

HACK AND SLASH RPG KIT_{v2.0}

Introduction

Thank you for purchasing, Hack and Slash RPG kit asset. We built this package to create a structure of Hack and Slash RPG that we can change character's equipments with different parts. The one of main concern of this package is that use one model of character with multiple equipment to create not only the playable character but also the enemies as well. Therefore, we create some functionality of creating random characters with random choice of the equipment so that we can create various characters.

Contents

- One 3D model with multiple equipments (16 Weapons, 10 shields and 4 sets of armor sets with 4 facial hair set)
- Animations for different weapons (two hands, one hands and shield, dual weild and unarmed).
- Animator to transit between animations on different character state (move, run, attack, idle and damage reaction.
- C# Scripts for character control, equipment system, simple enemy AI, GUI control and animation related controls
- Two sample Scenes to present how this system works.

Added In V2

- Inventory & equipment system and character status system that will work with equipment system.
- Exp and Level up system.
- Using proper Unity3D UI framework.
- GUI for status, inventory, equipment, selected item detail, dropped item and damage info, also all the pervious buttons as well.



Instruction of C# scripts in this asset

Equipment System

changed from CharacterEquipments.class (Deprecated)

CharacterEquipmentsV2.class

-This is a class of character current equipment information. This is used for capturing all equipments that character has equipped now.

changed from Equipment.class (Deprecated)

EquipmentV2.class

-This is a class for each equipment, add this into the equipment model, this will use model name which is gameobject name to set up its type, part, side. Please make sure the name of the equipment has to be unique as it will get used as id. The equipment type will get populated by name, this action will be done in editor mode.

The Equipment naming rule is

"**Part_Side_Type_NameOfEquipment**(**_t**, **_d**, **_2**)"

(**_t** if you want to have a body parts to show with this equipment.)

(**_d** if This is dual weild weapon.)

(**_2** if This is for two hands weapon.)

ex 1) **Feet_Left_Armor_Chainmail_t**

ex 2) **Hand_Left_Weapon_SilverDagger_d**

ex 3) **Chest_Body** (If is body type then it doesn't require to have any name, also if the equipment doesn't have two sides(Left or Right) such as body, head then don't need to put Side value in the name.)

In V2.0

-Now EquipmentV2 will keep the icon of item(also icon size and rotation as well for the other inventory system view) , rarity.

-It is now extended class of the Item.class that will contain more information about the equipment.

Item.class (Newly added in V2.0)

This will keeps the details of Equipment, ex) attack power, defense power and attack/movement speed.

EquipmentControl.class (Newly added in V2.0)

This class will get mainly used for the BaseCharacterContoller for changing equipment place holder meshes.

changed from EquipmentManager.class (Deprecated)

EquipmentManagerV2.class

-This is a core class for equipment system. This script has to be attached in root gameobject with where contains equipments in child object. This script will load all the equipments from the children of the attached game object. This script provides change equipments (random choice, apperance choice, set armor equipment choice and weapon equipment choice). Also with this script we can spawn a cloned character with given equipment type, which is used in the sample scene as spawning a enemy with randomly chosen equipments. And the other main parts is equipment changes for the player.

in V2.

-Changing equipment visual parts are all moved into EquipmentControl.class, which now EquipmentManagerV2 extends from the EquipmentControl. Spawned character objects will only have EquipmentControl and placeholder objects.

-Change equipment mechanic has been changed, other then each equipment's game object setActive to false, now this script will populate place holder for the every each part and will simply change its SkinnedMeshRenederer's mesh. - which is a lot faster and quicker solution.

Eye.class

-This is a simple script to change eye color of the sample model class.

HACK and SLASH RPG Action control

changed from RPGCharacterObject.class (Deprecated)

BaseCharacterController

-This is a core class for Hack and Slash RPG style character action control. This class is an abstract class so that it can be extended into many type of character controls. In the example scene, this class is extended as **PlayerController** and **Enemycontroller** to perform different type of character action. This class provides moving, attacking, damaging and dying actions. dying action will trigger the rag doll so that will do more realistic physics driven reaction of dying animation.

changed from PlayerControl.class (Deprecated)

PlayerControllerV2.class

-This is a core class for Hack and Slash RPG style player action control. This class is an extended class of **BaseCharacterControl**, in the sample scene this class will get added into the player character gameobject from GameManager script which is in the BattleScene. This script is designed for touch and click to move and attack, it will move the position when player click/touch to the point, if player select the enemy then will move (double touch or double click will make character to run) to the enemy and attack it when it gets close enough, with this asset is more focused on the character selection part so that player damage/dying/level/inventory system is not implemented.

changed from CameraController.class(Deprecated)

CameraControllerV2.class

-This is a class for Hack and Slash RPG style top down player character centered camera. This class will follow the game object where the class of **PlayerControllerV2** is attached.

changed from Enemy.class(Deprecated)

EnemyControllerV2.class

-This is a class for the enemy action control. This class will get added from **GameMasnager** script which is attached to the object "GameMaster" in given sample Battle Scene. Enemy script will get used by **WanderController** script to deal with the action control (move, attack, damage, die and drop one of the equipped item into the scene).

changed from Wanderor.class(Deprecated)

WanderController.class

-This is a class for simple AI to wander around the map. This script will get added by **GameManager** script after the enemy is spawned to the scene. It has simple wandering AI that will move toward until it finds the object then will rotate left or right until it finds nothing in front. This class contains stamina that will sleep if it uses all stamina to move around. It has debugging features that if the debug is on then in the editor view to check where it is casting the ray to and what it finds.

changed from GameDBs.class(Deprecated)

GameDataManager.class

-This is a singleton class, this script will not get destroyed during scene changing. This class will carry player selection information of the player between scenes.

Effect control

changed from `BattleEffectDB.class`(Deprecated)

BattleEffectManager.class

-This is a singleton class in BattleScene sample, this script will provide easy access for any game object in the scene, this script will contains all the required game effect object(mostly particle system such as blood splitter, item dropped effect and selected target indicator. Also providing the methods that will spawn the desirable effect at given location.

CameraShaker.class

-This is used for critical hit effect, it will shake the camera. This script is referenced at **BallteEffectManager** in BattleScene.

SoftFlicker.class

-This is used for background fire effects. This script is used in ExampleMainScene.

MouseToRotate.class

-This is used for rotating character stands. This script is used in ExampleMainScene.

RandomAnimationEffect.class

-This is an extended class of **StateMachineBehaviour**, this class will randomly pick from 0~2 and will set as **Anim_Index** of **CharacterAnimationController**. This will get used for picking one animations of attacking and damaging from three choice of animations.

Sound control

changed from `BattleSoundsDB.class(Deprecated)`

BattleSoundsManager.class

-This is a singleton class in BattleScene sample, this script will provide easy access to provides all the audio clips for any game object in the scene so that wherever can get the reference of them and play desirable audio clip from it.

in V2.

This will be the one will contain audio source. So everywhere when the audio needs to be played will ask BattleSoundManager.

changed from `MovementSound.class(Deprecated)`

MovementSoundV2.class

-This is an extended class of **StateMachineBehaviour**, this class will play the movement sounds, if player is running then will play the sounds twice faster. This script will look for the instance of **BattleSoundsManager** in the scene.

Extension classes

ColorHSV.class

-This is a C# class for the creating random color in runtime, which is used for changing eye color of the player character.

Extension.class

-This is a C# extension class for providing extension methods for project. ex)
GetRandomElement, NextOf etc..

changed from SkinnedMeshToMeshV2.class(Deprecated)

SkinnedMeshToMeshV2.class

-This is a simple script to convert SkinnedMesh object to Mesh object. This can be very helpful script that when it is required to create Mesh renderer (no born attached) from SkinnedMesh renderer(born attached), so that transform can be controlled by gameobject not the animation. This is used for creating loot item from one of enemy's equipment after enemy is dead.

MeshFixer.class

-This is a simple script to fix when the SkinnedMesh got mixed up, simply check with the model object name and looks for the same named SkinnedMesh in the Scene.

RandomName.class

-This is a simple script to creating random name for random enemy, it is used in BattleScene.

GUI control

ClickableObject.class(Deprecated), CameraRelatedView.class(Deprecated) and ExitButton.class(Deprecated)

-We don't need to use this anymore as we are now using proper Unity3d GUI framework.

changed from GUIControl.class(Deprecated)

GUIController.class

-This is an extended class of the **MonoBehaviour**, this is used for buttons in ExampleMainScene, this script is dealing with all the button related control.

changed from BattleGUIControl.class(Deprecated)

BattleGUIController.class

-This is a singleton class in BattleScene sample, this script will provide easy access to the GUI object to any game objects in the scene.

changed from PlayerInfoControl.class(Deprecated)

PlayerInfoController.class

-This is a singleton class in BattleScene sample, this script will provide easy access to the GUI object for player information presenter. This script will require to setPlayer with the BaseCharacterController object (Player RPG information) to set the player status change listener to keep player status up to date.

HPBarControl.class

-This is an extended class of the **MonoBehaviour** that used for showing targeted enemy information (Name, Max Hp and Current Hp). Used in BattleScene.

BaseGameUIView.class (Newly added in V2.0)

-This contains base functionality of the UI that contains updateUI and init method.

-Extended classes

DamageInfoView, DroppedItemView, ItemDetailsView, PlayerBagView, PlayerStatusView and PlayerEquipmentView.

BaseGameUIController.class (Newly added in V2.0)

-This contains base functionality of the UI controller that contains show and init method and **BaseGameUIView**'s instance so that it will control the ui of this controller.

-Extended classes

DroppedItemViewController, ItemDetailscontroller, PlayerBagViewController, PlayerEquipmentViewController and PlayerStatusViewController.

BaseItemIconView.class (Newly added in V2.0)

This script is for the Inventory slot and Equipment slot view, which deals with Equipment Icon.

This script controls of setting equipment into the slot or remove it from it.

-Extended classes

InventorySlot and EquipmentSlot

Editor and helper class

LayerUtil.class

-This script will run automatically when whenever game requested to run it will set up layers that is required for V2.

Equipment Editor

This script adds editor button that we can run init method Equipment in inspector view.

EquipmentManagerEditor

This script adds these button to the EquipmentManagerV2 in Inspector view.

Initialize button (Only showing when this script is not initialized.)- If there are problem simply tick off initialized tick from inspector and run initialize by clicking button.

Random Equipment button - now in inspector we can just run random equipment without run the game.

Spawn Character button - we can now spawn current equipment character from equipment manager in editor view. Will take SpawnCharacterPos position to spawn.

Save Equipment to Prefabs button -this will extract all the equipment (Armor and Weapon type only) and save them as Prefab, will take the path from EquipmentPrefabTargetPath which has to be exist in order to complete this process.

FindMissingScripts

Credits

(Team Hwants)

Director: Hwan Kim

3D Models/Animations: Hwan Kim

Programmer: Hwan Kim

(Sound Effects)

59992__qubodup__swosh-sword-swing, 60004__qubodup__fast-sword-swing-sound

Author: qubodup

Source : <http://www.freesound.org/people/qubodup/sounds/59992/>

Licence : Creative Commons 0

195954__minian89__death-blood-splatter

Author : minian89

Sorcue: <http://www.freesound.org/people/minian89/sounds/195954/>

Licence : Creative Commons 0

166511__yoyodaman234__dirt-gravel-footstep-1

Author : Yoyodaman234

Sorcue: <http://www.freesound.org/people/Yoyodaman234/sounds/166511/>

Licence : Creative Commons 0

118513__thefsoundman__punch-02

Author : thefsoundman

Sorcue: <http://www.freesound.org/people/thefsoundman/sounds/118513/>

Licence : Creative Commons 0

87535__flasher21__splat

Author : Flasher21

Sorcue: <http://www.freesound.org/people/Flasher21/sounds/87535/>

Licence : Creative Commons 0

63529__florian-reinke__button-on

Author : florian_reinke

Sorcue: http://www.freesound.org/people/florian_reinke/sounds/63529/

Licence : Creative Commons 0

(Images)

Fire Texture

Author: Mixclipart Mari

Sorcue: <http://mixclipart.com/fire-free-psd-file/#more-798>

License: Free for personal and commercial purposes.

(Images)

GUI element

Author: Wurmheart

Sorcue: <http://opengameart.org/content/rpg-game-ui>

License: Creative Commons 0 (CC0 1.0)

Ring

Author: Luke.RUSTLTD

Sorcue: <http://opengameart.org/content/4-summoning-circles>

License : Creative Commons 0

WaterDrop

Source : <https://pixabay.com/en/drops-rain-raindrops-water-148870/>

Licence : Creative Commons 0

(Texters)

SandPebbles0094_5_M, marble

Source : <http://www.cgtextures.com/>

Licence : Creative Commons 0

"One or more textures on this 3D model have been created with photographs from Textures.com. These photographs may not be redistributed by default; please visit www.textures.com for more information."

(3D model)

rocks_03 and related textures

Author : Nobiax

Sorcue: <http://opengameart.org/content/rocks-03>

Licence : Creative Commons 0

(Font)

Ferrum font — Created in 2008 by **arro**

Sorcue: <http://www.fontspace.com/arro/ferrum>

Licence : Creative Commons 0



**Team
Hwants**

