

# Dungeon Breaker

## ***Starter Kit 2.2***

The Action RPG Starter Project. Available on mobile devices  
By Rachan Neamprasert

*Important: There's many changes in this version. Please backup your project before update.*

### **New Update! 2.2**

- Add Mecanim Suport
- Add TopDown styled game
- Optimize Character Motor
- Cleaned masses stuff Make it easier to use
- Bug Fixes

## **Project Included**

- Multi Game View (TopDown and Third Person)
- Mecanim Support
- Character Action and Combat System
- Inventories and Items embbed
- Skills System
- NPC Quest System
- NPC Shop System
- Teleporter
- Save/Load Game
- Particle FX
- AI Friend / Enemy with Skills
- Game Manager System
- Mini map
- Mobile Devices Support

## **Controller**

- W A S D : Move
- MouseR : Spell
- MouseL : Attack
- Tab : Open Inventory

## **Information**

Web Demo: <http://www.gamekak.com/unity/DB/>

VDO: <http://youtu.be/AfEcrnhdD08>

Online Manual: <http://www.hardworkerstudio.com/db>

Support: [hwrstudio@gmail.com](mailto:hwrstudio@gmail.com)

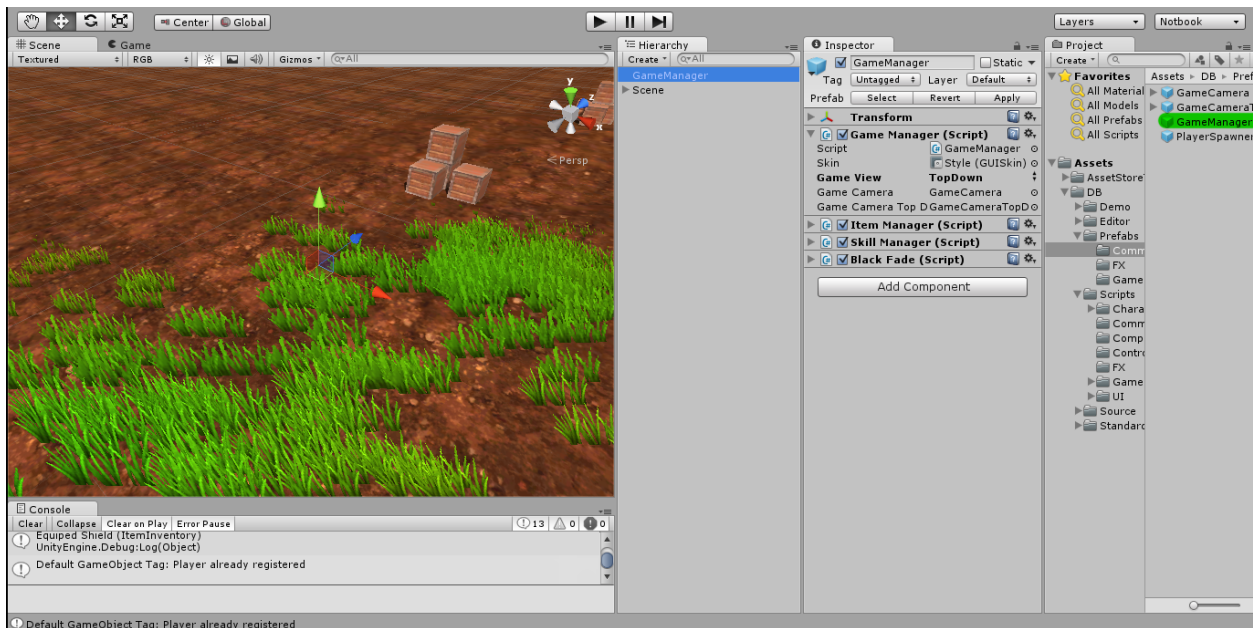
# Basic

Basic tutorial will show you how to preparing game environment and characters with few steps.

## Game Manager Setup

In Dungeon Breaker. **Game Manager** is the importance object. it's will collected all of game data and player management. You have to add **Game Manager** prefab into a starter scene.

1. Add GameManager prefab into the scene  
(DB/Prefabs/Common/GameManager)

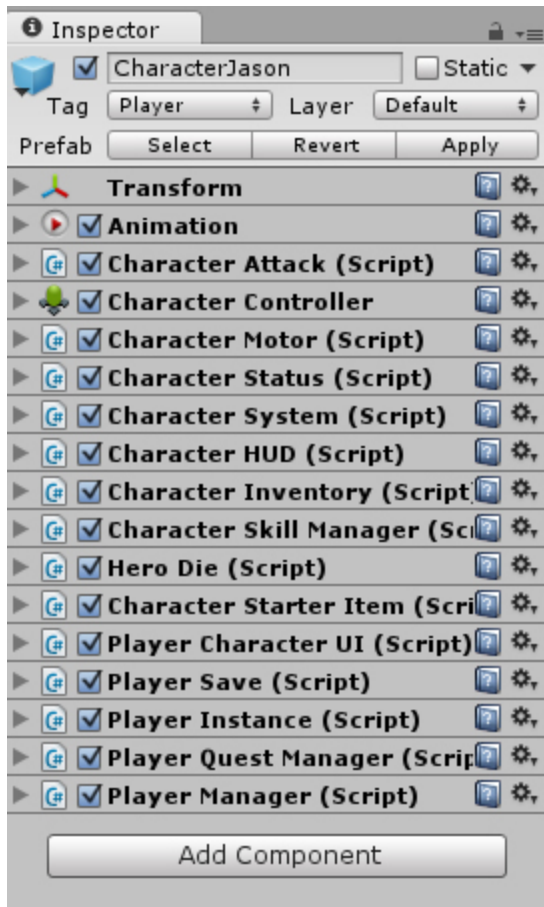


### Note.

See **Game View in Game Manager**

You can switch to **TopDown Styled** or **Third Person Styled**

# Create Character



*all components will added automatically.*

## Note.

*Tag is importance! all characters must Tagged. the tags are using for AI system.*

e.g

is **Player** Tag set to “**Player**”

is **Enemy** Tag set to “**Enemy**”

## Add Component

(DB/Scripts/Common/)

1. Add [PlayerInstance.cs](#) or [PlayerManager.cs](#) into character for **Player**

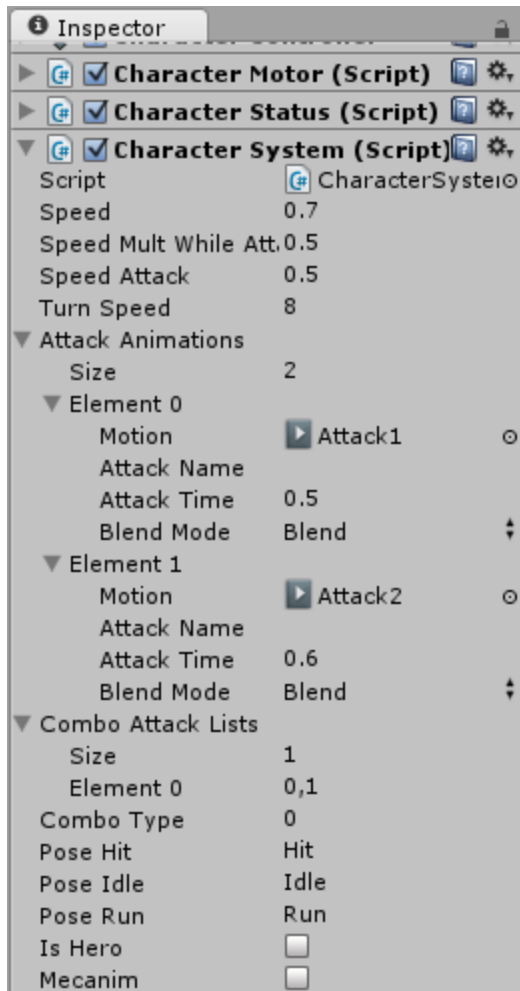
or Add [AllInstance.cs](#) to character for **AI**

2. Setup **Collision size** in **Character Controller**

3. See **Character System**

## Parameters

- **Speed** : Character move speed
- **Speed Attack** : Attack speed
- **Turn Speed** : Character turn speed
- **Attack Animations** : list of attack animation
- **Combo Attack Lists** : list of combo animations
- **Combo Type** : Combo type
- **Pose Hit** : Animation when got hit
- **Pose Idle** : Idle animation
- **Pose Run** : Run animation
- **Is Hero** : will not freeze when got hit



## Setup Combat System

### Setup Attack Animations

**Attack Animations[N]** create instance all of attack post

1. Setup attack pose

**Attack Name** with **animtion name**

or

**Motion** with **motion file**

2. Setup **Attack Time** to sync with attack animation

e.g. **Attack Time** = 0.5 push damage after 0.5 sec of animation.

3. Add attack animation element index to **Combo Attack Lists**

e.g. **Combo Attack Lists[0]** = 0,1

Play **Attack Animations[0]** and **Attack Animations[1]** in 1 combo.

4. **Combo Type** is the index of **Combo Attack Lists** element

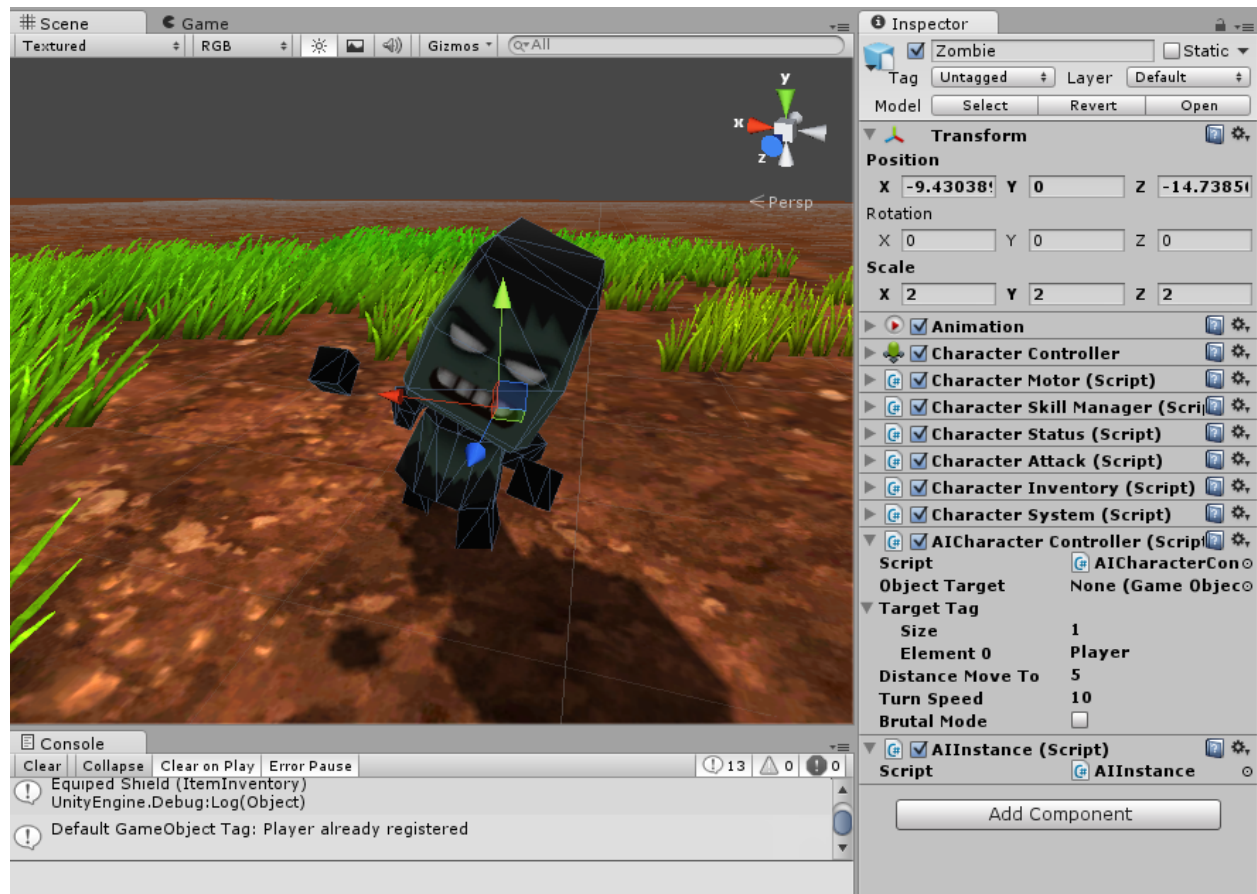
e.g. **Combo Type** = 0 Play **Combo Attack Lists[0]** on attacking.

# Create Enemy

Same as Create Character.

(DB/Scripts/Common/)

1. Add **AllInstance.cs** into your Character model. *all components will added automatically*



2. Setup **collision size** in **Character Controller** and **Combat system** in **Character System** (See Setup Combat System) Set **Tag** to “**Enemy**”

3. See **AICharacter Controller**

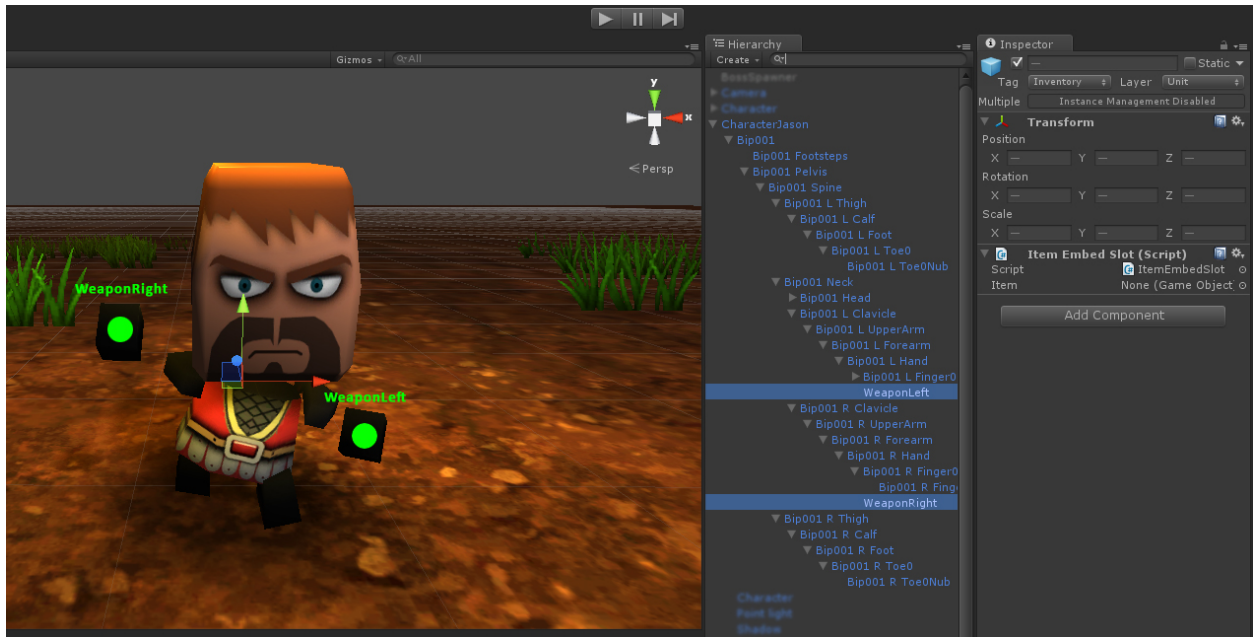
Set **Target Tag** as his opposits characters.

e.g. is he a Zombie? **Taget Tag** must set to “**Player**”.

**Basic Tutorial Completed. Let's Play Test!**

# Advanced

Advanced tutorial will show you how to create equipment weapons and embed into your character skeleton. also how to setup Items Database and Skills Database.



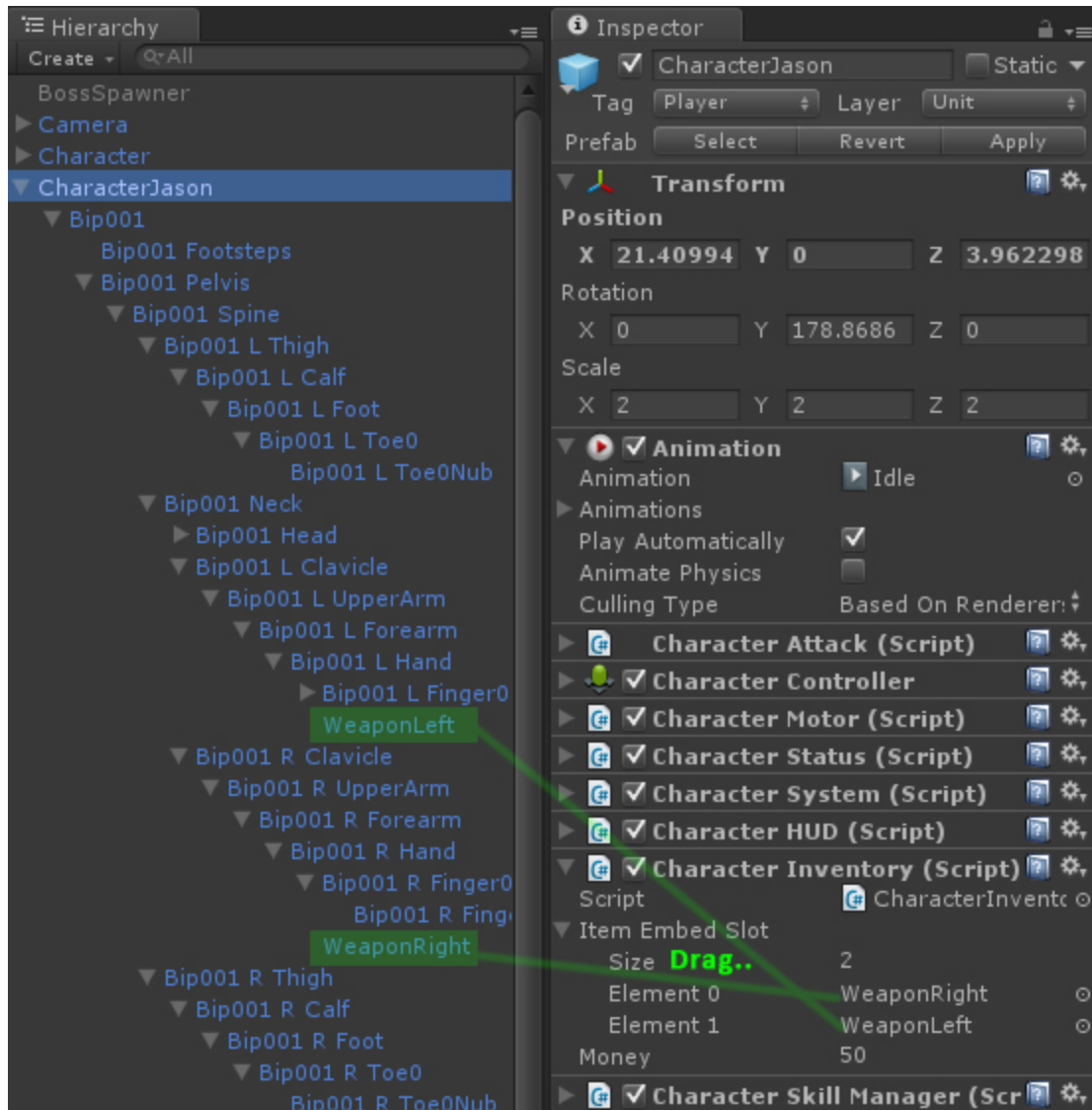
## Setup Character Inventory

(DB/Scripts/GamePlay/Item/Base/)

*e.g. Create Left and Right Weapon*

1. Create **GameObject** named RightWeapon and LeftWeapon.
2. Add [ItemEmbedSlot.cs](#) to the RightWeapon and LeftWeapon.
3. Add Left bone and Right bone to hierarchy





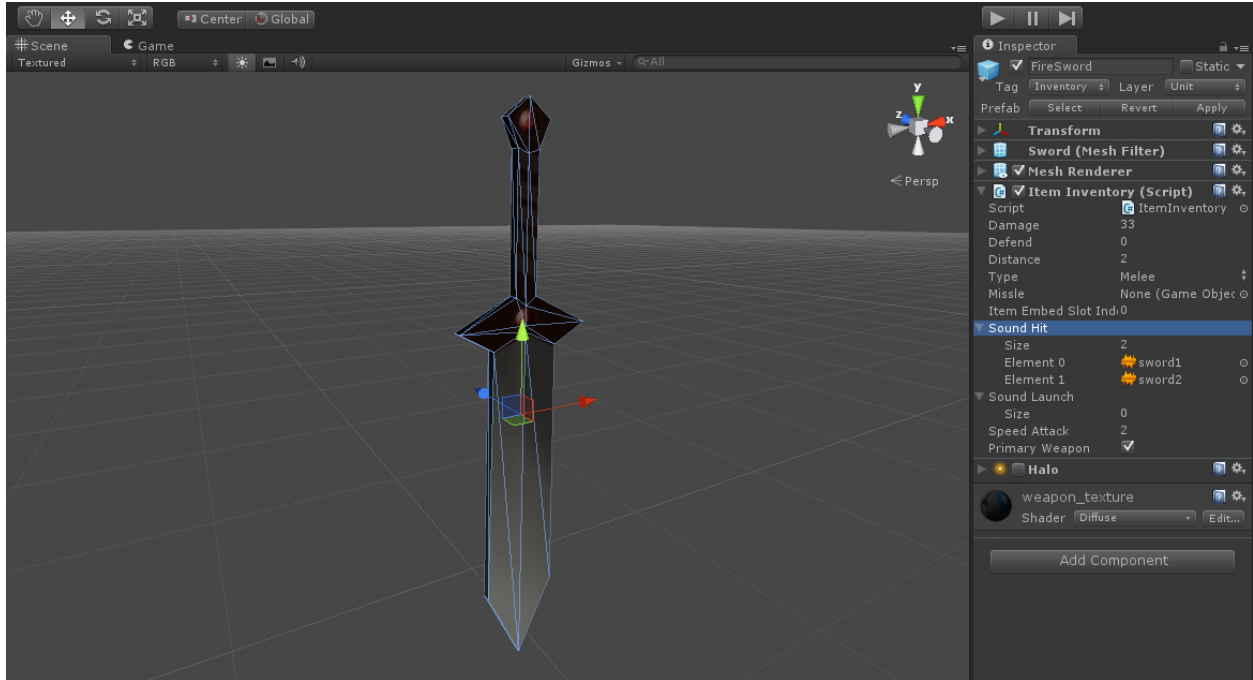
## Setup Character Inventory

1. See **Character Inventory**
2. Drag **WeaponLeft** and **WeaponRight** to **Item Embed Slot**
3. Add start money

if you have many inventory items than this sample you can do the same way and add all of them into **Item Embed Slot**

**Item Embed Slot[N]** N element index using to reference with weapon item.





# Create Weapons

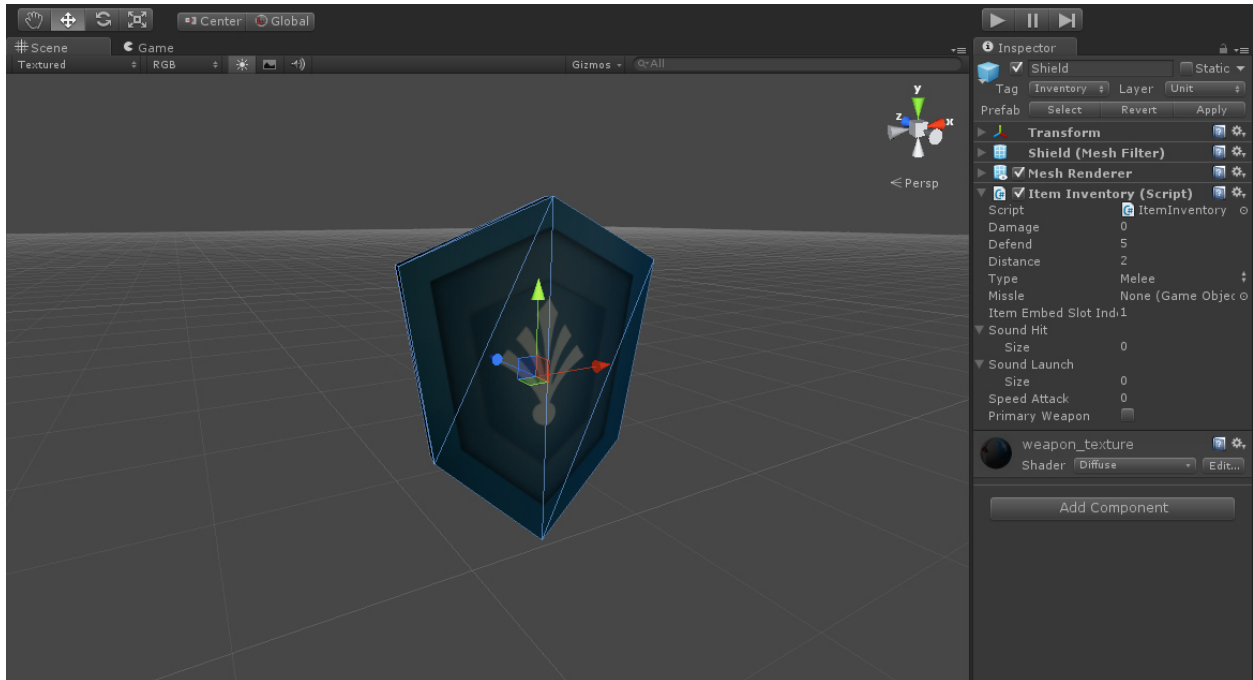
(DB/Scripts/GamePlay/Item/Base/)

## e.g. Create Sword

1. Using Sword model
2. Add [ItemInventory.cs](#) to Sword model
3. Setting parameters
4. Save to Prefab named Sword

## Parameters

- **Damage** Damage point of this weapon
- **Defend** Defend point of this weapon
- **Distance** Distance of this weapon only use in Ranged type
- **Type** type of weapon (Melee , Ranged)
- **Missile** Missile object using to launch only use in Ranged type
- **Item Embed Slot Index** where is this item attach to. see index of **Item Embed Slot[index]** in **Character Inventory** in Setup Character Inventory
- **Sound Hit** Sound when hit the target
- **Sound Launch** fire sound
- **Speed Attack** attack speed base of this weapon
- **Primary Weapon** Active when this is a weapon



## e.g. Create Shield

1. Using Shield model
2. Add [ItemInventory.cs](#) to Shield model
3. Setting parameters
4. Save to Prefab named Shield

## Parameters

- **Damage** 0
- **Defend** 5
- **Distance** 0
- **Type** Melee
- **Missile** None
- **Item Embed Slot Index** 1 \*\* index 1 is **WeaponLeft** in **Character Inventory**
- **Sound Hit** None
- **Sound Launch** None
- **Speed Attack** 0
- **Primary Weapon** false

# Setting Items Database

Importance\* **Item Manager** must setting up before playing.



1. go to **Setting** Scene
2. See [Item Manager](#) in **GameManager** prefab
3. Create Items Element
4. Add all parameter

## Parameters

- Icon** Item Icon Texture2D
- Name** Item Name
- Description** Item Description
- Price** Item Price
- Item Type** Item Type
- Item Prefab** Sword and Shield from *Setup Item for Character Inventory step* adding here
- Item Prefab Drop** Item prefab using to drop

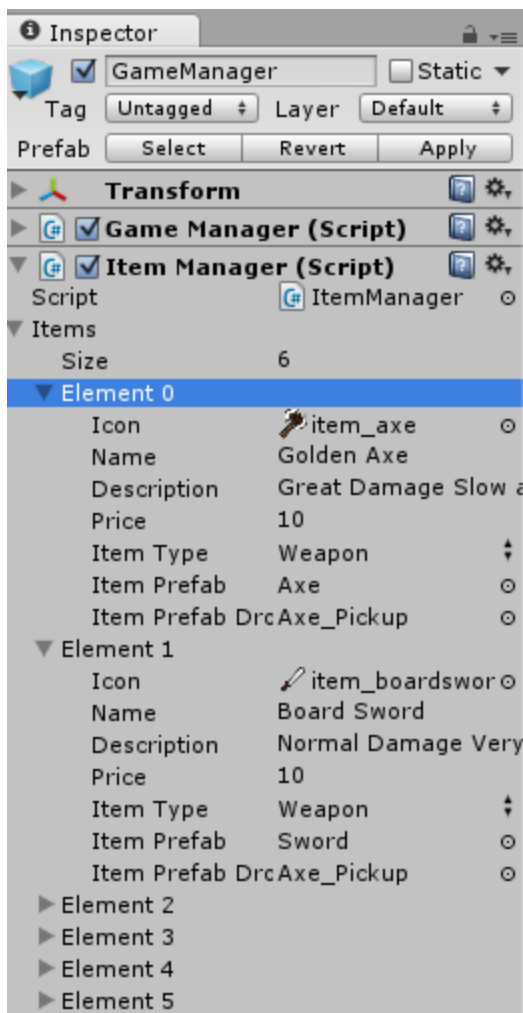
# Add Starter Items

(DB/Scripts/Character/CharacterBase/Base/)

1. Add [CharacterStarterItem.cs](#) into your Character
2. Setting **Parameters**

## Starter Item[ ]

The **Starter item** is reference as **Item Embed Slot** in Character Inventory  
e.g. **Starter item[0]** reference to **Item Embed Slot[0]**



## How to Set Items to Character

You can set **Starter item[N]** to any index of **Items[N]** in **ItemManager**

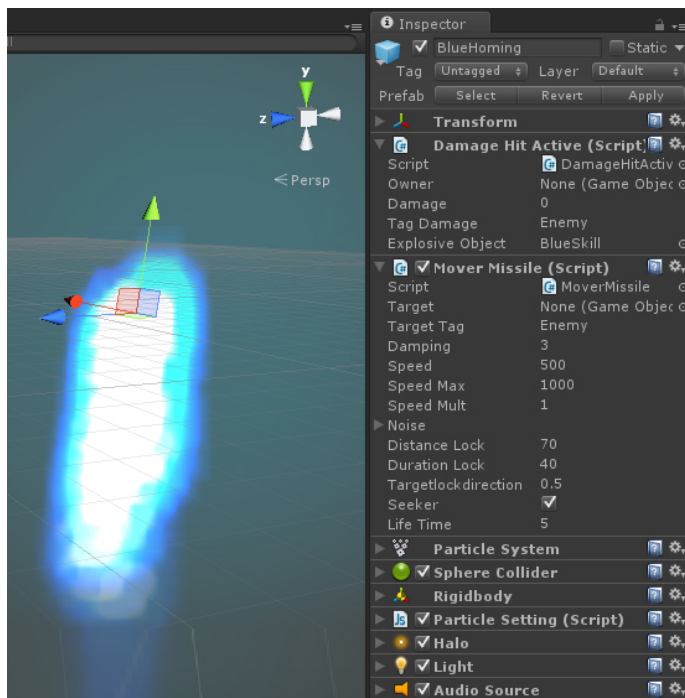
e.g.

if **Starter Item[0] = 1** then **Item Embed Slot[0]** will equip **Board Sword**

# Create Skills

## e.g. Blue Spirit Skill (Homing Missile)

1. Create particle system
2. Add [DamageHitActive.cs](#)
3. Setting Parameters



### Parameters

- Target
- Target Tag tag of game object to be a target
- Damping
- Speed normal speed
- SpeedMax max speed
- Speed Mult speed multiply
- Noise
- Distance Lock distance to lock
- Duration Lock duration before get locked
- Targetlock direction 1 is only front 0 is all around direction
- Seeker active is finding a target automatically
- Life Time

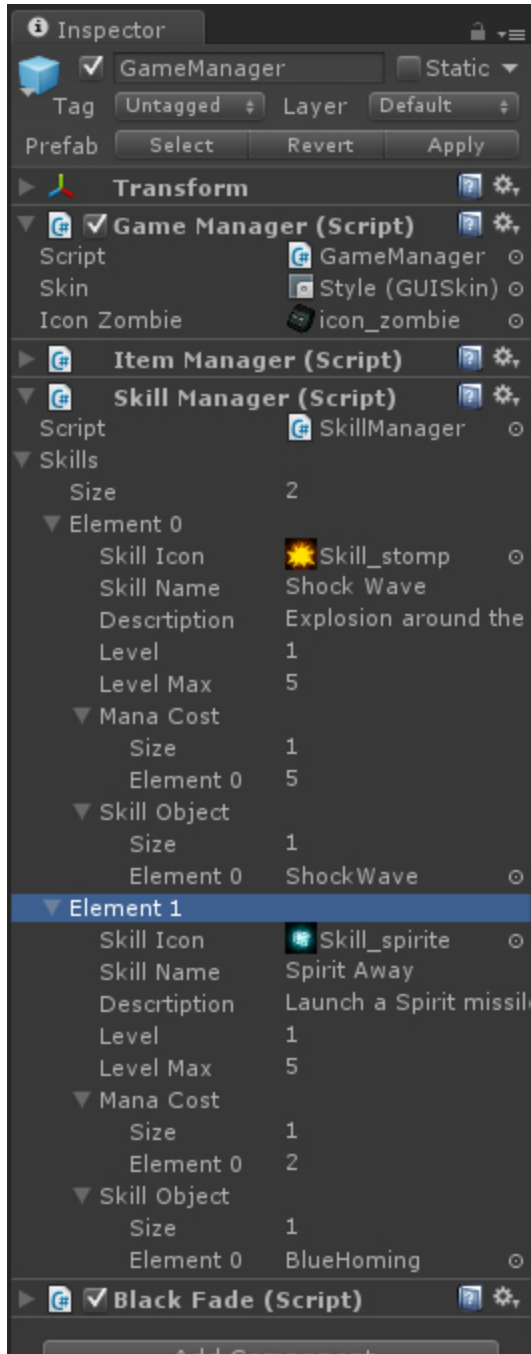
4. Add [DamageHitActive.cs](#)
5. Setting Parameters

### Parameters

- Owner launcher e.g. **player** (skill launcher will set this automatically)
- Damage Damage point
- Tag Damage Tag of game object to damage to
- Explosive Object Explosion object e.g. Explosion particle effect.

# Setting Skills Database

Importance\* **Skill Manager** must setting in **GameManager** before playing.

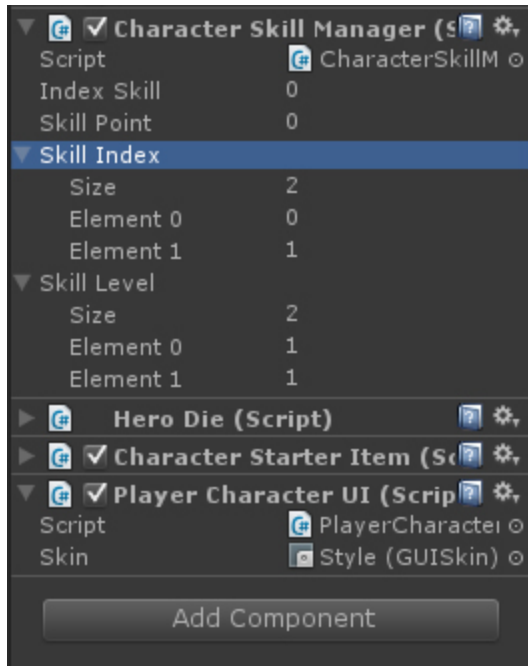


1. go to **Setting** Scene
2. See [Skill Manager](#) in **GameManager** prefab
3. Create Items Element
4. Add all parameter

## Parameters

- Skill Icon** Skill Icon Texture2D
- Skill Name** Skill Name
- Description** Skill Description
- Level** Start Level
- Level Max** Max Level
- Mana Cost** is Mana cost [Level-1]
- Skill Object** is Skill Object[Level-1] using to launch

# Add Skills to Character



1. See [Character Skill](#) in **Character**
2. Setting parameters

## Parameters

- Index Skill** Current skill index
- Skill Point** Skill level
- Skill Index** Skill index lists using index of element in Skill Database
- Skill Level** Level skills

## Thank you

if you have any question or need assistance please mail me  
[hwrstudio@gmail.com](mailto:hwrstudio@gmail.com)

Find more information here  
<http://www.hardworkerstudio.com>

Find vdo tutorial here  
<http://www.youtube.com/channel/UCOq7Q7ND97c0bJhkfzYIZLg>