PinaCollada Documentation

*What is PinaCollada?*

PinaCollada is an Asset Importer for Totally Accurate Battle Simulator. It has the ability to import multi-materialed static models, sound effects, and even textures into the game **without** replacing any in-game assets.

*What is coming in the future?*

The next update for PinaCollada will be a compatibility update to support non-static meshes (skinned meshes). The update(s) after that, (far in the future) will include importing armatures and replacing collision for GameObjects.

*What will be in this document?*

This document will include the most up-to-date information on PinaCollada, including information on how to use it, as well as information on updates and changes.

*Who created PinaCollada?*

PinaCollada was created by me, Derulan. I am a Unity developer who has been coding in C# and Python for ~4 years. PinaCollada was created in Unity 2019.1.0f2, and ported into TABS using DnSpy. UnitMakerHelper (UMH), which is included in PinaCollada, was **not** created by me. It was created by Fern. Show them some love on the TABS Modding Discord.

Documentation begins below.

Setup

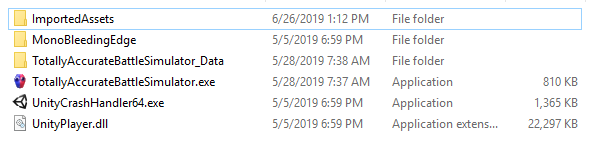
To use PinaCollada, you first need to install the Mod File, then you need to set up your *Imported Assets* folder. This is the folder that will hold all the assets that you will be importing into the game.

First, navigate to your TABS root directory, which should be something like:  
C:\Program Files (x86)\Steam\steamapps\common\Totally Accurate Battle Simulator

Then, create a new folder named “ImportedAssets”.

REMINDER: Everything PinaCollada uses is case sensitive. Make sure to capitalize the folder as shown here.

The folder should look something like this:



Now, navigate inside the newly created folder, and create three more folders, named

“Models”, “Sounds”, and “Textures” respectively.

Now you should be ready to start creating Assets for PinaCollada.

Importing 3D models

Importing models is pretty simple with PinaCollada, just make sure to follow the steps and requirements. This section will be broken up into two parts:

* Creating the model
* Importing the model

Creating the model:

When creating a model, do whatever you want, but keep these restrictions in mind:

* Do not try to import a model with multiple objects
* Make sure the model is as centered as possible
* PinaCollada will only import the diffuse color of your model, unless it is using a Phong shader.
* When exporting your model, make sure you rotate your model 90 degrees on the X axis (if using blender) and export using the “Collada” (.dae) file format.
* Models in the (.dae) format found online may be too old to work with PinaCollada. Make sure to re-import them into blender, and re-export them to fix version-related issues.
* Don’t use vertex colors.
* Make sure that your mesh is being triangulated manually or through the Collada exporter.

If your model fits all of these criteria, you can move onto the next step.

Importing the model:

To import the model, you need to place it in the ‘Models’ folder in the ‘ImportedAssets’ folder.

Now, we can move onto DnSpy. Open DnSpy and navigate to the UMH class.

For an example, let’s try and replace a weapon with a PinaCollada model.

Find the EditUnitWeapon method in UMH, and in it, place the following code:

This code will replace the weapon model(s) of a unit with the model with the same name in the Models folder. The third parameter is a simple scalar. This is for when your model is too big or too small. Instead of pain-stakingly resizing it in blender, you can conveniently resize it from the code. The other parameters are not shown here, but there are six in total. The third and fourth are a Rotation and Translation, allowing for last-minute adjustments to models, and the fifth allows the GameObject containing the model data to be named, for whatever purpose you have in mind.

Now, if you start up TABS, your custom unit should now be using your custom model!

Importing Sounds

Sounds are a bit complicated to import. First off, they **must** be in WAV format. Secondly, you can't just slap them into the ‘ImportedAssets\Sounds’ folder.

The reason for this is the way that TABS processes sound effects.

Sounds are represented by a SoundEffectInstance. Each SoundEffectInstance contains data like:

* Volume
* Distance cutoff
* Pitch Variation
* Spacial Blending

As well as an array of another type of class, the SoundEffectVariation, which has a materialType, and an array of AudioClips which actually carry the audio data. The materialType mentioned above is used for sounds that vary depending on what material they hit or are hit by. For example, you’d want a different sound for when a bullet hits wood, opposed to when it hits metal. Other sounds, like menu clicks or unit voices, only have one SoundEffectVariation with the type Default. Now with that out of the way, here is the folder structure for a sound effect:



As you can see, each variation has its own set of audio files, which are selected and played randomly by the TABS SoundPlayer. If you were making a sound effect that doesn’t have variants, like a unit voice, you would only make the ‘Default’ folder.

Now, if you look at the bottom left corner of the diagram, you can see a file called ‘EffectName.sfxml’. This is a Sound Effect Markup Language file, which contains other sound data, like pitch and volume. Here is the code inside.



These files are pretty simple, however, if anything is misspelled, PinaCollada will not run properly.

*Note: Some attributes of Sound Effects are not listed here, such as lengthInMesures and transitionMesures, which have to do with songs. Their sfxml tags are ‘song\_length’ and ‘transitions’ respectively. However, these are not very useful when working with sound effects, and should be used at your own risk.*

To use an imported sound, replace the soundRef variable of whatever class you are editing with “PinaCollada/NameOfTheSoundEffect”. The sound should now be able to be heard in-game.

Importing Textures

Importing textures is very simple. Take pictures you have (which must be in .png format) and put them in the ‘ImportedAssets\Textures’ folder.

To access them, either index the imported\_textures array with the number responding to the texture’s place in the folder alphabetically, or use PinaCollada’s GetSprite method, which creates a sprite from a texture in the Textures folder with the same name.

Have in mind that PinaCollada does not have a method to texture objects with imported textures, so you will have to do that manually.

*Thanks for reading, I hope this helps.*

*If you have any trouble, feel free to @ me in the*

*#modder-questions channel of the discord server.*

*Happy Modding!*

*Derulan.*

Changelog

> (1/7/19) PinaCollada 0.1b released for TABS stable 0.3.1

> (9/7/19) PinaCollada 0.2b released for TABS stable 0.3.1

* SetStaticMesh() now takes a string instead of a mesh index
* GetSprite() method added to make using imported Textures easier
* Various bug fixes
  + Stopped sound system from breaking if no sfxml file is found

> (1/7/19) PinaCollada 0.3b released for TABS stable 0.3.1

* Fixed parsing issues due to local number systems

> (13/7/19) PinaCollada 0.3.1b released for TABS stable 0.3.1

* Fixed bug where translation was not local to the unit

> (1/8/19) PinaCollada 0.3.2b released for TABS stable 0.4.3

* Updated to newest TABS version and newest version of Fern’s UMH

> (6/8/19) PinaCollada 0.3.3b released for TABS stable 0.4.3

* Updated to include UMH updates

> (9/8/19) PinaCollada 0.3.4b released for TABS stable 0.5.1

* Updated to newest TABS version and newest version of Fern’s UMH