OHL Technical Document

# Character System

Overview - Character Manager has all the information even enemies. On changing the scene, we can get the attributes that saved in the Character Manager. So, we must save the data into the Character Manager whenever we change the value in the character.

# Classes

**User.cs**

Purpose : User class have one inventory and multiple characters

|  |  |  |
| --- | --- | --- |
| Type | Value Name | Description |
| Float | Gold | Gold |
| **CharacterManager** | characterManager | You can access character manager using GetCharacter(ID) |
| **Inventory** | inventory | You can access our item list in the inventory |

**Entity.cs**

Purpose : Entity class have basic elements what they need both character and enemy.

|  |  |  |
| --- | --- | --- |
| Type | Value Name | Description |
| **Int** | ID | Each character and enemy have a unique ID |
| **Float** | HP | HP |
| **Attributes** | attribute | There are 5 main attributes (con, wis, str, dex, int) in “Attributes” class |
| **GameObject** | target | Each character and enemy need to know who they are attacking |

**CharacterManager.cs (Singleton)**

Purpose : Saving data during the scene changing. On changing the scene, we can get the attributes that saved in the Character Manager. To use this, we need to specific ID and we should call CharacterManager.instance.GetCharacter(ID) and then we can handle the all values in that character.

Notice : To use this CharacterManager, you have to add character.prefab that we already made into the player GameObject in the CharacterManager. And we will get this prefab value when we start the scene.

|  |  |  |
| --- | --- | --- |
| Part | Fucntion Name | Description |
| **Get (important)** | Character GetCharacter(int \_characterID) | You can get the all character information using Getcharacter(ID) |
| **Get** | Enemy GetEnemy(int \_enemyID) | You can get the all enemy information using GetEnemy(ID) |
| **Set** | void SetAttributePoint(int id, int val) | When we level up, God give us the attribute point.  we can add our attributes using this attribute point. |
| void SetLevel(int id, int val) | level |
| Void SetExp(int id, int val) | Experience |
| Void SetExpRequired(int id, int val) | Experience for level up |
|  | Void SetMaxHealth(int id, float val) | Maximum Health |
| Void SetArmor(int id, float val) | Armor |
| Void SetMana(int id, float val) | Mana |
| Void SetDamage(int id, float val) | Damage |
| Void SetSpeed(int id, float val) | Attack Speed |
| **Get** | Int GetAttributePoint(int id) | Attribute point |
| Int GetLevel(int id) | Level |
| Int GetExp(int id) | Experience |
| Int GetExpRequried(int id) | Experience for level up |
|  | Float GetMaxHealth(int id) | Maximum Health |
| Float GetArmor(int id) | Armor |
| Float GetMana(int id) | Mana |
| Float GetDamage(int id) | Damage |
| Float GetSpeed(int id) | Attack Speed |
| **Add** | void AddMaxHealth(int id, float val) | Maximum Health |
|  | void AddArmor(int id, float val) | Armor |
|  | void AddMana(int id, float val) | Mana |
|  | void AddDamage(int id, float val) | Damage |
|  | void AddSpeed(int id, float val) | Attack Speed |

**Character.cs**

Purpose : Character have level and xp .

|  |  |  |
| --- | --- | --- |
| Type | Value Name | Description |
| **AttributeDistribution** | AD | They have attribute point for calculating in the AttributeDistribution class. |
| **String** | Name | Name |
| **Int** | attributePoint | Value for upgrade. |
| **Int** | level | Each character and enemy need to know who they are attacking |
| **Int** | Xp | Character exp |
| **Int** | XP\_required | Character exp for level up |
| **Float** | FlatPoint | Base FlatPoint |
| **Item** | Items | Equipped Item list |

|  |  |  |
| --- | --- | --- |
| Part | Function Name | Description |
| **Add** | Void AddMaxHealth() | Add one max health attribute point. . |
| Void addArmor() | Add one armor attribute point. |
| Void Addmana() | Add one mana attribute point. |
| Void AddDamage() | Add one damage attribute point. |
| Void AddSpeed() | Add one attack speed attribute point. |
|  | Void AddAttributePoint() | Add five attribute point for test. |
|  | Void AddEXP() | Add one hundred point for test. |
| **Button** | Void OnAddButtonCON() | Add attribute button. |
| Void OnAddButtonWIS() |
| Void OnAddButtonSTR() |
| Void OnAddButtonDEX() |
| Void OnAddButtonINT() |
| **Reset** | Void RESET\_ATTRIBUTE() | Reset all attribute. |
| **Equip** | Void Equip(Equipment newItem) | Equip the item |
| **Unequip** | Void UnEquip(int slotIndex) | UnEquip the item |

**AttributeDistribution.cs**

Purpose : For calculating current attribute with accumulated point.

|  |  |  |
| --- | --- | --- |
| Part | Function Name | Description |
| **Add**  **(Button)** | float AD\_MaxHealth(float init) | init + (init\*MaxHealth\* point.MaxHealth) |
| float AD\_Armor(float init) | init + (init\*Amror\* point.Amror) |
| float AD\_Mana(float init) | init + (init\*Mana\* point.Mana) |
| float AD\_Damage(float init) | init + (init\*Damage\* point.Damage) |
| float AD\_Speed(float init) | init + (init\*Speed\* point.Speed) |

**Attributes.cs**

Purpose : attributes in our characters.

|  |  |  |
| --- | --- | --- |
| Type | Value Name | Description |
| **Float** | MaxHealth | Our attributes for combat. |
| **Float** | Armor |
| **Float** | Mana |
| **Float** | Damage |
| **Float** | Speed |

# Item System

Overview –

# Classes

**Item.cs (Scriptable Object)**

Purpose : Generate Item

Notice : You can use “create -> Inventory/Item or Equipment” in the unity project.

|  |  |  |
| --- | --- | --- |
| Type | Value Name | Description |
| **String** | name | Default is “New Item” |
| **Sprite** | icon | For item icon |
| **Bool** | isDefaultItem | Initialized false |

|  |  |  |
| --- | --- | --- |
| Part | Function Name | Description |
| **Equip** | void Equip(int id) | Equip item to Character(id) |
| **Unequip** | Void Unequip(int id) | Unequip from character |
| **Remove** | Void RemoveFromInventory() | Remove item from inventory |

|  |  |  |
| --- | --- | --- |
| Type | Value Name | Description |
| **EquipmentSlot (enum)** | equipSlot | For the type {Armor, Weapon, Accessories} |
| **Rarity**  **(enum)** | rarity | {Normal, Arcane, Rare, Legendary, Divine} |
| **Float** | MaxHealthModifier | For modify the attributes. |
| **Float** | ArmorModifier |
| **Float** | ManaModifier |
| **Float** | DamageModifier |
| **Float** | SpeedModifier |

**RewardManager.cs (singleton)**

Purpose : for rewarding and using in the UI.

|  |  |  |
| --- | --- | --- |
| Type | Value Name | Description |
| **Image** | icon | For the item icon in the reward scene. |
| **List<Equipment>** | items | For the random generate item. |
| **User** | user | To get the gold |
| **Int** | Xp | We receive 100 point. |
| **Int** | Gold | For reward gold – random |

|  |  |  |
| --- | --- | --- |
| Part | Function Name | Description |
| **Item** | Item GetRandomItem() | Get the random item |
| **XP** | Int GetXP() | Return xp //100 |
| **Gold** | Int GetRandomGold | It will be return random reward gold |
| **Generate** | Void RandomSpawn() | Generate random item |
| **Save** | Void Save() | Add item into inventory and save exp and gold |

# Inventory System

Overview –

# Classes

**Inventory.cs (singleton)**

Purpose : Inventory .

|  |  |  |
| --- | --- | --- |
| Type | Value Name | Description |
| **Delegate** | delegate void OnItemChanged() | Using delegate, we can check the item added or remove. |
| **OnItemChanged** | onItemChangedCallback |
| **List<Item>** | items | Save item into inventory |

|  |  |  |
| --- | --- | --- |
| Part | Function Name | Description |
| **Add** | Bool add(Item item) | Add item into item list. |
| **Remove** | Void Remove(Item item) | Remove item from the list. |
| **List<Item>** | GetItemList() | To get all Item in the inventory |

**InventorySlot.cs**

Purpose : For each slot in Inventory.

|  |  |  |
| --- | --- | --- |
| Type | Value Name | Description |
| **Image** | icon | Icon in the inventory |
| **Button** | removeButton | For Remove button |

|  |  |  |
| --- | --- | --- |
| Part | Function Name | Description |
| **Add** | Void AddItem(Item newItem) | Add new item into private item for using. And change the icon sprite and removebutton will be enabled. |
| **Clear** | Void ClearSlot | Make item and icon is null. |
| **RemoveButton** | Void OnRemoveButton() | Remove item from the Inventory. |
| **Use** | Void UseItem() | Use the item. |

**InventoryUI.cs**

Purpose : For temporary UI .

Notice : you can use keyboard “M” to open/close the Inventory UI.

|  |  |  |
| --- | --- | --- |
| Type | Value Name | Description |
| **Transform** | itemsParent | Get the inventory slot root in the canvas. |
| **GameObject** | inventoryUI | For Open/Close inventory using keyboard (M) |
| **Transform** | equipParent1 | Get the equip slot in the character 1 |
| **Transform** | equipParent2 | Get the equip slot in the Character 2 |
| **GameObject** | equipUI | For Open/Close equipment UI using Keyboard (Tab) |

|  |  |  |
| --- | --- | --- |
| Part | Function Name | Description |
| **Update** | Void UpdateUI | Add or Clear item in the inventory. |
|  | Void UpdateEquipUI | Updating Character 1 ‘s and 2‘s equip slot |

# Drawing

**Character System**

스크린샷이(가) 표시된 사진

자동 생성된 설명

**Item System**

스크린샷이(가) 표시된 사진

자동 생성된 설명