



RPG ALL-IN-ONE (RPGAIO) (v1.0)

ZERO Code – ONE Game

Getting Started Guide

Contents

1.	Information	2
2.	Installation and Third Party Assets	4
3.	Setting Up Your Workspace	5
4.	Setting up Our Scene	6
5.	Customising Settings.....	8
6.	Adding Game Data	11
7.	Creating a Quest	24
8.	Putting it all together	28
9.	Where to go from here	35

1. Information

Welcome to the RPG All in One framework for developing games. Included is a set of tools to allow you to create a simple or complex RPG game without writing a single line of code.

Included here is documentation guiding you on how to get started with the framework, with video tutorials at: https://www.youtube.com/playlist?list=PLgDkDr4naeh-G1aingqNF_v9dL3kMnxPG

November 2017 Update:

Firstly, I would recommend all users to sign up to our forums for support, RPGAIO discussion, feature requests and more: <http://rpgaio.logicspawn.co.uk/forums/index.php>

This documentation should be useful for current versions of RPGAIO, but may be slightly updated. We are planning over the next month or so to release an up-to-date full documentation for RPGAIO including a new getting started guide and detailed information on how every system in RPGAIO works. If you run into any problems, post a message on the forums here: <http://rpgaio.logicspawn.co.uk/forums/index.php> or email us here: info@logicspawn.co.uk

RPG All-In-One (RPGAIO) is a **huge** creation tool for RPG games in continual development. It boasts a fully **code-less interface** from start to finish to create your RPG game. From main menu to end credits, an entire game can be created without writing a single line of code.

Unique Features of RPG AIO:

- Create an entire game, start to finish writing a total of **zero** lines of code
- Contains over 20 features and systems featured in well known RPG games like Skyrim, Diablo and Final Fantasy (see below)
- Customise all features of your created RPG through included EditorWindows and UI
- Customise calculations, create cutscenes and events and write dialog using the Node System
- All code and required assets : Under 20mb!
- Video tutorials (see link below) and plenty of support available (Email below)
- Download example projects from the website.

RPG Features included:

- Robust Save Load System
- Customisable Stats (Attributes, Vitals, Statistics and Traits)
- Highly Advanced Skill System (AOE , Projectile, Melee, Aura, Ability, Talents, Status Effects and Traits)
- Advanced Quest System (Multiple quest condition types as well as quest chains)
- Inventory System with a multitude of Item Types (e.g. readable books and audiobooks)
- NPC System (Including dialog, allied npc and reputations)
- Cutscenes
- Detailed Loot System (Per-monster loot as well as loot tables)
- Achievements system
- Harvesting System
- Crafting System (Including dismantling)
- Vendor System
- ToolTip system for items.

- More Info
- Video Tutorials

NOTE: This is the source code version. As the source is not designed to be modified (it's codeless,
remember !), the project works entirely from the Unity Editor and custom RPGAIO editor
windows, so no documentation is included for the code.

2. Installation and Third Party Assets

As of Update 1.3.0, RPGAIO should not require any additional installation. If you do run into any issues, then please **ensure** you read the included InstallationGuide.txt file included inside the Assets folder.

Once done, to follow this tutorial you should import some assets, in this tutorial I used the following from the asset store:

Enemy model: Allosaurus: <https://www.assetstore.unity3d.com/en/#!/content/7477>

Environment model: Dead Tower Scene: <https://www.assetstore.unity3d.com/en/#!/content/1817>

Player model: Dragon Warlord Bruce: **(No Longer Available)**

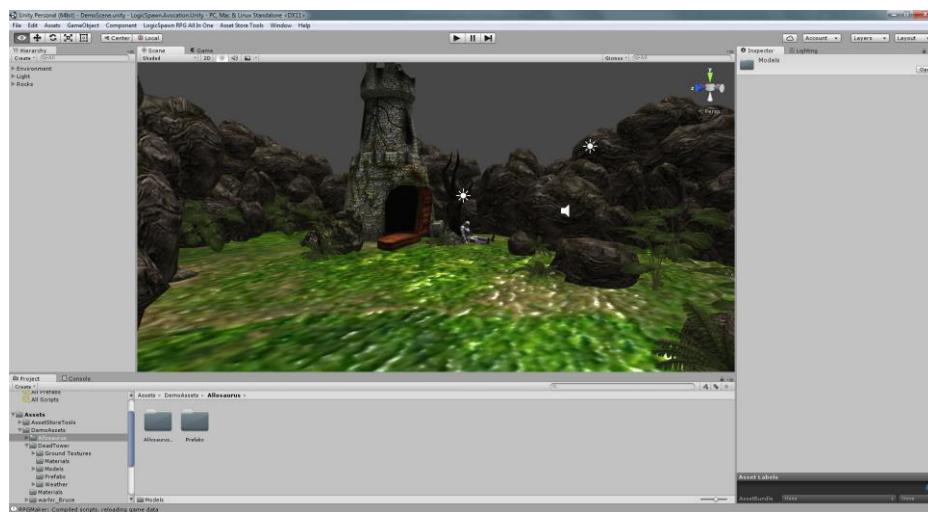
Instead of the Dragon Warlord Bruce model, which is no longer available, find any model that uses legacy animations (as this tutorial covers the legacy animation feature of RPGAIO not Mecanim)

Alternatively, you can use a Mecanim-enabled character and replace the animation section of this tutorial (for your Mecanim-model) by following the Mecanim tutorial here:

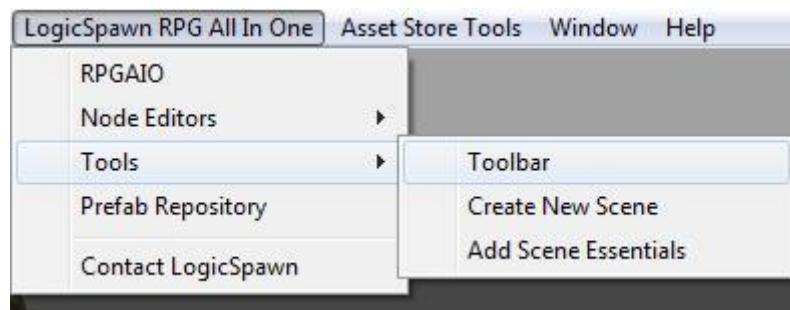
https://www.youtube.com/watch?v=0cvwtk8Y_Uk

3. Setting Up Your Workspace

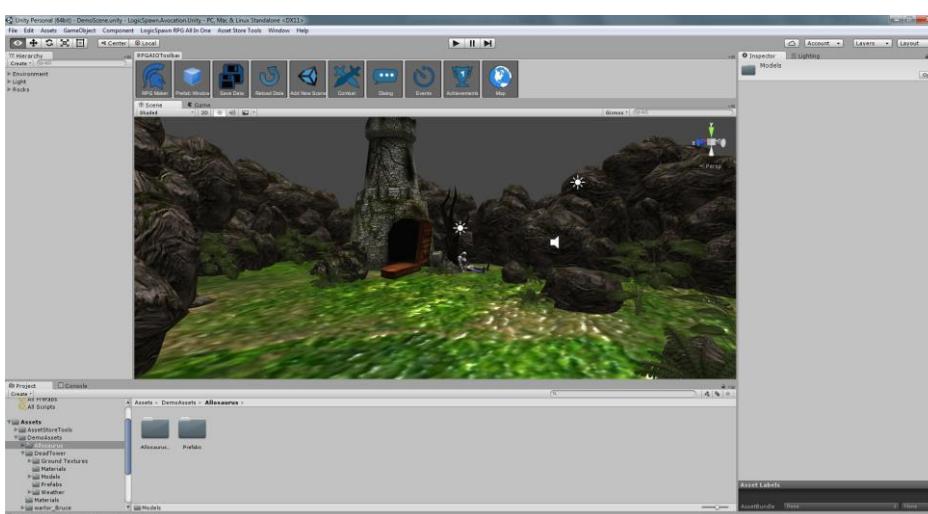
RPGAIO comes with a handy toolbar to improve productivity while developing. When you first import the project, the layout will look similar to this, without the toolbar:



Click ‘LogicSpawn RPG All-In-One’ under Tools on Unity Editor’s menu bar and select Tools > Toolbar (In recent updates, this is now under Tools in the toolbar of Unity [Tools > LogicSpawn RPG All in One > Tools > Toolbar])

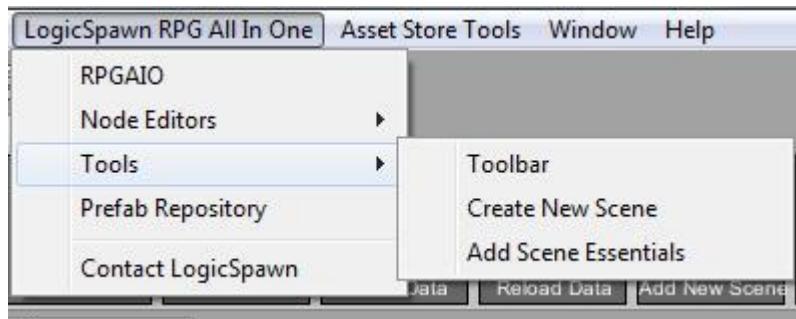


Once it is open, drag and drop it somewhere preferable so you have access to it's tools.

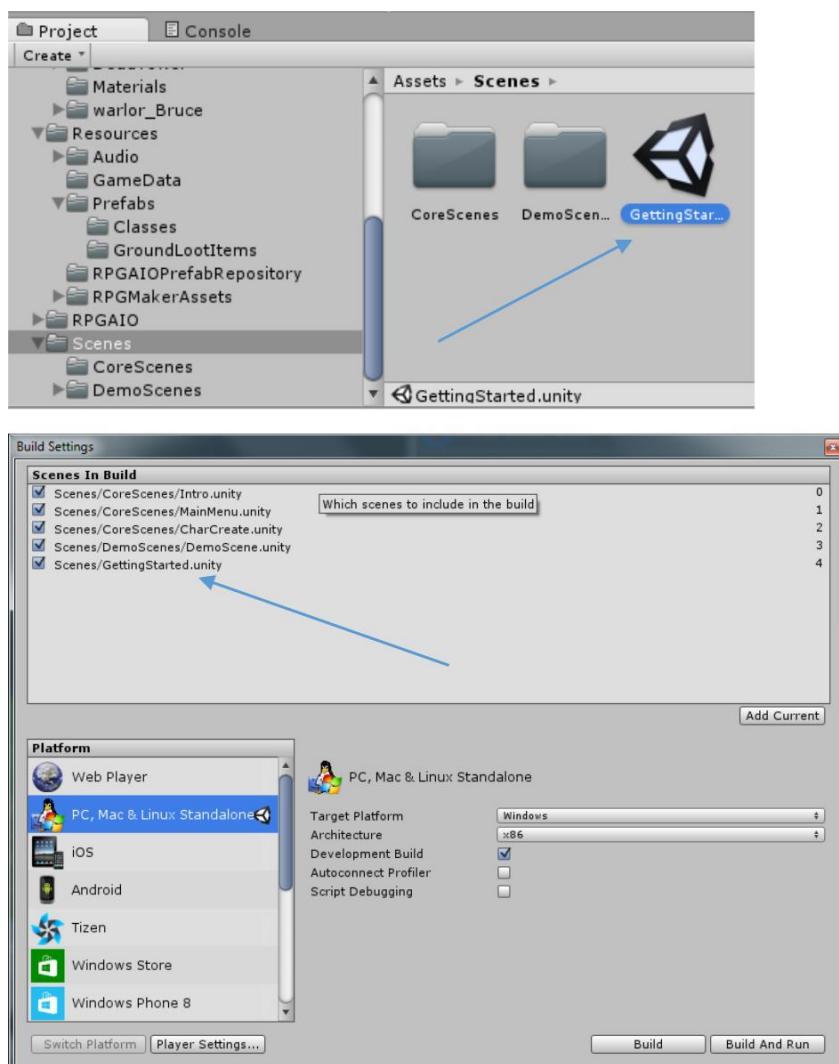


4. Setting up Our Scene

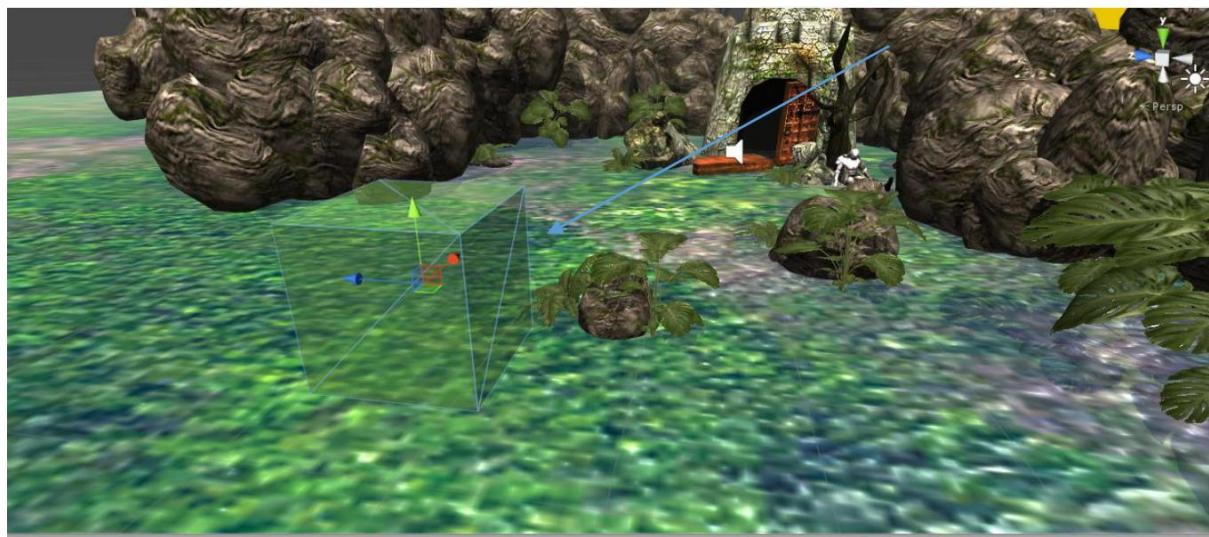
I've chosen to use an existing scene for our demo. The Dead Tower scene hasn't got any of the required game objects we need for our RPG to work. Selecting the scene, simply go to the menu bar : LogicSpawn RPG All In One > Tools > Add Scene Essentials.



We'll then be notified we'll need to add it to build settings also. While we're here we can optionally move it our Scenes folder for better organisation.



Finally, we can select the spawn position object in the hierarchy and move it somewhere suitable:

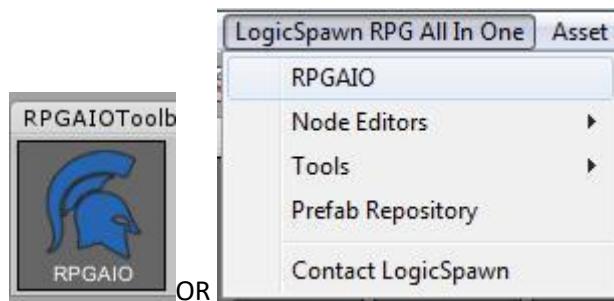


5. Customising Settings

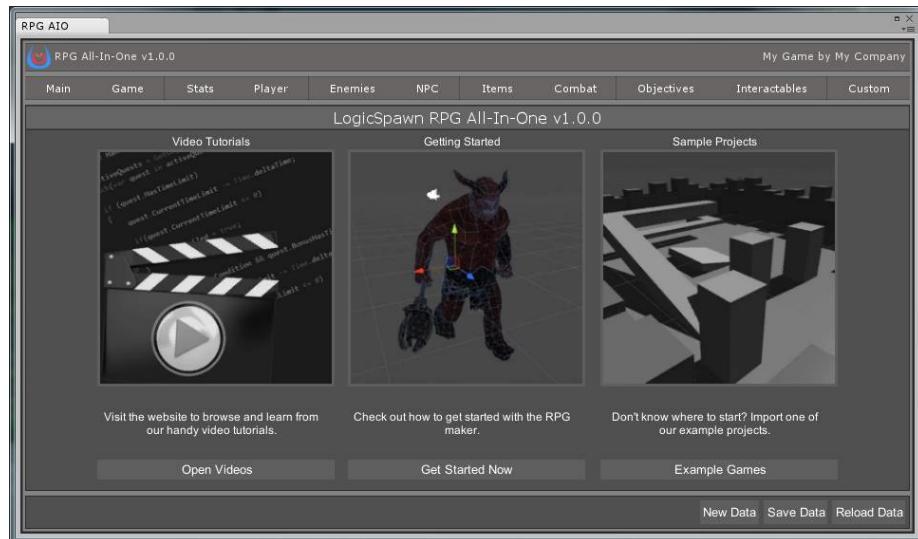
Let's begin customising settings. RPGAIO has a whole suit of customisations you can make, most of which are self-explanatory. We will cover a few here.

Auto-save

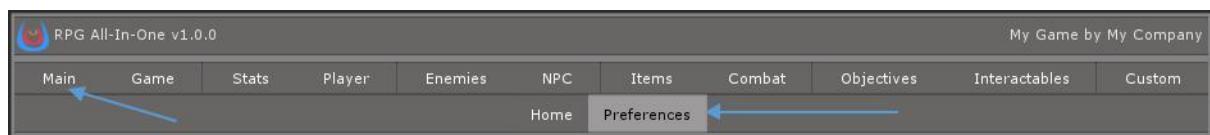
First we will enable an autosave, so our game data is saved every 60s. Let's first open the RPGAIO window. You can either click the RPGAIO button on the toolbar, or access it through the menu bar.



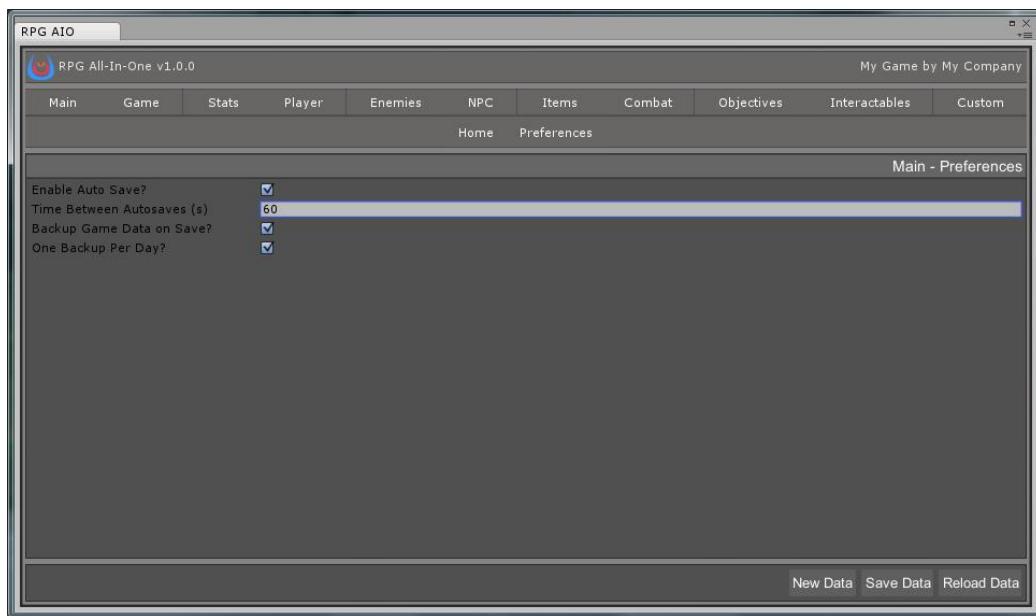
Once it's open it will look something like this:



We'll open the 'Main' menu option and choose 'Preferences'



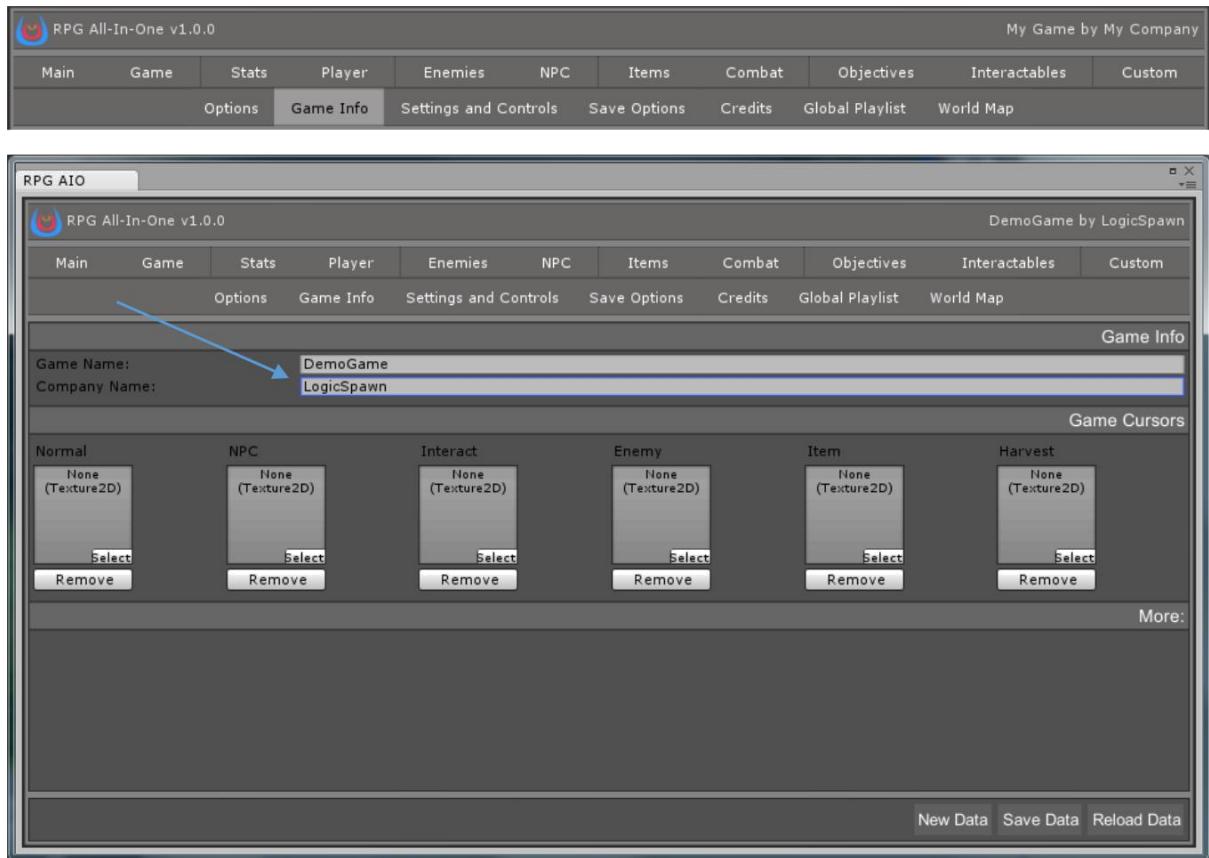
Then we can simply enable auto save, and set it to 60s



As you can see, you can also choose whether to back-up the game data every time we save, and whether or not we make just one backup a day. Depending on your game data you may not want more than 1 backup a day (the file size can add up if you're game data is really huge)

Game Details

Let's also set the name and company of our game, useful for the Main Menu by going to 'Game' and then 'Game Info'

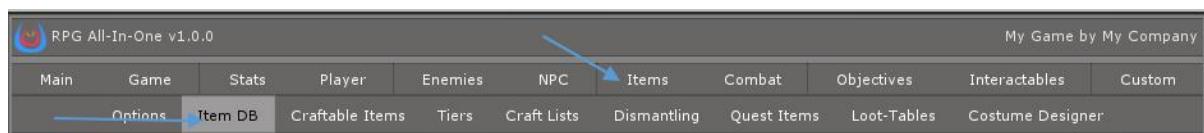


6. Adding Game Data

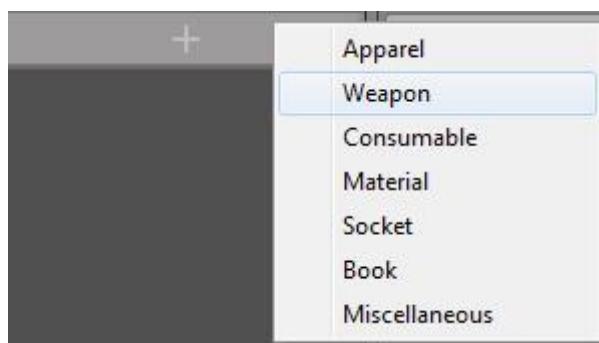
What's an RPG Game without data? Let's go ahead and add a few items, a loot table, a class and an enemy.

Items

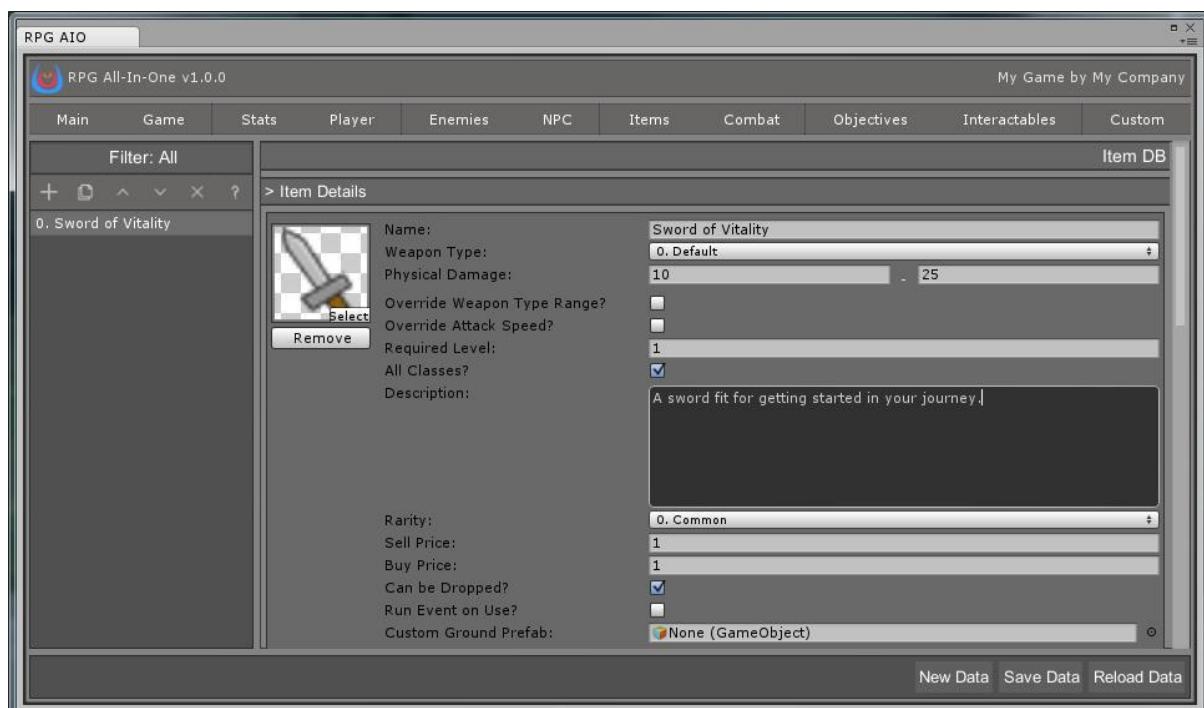
We can add items by selecting the 'Items' option on the menu, and then choosing 'Item Db'



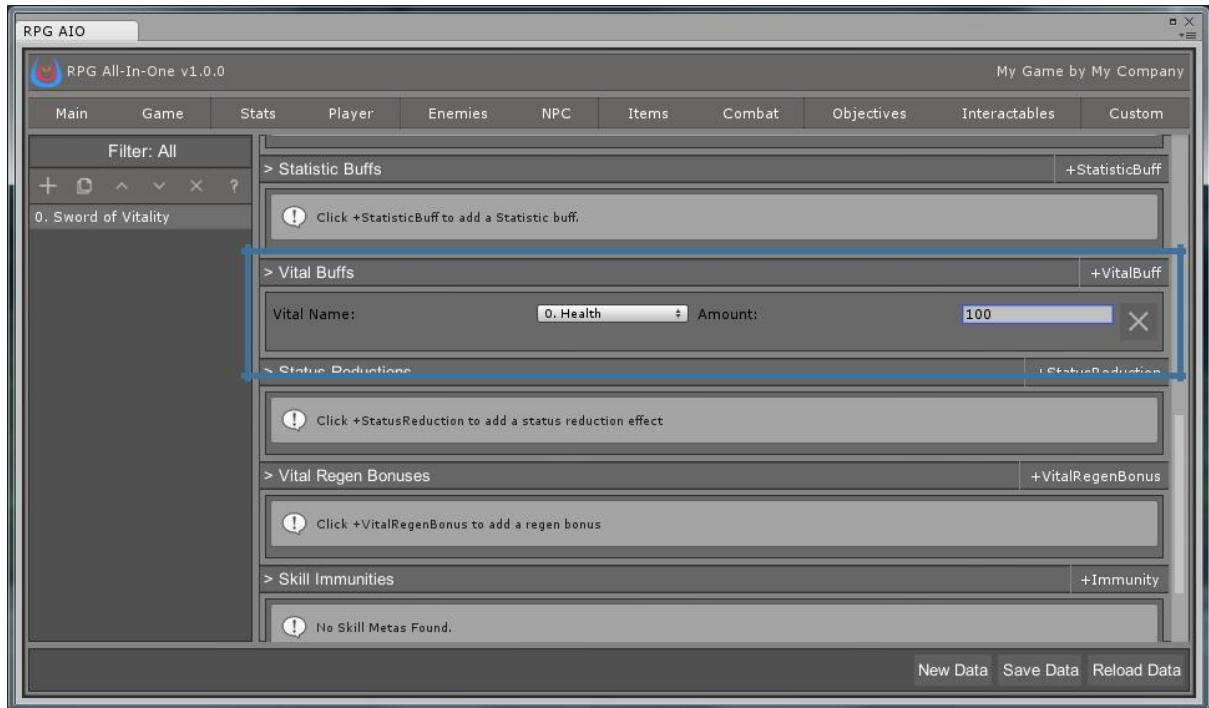
We'll add a new item by clicking the + button and choosing Weapon



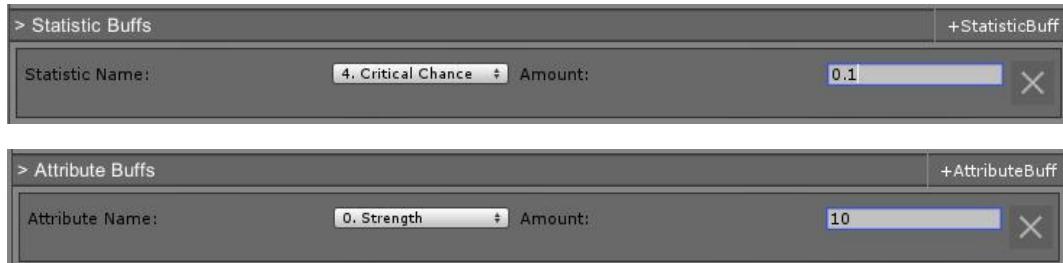
We'll give it some basic properties for now: a name, 10-25 damage, an icon, and a short description and make it give our player 100 health. Here's how it looks filled in:



And we can scroll down to add the vital (health, mana, etc) buff:



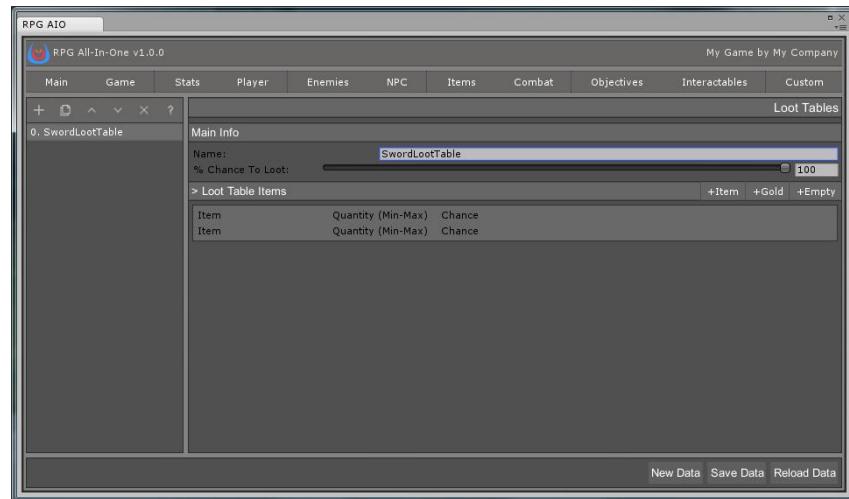
We'll repeat this process but instead we can make a sword which gives a statistic (critical chance, movement speed, etc.) buff of +10% critical chance and a sword which gives an attribute (Strength, Dexterity, etc.) buff of +10 strength.



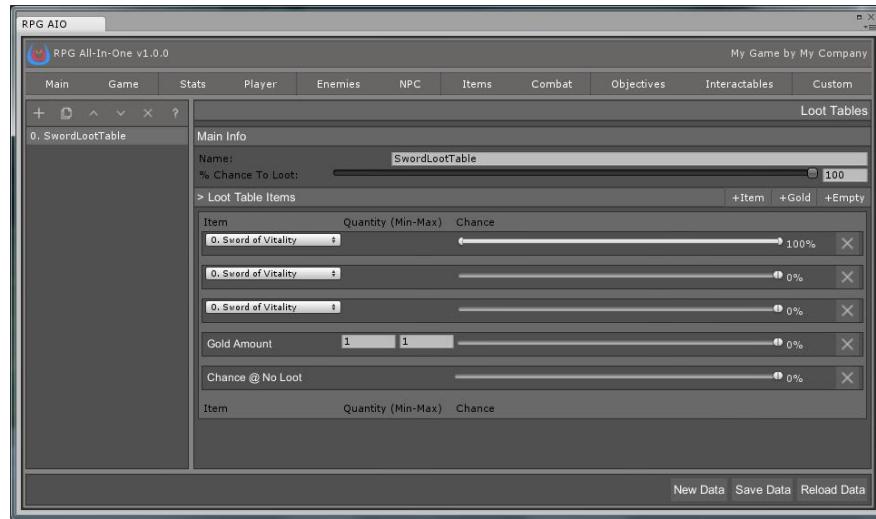
Loot Tables

Loot tables are a powerful way to handle the way items are dropped. Let's create one that gives a chance at gold, and the three swords we've made. You can open the loot tables by choosing the 'Loot-Tables' option under 'Items'

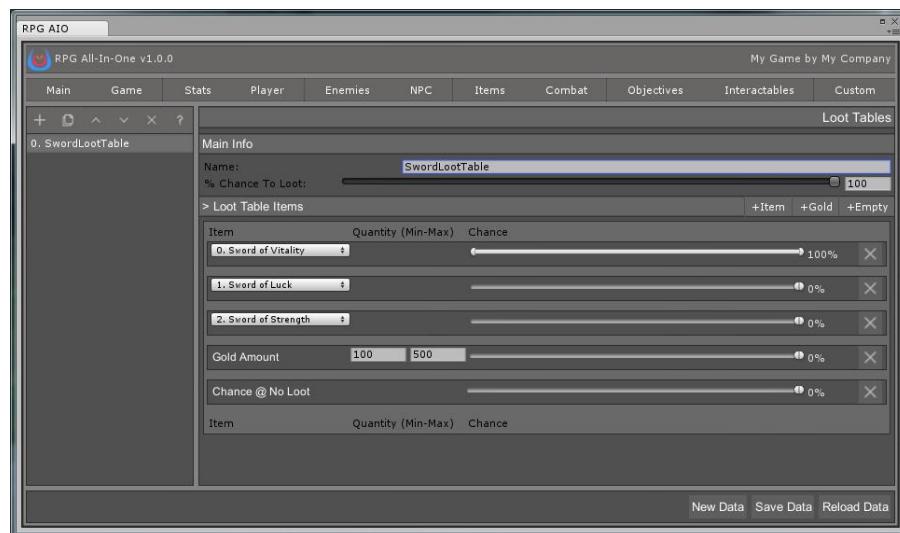
We'll add a new loot table by hitting the + button and give it a name.



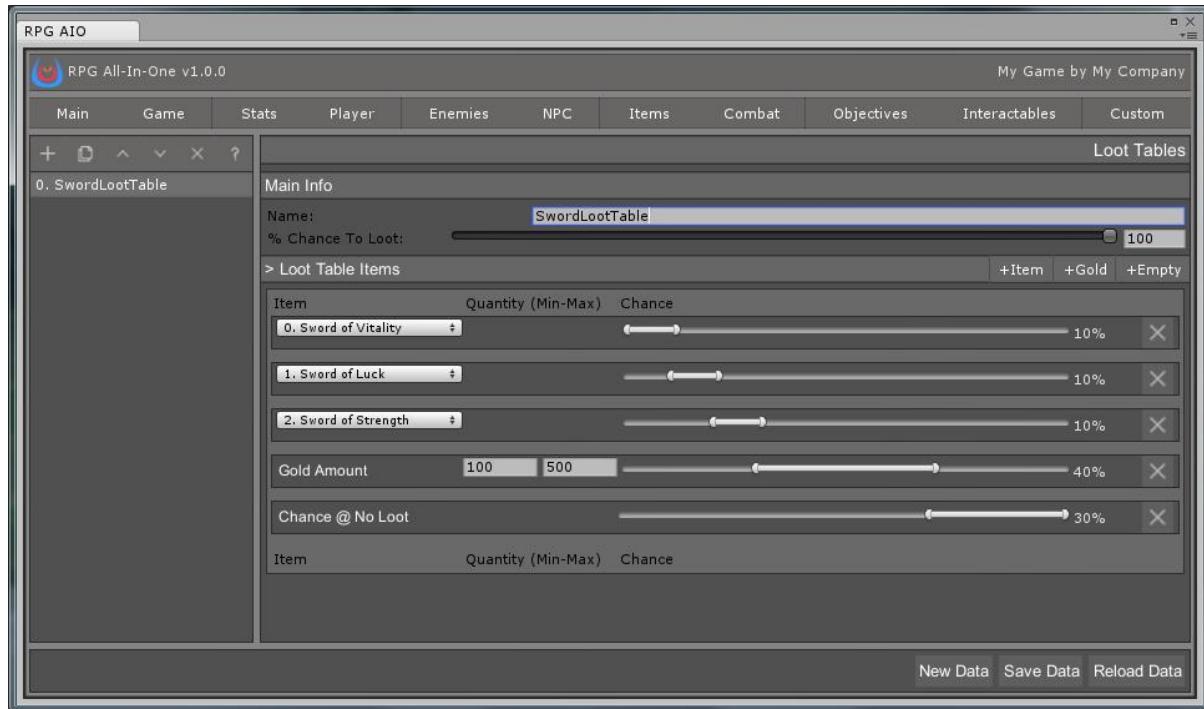
We can then add some loot options. We'll add 3 item loot options, 1 gold option and 1 empty – keeping the % chance to loot at 100.



Next, using the dropdowns and text-fields, we'll set the 3 items to the 3 swords we've made and the gold to 100-500.



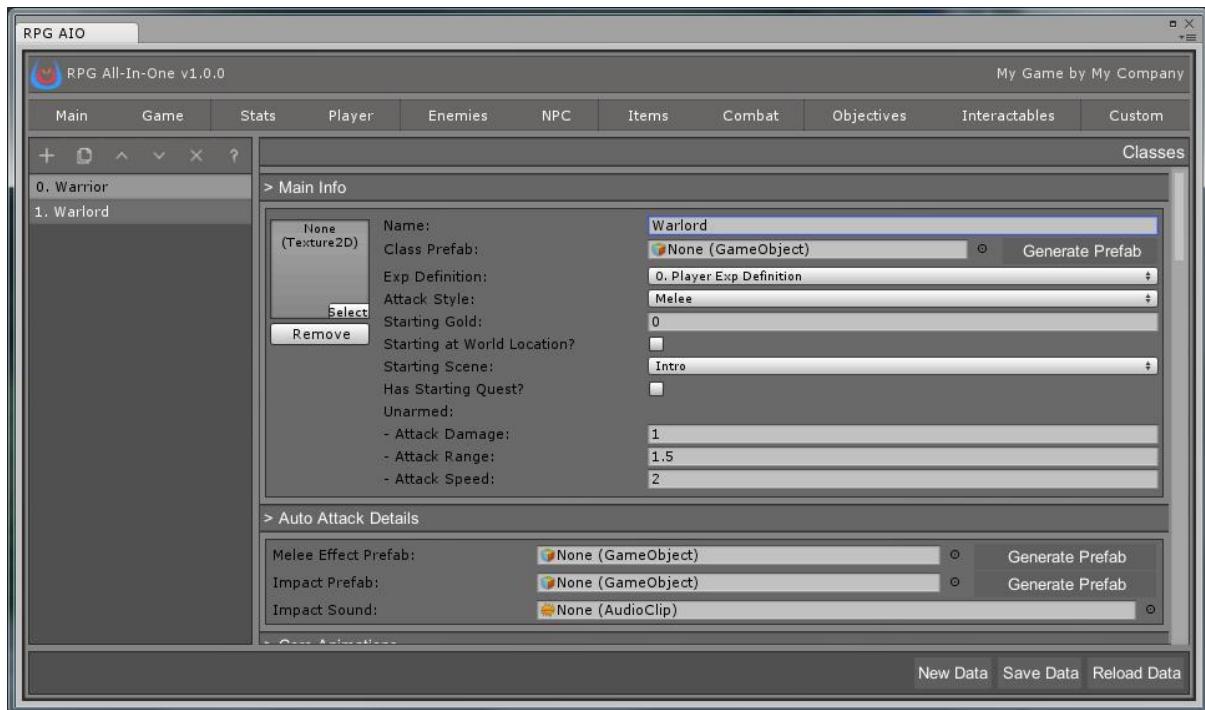
Finally we'll adjust the sliders to set the percentages. Making it so the 3 swords each have a 10% chance to drop, gold a 40% chance and the chance of no loot being 30%.



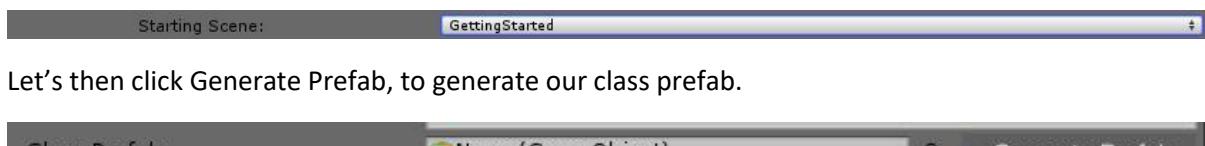
Just like that we've created a loot table we can later assign to one of our enemies in the game.

Creating a class

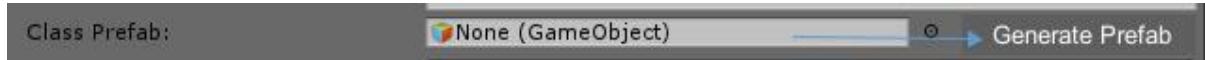
Next let's create a class in our game under 'Player' > 'Classes'. We'll add a new class by clicking the '+' button and give it a name. For now we'll leave the rest as default.



Make sure to set our starting scene to a valid one (an actual game level, with our essentials), the getting started scene we created earlier as opposed to 'Intro':



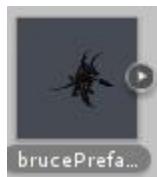
Let's then click Generate Prefab, to generate our class prefab.



This will generate the prefab that will be spawned for the player once they enter the level.

Note: We can rename the prefab to make it easier to find. Click the GameObject field and it will show you the prefab in the browser. You can then simply rename it and drag and drop it back into the GameObject field as once you rename it, it clears the field.

Selecting the prefab now in the scene, which I have renamed 'Warlord', you can then drag and drop your model as a child of the prefab:



Here is the model, all it has attached on the parent is an animation component.

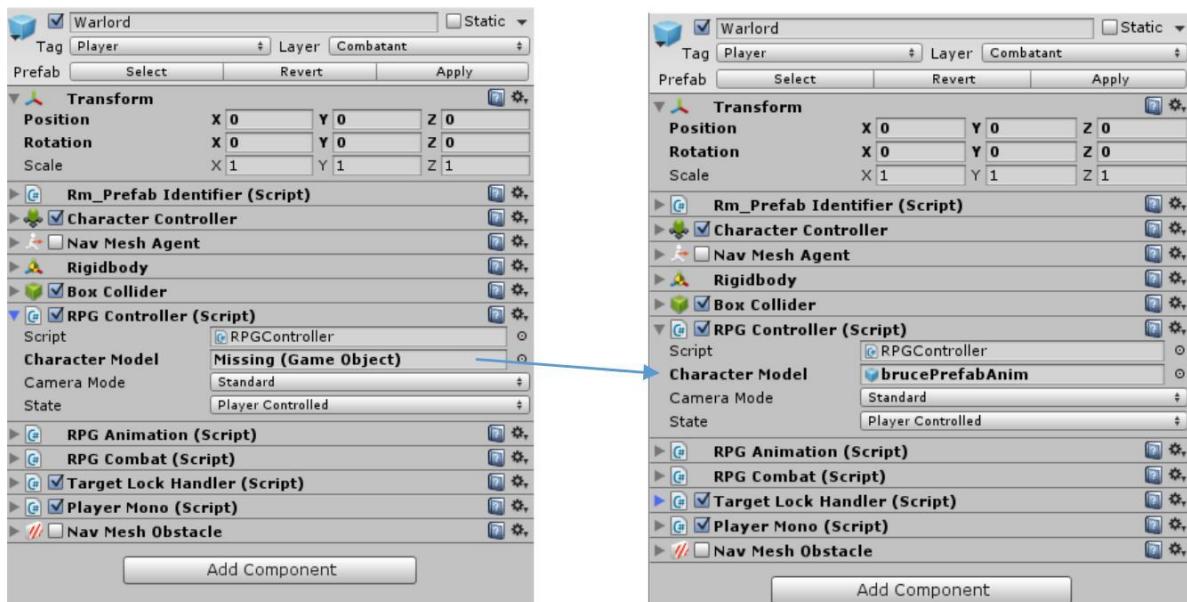


We can then remove the capsule model, as this is no longer required, and then reset the position and rotation of our model (brucePrefabAnim) and apply the prefab to save the changes.

We should end up with this:



Now select the Warlord gameobject (the parent) and we can assign the animationModel:



Once again click Apply to apply the prefab changes.

We can then go back to our RPGAIO menu with our brucePrefabAnim selected (as it has the animation component) and assign our animations:

RPG AIO

RPG All-In-One v1.0.0

DemoGame by LogicSpawn

Main	Game	Stats	Player	Enemies	NPC	Items	Combat	Objectives	Interactables	Custom
Classes										
0. Warrior	Unarmed	LeftStraightBlow	1	None (AudioClip)	<input checked="" type="radio"/>	<input type="checkbox"/>				
1. Warlord	Walk	Walk	1	None (AudioClip)	<input checked="" type="radio"/>	<input type="checkbox"/>				
	Walk Back	MoveBack	1	None (AudioClip)	<input checked="" type="radio"/>	<input type="checkbox"/>				
	Run	Run	1	None (AudioClip)	<input checked="" type="radio"/>	<input type="checkbox"/>				
	Jump		1	None (AudioClip)	<input checked="" type="radio"/>	<input type="checkbox"/>				
	Turn Right	LookRight	1	None (AudioClip)	<input checked="" type="radio"/>	<input type="checkbox"/>				
	Turn Left	LookLeft	1	None (AudioClip)	<input checked="" type="radio"/>	<input type="checkbox"/>				
	Strafe Right	MoveRight	1	None (AudioClip)	<input checked="" type="radio"/>	<input type="checkbox"/>				
	Strafe Left	MoveLeft	1	None (AudioClip)	<input checked="" type="radio"/>	<input type="checkbox"/>				
	Idle	CombatMode	1	None (AudioClip)	<input checked="" type="radio"/>	<input type="checkbox"/>				
	Combat Idle	CombatMode	1	None (AudioClip)	<input checked="" type="radio"/>	<input type="checkbox"/>				
	Take Hit	TakeHit	1	None (AudioClip)	<input checked="" type="radio"/>	<input type="checkbox"/>				
	Falling		1	None (AudioClip)	<input checked="" type="radio"/>	<input type="checkbox"/>				
	Death	Dieing	1	None (AudioClip)	<input checked="" type="radio"/>	<input type="checkbox"/>				
	Knock Back	TakeHit	1	None (AudioClip)	<input checked="" type="radio"/>	<input type="checkbox"/>				
	Knock Up	TakeHit	1	None (AudioClip)	<input checked="" type="radio"/>	<input type="checkbox"/>				

New Data Save Data Reload Data

If we scroll down further we can then apply some attack animations as well:

> Default Attack Animations

+Animation

ID	Animation	Speed	Impact	Sound	Reverse?
----	-----------	-------	--------	-------	----------

Let's click +Animation a few times:

> Default Attack Animations

+Animation

ID	Animation	Speed	Impact	Sound	Reverse?
Attack 0	None	1	0.7	None (AudioClip)	<input checked="" type="radio"/>
Attack 1	None	1	0.7	None (AudioClip)	<input checked="" type="radio"/>
Attack 2	None	1	0.7	None (AudioClip)	<input checked="" type="radio"/>

And then assign some animations

> Default Attack Animations

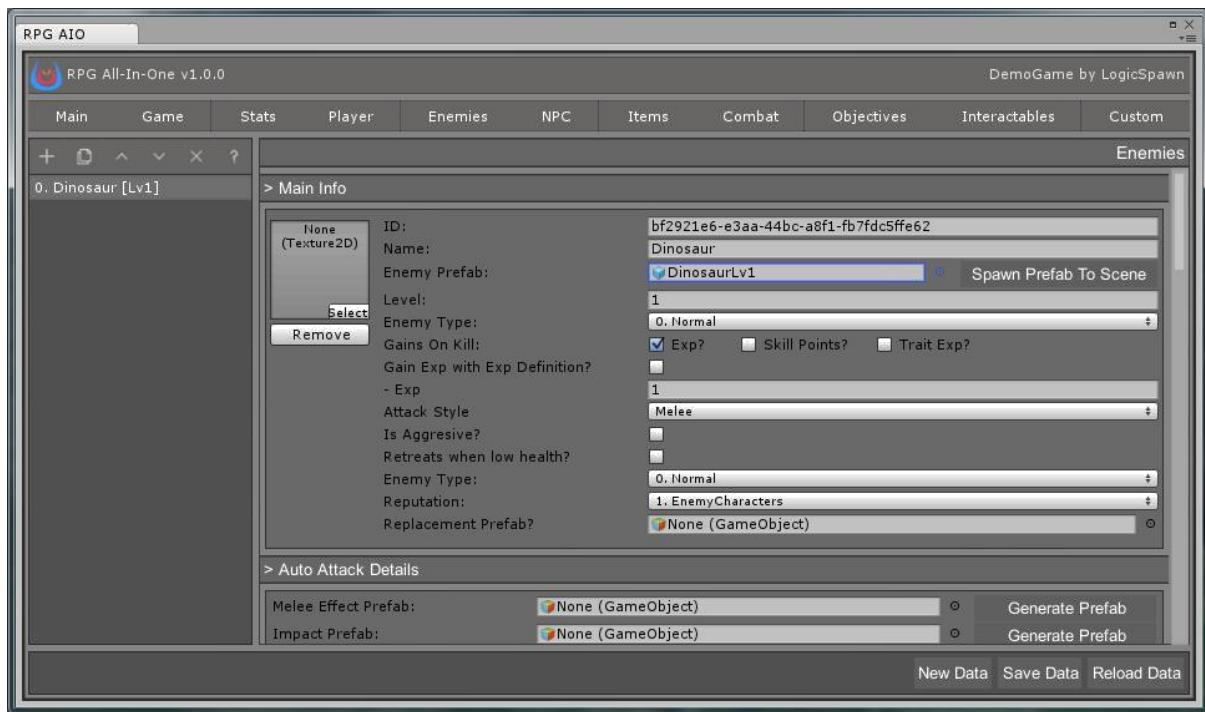
+Animation

ID	Animation	Speed	Impact	Sound	Reverse?
Attack 0	LeftStraightBlow	1	0.7	None (AudioClip)	<input checked="" type="radio"/>
Attack 1	RightStraightBlow	1	0.7	None (AudioClip)	<input checked="" type="radio"/>
Attack 2	ScissorsBlow	1	0.7	None (AudioClip)	<input checked="" type="radio"/>

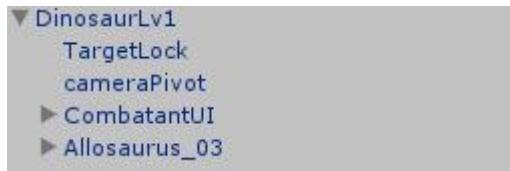
Hit save data, and for now, we're done with creating our class.

Creating an enemy

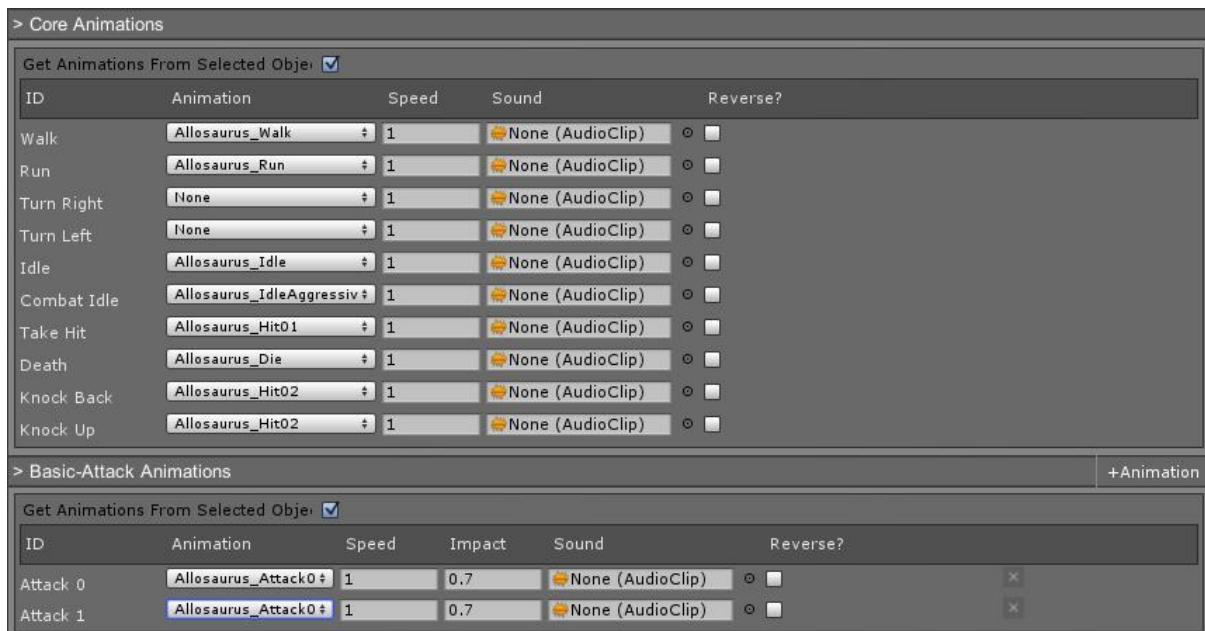
Creating the game data and prefab for an enemy is almost exactly the same, but we'll cover it again briefly. First add a new enemy by going to 'Enemies > Enemies', give it a name and click generate prefab.



Just like our class prefab, we can drag and drop our enemy model with animations as a child of the new prefab in the scene deleting the capsule. Just like before I've renamed it and reassigned it to the 'Enemy Prefab' field.



Once again assign the model (Allosaurus_03) as the animation model, apply the prefab changes and select the model so we can use its animations. Once we're done adding some the core and basic attack animations we can set up more details:



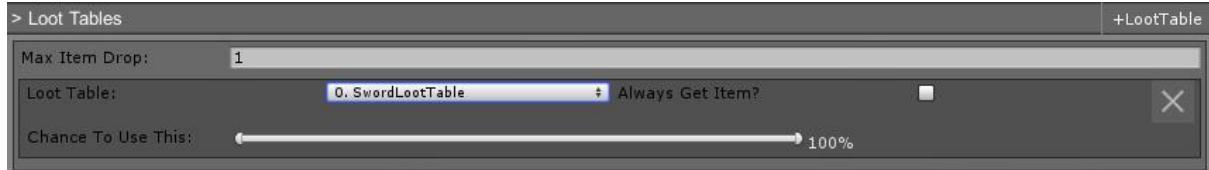
Let's scroll down the enemy definition and change the damage of the enemy to 2-5 and the health of the enemy to 20:

The screenshot shows the configuration interface for an enemy. It includes sections for 'Damage' (Physical Damage: 2-5), 'Base Attributes' (Strength: 10, Dexterity: 10, Intelligence: 10, Vitality: 10), and 'Base Vitals' (Health: 20, Mana: 0).

If we scroll down further we can see the loot options:

The screenshot shows the 'Loot Tables' section where 'Max Item Drop' is set to 1, and a note says 'Click +LootTable to add a loot table that the enemy can drop items from.' Below it is the 'Guaranteed Loot and Gold' section, which contains a note 'Click +Loot to add loot that will always drop from this enemy. Click +Gold to add gold drops.' with buttons for '+Loot' and '+Gold'.

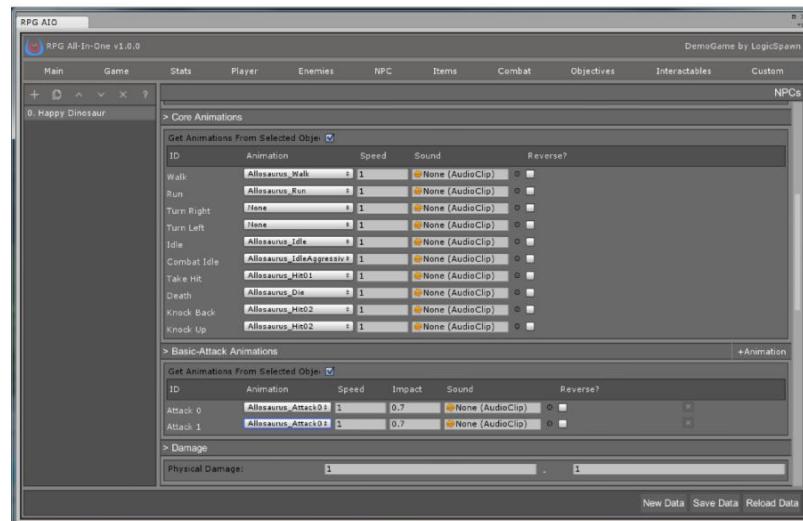
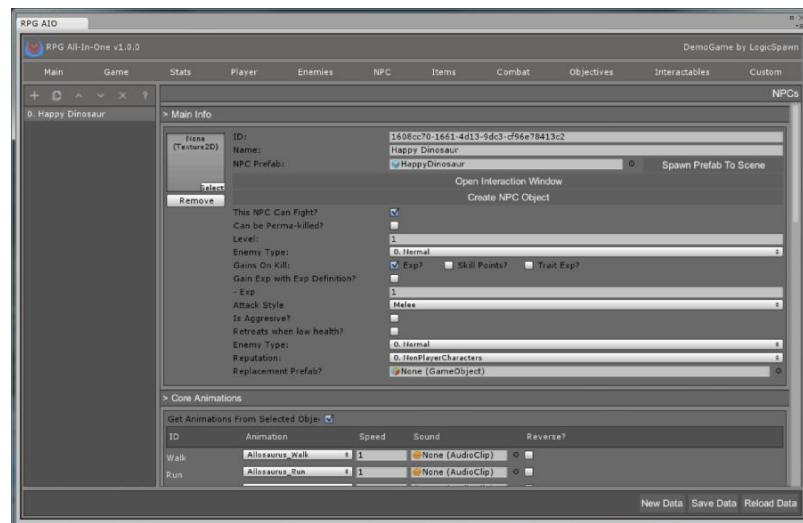
Let's assign the loot table we created earlier, and leave the guaranteed loot as default. We can do this by clicking the **+LootTable** button and choosing our loot table from the drop-down. We have some additional options to customise here, such as always getting an item (simply ignores the loot table's % chance of loot, so you can still get an empty item) or the chance of using that loot table (if we have multiple loot tables assigned to an enemy) but we'll leave them default for now.



That's it for game data for now!

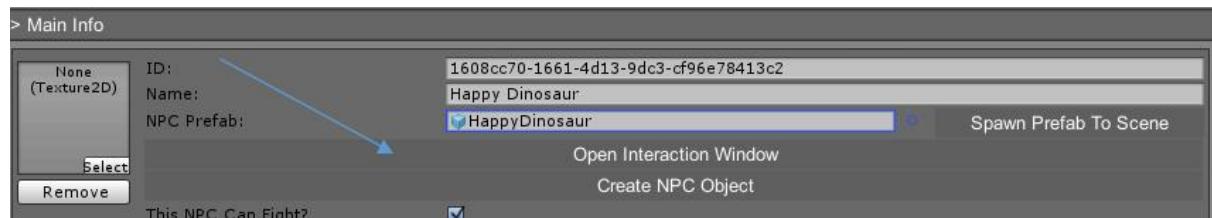
Creating an NPC

Good news, creating an NPC is exact same process as classes and enemies. So go ahead and add a new NPC definition under **NPC > NPCs**, giving it a name and generating a prefab. Assign your character model to the NPC under its parent, delete the capsule and assign the character model to the **RPG Controller** component:

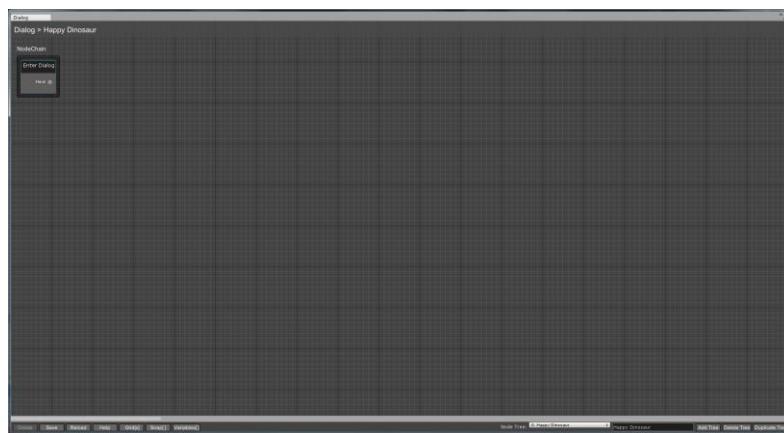




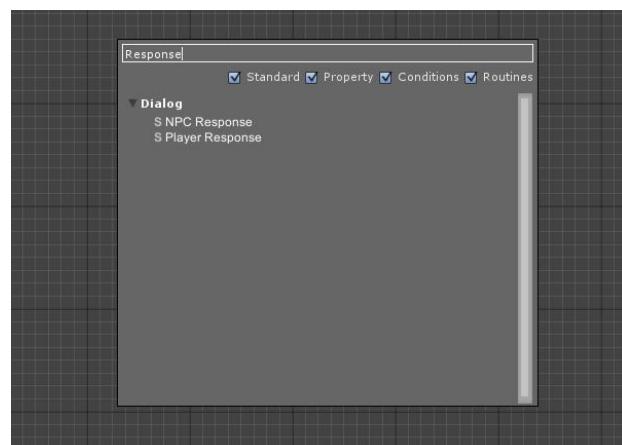
Next let's give our NPC some basic dialog, first click 'Open Interaction Window'



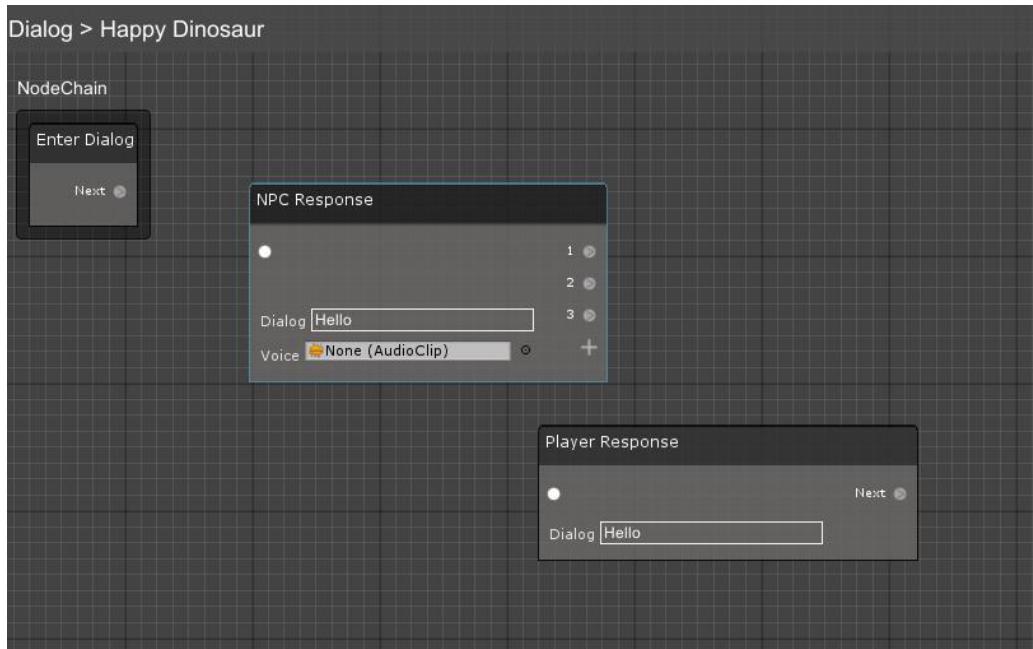
This will open up our RPG Node Editor. If the enter dialog button is in the corner, just drag it out into the main area:



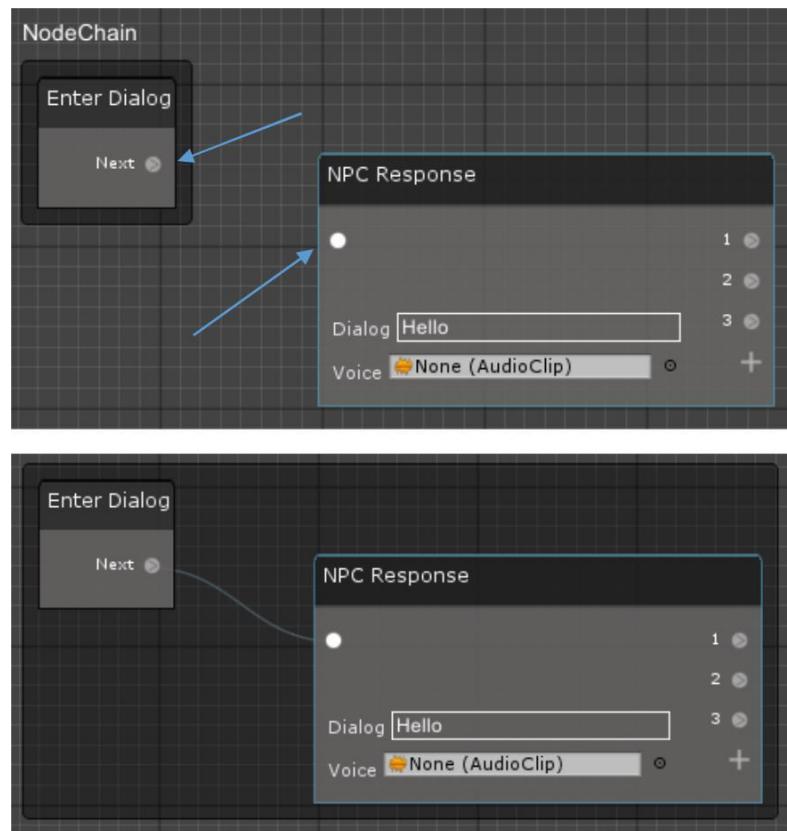
Let's go ahead begin creating our dialog, first we'll need an NPC response. To add a node simply right click an empty area. You can begin typing straight away to search through the available nodes. Type in: 'Response' and you should see 'NPC Response' and 'Player Response' listed. We'll be using both of these.



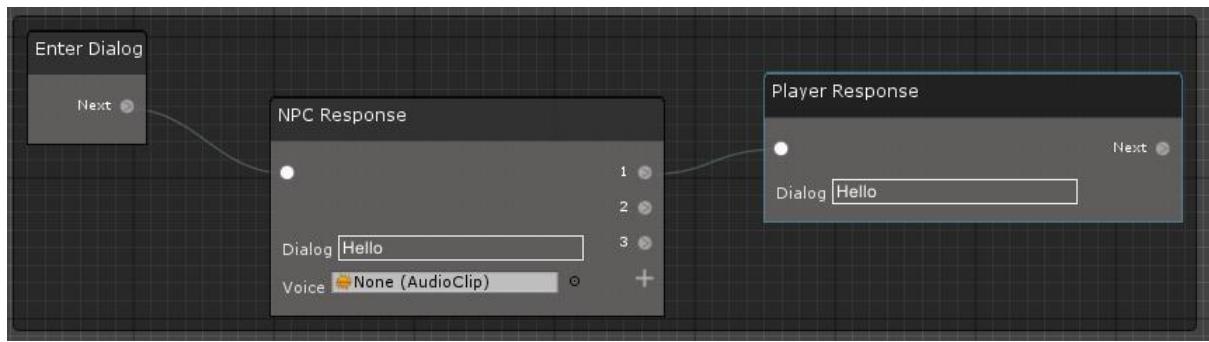
Click NPC Response and it will create a node roughly where your mouse was placed when you right click. Let's repeat this and instead add a Player Response node, so we have this:



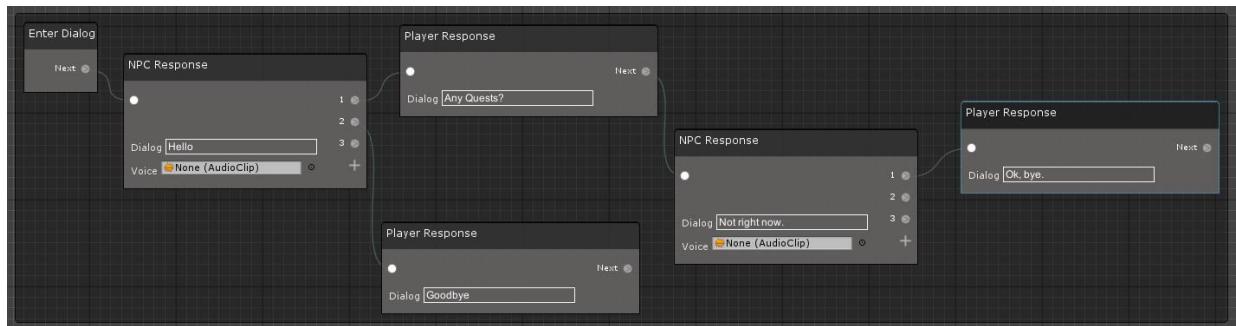
To link nodes, simply click on a grey circle with a '>' and then click on a white circle to link:



Let's also link our NPC Response to one of our available player responses:

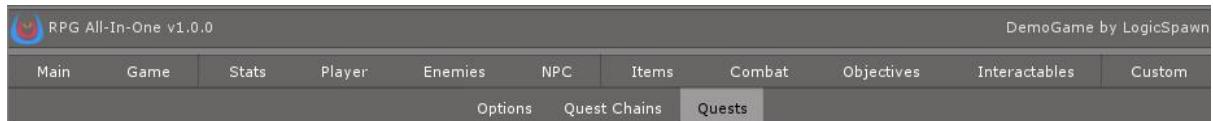


By doing this, and adding more nodes, we can create a simple dialog:

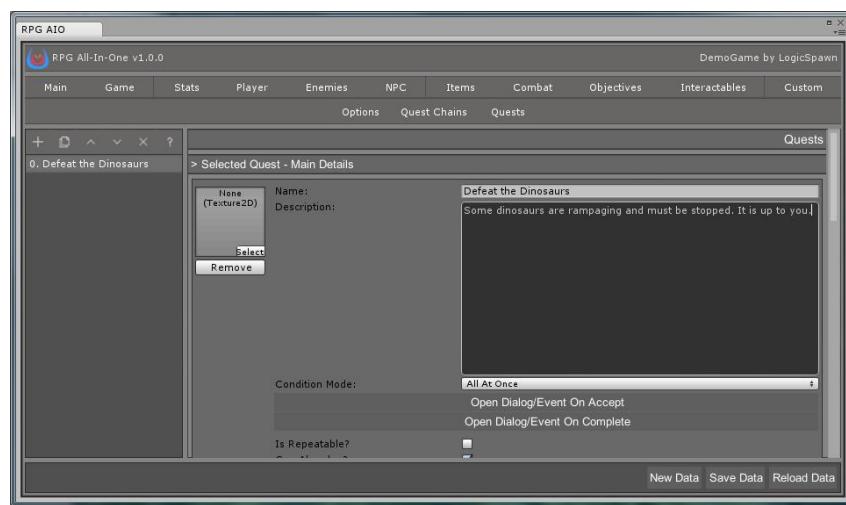


7. Creating a Quest

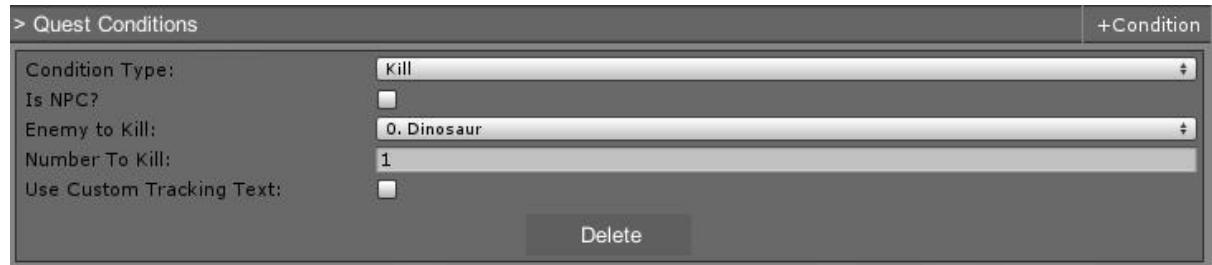
Let's finish us by giving the player something to do. In the RPGAIO menu, let's add a new quest and give it a name by going to Objectives > Quests:



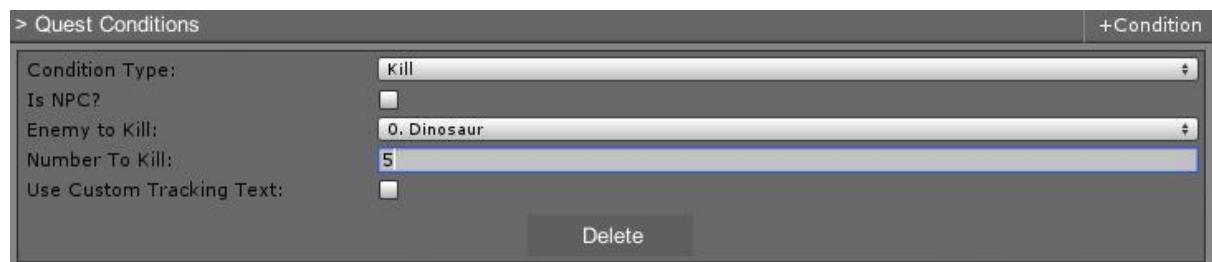
Add the new quest by clicking the + button and give it a name and description:



Next let's add some conditions to complete this quest. Scroll down to quest conditions and click the +Condition button to add a new condition:



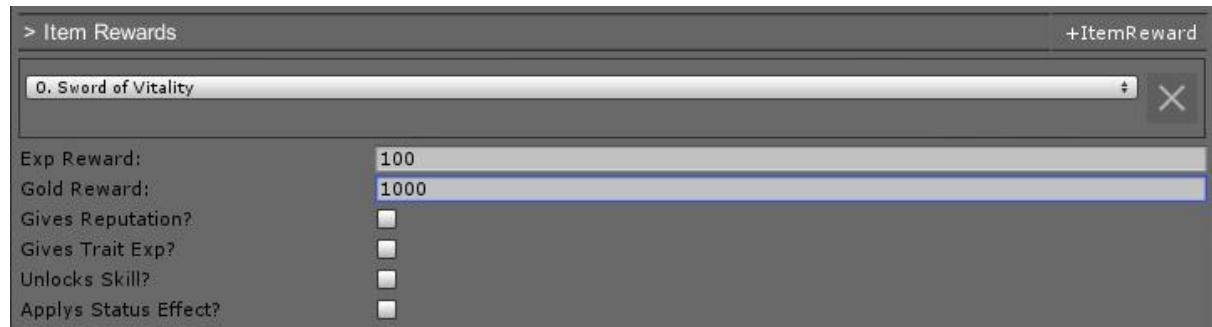
For now we'll keep it to a kill quest. As we only have 1 enemy at the moment, it luckily defaults to our Dinosaur enemy. Let's change the amount to kill to 5.



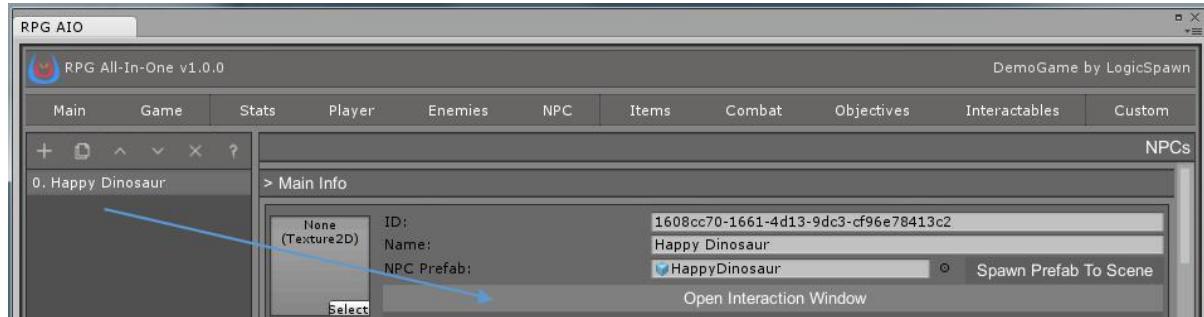
Let's now add some rewards for completing this quest, scroll down to the rewards section and click +ItemReward. Choose the Sword of Vitality, just in case our player doesn't manage to get one as a drop!



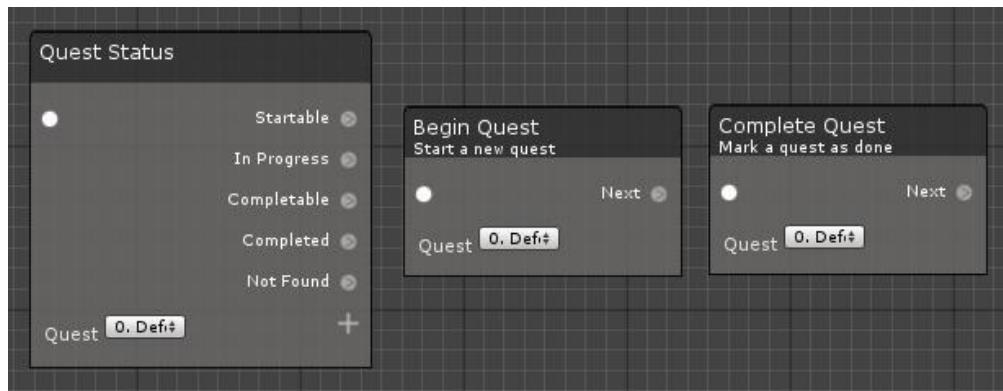
Let's also give the player 100 experience, and 1000 gold.



Just like that we've created a quest. Let's now add it into the game by going back to our NPC we created earlier and opening his interaction window:



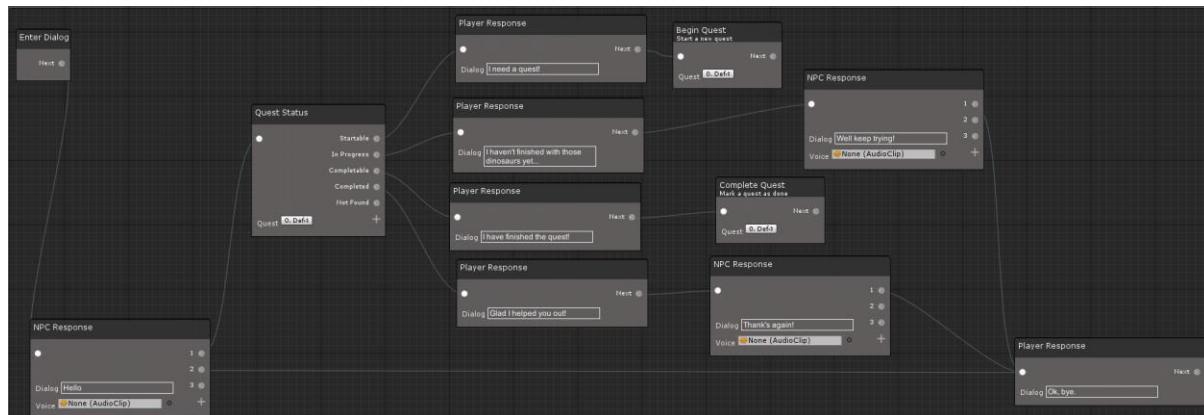
If we right click and search for nodes, we can add a new one called 'Quest Status' to check what the status of a quest is. Begin Quest to start a quest, and Complete Quest, to finish a quest once conditions are met.



Let's integrate these into our dialog so it reacts accordingly, the quest we're checking has already been set to 'Defeat the Dinosaurs' for all 3 nodes, if not, set it with the drop down:



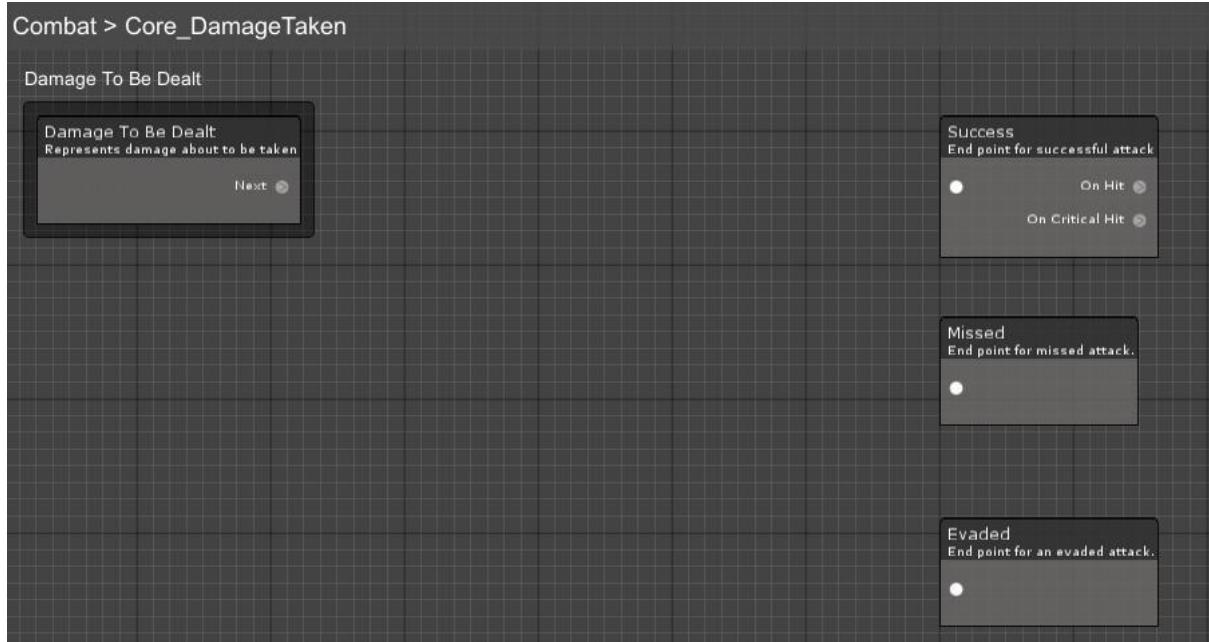
Let's now create some dialog:



When we speak to the NPC, we'll have different responses depending on our status with the quest. Complete Quest is only accessible if the quest status is completable (conditions met, but not turned in). We can also notice that multiple NPC Responses go to our 'Ok, Bye' response as well as the Begin Quest and Complete Quest not requiring any player responses after it. Though you could continue by adding an NPC response after them if you decided to.

8. Putting it all together

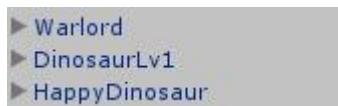
Before we put our characters in our scenes, be sure to check your combat nodes are setup correctly. Open the Combat node window by clicking the icon on the toolbar or LogicSpawn RPG All In One > Node Editors > Combat:



Ensure that your damage to be dealt node is linked to the success node. Later you can customise this for missed/evaded attacks.



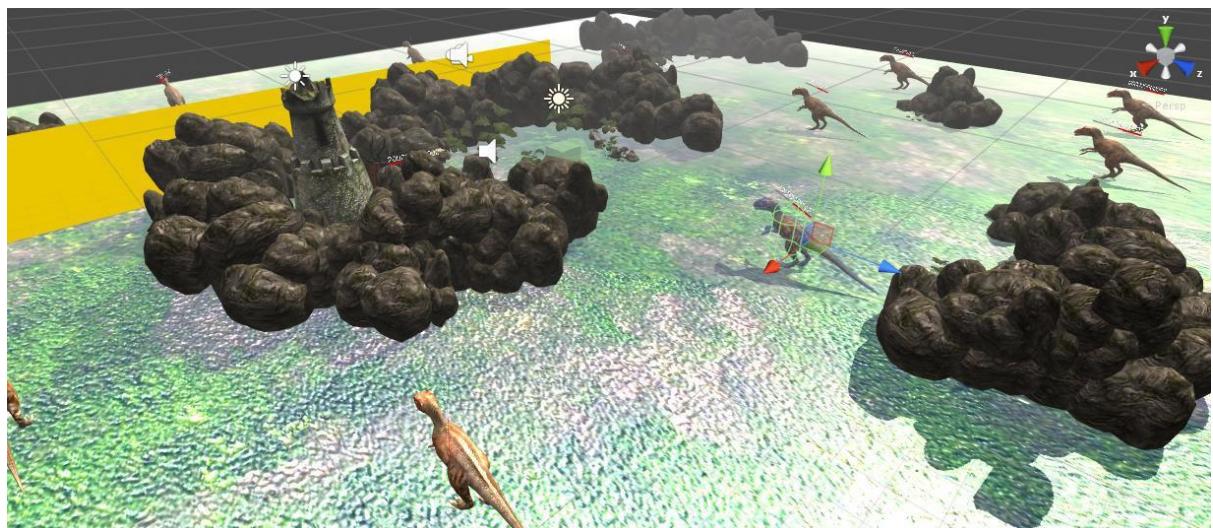
Next let's go in our scene. We should still have our prefabs from earlier here, if not, let's add them all back into the scene (our class prefab, enemy prefab and npc prefab)



Delete the class prefab, as this is spawned by RPGAIO. Next move the HappyDinosaur (our NPC) into a suitable location:



Let's duplicate our enemy and spread them around the map:



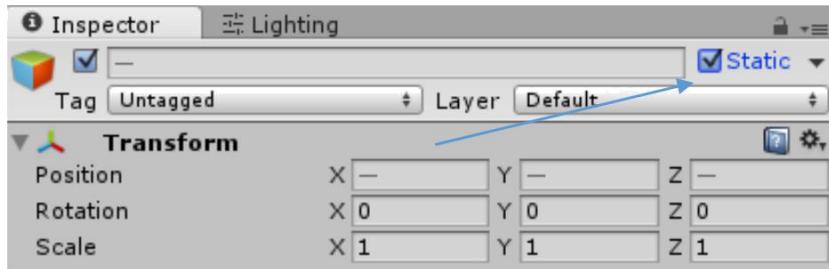
Select all static objects in the scene (things that won't move, such as the floor and rocks in this scene) and ensure they are marked as static in the inspector:

The image shows the Unity Editor's Inspector window. At the top, there is a list of selected objects: Environment, Light, Rocks, and Dinosaurv1. Below this, the Transform component is selected. In the Transform section, there is a checkbox labeled "Static". An arrow points from the text "ensure they are marked as static" in the previous step to this "Static" checkbox. The Inspector also shows Position, Rotation, and Scale values for the selected object.

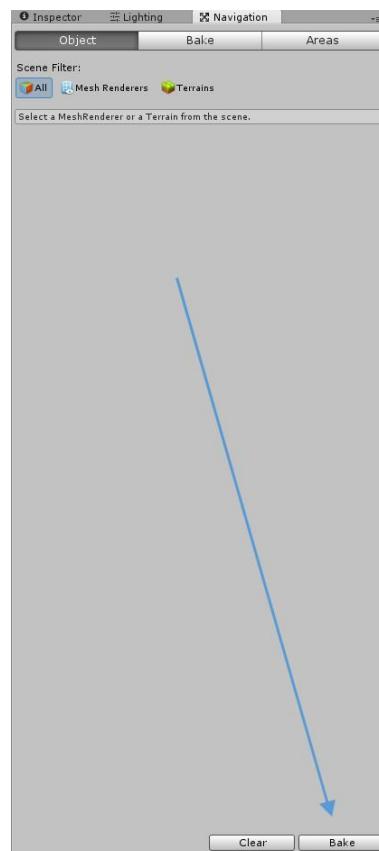
Position	Y	Z
X	-	-
X	0	0
X	1	1

Rotation	Y	Z
X	0	0

Scale	Y	Z
X	1	1



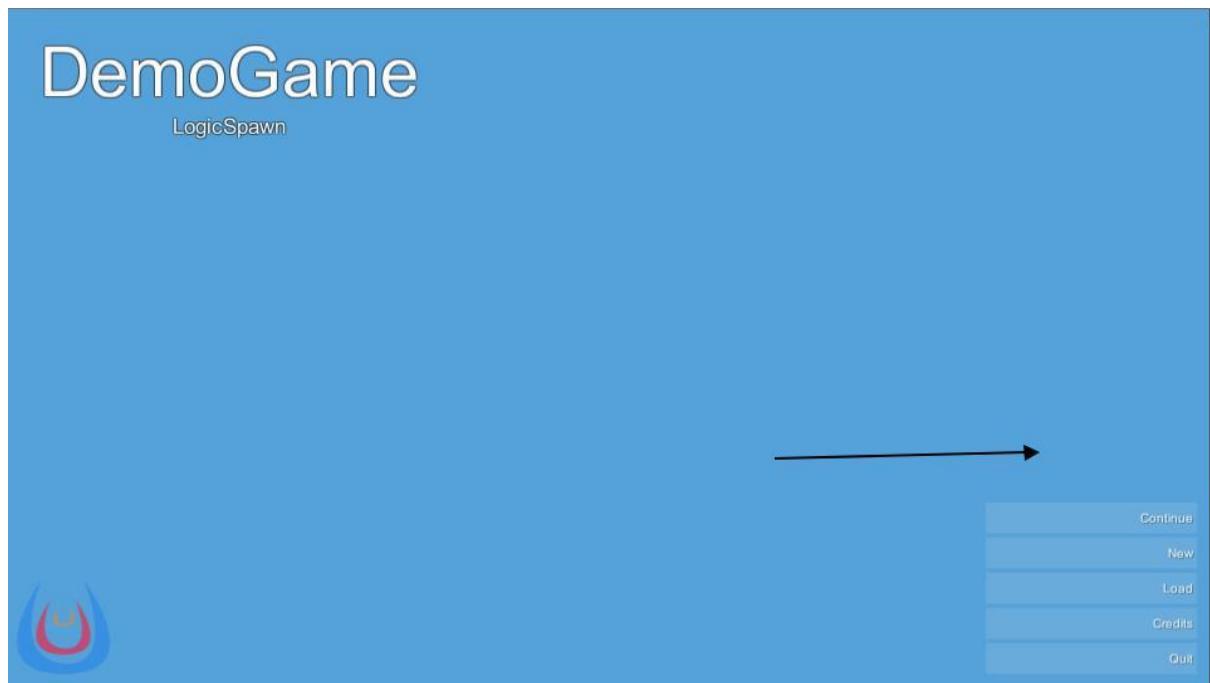
Then bake the NavMesh by accessing the Editor menu bar > Window > Navigation:



And clicking 'Bake' at the bottom right. This will give the AI the knowledge of where it can roam.

You're done! Let's test it out by starting from the main menu and creating a new 'Warlord' character.

Let's start with the main menu/intro scene, this is the default design which you are free to customise. Let's hit new.



Default character creation, pick your class (Warlord) a name and hit create.



And there we have it, we're now in the game. Once again the UI is fully customisable without coding:



Let's check the dialog without accepting the quest:



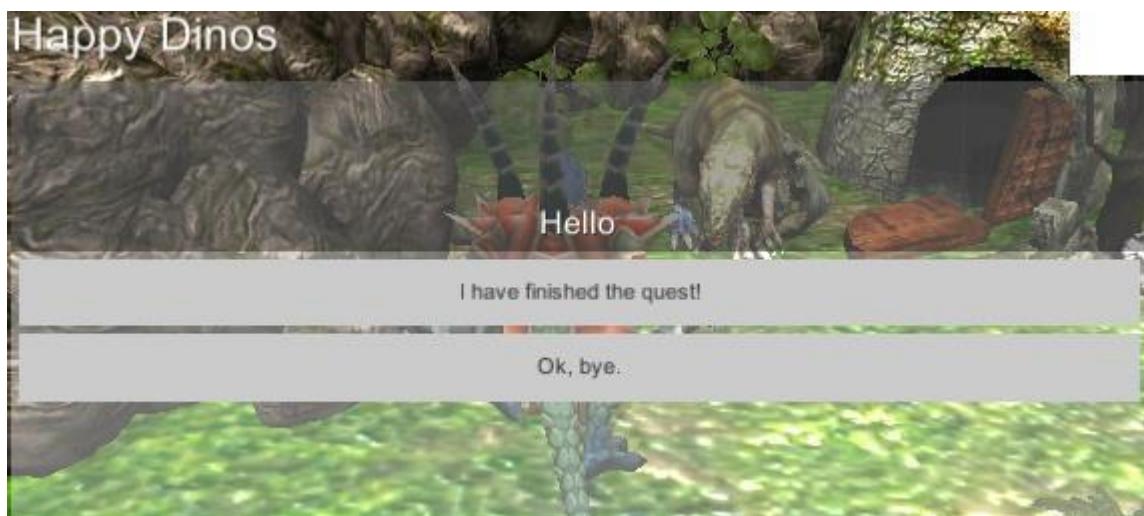
After accepting the quest:



Defeating the dinosaurs:



Dialog with the quest complete:



Quest done!



9. Where to go from here

The best thing to do from here is experiment with the customisation options available inside the RPGAIO main window. There is tons of uncovered content in this getting started guide. Customise the main menu by adding a list of credits of who has worked on your game. Add in tons of new enemy's with unique characters such as immunities to certain types of spells. Create spells for your player to use. Add harvestable crops that will give you items which you can use in crafting items. There's tons to explore inside of RPGAIO. New versions will be released often with bug fixes, optimisations and new features.

Good luck, and have fun coding for zero hours, and creating your one game