Overview

This document contains all of the information needed for Touch Designer Integration into the rest of the Art Generator Project. Information such as connectivity, recording, and patch enumerations can be found in this document.

Connectivity

Required Objects

This section describes the required objects for full Python integration for starting and closing Touch Designer. Table 1 includes the full list of objects required for full functionality. Table 2 includes all objects that are dependent on Python files, and the associated Python files. There is an additional section, “Optional Objects”, that discusses objects that can be added to use for debugging the Touch Designer art generation.

Object Name	**Object Type**
TD_InitializePatch	CHOP Execute
AGD_LaunchGeneration	Text
AGD_LaunchGeneration_exec	Execute
AGD_StartTrigger	Constant
AGD_Timer	Timer
AGD_RecordOutput	Movie File Out

Table 1: Required Objects for Touch Designer Integration

TD_InitializePatch

The TD_InitializePatch object is a CHOP Execute object. The purpose of this object is to execute upon start-up to properly initialize all Touch Designer elements needed for execution and to start the process of execution in Touch Designer.

Unlike all other Python scripts that are utilized in Touch Designer, this the python script cannot be loaded in during run-time. This means that any time a change to TD_InitializePatch.py is made, it must be re-uploaded manually to each Touch Designer patch. Since Touch Designer closes after a certain period of time, in order to make any manual change to a patch, the developer must temporarily remove the ‘quit()’ line in AGD_Utilities.py to prevent the timer from exiting the patch.

AGD_LaunchGeneration

This object is responsible for loading in all of the required data for execution, as well as starting the actual processing of the Touch Designer patch. This file will be loaded in every time Touch Designer is run by TD_InitializePatch. This means a developer does not have to manually load in a file in the same manner that is done in TD_InitializePatch. This object should be set to “Load on Start” based off of current implementations.

AGD_LaunchGeneration_exec

This object is responsible for launching AGD_LaunchGeneration. This object is directly started from TD_InitializePatch. It may be possible to not use this execute and directly start AGD_LaunchGeneration from TD_InitializePatch, but this has not been tested yet. The file that controls this object is shared between this module and AGD_TimerCallbacks. This object should be set to “Load on Start” based off of current implementations.

AGD_StartTrigger

This object is a constant, that when switched from zero to one, starts AGD_Timer. There are no special settings required for this object.

AGD_Timer

This object acts as a timer that counts down for the desired video length. There is an associated callback (AGD_TimerCallback) that is used to actually shutdown Touch Designer once the desired generation has completed. All of the required parameters for this object are set in the AGD_ArtGeneratorInstance class when it is initialized.

AGD_TimerCallbacks

This object is responsible for closing down Touch Designer when AGD_Timer has reached its expected count. This object should be set to “Load on Start” based off of current implementations.

AGD_RecordOutput

This object is responsible for recording the generated art from Touch Designer. All of the required parameters for this object are set in the AGD_ArtGeneratorInstance class when it is initialized.

Object	File Path
TD_InitializePatch	<root>/code/TouchDesigner/src/TD_InitializePatch.py
AGD_LaunchGeneration	<root>/code/Backend/ArtGenerationDriver/src/AGD_LaunchGeneration.py
AGD_LaunchGeneration_exec	<root>/code/Backend/ArtGenerationDriver/src/AGD_Utilities.py
AGD_TimerCallbacks	<root>/code/Backend/ArtGenerationDriver/src/AGD_Utilities.py

Table 2: List of Files for Touch Designer Objects

Optional Objects

This section describes the optional objects that can be added to Touch Designer to simulate art generation without having to run the system using the Python software. Table 3 contains a list of all additional objects needed for the optional debugging functionality.

Object Name	Object Type
SIMStart	Button
IN	In
NULL	Null

Table 3: List of Objects for Debugging Functionality

// Insert stuff here

Patch Enumerations

In order to have a clean connection between the front-end, back-end, and Touch Designer, there needs to be a defined enumeration that correlates between users selecting art generation on the front-end, to what patch gets run in Touch Designer in the back-end. Table 4 highlights the proper enumeration that should be used by all code to align the front-end, back-end, and Touch Designer for art generation.

Patch	Value
none (debugging patch)	0
loop	1
shore	2
instance	3
hex_quake	4
watercolor	5

Table 4: Patch Enumerations

NOTE: The None patch should not be accessible by any user. This means that selections on the front-end must start as index at one.

Additional Information

Disabling Automatic Exit

Once Touch Designer has been integrated with the objects and Python control as mentioned in “Required Objects”, any Touch Designer patch will automatically exit after the fixed art generation length timer has completed. This can make developing or changing patches difficult. To disable this functionality, go to <root>/code/Backend/ArtGenerationDriver/src/AGD_Utilities.py and disable the line that says “quit()” by commenting it out. Re-Open the Touch Designer patch. It should no longer quit out automatically after the fixed

amount of time. When done, remove the comments from quit(), because it is needed for the full functionality of the system.

Touch Designer Configuration

In Touch Designer, the overall application can have various settings controlling aspects of speed and direction of art generation. Ensure for proper execution of a Touch Designer patch, that on the bottom toolbar, forward playback is enabled and looping is enabled. Additionally, enable real-time playback on the top toolbar.

Initially, it was thought that disabling real-time processing would decrease the amount of time to process graphics. However, what it actually does is slow down the generation to get better quality output. This is not desired as we are already taking around 5 seconds to generate art, which can lead to a clogged queue if many users want to generate art.