2D RPG KIT

Documentation

Preface

This is my first game-related project that is publicly available. So thank you for purchasing the 2D RPG Kit and supporting this project!

I am constantly working on improving this asset pack by adding new features and fixing bugs through future updates. I still have a lot of ideas to make this project a true RPG Kit worthy of its name. So if you have questions, ideas for more features or encounter any bugs, just contact me through artktentrion@gmail.com and there will be a good chance your feature/bug will be handled with the next update.

I am aware that this project is still not in a perfect state. I put a lot of time, effort and even more love into the 2D RPG Kit and I think even in this state it is already a very useful tool and enables you to build the RPG of your dreams. With this in mind, I released this project so we can build an awesome community and by providing your feedback I will be able to improve this asset pack over time. Please consider joining the dedicated Discord server here: https://discord.gg/Fp2K55

Please also consider following me on my social network channels in order to stay up to date with the 2D RPG Maker kit and other projects. I would also be very pleased to see how you used this tool to build your own RPG. Send me your screenshots/videos and I will share them on my channels!

Twitter
Instagram
itch.io

The background music and sound included in this project are free to use!
Credits for the background music goes to David Vitas! -> Twitter: @davidvitas

With this said, a warm welcome to the 2D RPG Kit!

Demo Game

You can try out a demo game that acts as a showcase to a lot of features that the 2D RPG Kit provides. Within unity open and play the scene “TitleScreen” from the folder “Assets/Scenes/System Maps”. Or download the demo game from itch.io. I suggest playing the demo game first in order to get a better understanding of how things look and feel.

Move: Arrow Keys, WSAD
Interact: Return
Menu: X
Cancel: Esc

Project Structure

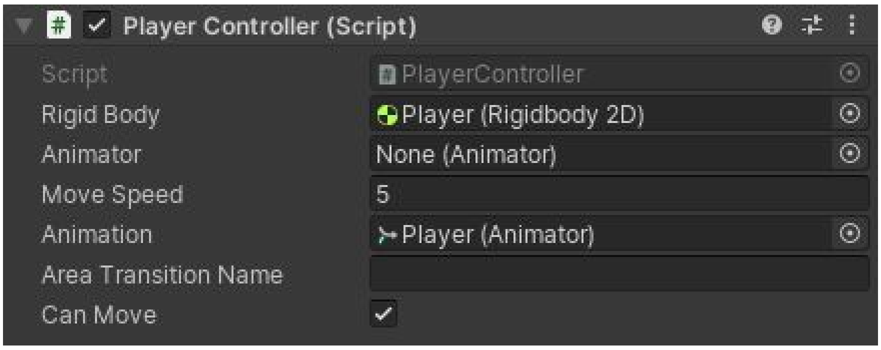
The project is structured in a way that makes the 2D RPG Kit as user-friendly as possible and is suitable for beginners as well as advanced Unity users. Once you got familiar with the features, you will be able to easily and effectively create awesome RPGs!

The folder structure is as follows:
Assets/2D RPG Kit

/Animations	This folder contains all animations including player and battle animations as well as effects. Character battle and enemy battle animations are stored individually in subfolders.
/Audio	Contains background music and sound effects.
/Fonts	Contains the font which is used for in-game texts.
/Materials	Contains a material that can be used on sprites to enable lighting.
/Prefabs/Battle	Contains all battle-related prefabs such as player characters, enemies and battle effects. You will create more characters, enemies and effects as you build your game.
/Prefabs/Items	Contains all item prefabs that can be consumed or equipped. You will create more items as you build your game.
/Prefabs/Main	Contains the main prefabs that need to be active in almost every scene to properly work. Automatic loading into scenes is already handled with the prefab “Main Objects Loader”.
/Prefabs/Objects	Contains all prefabs that can be used by a simple drop into the scene.
/Scenes/Demo	Contains all the scenes used for the demo game. You can open every scene individually and explore the contents for your reference.
/Scenes/System Maps	Contains non-playable scenes like the title screen and the game over screen.

/Scenes/World Maps	You can put all your scenes here by duplicating and editing the contained template scene.
/Scripts	Contains all scripts used by the 2D RPG Kit.
/Sprites	Contains all the graphics and UI elements, stored in individually subfolders.

Player



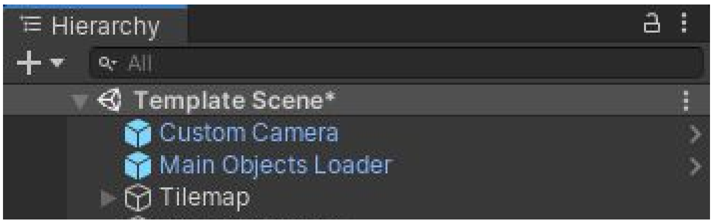
Player Controller (Script)

Script: PlayerController
Rigid Body: Player (Rigidbody 2D)
Animator: None (Animator)
Move Speed: 5
Animation: Player (Animator)
Area Transition Name:
Can Move: ✓

"Player" is a main prefab (stored in **Assets/2D RPG Kit/Prefabs/Main**). Attached to this prefab is the "Player Controller" script. There, you are able to edit the move speed through the inspector. You don't need to drag this prefab into every scene since the "Main Object Loader" already loads this prefab into the scene. During play, you can find the player prefab in the hierarchy under **DontDestroyOnLoad -> Player(Clone)**.

The player works with four directional animations for walking and idling which are set up through Unity's Blend Tree. The animation is stored in **Assets/2D RPG Kit/Animations/Player**. You can easily swap out the sprites in **Assets/2D RPG Kit/Sprites/Characters/Character1** to create your own player character.

Scenes

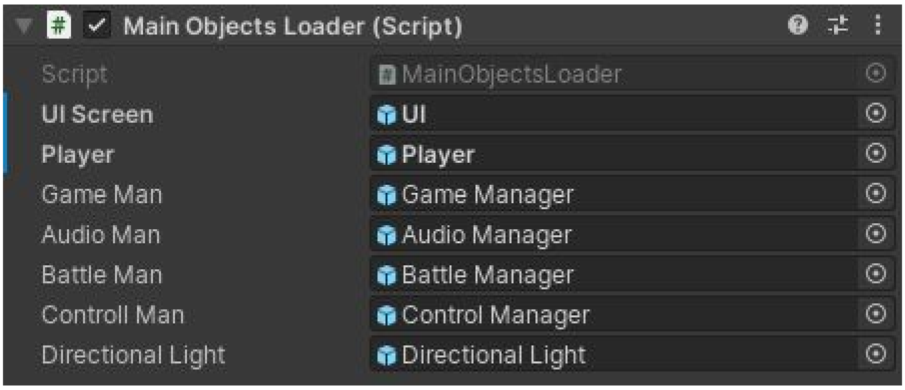


Hierarchy
All
Template Scene*
Custom Camera
Main Objects Loader
Tilemap

Use the template scene stored in **Assets/2D RPG/Scene/World Maps** to create your own scenes. Duplicate the "Template Scene" (STRG+D) and open it. The scene already contains the "Custom Camera", "Main Objects Loader" and a tilemap component to be able to be played.

Main Objects Loader

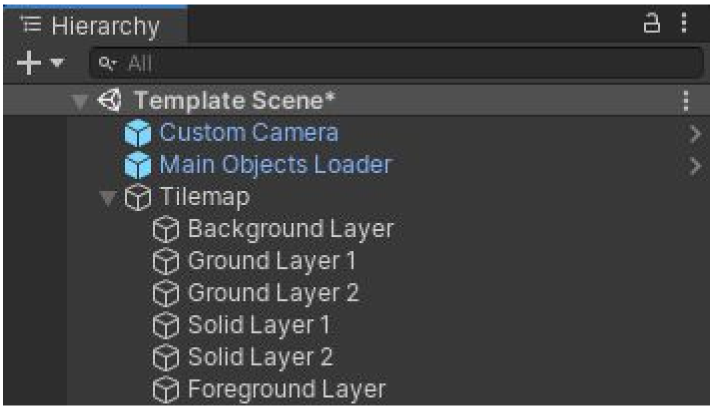
The "Main Objects Loader" is another main prefab (stored in **Assets/2D RPG Kit/Prefabs/Main**) and is essential for every playable scene since it loads in all needed prefabs automatically. Jump to the respective chapter to find out more about each prefab loaded with the "Main Objects Loader".



Main Objects Loader (Script)

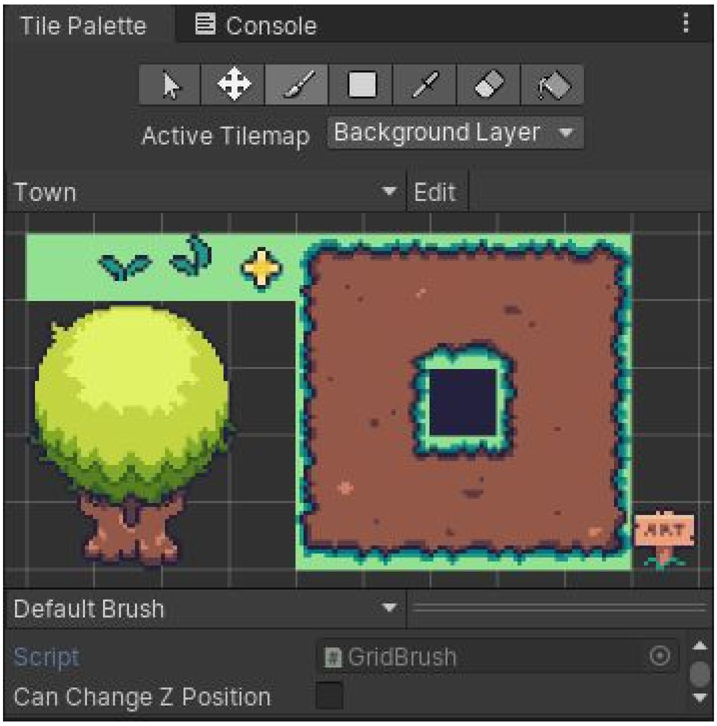
Script	MainObjectsLoader
UI Screen	UI
Player	Player
Game Man	Game Manager
Audio Man	Audio Manager
Battle Man	Battle Manager
Controll Man	Control Manager
Directional Light	Directional Light

Tilemaps



Hierarchy

- Template Scene*
  - Custom Camera
  - Main Objects Loader
  - Tilemap
    - Background Layer
    - Ground Layer 1
    - Ground Layer 2
    - Solid Layer 1
    - Solid Layer 2
    - Foreground Layer



Active Tilemap: Background Layer

Town

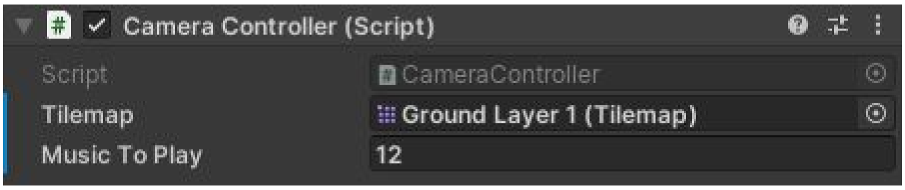
Default Brush

Script	GridBrush
Can Change Z Position	□

Scenes are built with tilemaps. To build your own environments, expand the “Tilemap” component in the hierarchy of your scene. There, you will find a number of layers. Use the layers to build maps with differently overlapping objects. Choose one of the layers from the hierarchy and select a tile with the brush tool from the tile pallet. You can then draw with this tile over the map in the scene view. The player sprite will be rendered between “Ground Layer 2” and “Solid Layer 1” -> Everything on “Ground Layer 2” and below is rendered below the player / Everything on “Solid Layer 1” and above, is rendered above the player.

Camera

The Camera in this project (stored in **Assets/2D RPG Kit/Prefabs/Main**) is a slight alternation to Unity’s default camera component but is still needed for each scene in order to render the graphics. Attached to this prefab is the “Camera Controller” script. There, you can assign the tilemap of the current scene which is important to calculate the boundaries. You don’t have to assign the tilemap to the camera if you are working with the template scene. If you are not using the template scene, simply drag and drop a layer from your scenes tilemap into “Tilemap” of the “Camera Controller” script. This will calculate the borders of your map and will prevent the player from going out of bounds. You can also set the background music of your scene here. Simply enter the number corresponding to the music would like to play in this scene (see “Audio Manager” for more info).

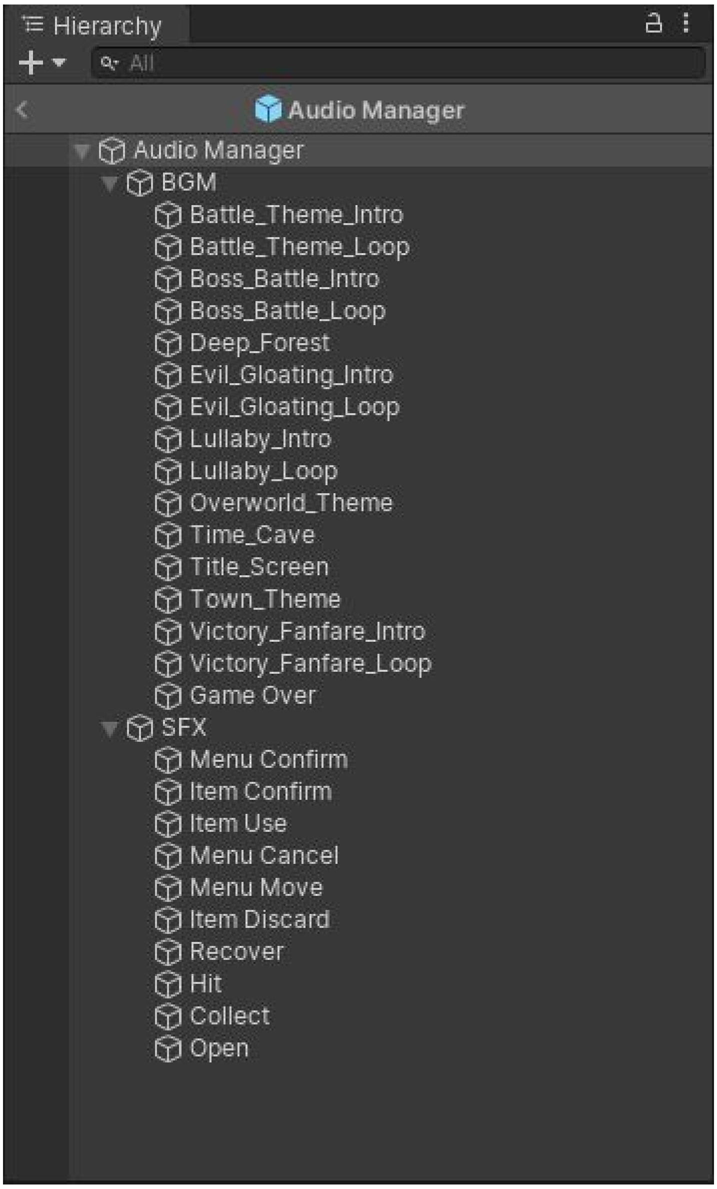

Camera Controller (Script)

Script: CameraController
Tilemap: Ground Layer 1 (Tilemap)
Music To Play: 12

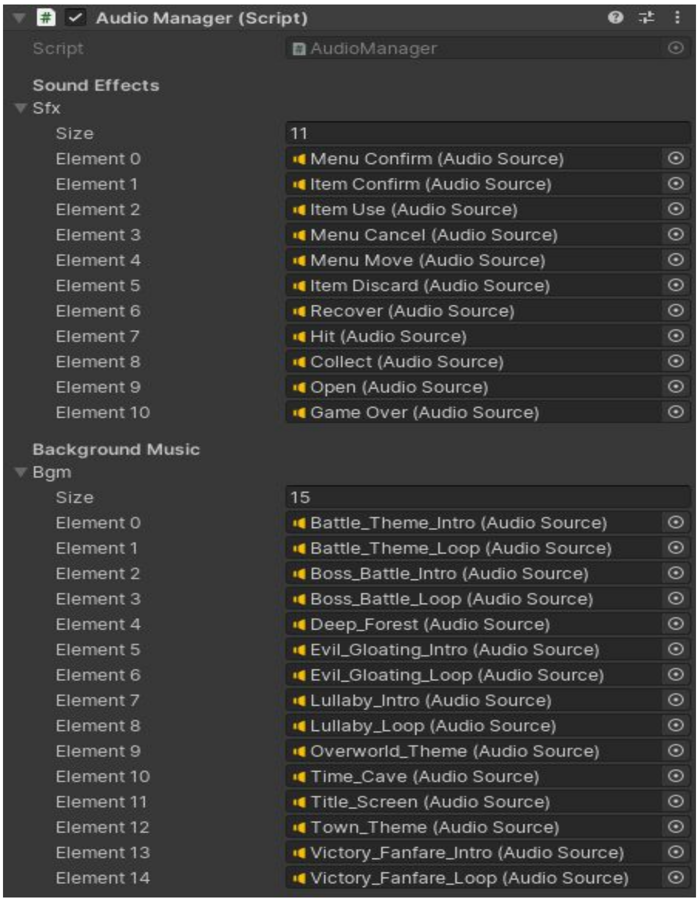
Audio Manager

“Audio Manager” is a main prefab (stored in **Assets/2D RPG Kit/Prefabs/Main**). This prefab loads in the music and sound data used in your project. A selection of free to use sounds and music comes with this project and is stored in **Assets/2D RPG Kit/Audio**. You don’t need to drag this prefab into every scene since the “Main Object Loader” already loads this prefab into the scene. If you want to add more music/sounds to your project, simply copy your music files into the respective folders within the Audio directory. From there, drag and drop the audio files directly into the hierarchy of the “Audio Manager”. Make the music a child of “BGM” and sounds a child of “SFX”.

Finally, drag and drop the audio files from the hierarchy into the “Audio Manager” script within the inspector. Again, music goes into “BGM” and sound goes into “SFX”. The element number next to each sound/music is how you can reference the audio files. So if you want to play the overworld theme in your scene, enter a “9” into “music to play” of your scene’s camera component. During play, you can find the audio manager prefab in the hierarchy under **DontDestroyOnLoad -> Audio Manager(Clone)**.



```
Audio Manager
  BGM
    Battle_Theme_Intro
    Battle_Theme_Loop
    Boss_Battle_Intro
    Boss_Battle_Loop
    Deep_Forest
    Evil_Gloating_Intro
    Evil_Gloating_Loop
    Lullaby_Intro
    Lullaby_Loop
    Overworld_Theme
    Time_Cave
    Title_Screen
    Town_Theme
    Victory_Fanfare_Intro
    Victory_Fanfare_Loop
    Game Over
  SFX
    Menu Confirm
    Item Confirm
    Item Use
    Menu Cancel
    Menu Move
    Item Discard
    Recover
    Hit
    Collect
    Open
```



Audio Manager (Script)

Script AudioManager

Sound Effects

Sfx

Size	11
Element 0	Menu Confirm (Audio Source)
Element 1	Item Confirm (Audio Source)
Element 2	Item Use (Audio Source)
Element 3	Menu Cancel (Audio Source)
Element 4	Menu Move (Audio Source)
Element 5	Item Discard (Audio Source)
Element 6	Recover (Audio Source)
Element 7	Hit (Audio Source)
Element 8	Collect (Audio Source)
Element 9	Open (Audio Source)
Element 10	Game Over (Audio Source)

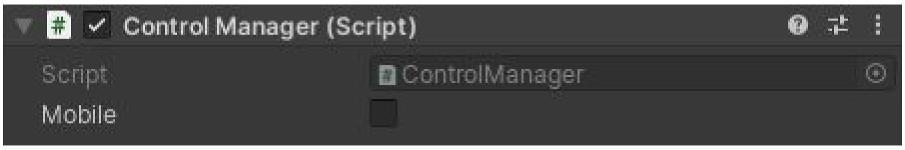
Background Music

Bgm

Size	15
Element 0	Battle_Theme_Intro (Audio Source)
Element 1	Battle_Theme_Loop (Audio Source)
Element 2	Boss_Battle_Intro (Audio Source)
Element 3	Boss_Battle_Loop (Audio Source)
Element 4	Deep_Forest (Audio Source)
Element 5	Evil_Gloating_Intro (Audio Source)
Element 6	Evil_Gloating_Loop (Audio Source)
Element 7	Lullaby_Intro (Audio Source)
Element 8	Lullaby_Loop (Audio Source)
Element 9	Overworld_Theme (Audio Source)
Element 10	Time_Cave (Audio Source)
Element 11	Title_Screen (Audio Source)
Element 12	Town_Theme (Audio Source)
Element 13	Victory_Fanfare_Intro (Audio Source)
Element 14	Victory_Fanfare_Loop (Audio Source)

Control Manager

The “Control Manager” can make your game playable on mobiles via touchscreen input and is also a main prefab (stored in **Assets/2D RPG Kit/Prefabs/Main**). Though the mobile feature is still in a very early stage and has a few bugs. To test the touch input, just tick the “mobile” checkbox directly in the prefab’s inspector before playing. This will display a touch interface that you can test using the mouse. Gamepad/keyboard input will be disabled when the “mobile” checkbox is checked. You will also need the “Standard Asset Pack” which you can get from the Unity Asset Store for free in order for the touch input to work. Also, don’t forget to switch to the Android target platform in your build settings and enable mobile input via “Mobile Input” -> ”Enable” in the window navigation bar!



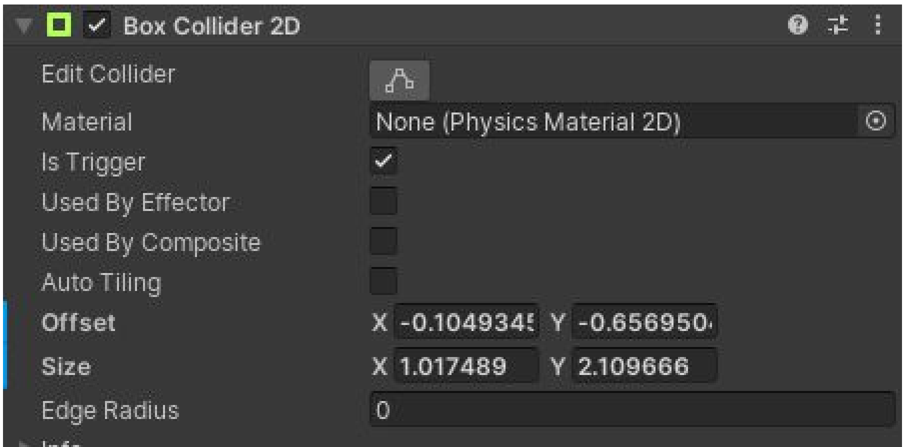
Control Manager (Script)

Script ControlManager

Mobile □

Scene Transition

To teleport between scenes, you can use the “Teleport to” prefab (stored in **Assets/2D RPG Kit/Prefabs/Objects**). This prefab connects two scenes and defines an exit and entry point. Drag and drop this prefab into your scene and place it where you want to create an entrance to another scene. The “Box Collider 2D” component attached to this



Box Collider 2D

Edit Collider

Material None (Physics Material 2D)

Is Trigger ☑

Used By Effector □

Used By Composite □

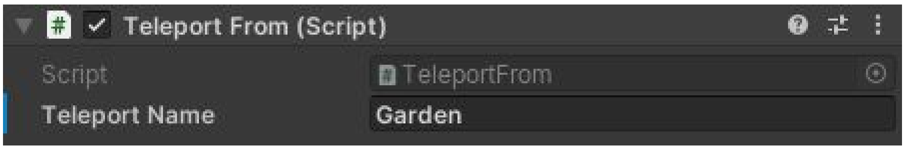
Auto Tiling □

Offset X -0.1049345 Y -0.656950

Size X 1.017489 Y 2.109666

Edge Radius 0

prefab, acts as the trigger for the scene transition. You can resize the collider to your needs by clicking the “Edit Collider” button within the inspector. Below the “Box Collider 2D” component you will find the “Teleport To” script. There, under “Scene”, you can enter the name of the scene where you wish to teleport to. Under “Teleport Name”, you can assign a unique teleport ID. Next, expand the “Teleport To” prefab in the scene’s hierarchy and select the “Teleport From” component. Within the inspector, under “Teleport Name” enter the same teleport ID as before. In your scene, move the “Teleport From” object to where you want to have the entry point from another scene. Keep in mind that the “Teleport From” object should not overlap with the “Box Collider 2D” of the “Teleport To” object. Now open the scene you wish to teleport to and repeat the steps. Again, use the same teleport ID from the previous scene.


Teleport From (Script)

Script TeleportFrom

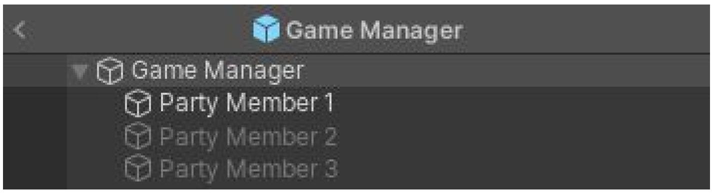
Teleport Name Garden

Game Manager

The “Game Manager” is another main prefab (stored in **Assets/2D RPG Kit/Prefabs/Main**). This prefab handles important values regarding the player stats and some world settings. Additionally, this prefab handles the management of chests, quests and events (see “Chest /Chest Manager, “Quest Manager”, “Event Manager” for more info). You don’t need to drag this prefab into every scene since the “Main Object Loader” already loads this prefab into the scene. During play, you can find the game manager prefab in the hierarchy under **DontDestroyOnLoad -> Game Manager(Clone)**. For this section of the documentation, we will focus on the “Game Manager” script component.

Initialization

Here, you will find Character 0 - 2 and the character status list. These objects are the characters that can join the player’s party during the game and are child objects of the “Game Manager” prefab (see “Character Status” for more info).


Game Manager

Game Manager

Party Member 1

Party Member 2

Party Member 3

Currently Active Menus

Here, you can confirm which menus are active at any time for debugging purposes.

Character Bools

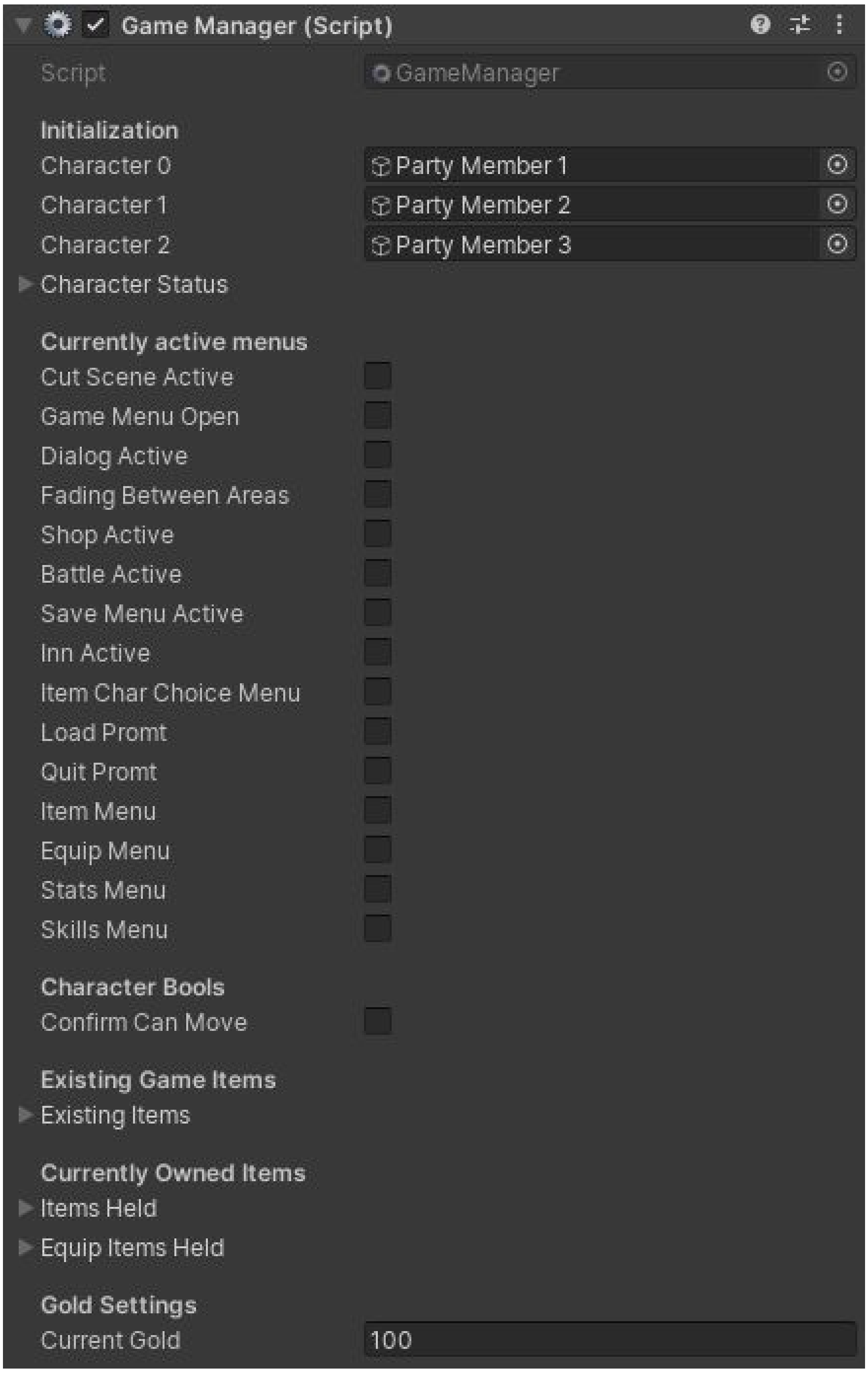
Here, you can check if anything is blocking the player movement at any time for debugging purposes.

Existing Game Items

This is a list of every item and equipment that will be available in your game. Items are stored as prefabs in **Assets/2D RPG Kit/Prefabs/Items** (see “Items” for more info). Just drag and drop the prefab into the item list.

Currently Owned Items

Here you can confirm during play which items/equipment the player is currently holding for debugging purposes. You can also enter an item or equipment before playing/compiling to make the player start out with specific items/equipment (see “Items” for more info). Items/equipment are entered by hand so you don’t need to drag and drop prefabs into the list. Note that the item/equipment name has to be spelled correctly in order for the game manager to recognize the referenced item/equipment.



Game Manager (Script)

Script: GameManager

Initialization

Character 0: Party Member 1

Character 1: Party Member 2

Character 2: Party Member 3

▶ Character Status

Currently active menus

Cut Scene Active ☐

Game Menu Open ☐

Dialog Active ☐

Fading Between Areas ☐

Shop Active ☐

Battle Active ☐

Save Menu Active ☐

Inn Active ☐

Item Char Choice Menu ☐

Load Promt ☐

Quit Promt ☐

Item Menu ☐

Equip Menu ☐

Stats Menu ☐

Skills Menu ☐

Character Bools

Confirm Can Move ☐

Existing Game Items

▶ Existing Items

Currently Owned Items

▶ Items Held

▶ Equip Items Held

Gold Settings

Current Gold: 100

Character Status

With the “Character Status” script it is possible to create customized playable characters by giving them different strengths and weaknesses. This project contains three predefined characters. they are stored as child objects of the “Game Manager” prefab (stored in **Assets/2D RPG Kit/Prefabs/Main**). The “Party Member 1” object always refers to the main character and has to be always active for the battles to work. The other to child objects refer to the characters that can join the player during the game. By activating one or two of the other party member objects before playing/compiling, you can unlock the other characters for the player directly from the beginning. Use the NPC prefab if you want a character to join the player at a specific point of your game (see “NPC” for more info).

By entering the following attributes in the “Character Status” component, you can create customized characters:

Attribute	**Description**

Character Name	Enter the name of the character (will be displayed in the game menu)
Level	Enter the level this character should start with
Max Level	Enter the max amount of levels this character can reach
Current HP	Enter the amount of HP this character should start with
Max HP	Enter the max amount of HP this character can have with the level entered above
Current SP	Enter the amount of SP this character should start with
Max SP	Enter the max amount of SP this character can have with the level entered above
Current EXP	Enter the amount of EXP this character should start with
Skills	Enter the skills this character should start with (see “Skills” for more info)
First Next Level EXP	If you wish the amount of EXP for each next level to be calculated automatically, enter the amount of EXP needed to reach the fist next level here. The amount of EXP for each level will then be calculated automatically using the provided multiplication factor
Multiplication Factor	Enter the multiplication factor used for the automatic calculation of EXP for each level
Manual EXP	Check this box if you like to enter the needed amount of EXP for each level by hand
EXP to Next Level	If you checked the “Manual EXP” box, enter a list of needed EXP to reach each level here. Note that the list should be of the same length as the max level this character can reach.
Manual HP Bonus	Check this box if you like to enter the HP bonus granted fo reaching each level by hand
HP Level Bonus	If you checked the “Manual HP Bonus” box, enter a list of HP bonuses for each level here. These values will be then added to the max HP whenever a level was reached. Note that the list should be of the same length as the max level this character can reach.
Manual SP Bonus	Check this box if you like to enter the SP bonus granted fo reaching each level by hand
SP Level Bonus	If you checked the “Manual SP Bonus” box, enter a list of SP bonuses for each level here. These values will be then added to the max SP whenever a level was reached. Note that the list should be of the same length as the max level this character can reach.

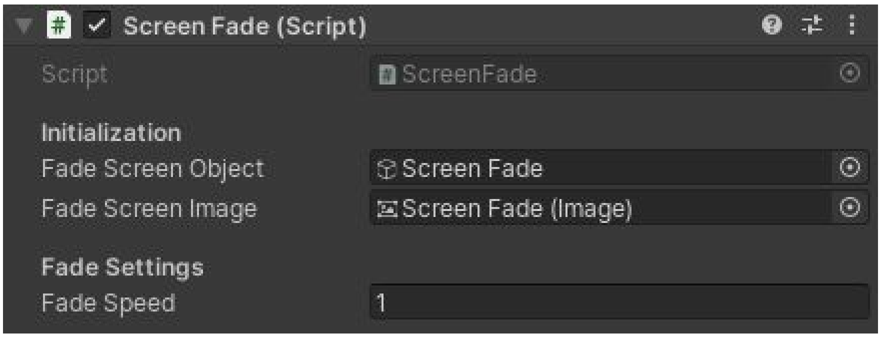
Strength	Enter the strength this character should start with. This value will be used to calculate the damage dealt to the enemy during battle
Defense	Enter the defense this character should start with. This value will be used to calculate the damage dealt by the enemy during battle
Offense Strength	You can leave this value at 0 since it will be defined by the weapon this character is equipped with. You can check this value during play for debugging purposes.
Defense Strength	You can leave this value at 0 since it will be defined by the Armor this character is equipped with. You can check this value during play for debugging purposes.
Equipped Offense Item	Enter the equipment this character should start with or leave blank. Can be used to confirm equipment during play for debugging purposes
Equipped Defense Item	Enter the equipment this character should start with or leave blank. Can be used to confirm equipment during play for debugging purposes
Character Image	Drag and drop the character sprite used as the portrait in the game menu. Character images are stored in **Assets/2D RPG Kit/Sprites/Characters**

UI

The “UI” is another main prefab (stored in **Assets/2D RPG Kit/Prefabs/Main**). This prefab handles the complete user interface with all its’ menus. You don’t need to drag this prefab into every scene since the “Main Object Loader” already loads this prefab into the scene. During play, you can find the “UI” prefab in the hierarchy under **DontDestroyOnLoad -> UI(Clone)**.

Screen Fade

The screen fade script controls the “Screen Fade” child object of the UI prefab and is utilized by several other scripts in order to activate a fade out and fade in of the screen in situations like going from one scene to another. The fade speed is adjustable.



Screen Fade (Script)

Script: ScreenFade

Initialization

Fade Screen Object: Screen Fade

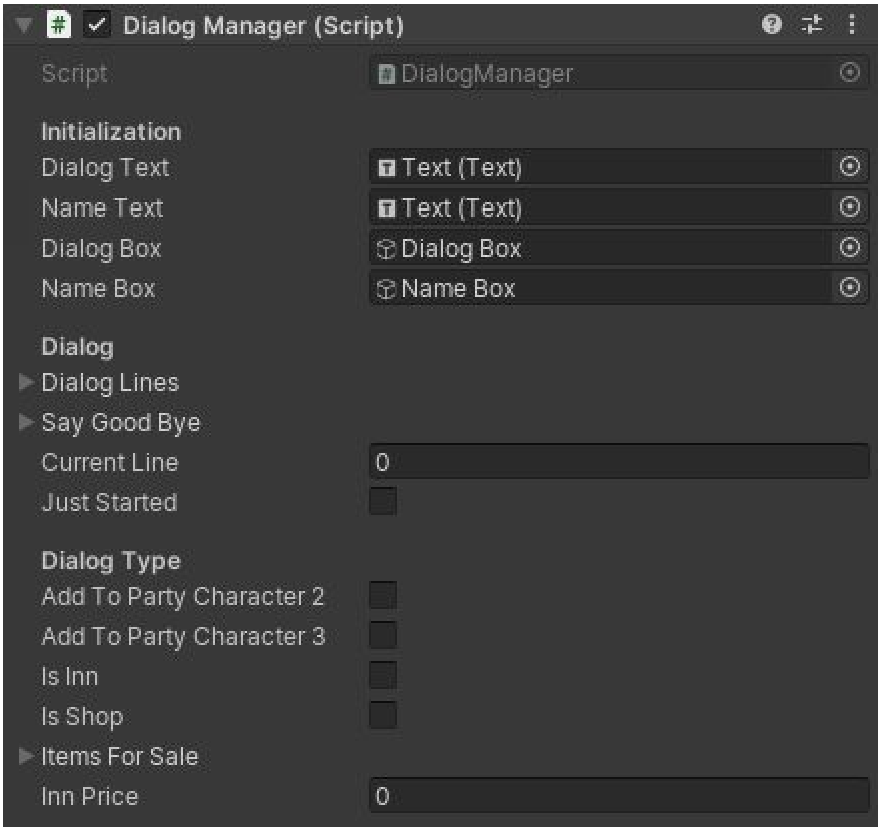
Fade Screen Image: Screen Fade (Image)

Fade Settings

Fade Speed: 1

Dialog Manager

The “Dialog Manager” script controls the “Dialog Box” child object of the UI prefab and can display two lines of dialog as well as a name. You can resize the dialog box and the text component if you want to be able to display more text at once. This script only displays the dialog lines specified via the “Dialog Starter” script and progresses the dialog by pressing the confirm button (see NPC for more info). Here, you can confirm the currently displayed dialog lines and several other settings during play for debugging purposes.



Dialog Manager (Script)

Script: DialogManager

Initialization

Dialog Text: Text (Text)

Name Text: Text (Text)

Dialog Box: Dialog Box

Name Box: Name Box

Dialog

▶ Dialog Lines

▶ Say Good Bye

Current Line: 0

Just Started: □

Dialog Type

Add To Party Character 2: □

Add To Party Character 3: □

Is Inn: □

Is Shop: □

▶ Items For Sale

Inn Price: 0

Game Menu / Shop / Inn / Reward Screen / Save

These scripts are responsible for showing the correct data within the game menus. There are a lot of objects and the scripts currently look a bit wild but you can check each connection by clicking the objects within the inspector. The menu is set up in a way that it can display up to three party members and a total amount of ten items and another ten pieces of equipment. But, since these objects were saved within lists, it is possible to expand them by duplicating the objects and assigning them into their respective lists. The code automatically adjusts to the new objects.

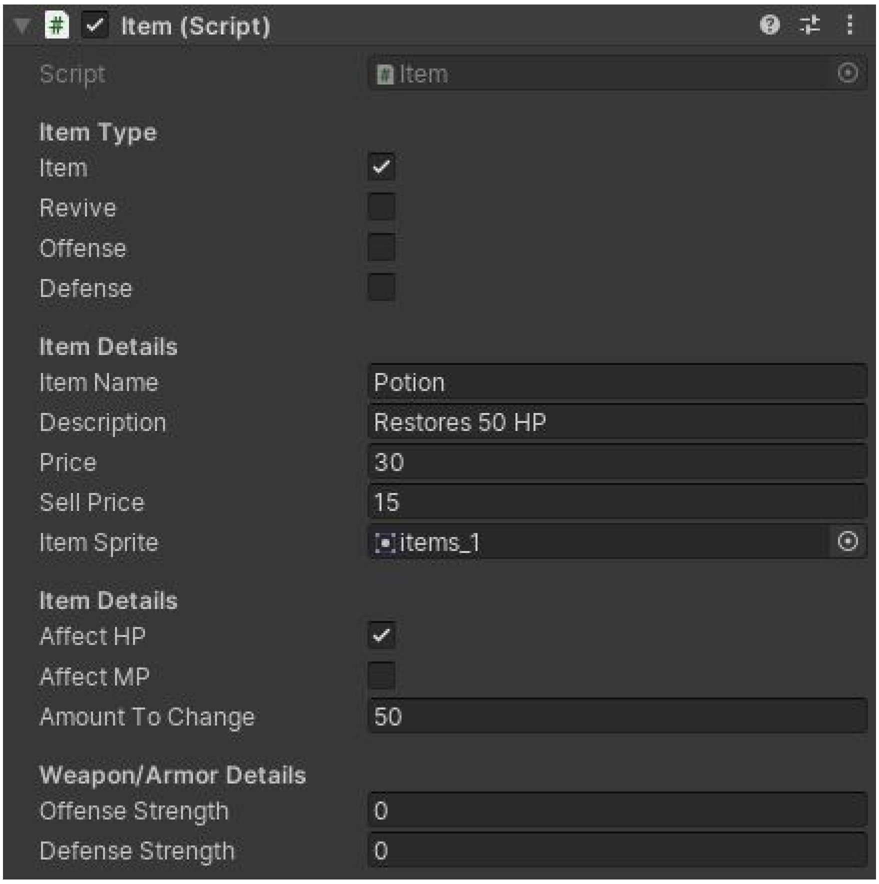
Items

Items are stored as prefabs in **Assets/2D RPG Kit/Prefabs/Items**. Here, you can find three predefined consumable items and two predefined equipable items. In order to create another item, you can just duplicate an existing item and change the values of the “Item” script within the inspector. Every item has to be registered in the “Game Manager” script under “Existing Items” of the “Game Manager” prefab in order to be usable in the game.

The “Item” script contains the following attributes to create customized items:

Attribute	**Description**
Item	Check this box if the item is a consumable. Leave blank if it is an equipable item

Revive	If you declared this item as a consumable, please decide if it also should be able to revive fallen members by checking the box
Offense	If you declared this item as equipable, you can choose it to be equipped as a weapon by checking this box
Defense	If you declared this item as equipable, you can choose it to be equipped as armor by checking this box
Item Name	Give this item a unique name. Give this prefab the same name within its hierarchy
Description	Enter a description to be displayed within menus
Price	Enter a price for which this item should be available in shops
Sell Price	Enter a price for which this item can be sold to shops
Item sprite	Drag and drop the item sprite used as the image in the game menu. item sprites are stored in **Assets/2D RPG Kit/Sprites/Objects**
Affect HP	If you declared this item as a consumable, please decide if it should affect HP by checking the box
Affect MP	If you declared this item as a consumable, please decide if it should affect MP by checking the box
Amount To Change	If this item affect HP/MP, enter the amount to heal
Offense Strength	If you declared this item as equipable, enter the amount of weapon strength
Defense Strength	If you declared this item as equipable, enter the amount of armor strength



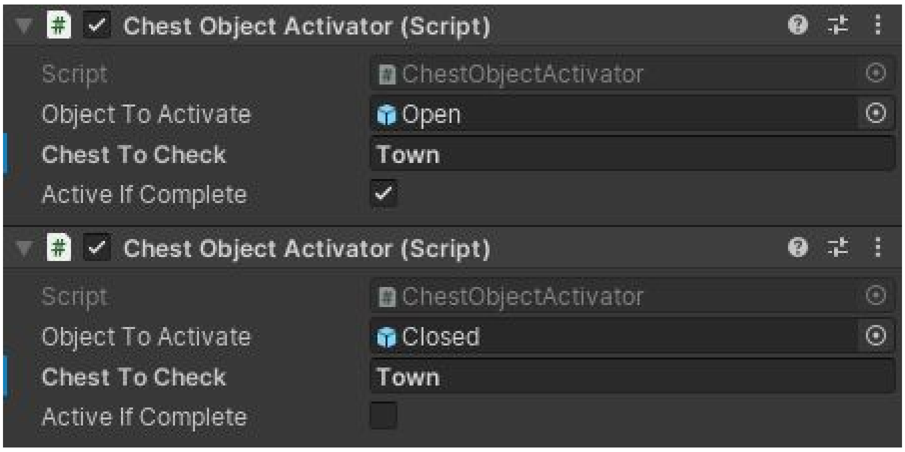
Item (Script)

Script	Item
Item Type	
Item	☑
Revive	□
Offense	□
Defense	□
Item Details	
Item Name	Potion
Description	Restores 50 HP
Price	30
Sell Price	15
Item Sprite	items_1
Item Details	
Affect HP	☑
Affect MP	□
Amount To Change	50
Weapon/Armor Details	
Offense Strength	0
Defense Strength	0

Chest / Chest Manager

The “Chest” is an object prefab (stored in **Assets/2D RPG Kit/Prefabs/Objects**). It can be dragged and dropped directly into the scene. A chest will give the player an item and the “Chest Manager” marks every opened chest as complete within the “Game Manager. This means that opened chests stay open even after leaving and re-entering a scene and also when restarting the game and loading the save. In order for this to work correctly, you have to assign a unique name to the chest under “Chest To Check” within the two “Chest Object Activator” components of the “Chest” object and register the chest within the “Chest Manager” Component of the “Game Manager” prefab.

Chest Prefab:



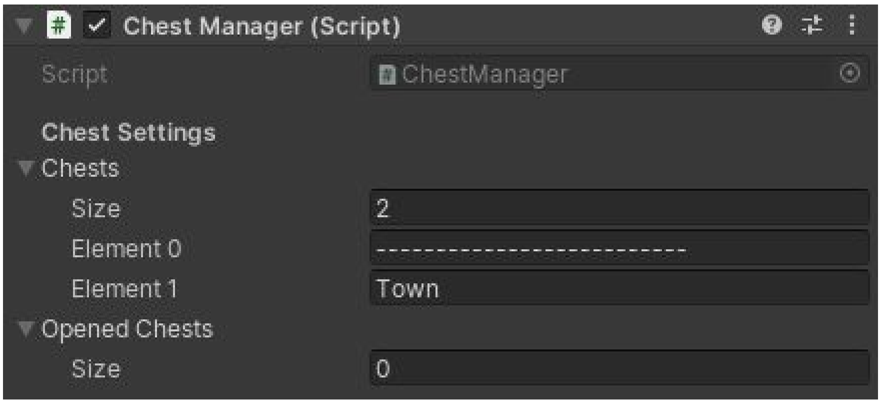
Chest Object Activator (Script)

Script	ChestObjectActivator
Object To Activate	Open
Chest To Check	**Town**
Active If Complete	☑

Chest Object Activator (Script)

Script	ChestObjectActivator
Object To Activate	Closed
Chest To Check	**Town**
Active If Complete	□

Game Manager Prefab:



Chest Manager (Script)

Script: ChestManager

Chest Settings

Chests

Size: 2

Element 0: -------------------------

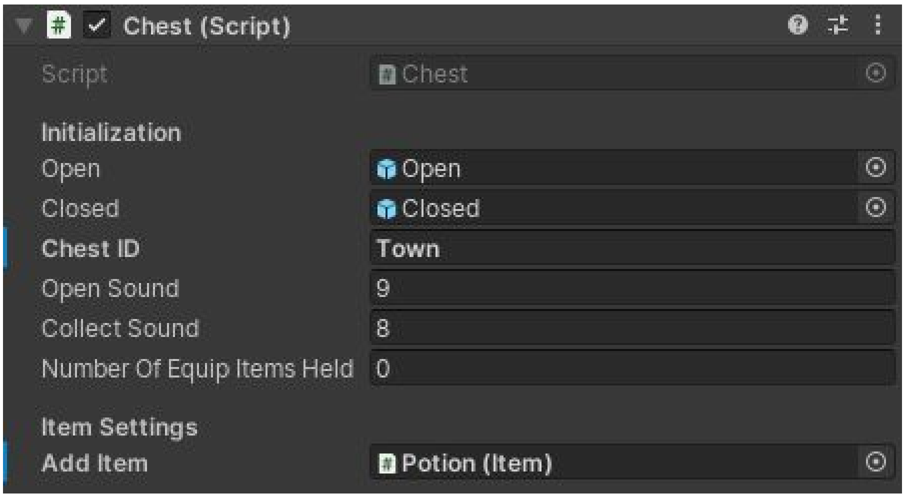
Element 1: Town

Opened Chests

Size: 0

Don’t use “Element 0” to register chests within the manager

After setting up the “Chest Manager” correctly, expand the “Chest” component of the prefab within its hierarchy and select the “Chest States” component.
Here, again enter the chest Name under “Chest ID”, select a sound for opening the chest and for collecting the item. Lastly, drag the item prefab that this chest should contain into “Add Item”.



Chest (Script)

Script: Chest

Initialization

Open: Open

Closed: Closed

Chest ID: Town

Open Sound: 9

Collect Sound: 8

Number Of Equip Items Held: 0

Item Settings

Add Item: Potion (Item)

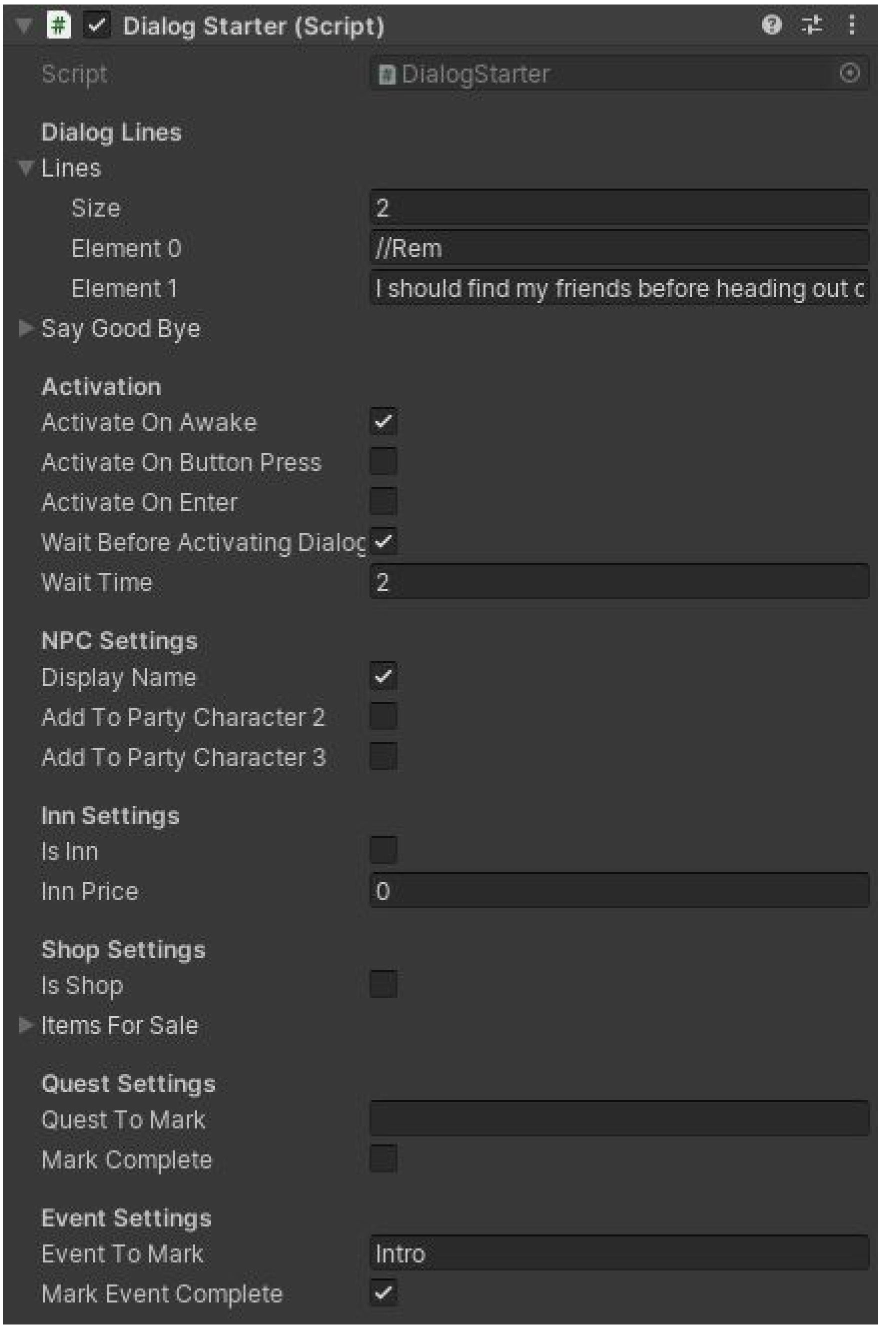
Shop / Inn / NPC

Stored in **Assets/2D RPG Kit/Prefabs/Objects**, you will find the “Shop Keeper”, “Inn Keeper” and the “NPC” prefabs. They can also be dragged and dropped directly into the scene. Even though all three prefabs are made of the same components (2x Box Collider 2D and “Dialog Starter” script), they are behaving very differently in the game. Approaching them and pressing the “Interact” button, opens either a dialog, a shop or an inn menu. They all utilize two Box Collider 2D components (one trigger and one collider). The other component those three prefabs use is the “Dialog Starter” script. Depending on how you use the options within this script, you will get different results. To make it easier, I included these three predefined prefabs to the project.

The “Dialog Starter” script contains the following attributes:

Attribute	**Description**
Lines	Enter the dialog line this NPC should say. Shops and inns will have a welcome dialog. Entering ”//” at the beginning of a line puts the following text into the name area of the dialog box.
Say Good Bye	Shops and inns will have a goodbye dialog when exiting them. This works the same as the dialog lines
Activate On Awake	Checking this box will start the defined dialog as soon

	as this object was activated
Activate On Button Press	Checking this box will start the defined dialog when the player stays within the trigger zone of the Box Collider 2D component and presses the “Interact” button
Activate On Enter	Checking this box will start the defined dialog as soon as the player enters the trigger zone of the Box Collider 2D component
Wait Before Activating Dialog	Checking this box will start the defined dialog with a delay. Useful for cutscenes
Wait Time	Enter the number of seconds the dialog should be delayed by
Display Name	Check this box if you want to show the name box within the dialog box
Add to Party Character 2	Lets the character “Party Member 2” join players party (child object of “Game Manager” prefab)
Add to Party Character 3	Lets the character “Party Member 3” join players party (child object of “Game Manager” prefab)
Is Inn	Check this box if this NPC should be an inn keeper (opens inn menu)
Inn Price	If this NPC is an inn keeper, enter the price for one night at this inn
Is Shop	Check this box if this NPC should be a shop keeper (opens inn menu)
Items for sale	If this NPC is a shop keeper, enter a list of items this shop should sell
Quest To Mark	If the completion of this dialog should complete a quest, enter the name of the quest here (see “Quests / Quest Manager” for more info)
Mark Complete	Check this box to complete the quest named above
Event To Mark	If the completion of this dialog should complete an event, enter the name of the event here (see “Event Manager” for more info)
Mark Event Complete	Check this box to complete the event named above



☑ Dialog Starter (Script)

Script: DialogStarter

Dialog Lines

Lines
Size: 2
Element 0: //Rem
Element 1: I should find my friends before heading out c

Say Good Bye

Activation

Activate On Awake ☑
Activate On Button Press ☐
Activate On Enter ☐
Wait Before Activating Dialog ☑
Wait Time: 2

NPC Settings

Display Name ☑
Add To Party Character 2 ☐
Add To Party Character 3 ☐

Inn Settings

Is Inn ☐
Inn Price: 0

Shop Settings

Is Shop ☐
Items For Sale

Quest Settings

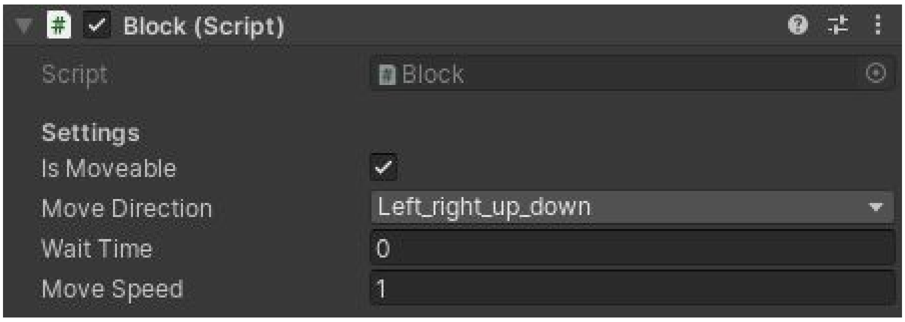
Quest To Mark:
Mark Complete ☐

Event Settings

Event To Mark: Intro
Mark Event Complete ☑

Pushable Block

The “Pushable Block” is an object prefab (stored in **Assets/2D RPG Kit/Prefabs/Objects**). It can be dragged and dropped directly into the scene to create blocks that the player can push around by walking into them. Under “Move Direction” you can choose in which direction this block should be movable. With this, you can lock the block into certain axes. With “Wait Time” you can decide how long the player has to walk against the block until it moves.



☑ Block (Script)

Script: Block

Settings

Is Moveable ☑
Move Direction: Left_right_up_down
Wait Time: 0
Move Speed: 1

Battle Manager

The “Battle Manager” is another main prefab (stored in **Assets/2D RPG Kit/Prefabs/Main**). This prefab handles important objects regarding the battle menu and other battle settings. You don’t need to drag this prefab into every scene since the “Main Object Loader” already loads this prefab into the scene. During play, you can find the game manager prefab in the hierarchy under **DontDestroyOnLoad -> Battle Manager(Clone)**. Battles can be set up to occur randomly or on collision with other objects (see “Battle Area” for more info). Battles are turn-based and it is randomly decided who attacks first. The player can attack normally, use skills and items and retreat from battle. The battle menu is set up in a way that up to three characters and up to four enemies can participate. It is possible to increase these numbers by duplicating UI elements and assigning them to their dedicated lists within the inspector of the “Battle Manager” script. You also need to register newly created skills and enemies here (see “Skills” and “Enemies” for more info).

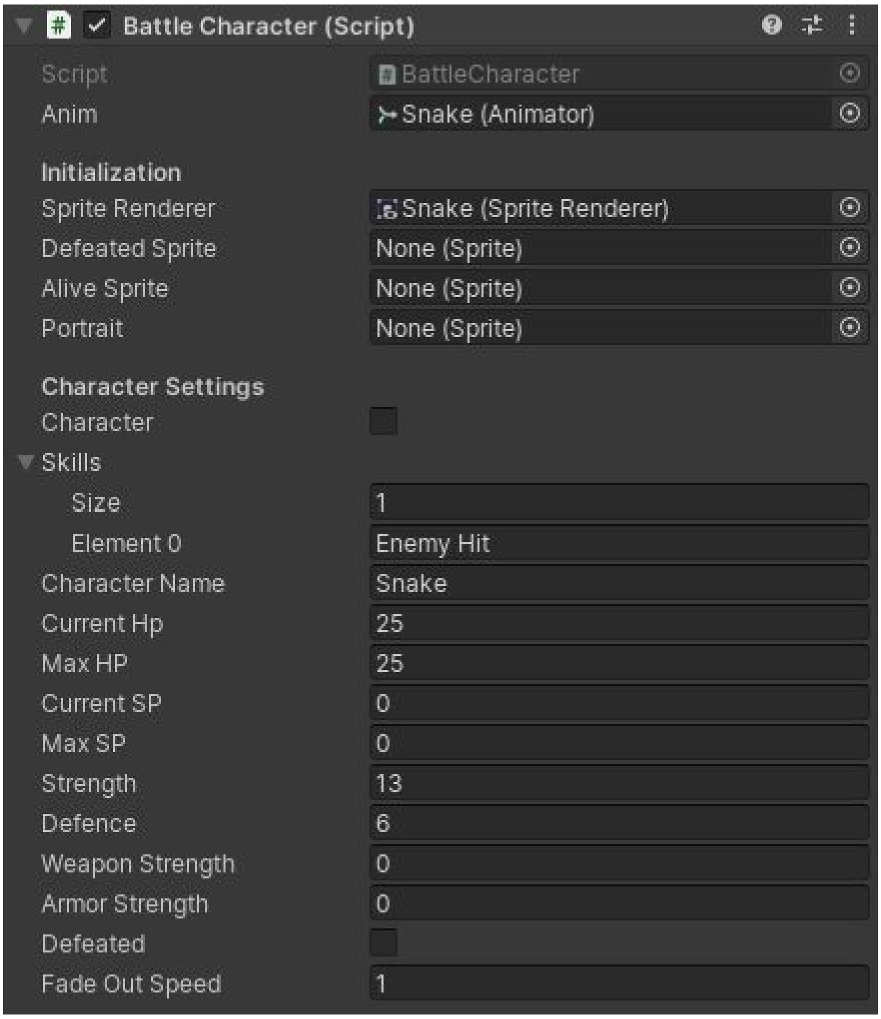
Enemies

Enemies are battle prefabs (stored in **Assets/2D RPG Kit/Battle/Enemies**). As you build your game, you will create a lot of different enemies for the player to battle. You can duplicate existing enemies and adjust their attributes within the inspector of the “Battle Character” script. Register all created enemies within the “Battle Manager” prefab under “Enemy Prefabs”.

The “Battle Character” script contains the following attributes:

Anim	The animation system used by this enemy during battle. You will have to create new animations if you want to have unique animations for each enemy since altering the animation from a duplicated enemy will also affect the animation of the original enemy prefab.
Sprite Renderer	The Sprite Renderer component of this prefab. You can assign the enemy sprite within this component.
Defeated Sprite	Not used by enemies
Alive Sprite	Not used by enemies
Portrait	Not used by enemies
Character	Let blank for enemies
Skills	Enter the list of skills this enemy should use during battle. The enemy will use a random skill from this list each turn.
Character Name	Enter a unique name
Current HP	Enter the amount of HP this enemy should go into battle with
Max HP	Enter the amount of Max HP this enemy should have
Current SP	Not used by enemies
Max SP	Not used by enemies
Strength	Enter the strength this enemy’s attacks should have

Defense	Enter the defense this enemy should have
Weapon Strength	Not used by enemies
Armor Strength	Not used by enemies
Defeated	During play, you can confirm if this character was defeated for debugging purposes
Fade Out Speed	Enter the fade-out speed for when this enemy is defeated


☑ Battle Character (Script)

Script	BattleCharacter
Anim	Snake (Animator)
Initialization	
Sprite Renderer	Snake (Sprite Renderer)
Defeated Sprite	None (Sprite)
Alive Sprite	None (Sprite)
Portrait	None (Sprite)
Character Settings	
Character	☐
▼ Skills	
Size	1
Element 0	Enemy Hit
Character Name	Snake
Current Hp	25
Max HP	25
Current SP	0
Max SP	0
Strength	13
Defence	6
Weapon Strength	0
Armor Strength	0
Defeated	☐
Fade Out Speed	1

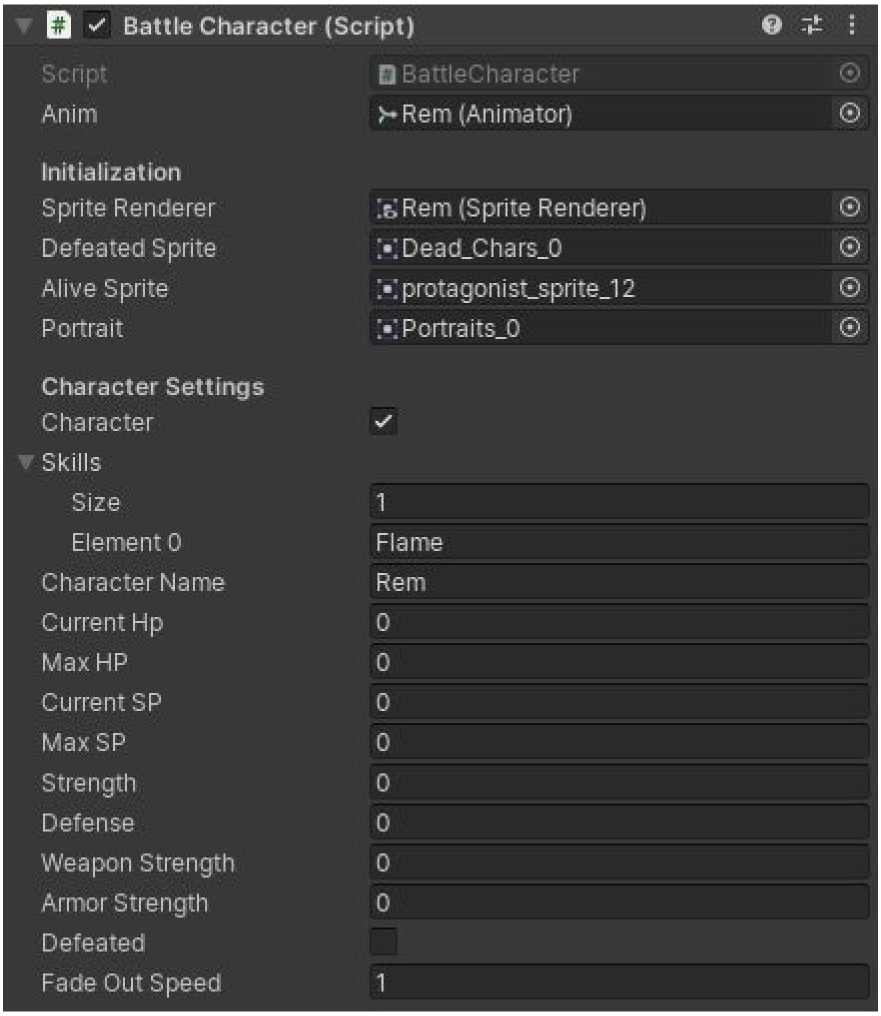
Battle Characters

Battle characters are battle prefabs (stored in **Assets/2D RPG Kit/Battle/Characters**). They share the same script as the enemy prefabs, but other than enemies, these characters are controlled by the player during battles. They represent the characters in the players' party, created with the "Character Status" script. Since the stats of the characters were already specified in the "Character Status" script, you can leave the most attributes within the script of the battle characters blank. You can duplicate existing battle characters and adjust their attributes within the inspector of the "Battle Character" script. Register all created battle characters within the "Battle Manager" prefab under "Character Prefabs".

The "Battle Character" script contains the following attributes:

Anim	The animation system used by this character during battle. You will have to create new animations if you want to have unique animations for each enemy

	since altering the animation from a duplicated enemy will also affect the animation of the original enemy prefab.
Sprite Renderer	The Sprite Renderer component of this prefab. You can assign the character sprite within this component.
Defeated Sprite	Assign the sprite for when this character is defeated
Alive Sprite	Assign the alive sprite for this character
Portrait	Assign the portrait sprite for this character
Character	Check this box for each battle character
Skills	Enter the list of skills this character can use during battle
Character Name	Enter the characters name
Current HP	Not used by playable battle characters
Max HP	Not used by playable battle characters
Current SP	Not used by playable battle characters
Max SP	Not used by playable battle characters
Strength	Not used by playable battle characters
Defense	Not used by playable battle characters
Weapon Strength	You can confirm this characters’ equipped weapon strength during play for debugging purposes
Armor Strength	You can confirm this characters’ equipped armor strength during play for debugging purposes
Defeated	During play, you can confirm if this character was defeated for debugging purposes
Fade Out Speed	Not used by playable battle characters

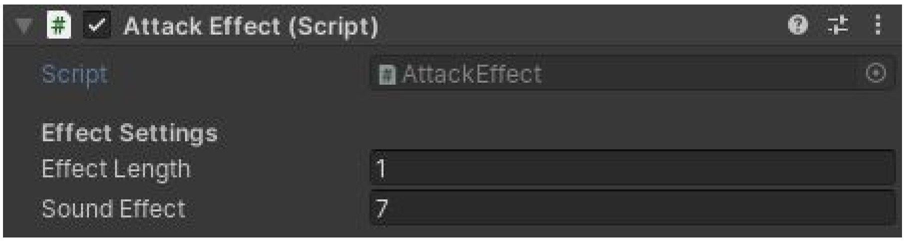


Battle Character (Script)

Script	BattleCharacter
Anim	Rem (Animator)
Initialization	
Sprite Renderer	Rem (Sprite Renderer)
Defeated Sprite	Dead_Chars_0
Alive Sprite	protagonist_sprite_12
Portrait	Portraits_0
Character Settings	
Character	☑
Skills	
Size	1
Element 0	Flame
Character Name	Rem
Current Hp	0
Max HP	0
Current SP	0
Max SP	0
Strength	0
Defense	0
Weapon Strength	0
Armor Strength	0
Defeated	☐
Fade Out Speed	1

Effects

Effects are stored as prefabs and are used for visualizing skills during battles (stored in **Assets/2D RPG Kit/Battle/Effects**). Create an animation and assign the “Attack Effect” script to the game object. Within the inspector, you can then choose the effect length. If you set the effect length to be longer than the animation, the animation will be simply repeated. Lastly, assign a fitting sound effect and save this animation as a prefab.

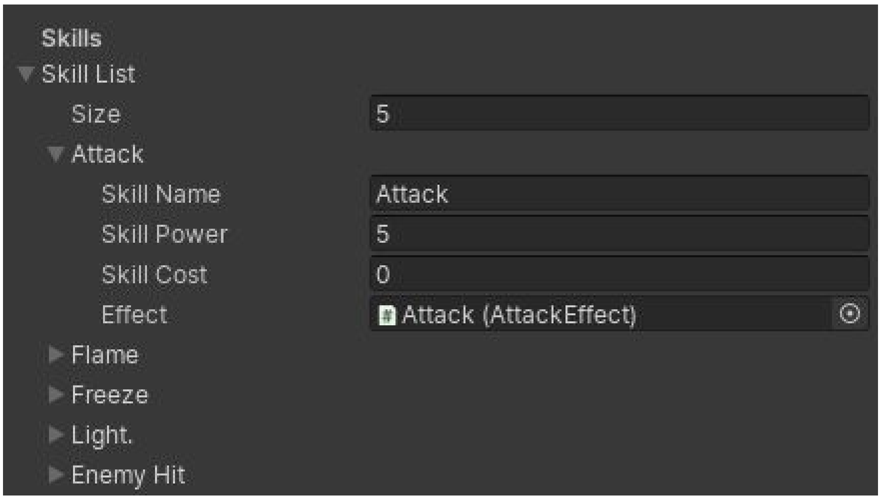


Attack Effect (Script)

Script	AttackEffect
Effect Settings	
Effect Length	1
Sound Effect	7

Skills

Skills can be created directly within the “Battle Manager” script of the “Battle Manager” prefab. There, you can enter a list of skills by assigning a unique name, the skill power and the skill cost. Lastly, assign a previously created effect prefab to the skill.



Skills
▼ Skill List
Size: 5
▼ Attack
Skill Name: Attack
Skill Power: 5
Skill Cost: 0
Effect: Attack (AttackEffect)
▶ Flame
▶ Freeze
▶ Light.
▶ Enemy Hit

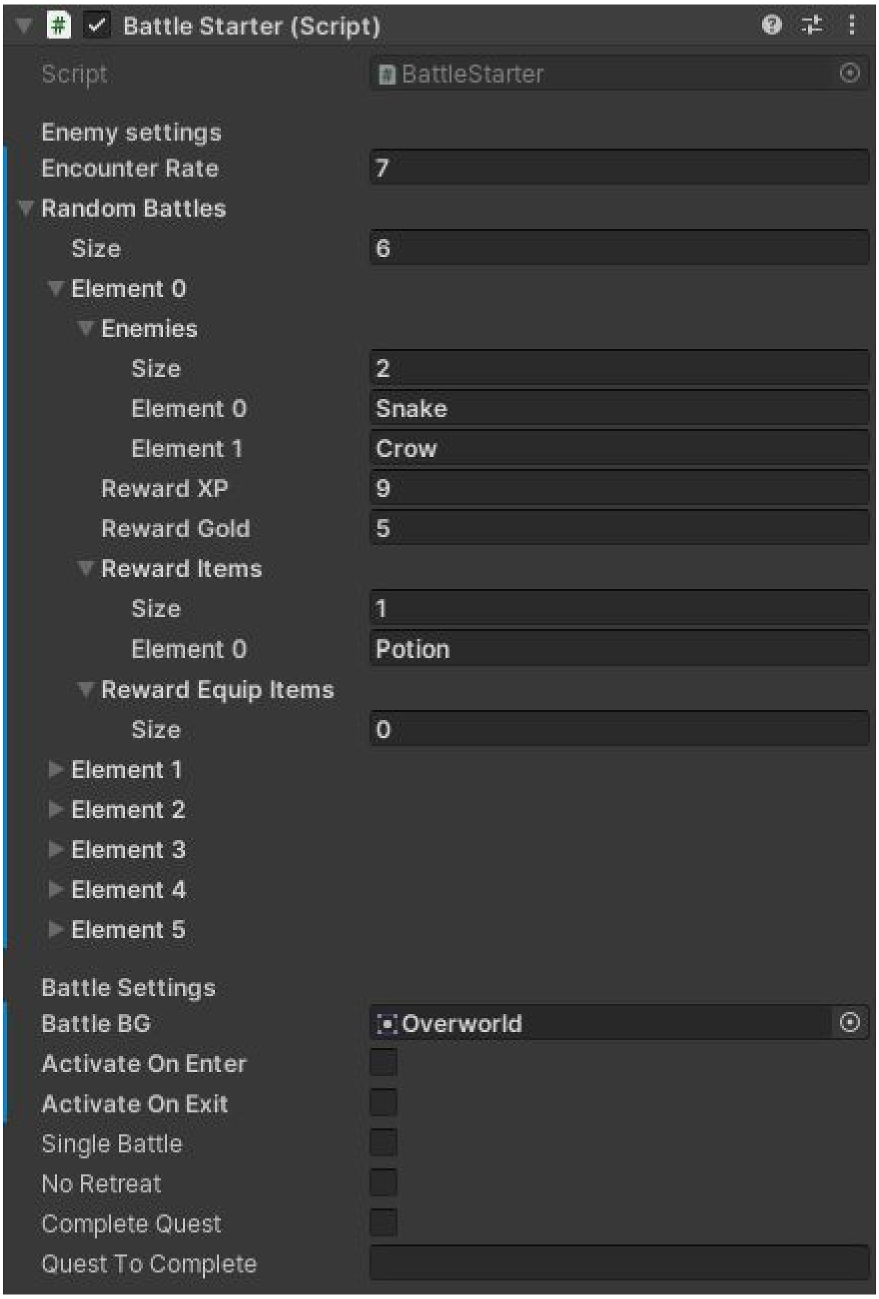
Battle Area

The “Battle Area” is an object prefab (stored in **Assets/2D RPG Kit/Prefabs/Objects**). It can be dragged and dropped directly into the scene to create areas within maps where the player can run into random encounters. Once dragged into the scene, you can resize the attached Box Collider 2D component to a fitting size. This will act as the trigger to start battles.

The attached “Battle Starter” script contains the following attributes:

Attribute	**Description**
Encounter Rate	Enter how fast battles should occur while walking through the battle area. The higher the number, the more steps between battles.
Random Battles	Enter how many different teams of enemies should be encountered
Enemies	A list of enemies within an enemy team. Enter up to four enemies for this team
Reward XP	Enter the amount of EXP gained by defeating this team of enemies
Reward Gold	Enter the amount of gold gained by defeating this team of enemies
Reward Items	Enter a list of expandable items gained by defeating this team of enemies
Reward Equip Items	Enter a list of equipable items gained by defeating this team of enemies
Battle BG	Set an image as the battle background for this battle area
Activate On Enter	Check this box if you want to start a battle as soon as the player walks into this battle area
Activate On Exit	Check this box if you want to start a battle as soon as the player exits this battle area
Single Battle	Check this box if you want to deactivate this battle area after the first battle

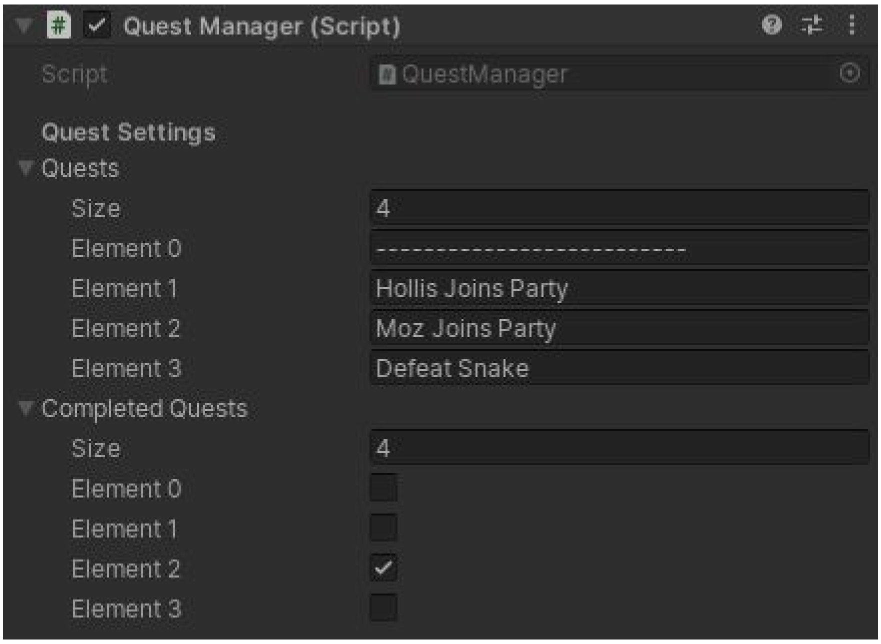
No Retreat	Check this box if you want to prevent retreating from battle
Complete Quest	Check this box if you want to complete a quest by fighting in this battle area
Quest To Complete	Enter the quest that should be completed



☑ Battle Starter (Script)	
Script	BattleStarter
Enemy settings	
Encounter Rate	7
▼ Random Battles	
Size	6
▼ Element 0	
▼ Enemies	
Size	2
Element 0	Snake
Element 1	Crow
Reward XP	9
Reward Gold	5
▼ Reward Items	
Size	1
Element 0	Potion
▼ Reward Equip Items	
Size	0
▶ Element 1	
▶ Element 2	
▶ Element 3	
▶ Element 4	
▶ Element 5	
Battle Settings	
Battle BG	Overworld
Activate On Enter	☐
Activate On Exit	☐
Single Battle	☐
No Retreat	☐
Complete Quest	☐
Quest To Complete	

Quests / Quest Manager

Quests can be created directly within the “Quest Manager” script of the “Game Manager” prefab. The “Quest Manager” should contain a list of all quests for your game so it can keep track of completed and uncompleted quests. During play, you can confirm which quest was completed by looking up the element number under “Completed Quests”.


Quest Manager (Script)

Script	QuestManager
Quest Settings	
Quests	
Size	4
Element 0	-------------------------
Element 1	Hollis Joins Party
Element 2	Moz Joins Party
Element 3	Defeat Snake
Completed Quests	
Size	4
Element 0	☐
Element 1	☐
Element 2	☑
Element 3	☐

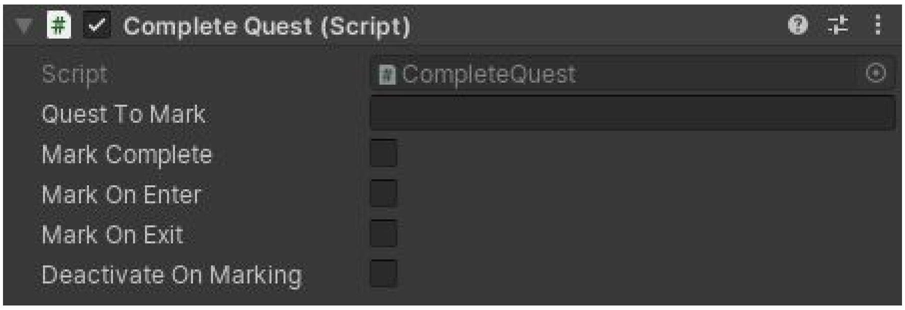
Don’t use “Element 0” to register quests within the manager

Complete Quest

Quests can be marked as completed with the “Dialog Starter” script (see “NPC” for more info), the “Battle Starter” script (see “Battle Area” for more info) and the following “Complete Quest” prefab (stored in **Assets/2D RPG Kit/Prefabs/Objects**). It can be dragged and dropped directly into the scene to create areas within maps that the player can enter to complete a given quest. Once dragged into the scene, you can resize the attached Box Collider 2D component to a fitting size. This will act as the trigger for completing quests.

The attached “Complete Quest” script contains the following attributes:

Attribute	**Description**
Quest To Mark	Enter the name of the quest that should be completed
Mark Complete	Confirm that the quest should be completed by checking this box
Mark On Enter	Check this box if you want the quest to be completed by entering this area
Mark On Exit	Check this box if you want the quest to be completed by exiting this area
Deactivate On Marking	Check this box if you want to deactivate this game object after the quest was completed


Complete Quest (Script)

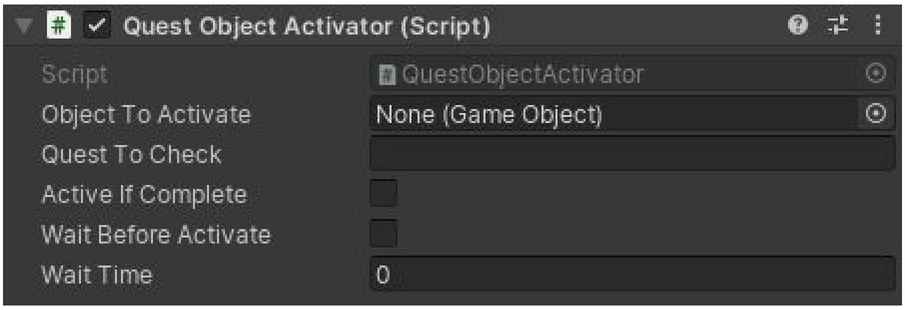
Script	CompleteQuest
Quest To Mark	
Mark Complete	☐
Mark On Enter	☐
Mark On Exit	☐
Deactivate On Marking	☐

Quest Object Activator

The “Quest Object Activator” is an object prefab (stored in **Assets/2D RPG Kit/Prefabs/Objects**). It can be dragged and dropped directly into the scene to manipulate game objects based on completed/uncompleted quests.

The attached “Quest Object Activator” script contains the following attributes:

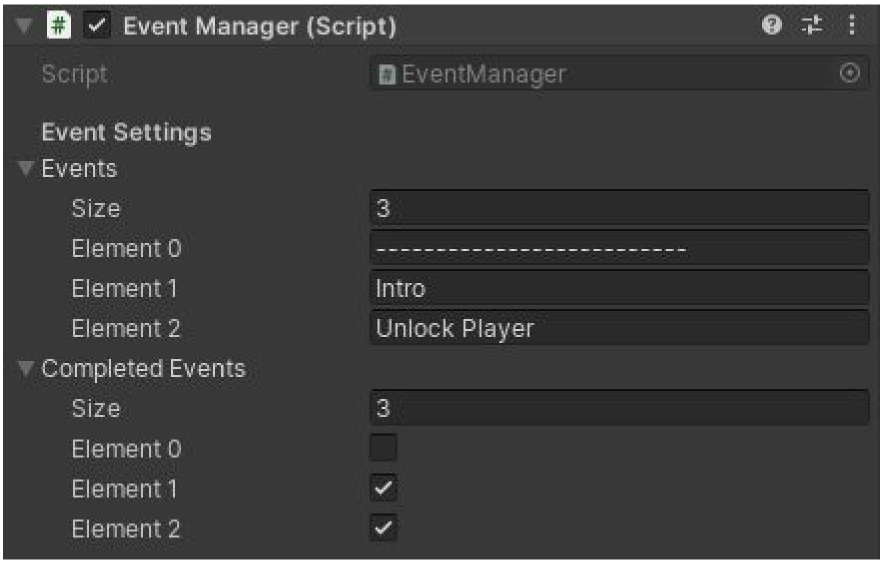
Attribute	**Description**
Object To Activate	Drag and drop the game object that should be activated / deactivated
Quest To Check	Enter the quest that should be checked
Active If Complete	Check this box to activate the linked game object as soon as the above mentioned quest was completed. Leave blank if you want to deactivate an active game object instead.
Wait Before Activate	Check this box to add a delay before the game object gets activated / deactivated
Wait Time	Enter the time to wait before activating / deactivating the linked game object (in seconds)



Script: QuestObjectActivator
Object To Activate: None (Game Object)
Quest To Check:
Active If Complete: □
Wait Before Activate: □
Wait Time: 0

Events / Event Manager

Events can be created directly within the “Event Manager” script of the “Game Manager” prefab. The “Event Manager” should contain a list of all events for your game so it can keep track of completed and uncompleted events. During play, you can confirm which event was completed by looking up the element number under “Completed Events”.
Basically, this manager works the same as the “Quest Manager”. I created two separate managers so you are able to manage other progression dependend aspects of your game, such as events and cutscenes, independently.



Event Manager (Script)

Script: EventManager

Event Settings

Events

Size	3
Element 0	-------------------------
Element 1	Intro
Element 2	Unlock Player

Completed Events

Size	3
Element 0	□
Element 1	☑
Element 2	☑

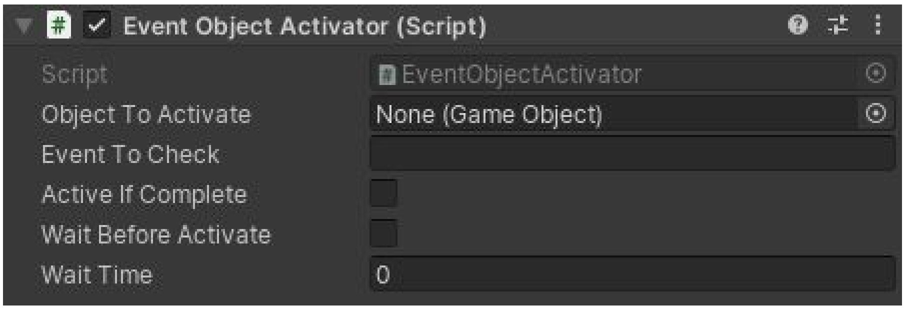
Don’t use “Element 0” to register events within the manager

Event Object Activator

The “Event Object Activator” is an object prefab (stored in **Assets/2D RPG Kit/Prefabs/Objects**). It can be dragged and dropped directly into the scene to manipulate game objects based on completed/uncompleted events.

The attached “Event Object Activator” script contains the following attributes:

Attribute	**Description**
Object To Activate	Drag and drop the game object that should be activated / deactivated
Event To Check	Enter the event that should be checked
Active If Complete	Check this box to activate the linked game object as soon as the above mentioned event was completed. Leave blank if you want to deactivate an active game object instead.
Wait Before Activate	Check this box to add a delay before the game object gets activated / deactivated
Wait Time	Enter the time to wait before activating / deactivating the linked game object (in seconds)



Event Object Activator (Script)

Script	EventObjectActivator
Object To Activate	None (Game Object)
Event To Check	
Active If Complete	□
Wait Before Activate	□
Wait Time	0

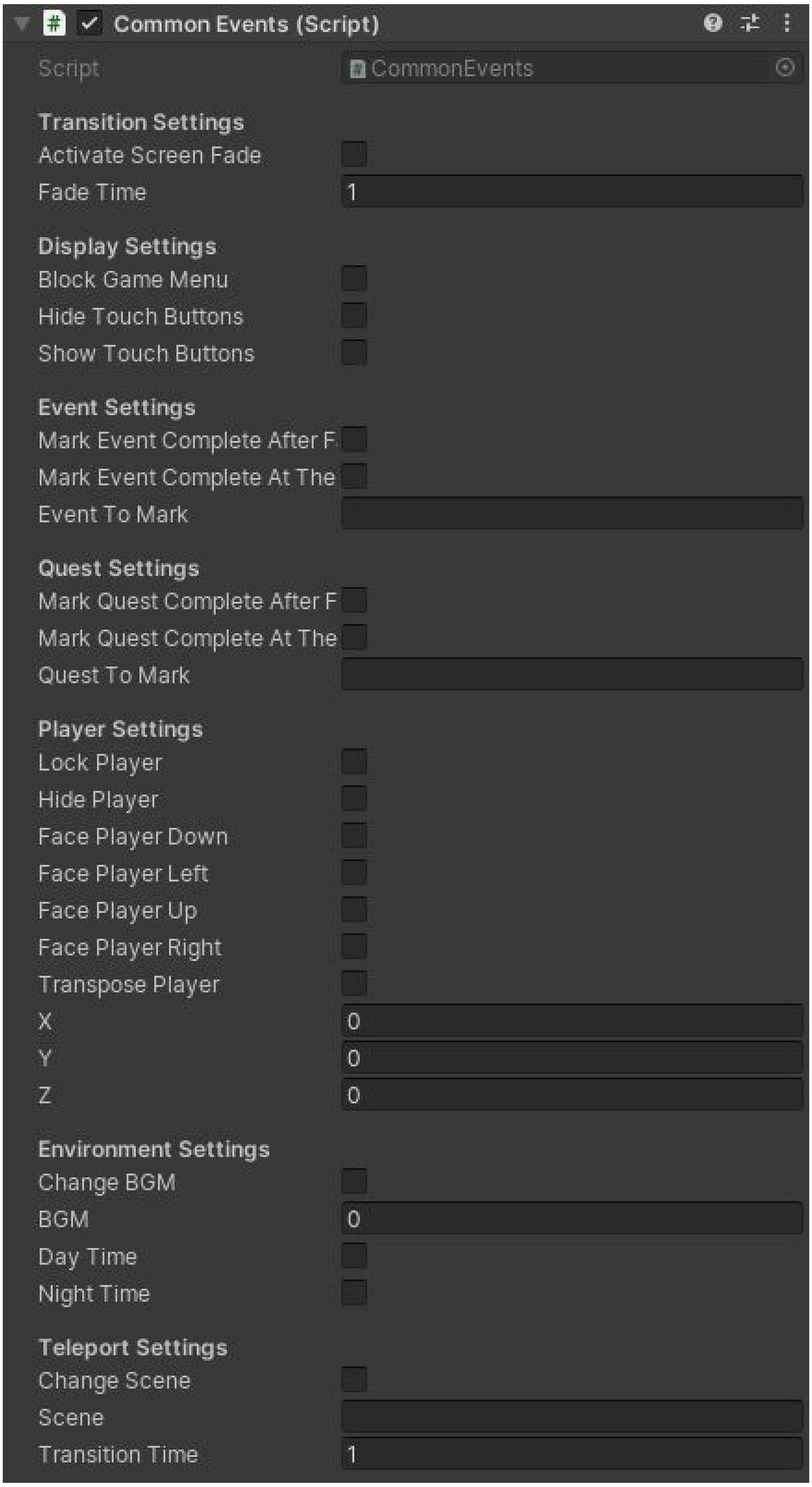
Common Events

“Common Events” is an object prefab (stored in **Assets/2D RPG Kit/Prefabs/Objects**). It can be dragged and dropped directly into the scene to create certain events during cutscenes or gameplay.

The attached “Common Events” script contains the following attributes:

Attribute	**Description**
Activate Screen Fade	Check this box if you want to activate a screen fade
Fade Time	Enter the screen fade length
Block Game Menu	Check this box if you want to disable the game menu
Hide Touch Buttons	Check this box if you want to hide the touch interface for a mobile version of your game
Show Touch Buttons	Check this box if you want to show the touch interface for a mobile version of your game
Mark Event Complete After Fade	Check this box if you want to mark an event as complete after a screen fade
Mark Event Complete At The End	Check this box if you want to mark an event as complete
Event To Mark	Enter the name of the event that should be completed
Mark Quest Complete After Fade	Check this box if you want to mark a quest as complete after a screen fade
Mark Quest Complete At The End	Check this box if you want to mark a quest as complete
Quest To Mark	Enter the name of the quest that should be completed
Lock Player	Check this box if you want to disable player movement
Hide Player	Check this box if you want to hide the player from the scene
Face Player Down	Check this box if you want to make the player face down
Face Player Left	Check this box if you want to make the player face left
Face Player Up	Check this box if you want to make the player face up
Face Player Right	Check this box if you want to make the player face right
Transpose Player	Check this box if you want to place the player on a specific location in this scene
X	If you chose to transpose the player, enter the X coordinate
Y	If you chose to transpose the player, enter the Y coordinate
Z	If you chose to transpose the player, enter the Z coordinate

Change BGM	Check this box if you want to change the background music
BGM	enter the BGM that should be played
Day Time	Check this box if you want to switch to day time
Night Time	Check this box if you want to switch to night time
Change Scene	Check this box if you want to teleport to another scene
Scene	Enter the name of the scene you wish to teleport to
Transition Time	Enter the length of the transition to the new scene



Common Events (Script)

Script: CommonEvents

Transition Settings
Activate Screen Fade □
Fade Time: 1

Display Settings
Block Game Menu □
Hide Touch Buttons □
Show Touch Buttons □

Event Settings
Mark Event Complete After F □
Mark Event Complete At The □
Event To Mark:

Quest Settings
Mark Quest Complete After F □
Mark Quest Complete At The □
Quest To Mark:

Player Settings
Lock Player □
Hide Player □
Face Player Down □
Face Player Left □
Face Player Up □
Face Player Right □
Transpose Player □
X: 0
Y: 0
Z: 0

Environment Settings
Change BGM □
BGM: 0
Day Time □
Night Time □

Teleport Settings
Change Scene □
Scene:
Transition Time: 1

Save/Load

The object prefab “Save Point” (stored in **Assets/2D RPG Kit/Prefabs/Objects**) can be dragged and dropped directly into the scene to create save points for the player. The player then can load their save data from either the game menu or the title screen.

The savepoint saves the following data:

- Scene
- Inventory
- Character stats

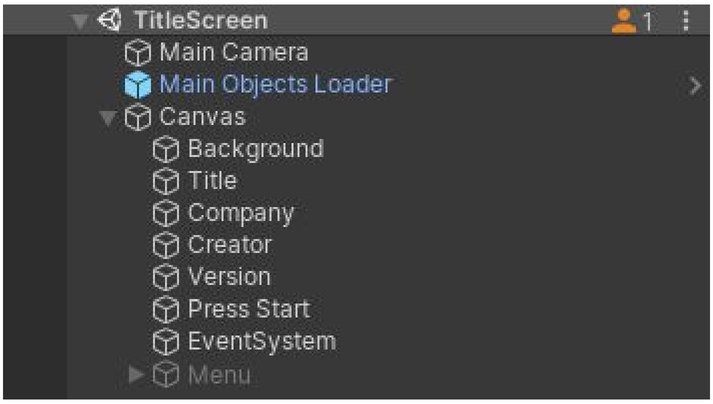
- Equipment
- Gold
- Party members
- Quests
- Chests
- Events

Loading opens the “Loading Scene” (stored in **Assets/2D RPG Kit/Scenes/System Maps**). The “Load” game objects within this scene then loads the saved data and brings the player to the last saved scene.

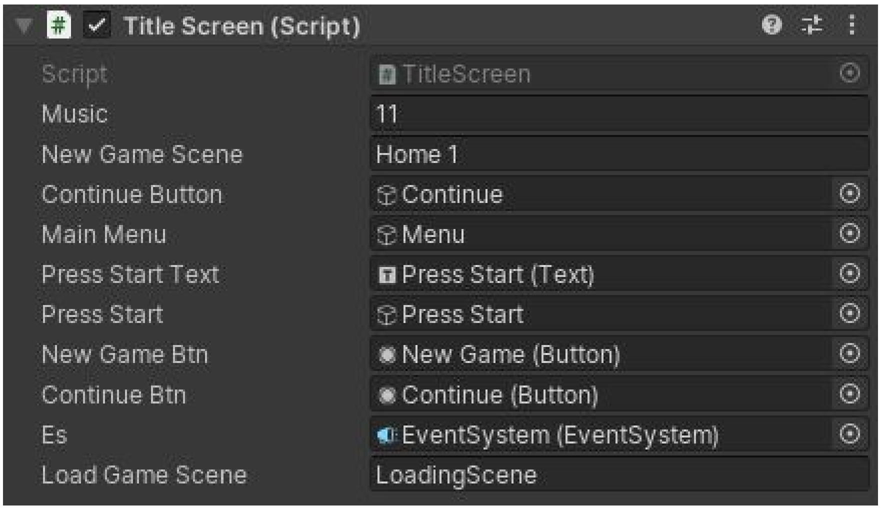
Title Screen

The “Title Scene” is a complete scene (stored in **Assets/2D RPG Kit/Scenes/System Maps**). From here, the player can choose to start a new game or load save data. You can specify the scene that should be loaded when the player decides to start a new game from the “Canvas” game object. There you will find attached the “Title Screen” script. Enter the scene under “New Game Scene” and choose a background music under “Music”. You can also customize the title screen by expanding the “Canvas” game object and changing the following elements:

- Background
- Title
- Company
- Creator
- Version



```
▼ TitleScreen                         1
    Main Camera
    Main Objects Loader
  ▼ Canvas
      Background
      Title
      Company
      Creator
      Version
      Press Start
      EventSystem
    ▶ Menu
```

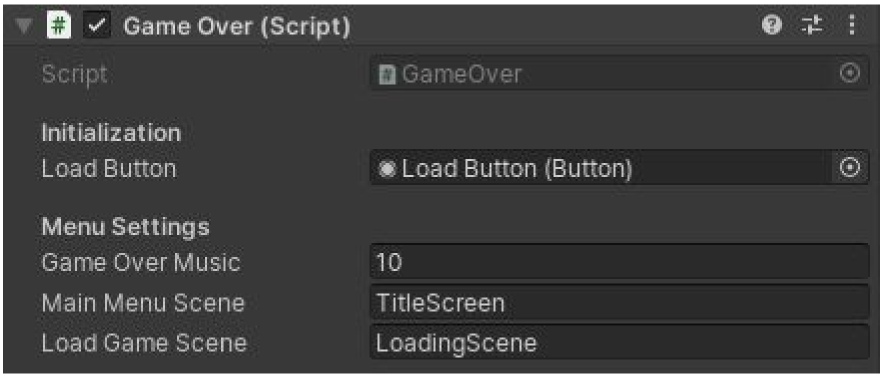


Title Screen (Script)

Script	TitleScreen
Music	11
New Game Scene	Home 1
Continue Button	Continue
Main Menu	Menu
Press Start Text	Press Start (Text)
Press Start	Press Start
New Game Btn	New Game (Button)
Continue Btn	Continue (Button)
Es	EventSystem (EventSystem)
Load Game Scene	LoadingScene

Game Over Screen

The “Game Over” scene is loaded automatically when all party members are defeated during battle (stored in **Assets/2D RPG Kit/Scenes/System Maps**). From here, the player can use to load the game or go to back to the title screen. The “Canvas” game object contains the “Game Over” script. Here, you can choose the background music, the scene to enter the title screen and the loading scene.



Game Over (Script)

Script: GameOver

Initialization

Load Button: Load Button (Button)

Menu Settings

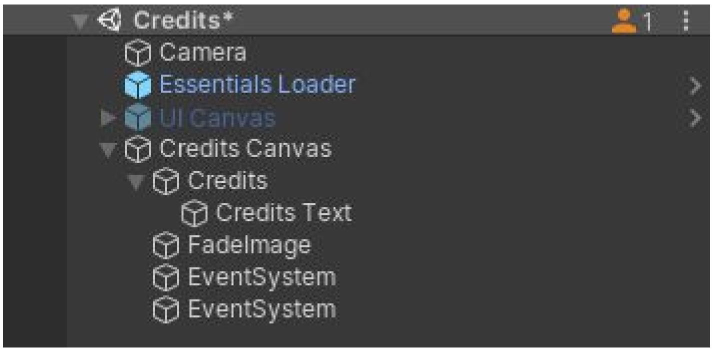
Game Over Music: 10

Main Menu Scene: TitleScreen

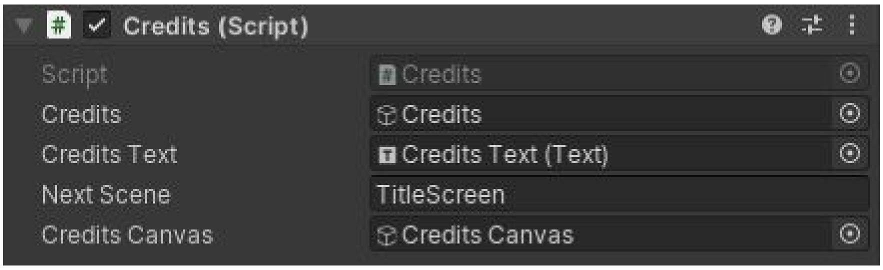
Load Game Scene: LoadingScene

Credits Screen

“Credits” is a complete scene (stored in **Assets/2D RPG Kit/Scenes/System Maps**). It can be entered when the player completed the game and will roll the typical credits. Just teleport to this scene via the “Common Events” prefab. Here you can customize the credits by expanding the “Credits Canvas” and editing the “Credits” text object. This object has a simple animation that you can also customize to your needs. From the “Credits Scene”, the player can enter the “Title Screen” by pressing the “Interact” button. This can also be changed in the “Credits” script within the “Credits Canvas” under “Next Scene”.



```
Credits*
  Camera
  Essentials Loader
  UI Canvas
  Credits Canvas
    Credits
      Credits Text
    FadeImage
    EventSystem
    EventSystem
```



Credits (Script)

Script: Credits

Credits: Credits

Credits Text: Credits Text (Text)

Next Scene: TitleScreen

Credits Canvas: Credits Canvas

Input

You can confirm the button mapping within Unity’s Input Manager. Here you will find the following 2D RPG Kit button settings:

Name	**Mapping**

RPGMenuPC	**Menu** button. Opens the game menu with the “x” key on the keyboard
RPGMenuJoy	**Menu** button. Opens the game menu with the joystick button 3
RPGConfirmPC	**Interact** button. Interact with the “return” key on the keyboard
RPGConfirmJoy	**Interact** button. Interact with the joystick button 1
RPGCanclePC	**Cancel** button. Cancel with the “escape” key on the keyboard
RPGCancleJoy	**Cancel** button. Cancel with the joystick button 2
Horizontal	**Move**. Horizontal control with left and right arrow keys + “a” and “d” buttons on the keyboard. Works also with the joystick
Vertical	**Move**. Vertical control with up and down arrow keys + “w” and “s” buttons on the keyboard. Works also with the joystick