ItemIndex = ItemTable.GetByIndex(0);
var ItemKey = ItemTable.GetByKey(30000001);
var ItemList = ItemTable.GetAllList();

```



If table name is Level?

You can use Item + Table as follows.

```

var LevelAll = LevelTable.GetAll();
var LevelIndex = LevelTable.GetByIndex(0);
var LevelKey = LevelTable.GetByKey(1);
var LevelList = LevelTable.GetAllList();

```



If table name is Player?

You can use Player + Table as follows.

```

var PlayerAll = PlayerTable.GetAll();
var PlayerIndex = PlayerTable.GetByIndex(0);
var PlayerKey = PlayerTable.GetByKey(100001);
var PlayerList = PlayerTable.GetAllList();

```

Please refer to "DisplayTables.cs" script.

```

var EnemyAll = EnemyTable.GetAll();
var EnemyIndex = EnemyTable.GetByIndex(0);
var EnemyKey = EnemyTable.GetByKey("200001");
var EnemyList = EnemyTable.GetAllList();

```

```

Debug.Log(" < ---EnemyTable Dictionary --->");

```

```

foreach (var item in EnemyAll)
    Debug.Log(string.Format("Key = {0}, m_ID = {1},m_Name = {2},m_BaseHealth = {3},m_PlayerHealth = {4},m_BaseDefense = {5},"
item.Key ,item.Value.m_ID ,item.Value.m_Name ,item.Value.m_BaseHealth ,item.Value.m_PlayerHealth ,item.Value.m_BaseDefense));

```

```

Debug.Log(" < ---EnemyTable Dictionary Index --->");

```

```

Debug.Log(string.Format("m_ID = {0},m_Name = {1},m_BaseHealth = {2},m_PlayerHealth = {3},m_BaseDefense =
{4}," ,EnemyIndex.m_ID ,EnemyIndex.m_Name ,EnemyIndex.m_BaseHealth ,EnemyIndex.m_PlayerHealth ,EnemyIndex.m_BaseDefense));

```

```

Debug.Log(" < ---EnemyTable Dictionary Key --->");

```

```

Debug.Log(string.Format("m_ID = {0},m_Name = {1},m_BaseHealth = {2},m_PlayerHealth = {3},m_BaseDefense =

```

```

{4}," ,EnemyKey.m_ID ,EnemyKey.m_Name ,EnemyKey.m_BaseHealth ,EnemyKey.m_PlayerHealth ,EnemyKey.m_BaseDefense));

Debug.Log(" < ---EnemyTable List --->");

foreach (var item in EnemyList)
    Debug.Log(string.Format("m_ID = {0},m_Name = {1},m_BaseHealth = {2},m_PlayerHealth = {3},m_BaseDefense = {4}," ,item.m_ID ,item.m_Name ,item.m_BaseHealth ,item.m_PlayerHealth ,item.m_BaseDefense));

var ItemAll = ItemTable.GetAll();
var ItemIndex = ItemTable.GetByIndex(0);
var ItemKey = ItemTable.GetByKey(30000001);
var ItemList = ItemTable.GetAllList();

Debug.Log(" < ---ItemTable Dictionary --->");

foreach (var item in ItemAll)
    Debug.Log(string.Format("Key = {0}, m_ID = {1},m_AddHealth = {2},m_AddAttack = {3},m_AddDefense = {4},m_Name = {5}," ,item.Key ,item.Value.m_ID ,item.Value.m_AddHealth ,item.Value.m_AddAttack ,item.Value.m_AddDefense ,item.Value.m_Name));

Debug.Log(" < ---ItemTable Dictionary Index --->");
Debug.Log(string.Format("m_ID = {0},m_AddHealth = {1},m_AddAttack = {2},m_AddDefense = {3},m_Name = {4}," ,ItemIndex.m_ID ,ItemIndex.m_AddHealth ,ItemIndex.m_AddAttack ,ItemIndex.m_AddDefense ,ItemIndex.m_Name));

Debug.Log(" < ---ItemTable Dictionary Key --->");
Debug.Log(string.Format("m_ID = {0},m_AddHealth = {1},m_AddAttack = {2},m_AddDefense = {3},m_Name = {4}," ,ItemKey.m_ID ,ItemKey.m_AddHealth ,ItemKey.m_AddAttack ,ItemKey.m_AddDefense ,ItemKey.m_Name));

Debug.Log(" < ---ItemTable List --->");

foreach (var item in ItemList)
    Debug.Log(string.Format("m_ID = {0},m_AddHealth = {1},m_AddAttack = {2},m_AddDefense = {3},m_Name = {4}," ,item.m_ID ,item.m_AddHealth ,item.m_AddAttack ,item.m_AddDefense ,item.m_Name));

var LevelAll = LevelTable.GetAll();
var LevelIndex = LevelTable.GetByIndex(0);
var LevelKey = LevelTable.GetByKey(1);
var LevelList = LevelTable.GetAllList();

Debug.Log(" < ---LevelTable Dictionary --->");

foreach (var item in LevelAll)
    Debug.Log(string.Format("Key = {0}, m_ID = {1},m_AddHealth = {2},m_AddAttack = {3},m_AddDefense = {4}," ,item.Key ,item.Value.m_ID ,item.Value.m_AddHealth ,item.Value.m_AddAttack ,item.Value.m_AddDefense));

Debug.Log(" < ---LevelTable Dictionary Index --->");
Debug.Log(string.Format("m_ID = {0},m_AddHealth = {1},m_AddAttack = {2},m_AddDefense = {3}," ,LevelIndex.m_ID ,LevelIndex.m_AddHealth ,LevelIndex.m_AddAttack ,LevelIndex.m_AddDefense));

```



```

Debug.Log(" < ---LevelTable Dictionary Key --->");
Debug.Log(string.Format("m_ID = {0},m_AddHealth = {1},m_AddAttack = {2},m_AddDefense = {3}," ,LevelKey.m_ID ,LevelKey.m_AddHealth ,LevelKey.m_AddAttack ,LevelKey.m_AddDefense));

Debug.Log(" < ---LevelTable List --->");

foreach (var item in LevelList)
    Debug.Log(string.Format("m_ID = {0},m_AddHealth = {1},m_AddAttack = {2},m_AddDefense = {3}," ,item.m_ID ,item.m_AddHealth ,item.m_AddAttack ,item.m_AddDefense));

var PlayerAll = PlayerTable.GetAll();
var PlayerIndex = PlayerTable.GetByIndex(0);
var PlayerKey = PlayerTable.GetByKey(100001);
var PlayerList = PlayerTable.GetAllList();

Debug.Log(" < ---PlayerTable Dictionary --->");

foreach (var item in PlayerAll)
    Debug.Log(string.Format("Key = {0}, m_ID = {1},m_Name = {2},m_BaseHealth = {3},m_BaseAttack = {4},m_BaseDefense = {5}," ,item.Key ,item.Value.m_ID ,item.Value.m_Name ,item.Value.m_BaseHealth ,item.Value.m_BaseAttack ,item.Value.m_BaseDefense));

Debug.Log(" < ---PlayerTable Dictionary Index --->");
Debug.Log(string.Format("m_ID = {0},m_Name = {1},m_BaseHealth = {2},m_BaseAttack = {3},m_BaseDefense = {4}," ,PlayerIndex.m_ID ,PlayerIndex.m_Name ,PlayerIndex.m_BaseHealth ,PlayerIndex.m_BaseAttack ,PlayerIndex.m_BaseDefense));

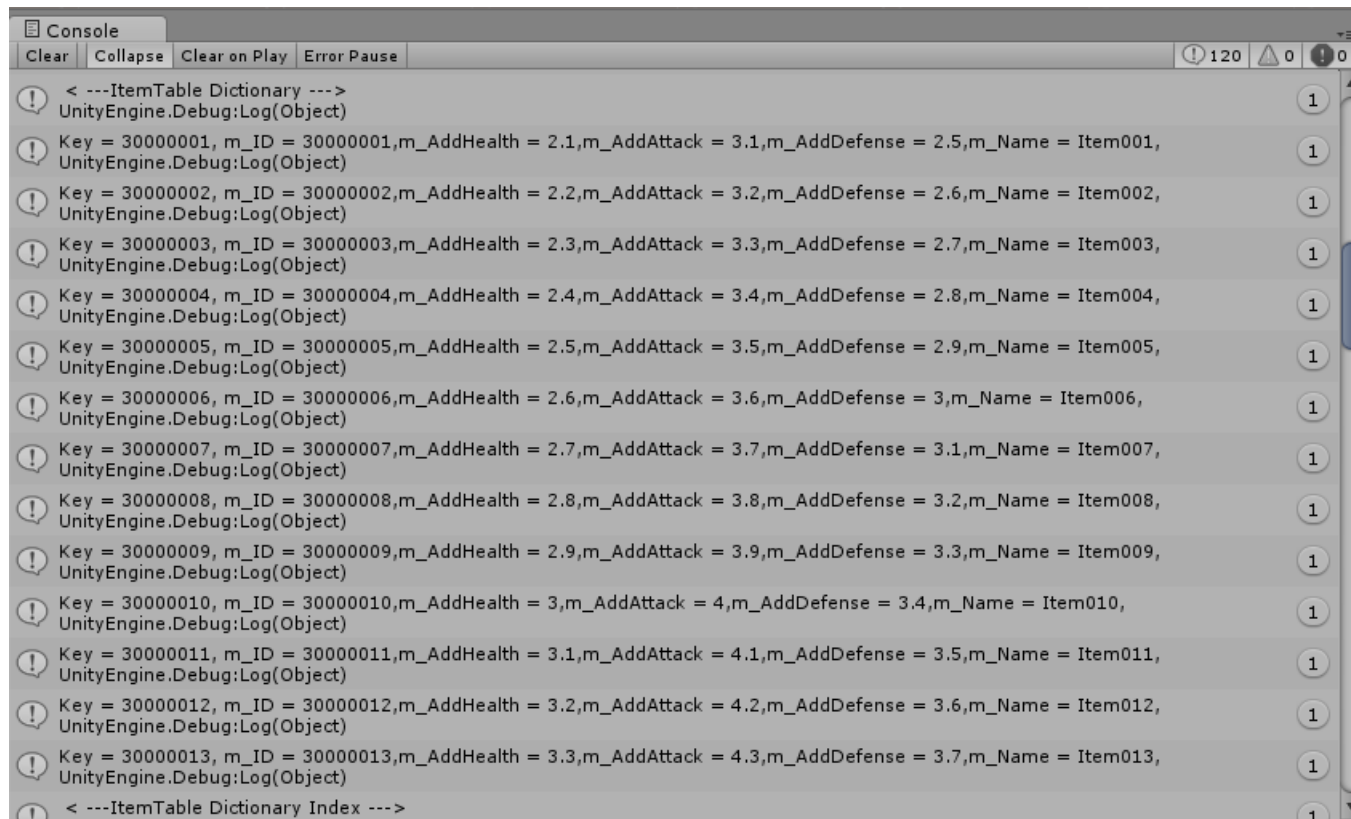
Debug.Log(" < ---PlayerTable Dictionary Key --->");
Debug.Log(string.Format("m_ID = {0},m_Name = {1},m_BaseHealth = {2},m_BaseAttack = {3},m_BaseDefense = {4}," ,PlayerKey.m_ID ,PlayerKey.m_Name ,PlayerKey.m_BaseHealth ,PlayerKey.m_BaseAttack ,PlayerKey.m_BaseDefense));

Debug.Log(" < ---PlayerTable List --->");

foreach (var item in PlayerList)
    Debug.Log(string.Format("m_ID = {0},m_Name = {1},m_BaseHealth = {2},m_BaseAttack = {3},m_BaseDefense = {4}," ,item.m_ID ,item.m_Name ,item.m_BaseHealth ,item.m_BaseAttack ,item.m_BaseDefense));

```

So easy. You can check all result in Console as follows.



The screenshot shows a Unity console window with a tab labeled 'Console'. The window has buttons for 'Clear', 'Collapse', 'Clear on Play', and 'Error Pause'. On the right side, there are status indicators: a warning icon with '120', an error icon with '0', and a debug icon with '0'. The console log contains the following entries:

```
< ---ItemTable Dictionary --->
UnityEngine.Debug:Log(Object)
Key = 30000001, m_ID = 30000001, m_AddHealth = 2.1, m_AddAttack = 3.1, m_AddDefense = 2.5, m_Name = Item001,
UnityEngine.Debug:Log(Object)
Key = 30000002, m_ID = 30000002, m_AddHealth = 2.2, m_AddAttack = 3.2, m_AddDefense = 2.6, m_Name = Item002,
UnityEngine.Debug:Log(Object)
Key = 30000003, m_ID = 30000003, m_AddHealth = 2.3, m_AddAttack = 3.3, m_AddDefense = 2.7, m_Name = Item003,
UnityEngine.Debug:Log(Object)
Key = 30000004, m_ID = 30000004, m_AddHealth = 2.4, m_AddAttack = 3.4, m_AddDefense = 2.8, m_Name = Item004,
UnityEngine.Debug:Log(Object)
Key = 30000005, m_ID = 30000005, m_AddHealth = 2.5, m_AddAttack = 3.5, m_AddDefense = 2.9, m_Name = Item005,
UnityEngine.Debug:Log(Object)
Key = 30000006, m_ID = 30000006, m_AddHealth = 2.6, m_AddAttack = 3.6, m_AddDefense = 3, m_Name = Item006,
UnityEngine.Debug:Log(Object)
Key = 30000007, m_ID = 30000007, m_AddHealth = 2.7, m_AddAttack = 3.7, m_AddDefense = 3.1, m_Name = Item007,
UnityEngine.Debug:Log(Object)
Key = 30000008, m_ID = 30000008, m_AddHealth = 2.8, m_AddAttack = 3.8, m_AddDefense = 3.2, m_Name = Item008,
UnityEngine.Debug:Log(Object)
Key = 30000009, m_ID = 30000009, m_AddHealth = 2.9, m_AddAttack = 3.9, m_AddDefense = 3.3, m_Name = Item009,
UnityEngine.Debug:Log(Object)
Key = 30000010, m_ID = 30000010, m_AddHealth = 3, m_AddAttack = 4, m_AddDefense = 3.4, m_Name = Item010,
UnityEngine.Debug:Log(Object)
Key = 30000011, m_ID = 30000011, m_AddHealth = 3.1, m_AddAttack = 4.1, m_AddDefense = 3.5, m_Name = Item011,
UnityEngine.Debug:Log(Object)
Key = 30000012, m_ID = 30000012, m_AddHealth = 3.2, m_AddAttack = 4.2, m_AddDefense = 3.6, m_Name = Item012,
UnityEngine.Debug:Log(Object)
Key = 30000013, m_ID = 30000013, m_AddHealth = 3.3, m_AddAttack = 4.3, m_AddDefense = 3.7, m_Name = Item013,
UnityEngine.Debug:Log(Object)
< ---ItemTable Dictionary Index --->
```

My Email is insaneoops288@gmail.com. If you have some question, or suggestion. Please mail me.

Thanks a lot.