SOQET
DOCUMENTATION

YOUR COMPREHENSIVE GUIDE TO USING THE STORY-OBJECTIVE-QUEST-EDITOR-TOOL IN UNITY
GABRIEL ALABI

Contents

INTRODUCTION

Welcome to the documentation for the Story-Objective-Quest-Editor-Tool (SOQET)! SOQET is a Unity tool designed to streamline the process of creating captivating stories, objectives, and quests for your game. Whether you're developing an RPG, adventure, or any narrative-driven experience, SOQET provides you with a user-friendly interface and robust features to bring your game world to life.

By using SOQET, you can easily define and manage storylines, objectives, and quests within your Unity projects. SOQET has reduced the manual scripting and tedious organization required to develop narrative/quest driven games. With SOQET, you can focus on crafting compelling narratives and engaging gameplay, while the tool handles the intricate logic and dependencies behind the scenes.

SOQET empowers you to create dynamic and immersive game experiences by offering a visual and intuitive approach to designing your stories, objectives, and quests. With its easy-to-use interface, you can seamlessly create/modify stories, objectives, quests, and subscribe to required events, allowing for a flexible and customizable game progression.

This documentation will guide you through the installation, setup, and usage of SOQET. Whether you're a beginner or an experienced game developer, you'll find everything you need to get started and make the most out of this tool. From creating your first story to integrating quests into your game, each section will provide detailed instructions, tips, and best practices to help you harness the full potential of SOQET.

We hope this documentation will serve as your companion on your journey to create captivating narratives and engaging gameplay experiences. Let's dive in and unlock the true potential of your game with SOQET!

SOQET is compatible with Unity 2020.3.9 or later.

INSTALLATION AND SETUP
Installing SOQET
1. Open your Unity project in the Unity Editor.
2. Download the latest version of SOQET from the Unity Asset Store.
3. Install SOQET into your project.

Unity will import and process the package, and SOQET will be added to your project.

Dependencies and Additional Packages

SOQET requires Unity version 2020.3.9 or later. Make sure your Unity Editor is up to date. There are no additional dependencies or packages required for SOQET.

Configuration and Initialization

Once SOQET is imported into your project, it's ready to use without any additional configuration or initialization steps.

USER INTERFACE

Components and Elements

Canvas: The main workspace where you design and arrange the objectives and quests for each story using a visual representation.

Connections: Lines connecting elements on the canvas represent relationships between different objectives and quests.

Buttons: Buttons are used to create or delete quests/objectives in a story. Each button is descriptive of the action it performs.
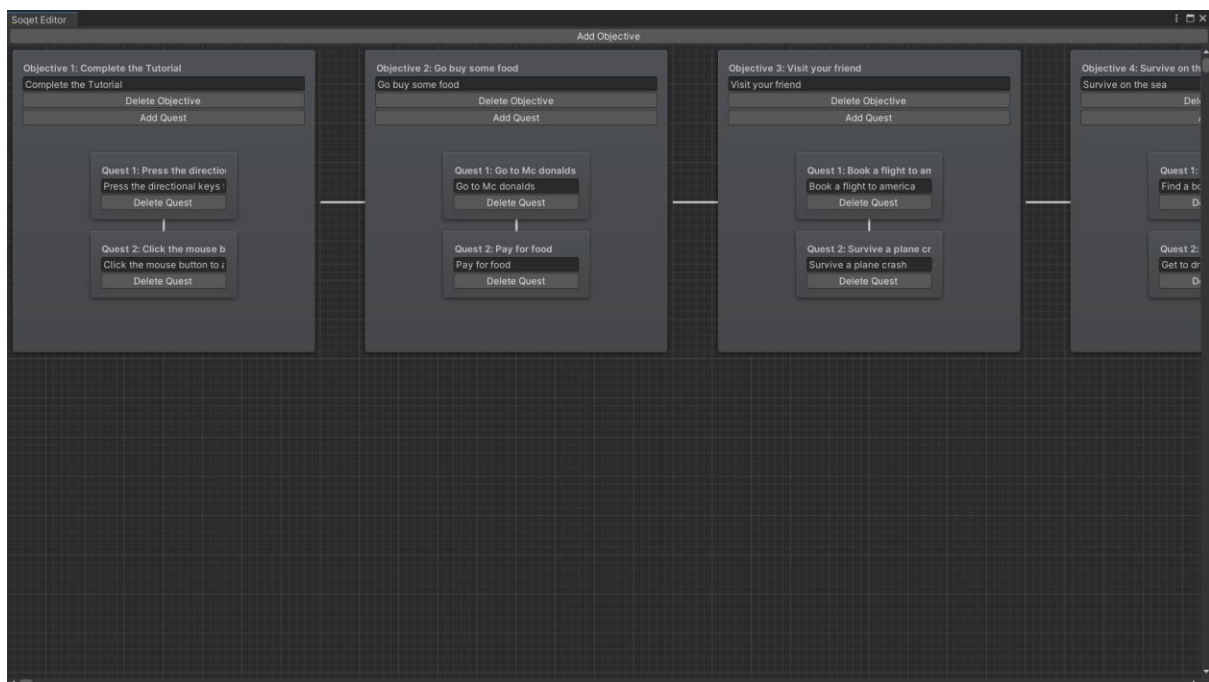

Figure 1: A canvas filled with Objectives and Quests from a story and static connections between each objective and quest.

Purpose and Functionality

Canvas: The canvas serves as a visual workspace where you can visually design and arrange your objectives, and quests. It offers an intuitive and interactive interface for creating and editing your game's progression.

Connections: Connections between objectives and quests visually represent relationships and dependencies, helping you define the logical flow and order of your game's narrative progression.

Quest/Objective connections are set automatically and cannot be modified.

Buttons: Buttons are used to create/delete quests or objectives.

CREATING STORIES, OBJECTIVES, and QUESTS

Creating Stories

You can create any story by right-clicking on an empty space in the project tab. Right-click on empty space in project tab -> create -> SOQET -> Story. Note press enter to open the newly create story in the SOQET editor.
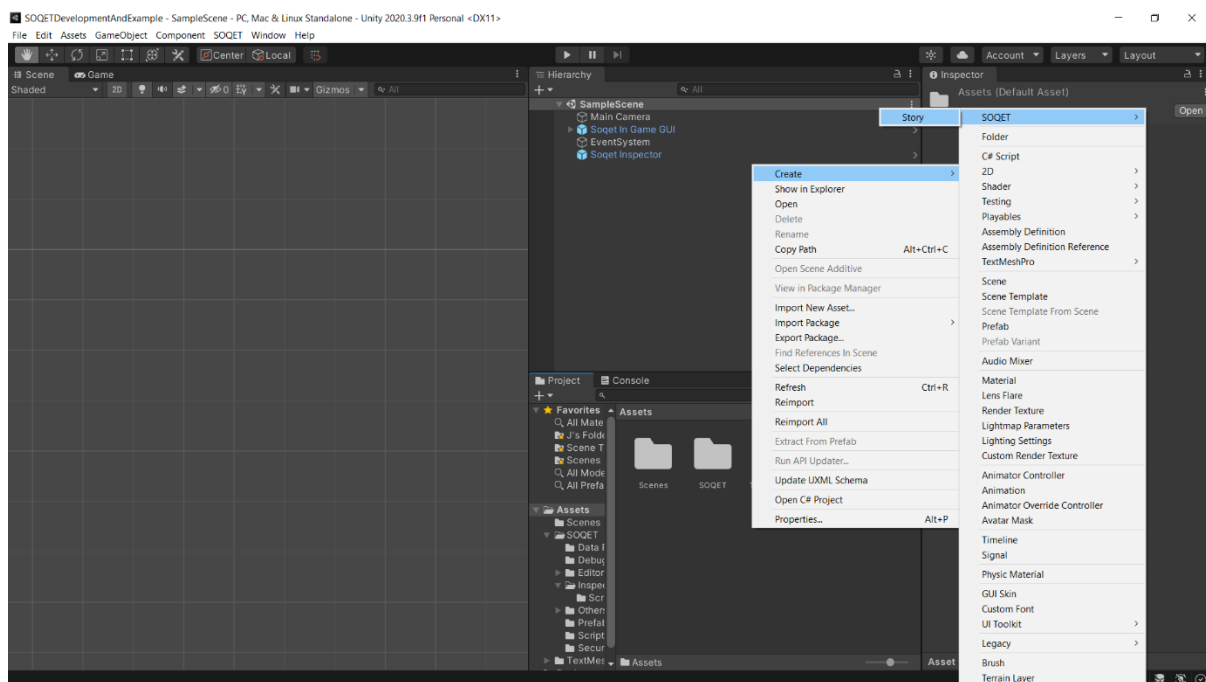

Figure 2: How to create a Story

Creating Objectives

You can create objectives by opening a story in the SOQET editor and left-clicking the "Add Objective" button. Note: Remember to save your changes by pressing ctrl + s(Windows) or cmd + s(MacOS).
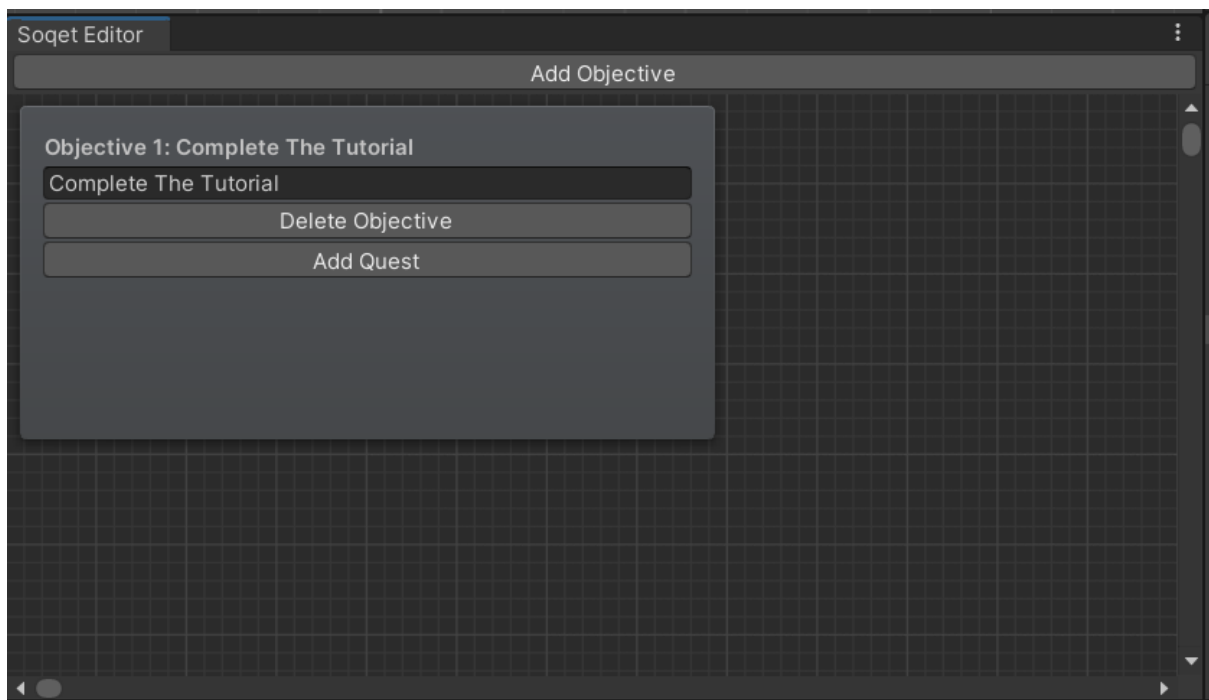

Figure 3: How to create objectives

Creating Quests

You can create any quest by opening a story in the SOQET editor, clicking on an existing objective and left-clicking the "Add Quest" button. Note: Name your Quests or Objectives as you want.
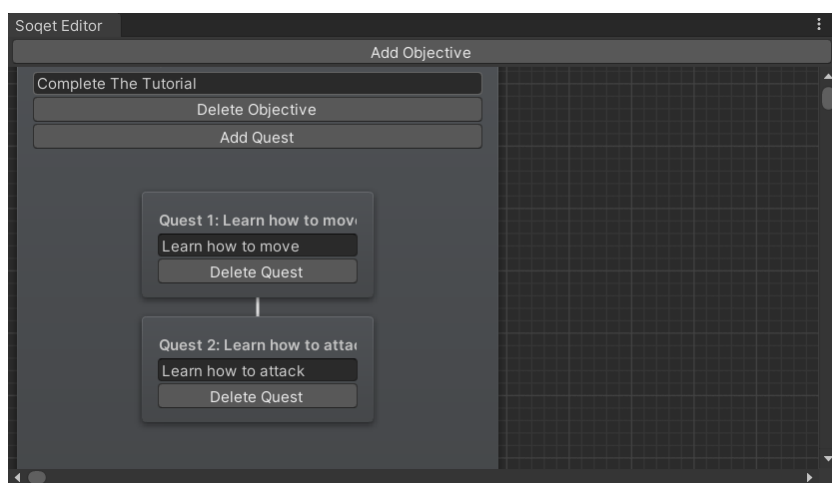

Figure 4: How to create quests

MODIFYING STORIES, OBJECTIVES, and QUESTS

Figure 5: Quest/Objective Inspector View

Note: All Stories, Objectives and Quests have attributes that can be modified in the inspector. The IsStarted and IsCompleted checkboxes and events can be modified in the inspector. The IsStarted and IsCompleted checkboxes are automatically managed by SOQET. The OnStart and OnComplete events can be manually added either through code or in the inspector.
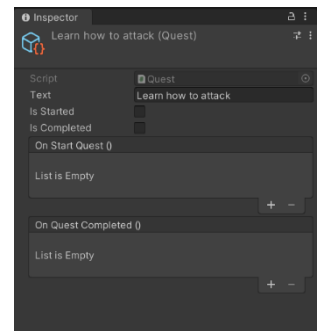
Modifying Stories

Each newly created story has soqet editor settings that can be modified in the unity inspector. Note: The "Save State" checkbox can be used to save the progress of the player. The "Enable Debug" checkbox is used to print out debug log statements for the SOQET editor which is used to notice problems in your quests and objectives. The "Enable Story" checkbox is used to enable or disable the Story. The "Encrypt Save File" is used to encrypt the saved story progress.

You can always delete the newly created story by right-clicking the story and clicking delete. You can also change the name of stories by by right-clicking the story and clicking rename.
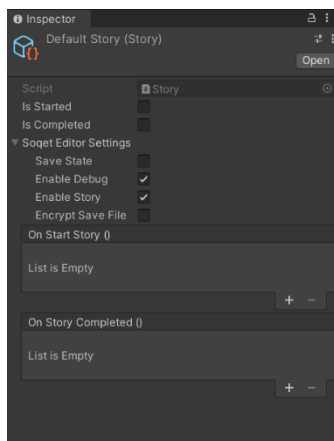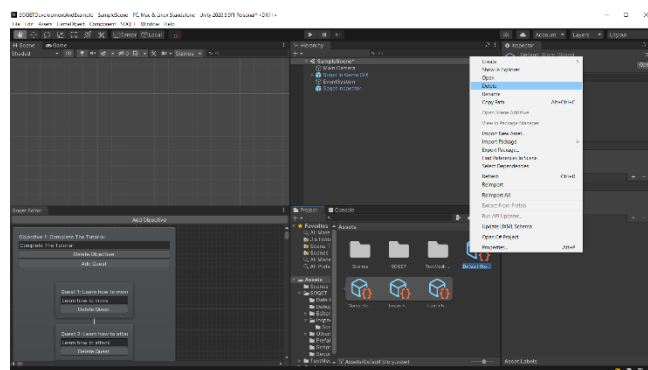

Figure 7: Story inspector View

Figure 6: How to rename/delete a Story

Modifying Objectives and Quests

You can always modify the name of any objective or quest. You can delete any quest and objective as you wish in the SOQET Editor. Figure 4 image gives a good visual description

SOQET INTEGRATION WITH UNITY

To work with SOQET in unity you will need to do 4 things.

1. You will need to create a story, objectives, and quests as was shown previously.
2. You will need to drag and drop the "SOQET Inspector" prefab into the scene of your choice.
3. You will need to setup the required events to trigger certain actions of your choice and measure player progress.
4. You will need to drag and drop the SOQET In Game GUI prefab into the scene of your choice to show the current objective and current quest.

SOQET INSPECTOR

SOQET inspector needs a story to function. You will drag the story you have created into the current story variable in the inspector. The SOQET inspector displays the current story information which will be used to easily subscribe to and unsubscribe from events present in the story, objectives and quests.
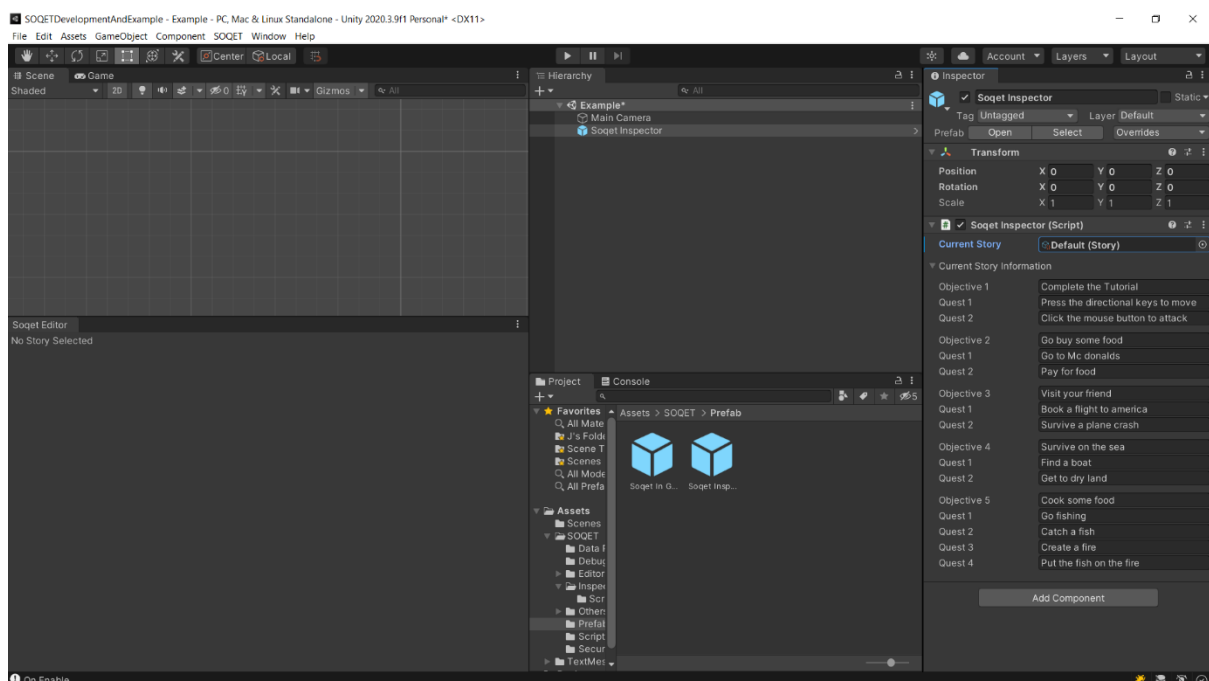

Figure 8: SOQET Inspector Prefab added to the scene youre working on

SOQET IN GAME GUI

Since In Game GUI logic will be different for each user, you will need to setup your own in game gui and connect it to SOQET.
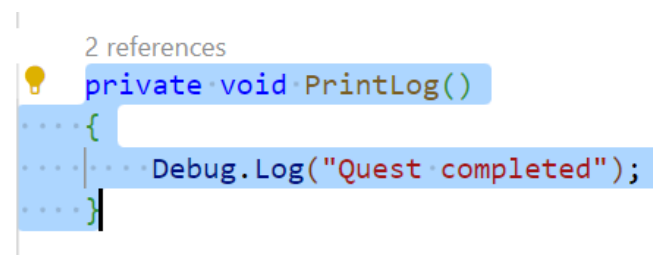
TRIGGERING EVENTS

There are 2 main events which include the OnStartEvent and OnCompleteEvent. The OnStartEvent is run when starting a story, an objective, and quest. The OnCompleteEvent is run when completing a story, an objective, and quest.
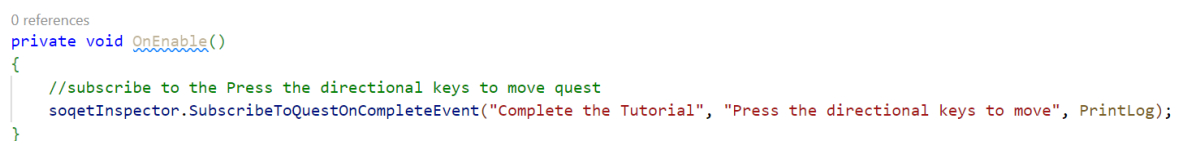
Step 1

Firstly, you must subscribe to an objective's or a story's or a quest's OnStart/OnComplete event in the OnEnable method by writing code. Create a new script or use your existing script as you wish.   An example below is that I am subscribing to a "Press the directional keys to move" quest OnComplete event which exists in the "Complete the Tutorial" objective.

```
2 references
private void PrintLog()
{
    Debug.Log("Quest completed");
}
```

Figure 9: Method that subscribes to OnStart/OnComplete Event

```
0 references
private void OnEnable()
{
    //subscribe to the Press the directional keys to move quest
    soqetInspector.SubscribeToQuestOnCompleteEvent("Complete the Tutorial", "Press the directional keys to move", PrintLog);
}
```
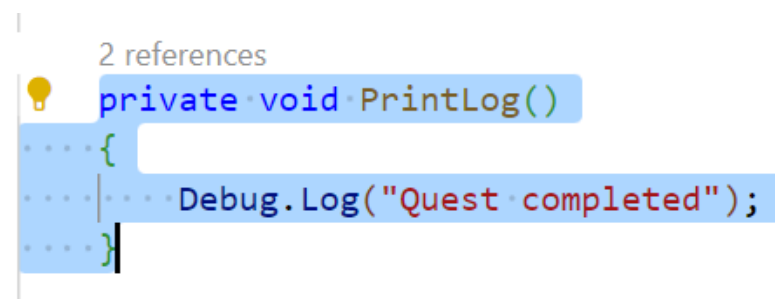
Figure 10:Subscribing to OnStart/OnComplete Events

Step 2

Secondly, you must unsubscribe from an objective's or a story's or a quest's OnStart/OnComplete event in the OnDisable method by writing code. Create a new script or use your existing script as you wish.   An example below is that I am unsubscribing from a "Press the directional keys to move" quest OnComplete event which exists in the "Complete the Tutorial" objective. ***Note failure to unsubscribe can cause memory leaks in your games***.

```
2 references
private void PrintLog()
{
    Debug.Log("Quest completed");
}
```

Figure 11:Method that unsubscribes from OnStart/OnComplete Event

```
0 references
private void OnDisable()
{
    //unsubscribe from the Press the directional keys to move quest
    soqetInspector.UnsubscribeFromQuestOnCompleteEvent("Complete the Tutorial", "Press the directional keys to move", PrintLog);
}
```

Figure 12:Unsubscribing from OnStart/OnComplete Events

Step 3

Secondly, you must complete player quest when the quest is complete by writing code. Create a new script or use your existing script as you wish.   An example below is that I am completing "Press the directional keys to move" quest which exists in the "Complete the Tutorial".

```
// Update is called once per frame
0 references
void Update()
{
    if(Input.GetAxis("Horizontal") > 0f)
    {
        horizontalCompleted = true;
    }

    if(Input.GetAxis("Vertical") > 0f)
    {
        verticalCompleted = true;
    }

    if(!movementTutorialCompleted && horizontalCompleted && verticalCompleted)
    {
        movementTutorialCompleted = true;
        soqetInspector.CompletePlayerQuest("Complete the Tutorial", "Press the directional keys to move");
    }
}
```

Figure 13:Completing Player Quest

Note: You can use the events to add rewards, feedback, achievements, and autosave. In conclusion, I hope SOQET helps streamline quests, objectives and stories in your games.