The GenvidTanks Sample

Overview

The GenvidTanks sample demonstrates how to create a simple, full-featured game with Unity and run it with the Genvid SDK. We use the Unity Tanks tutorial (https://learn.unity.com/project/tanks-tutorial) as a starting point and add the Genvid API to the project to power viewer streaming, game-state syncing, and viewer interactions. We also made some enhancements and modifications to the Unity tank example to showcase Genvid technology.

Gameplay & Viewer Features

The Unity Tank Tutorial is a player-versus-player (PVP) game. The first tank to score 5 points wins. In our version we have added a few features to highlight the capabilities of the Genvid SDK.

On the web page stream:

- Viewers can select a tank and view information about it.
- Viewers can influence the match by voting for the next loot drop to spawn. A simple loot spawning system and buff system was added to the Tank Tutorial to make this possible.
- Ads overlay a billboard to showcase how it is possible to modify the viewer's experience.
- Viewers can toggle a 2D map that shows tanks, shells, power ups and explosions to showcase an "Enthusiast" type of viewing experience.

Licensing

Unity and Genvid Code EULA

Please note that this GenvidTanks sample contains code from the Unity Asset Store and some code made by Genvid Technologies. Along with the Asset Store EULA, please review the accompanying license file and make sure you agree to its terms before proceeding: `Assets/GenvidTanks/GenvidSDKLicense.txt`.

See the Genvid Technologies website for more information about licensing.

Installation and Setup

Install the Genvid SDK

1. Go to the download section of the Genvid Technologies website. (https://www.genvidtech.com/sdk-download/)
2. Either sign in or create a free account to access the Developer Portal.
3. Download the version of the Genvid SDK that matches the version of GenvidTanks sample. **Current SDK version: 1.21.0.96**.
4. Run the installer and follow the directions to install the SDK.

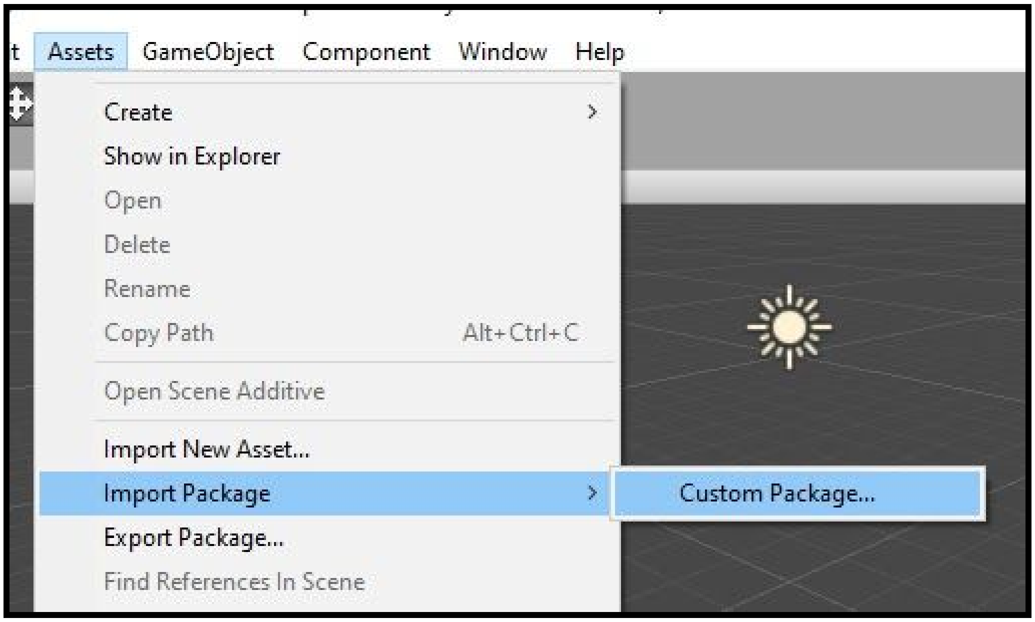
Set Up the Unity Project

1. Launch the Unity Editor. **Current Unity Project version: 2018.4.15f1 (LTS)**.
2. Create a new Unity project (or use an existing one).

3. Import the "GenvidTanks" Unity package from the Asset Store. (https://assetstore.unity.com/)

   If you already have the unitypackage file, you can manually import it.

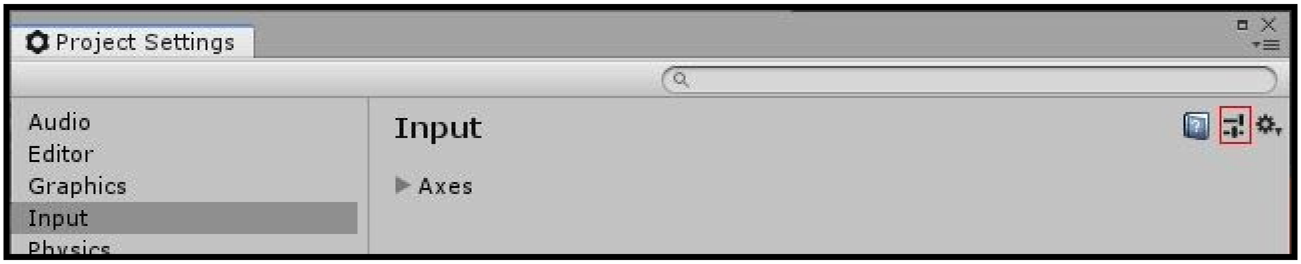
   1. Select **Assets --> Import Package --> Custom Package**
   2. Select the GenvidTanks unitypackage file.
   3. Click **Import** in the **Import Unity Package** window.



Apply Keyboard/Game Inputs

Apply Keyboard/Game Inputs to project using the GameInputFile provided.

1. Select **Edit > Project Settings**.
2. In the **Project Settings** window, select **Input**.
3. Click the slider icon in the top-right of the window.

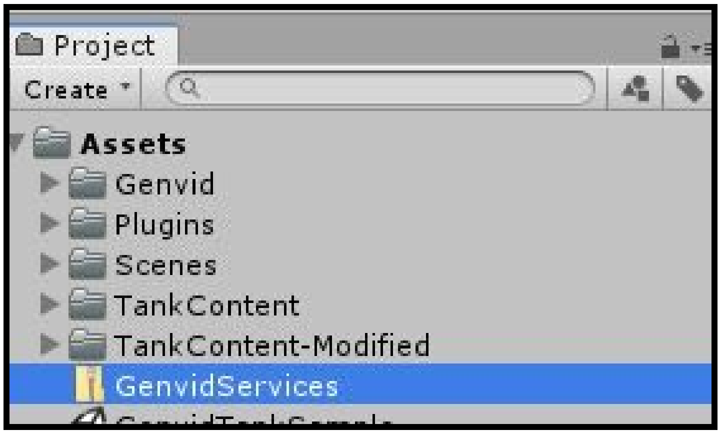


4. Double-click **GenvidTanks Inputs**.

Set Up the Genvid Services

Set Up the Directory Structure

1. In the **Project** window, right-click **GenvidServices**.



2. Select **Show In Explorer**.
3. Extract the **GenvidServices.zip** file in a directory outside your Unity Project folder.

   For example, after importing the GenvidTanks Unity package, all the directories created should be in a single directory. For example:

   `\Unity\GenvidTanks\GenvidTanksProject` (Contains the project Assets, ProjectSettings, etc.)

   So you should extract the contents of **GenvidServices.zip** into the root folder:

   `\Unity\GenvidTanks`

   This will put all the files for the Genvid Services in a single directory, like this:

   `\Unity\GenvidTanks\GenvidServices` (Contains all the files from GenvidServices.zip)

   GenvidServices creates folders and files (like the web **Node Modules** file) which can be large. Extracting it to a folder outside the main project folder ensures Unity doesn't read or generate any .meta files from GenvidServices.

4. Configure your project's name set in `\Unity\GenvidTanks\GenvidServices\config\game.hcl`
   - Open `game.hcl` with any text editor.
   - Change `GenvidTanksProject.exe` to what you named your project (for example: `NewUnityProject.exe)`.

   ```
   version = "1.7.0"

   job "unity" {
     dependencies = [
       "nats",
       "compose",
     ]
   }

   log "game" {
     job      = "unity"
     fileName = "unity.log"
   }

   log "gameerr" {
     job      = "unity"
   ```

```
  fileName = "stderr"
  logLevel = true
}

config {
  local {
    binary {
      unity {
        path = "{{env `UNITYPROJECTROOT` |
js}}\\Build\\GenvidTanksProject.exe"
      }
    }
  }
}
```

5. Set your project's path as an environment variable using your preferred shell.

   For example, in PowerShell, the command would be similar to this:

   ```
   $env:UNITYPROJECTROOT=’C:\[filepath]\GenvidTanks\[GenvidTanksProjectName]’
   ```

   This lets GenvidServices point to your Unity Project folder.

   NOTE: If you exit PowerShell, it erases this Environment Variable. You’ll need to set it again next time you reopen PowerShell and run these commands.

   To check the current setting for the env variable, run the following script from the GenvidServices directory:

   ```
   py ./unity.py env
   ```

   In PowerShell, you can view all your current environment variables by running this command:

   ```
   Get-ChildItem Env:
   ```

6. Close the Unity project.

Build and Run the Project

Build the Project

From your shell:

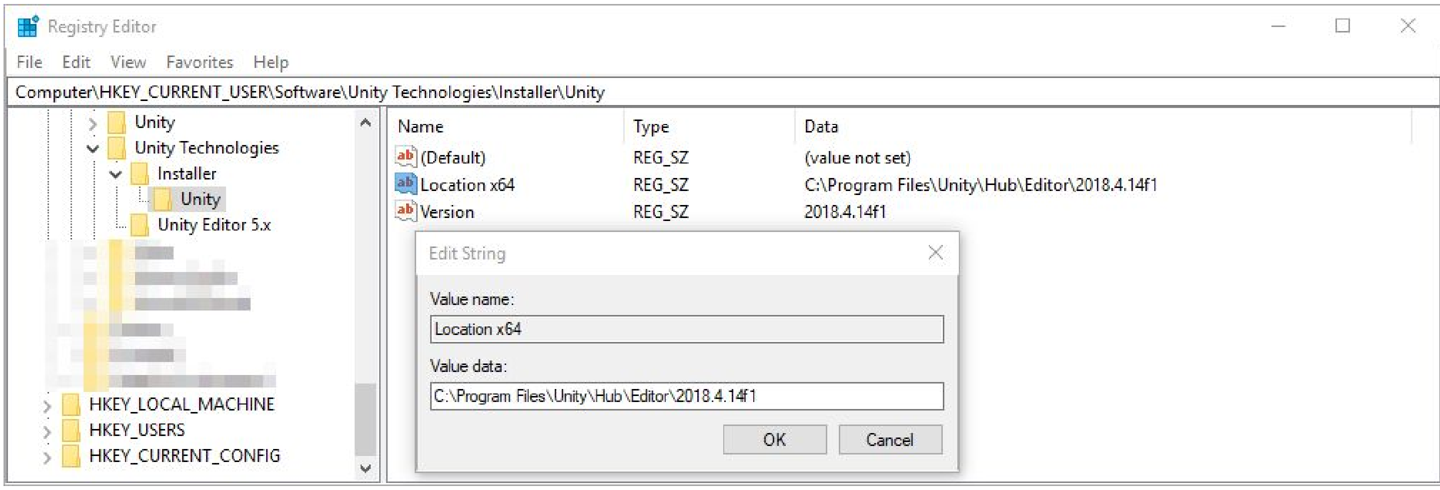
1. Go to your project’s **GenvidServices** directory.

2. Build the Unity project.

   ```
   py .\unity.py build
   ```

   If you get an error, close your Unity project. You also might need to update your Unity license.

   Note: This script uses the version of Unity specified in your Windows registry key, not your current project’s Unity version. If your Unity Project and Windows registry specify different versions of Unity, this script will change your Unity Project to match what is in the registry.



Computer\HKEY_CURRENT_USER\Software\Unity Technologies\Installer\Unity

Name	Type	Data
(Default)	REG_SZ	(value not set)
Location x64	REG_SZ	C:\Program Files\Unity\Hub\Editor\2018.4.14f1
Version	REG_SZ	2018.4.14f1

Value name: Location x64

Value data: C:\Program Files\Unity\Hub\Editor\2018.4.14f1

Start the Genvid Services

1. Install and set up the bastion server.

```
genvid-bastion install --bastionid localbastion --loadconfig
```

```
genvid-sdk setup
```

2. Make sure no old configuration data remains in the bastion.

```
genvid-sdk clean-config
```

3. Load the sample Genvid SDK configuration on the bastion.

```
genvid-sdk load-config-sdk
```

4. Open the Cluster UI page.

```
genvid-sdk monitor
```

Load and Run the Project

1. Load your project.

```
py .\genvid-project.py load
```

2. Start the Genvid Services.

`genvid-sdk start` (or click **Start All** within the Cluster-UI)

3. From Cluster UI, click the **Genvid Tanks Sample** link to open the web stream.

You should now be able to launch GenvidTanks and view/interact with the web stream!