Unity Summer of Code Proposal

Project: Cutscene Editor
Prepared by: Matthew Miner

July 11, 2009

Table of Contents

Executive Summary

The Problem

Creating cutscenes in Unity, while not necessarily difficult for the seasoned developer, can be a challenge for those more familiar with traditional filmmaking tools and techniques. As game projects grow larger, it's likely that trained filmmakers — cinematographers, editors, etc. — will assist with the creation of these cutscenes. Currently there's no simple system for working with elements like multiple shots and transitions; that is, there's no easy way to "cut together" a scene.

The Solution

A **Cutscene Editor**, a tool for Unity with the ability to "capture" an animated scene from multiple viewpoints and edit it together. These individual clips are placed in a timeline similar to those seen in non-linear editors like iMovie and Final Cut. Clips can be rearranged, trimmed, split, slowed down, sped up, and given special effect filters like sepia tone. Transitions like crossfades can be added between clips as well as title cards and subtitles. Dialogue and sound effects can be added on their own section of the timeline. Ideally a developer can create a cutscene without writing any code.

Pre-rendered vs. Realtime

The Cutscene Editor will not capture the scene in an actual video. Instead, the cutscene will be rendered at runtime using multiple cameras and in-game models. There are several advantages to this approach: the visuals will appear consistent with the rest of the game, scenes can be easily modified at a later date, and there won't be the overhead associated with storing potentially large video files. And, of course, excellent tools already exist for editing together video files.

Initial Release Goals

Naturally this tool will become quite complex as features are added. For a release in the six week timeframe, some features will be limited. Existing Unity assets like image effects scripts will be used while other advanced functions like multiple video and audio tracks will be incorporated at a later date. With that said, the following abilities are a priority:

- Record a scene that can be reliably played back
- Organize clips on a timeline
- Clip modifications: trim, split, effect filters, transitions
- Subtitle system
- Audio: dialogue, sound effects

Design

Taking Inspiration From The Best

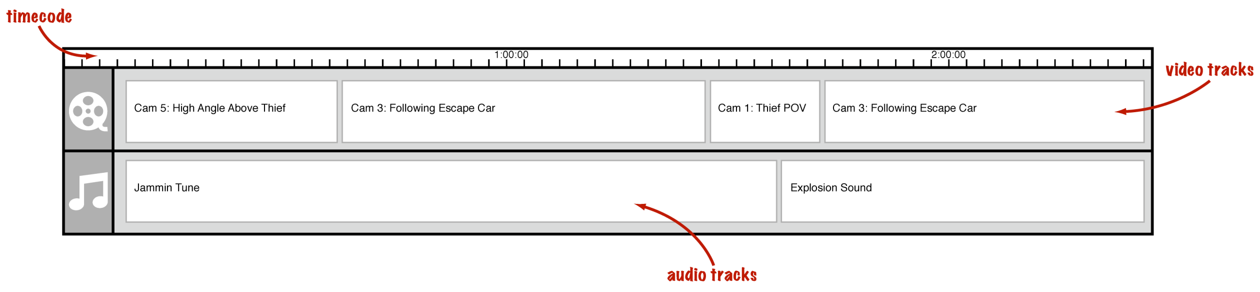
Users will create the movement how they normally would using Unity's built-in tools. They will then use the Cutscene Editor to place multiple cameras and switch between them. In order to create an environment that editors will instantly find familiar and intuitive, the design will draw heavily from non-linear video editing software like Apple's Final Cut. Also, much of the terminology used is taken from the world of video editing.

Graphical User Interface

The Cutscene Editor will be comprised of five separate windows: Timeline, Clips, Effects, Transitions, and Scene Preview. Some of these will be grouped in a tabbed interface to save screen space. Adding new cameras will be done from the Clips window or by dragging from the game object hierarchy. Subtitles will appear as an effect alongside visual filters. The tool will make use of drag-and-drop as much as possible.

Timeline

Arguably the most important part of the Cutscene Editor is the area where the clips are assembled. The bulk of the editor's time will be spent here. In this area clips can be reordered, trimmed, and split. There are two tracks: one for video and one for audio. A *timecode* shows how long the scene and individual clips are. In future versions it will be desirable to add the ability to have multiple video and audio tracks.



Editing Tools

Initially there will be two methods of modifying clips in the timeline.

1. A razor blade tool will split a clip at any arbitrary point. To use this tool, hover the cursor over the part of the clip where a split is desirable and click. The split will result in two separate clips which can be moved or deleted.
2. Hover the cursor over the front or the end of a clip and drag to extend or trim.

Camera Browser

This is where camera clips are stored, represented visually with a thumbnail. Clicking on one will show a preview of it in the Scene Preview window. There will be a button at the bottom to add a new camera/clip. Animating a camera will be up to the user using Unity's animation system.

Effect Filters

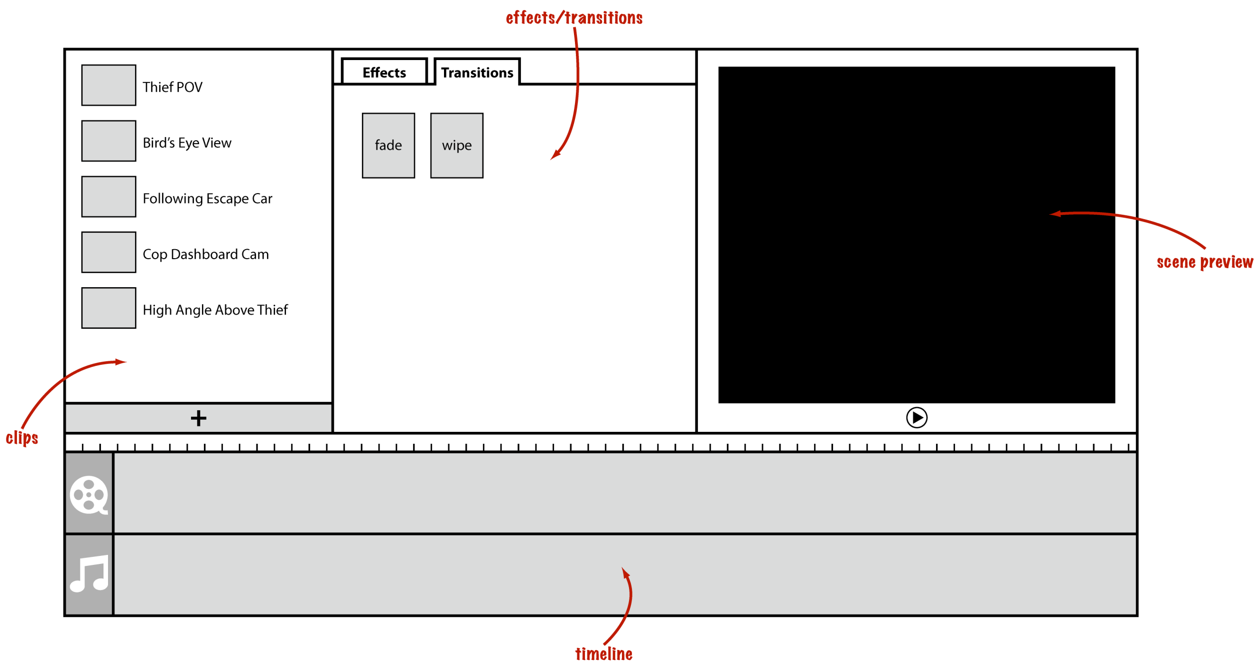
These are essentially full-screen post-processing effects, and as such will only be available to Unity Pro users. The user will be able to drag and drop the effect on a clip, which behind the scenes simply adds the script to the camera. Unity Pro already ships with several effects scripts, so virtually no work will need to be done to make these work.

Transitions

Several camera transitions like a crossfade already exist on the Unify Community Wiki (example). These will be presented to the user visually as draggable elements that can be placed between clips. When this happens, the transition script is attached to the first camera and references the second one.

Scene Preview

This window shows a preview of what the cutscene looks like. The Timeline will tell the Scene Preview which camera to render. It includes basic playback controls like stop.



Tentative Schedule

Date	Tasks
Week 1	Record camera views as selectable and playable “clips”
Week 2	Ability to drag and drop clips on a timeline
Week 3	Clip functions: trim, split; Scene Preview
Week 4	Audio: ability to add dialogue and sound effects
Week 5	Effects, transitions, subtitle system
Week 6	Tweak and polish user interface; fix inevitable bugs

Easy Tasks

- Visual effects; scripts already available
- Transitions; scripts already available, will simply need tweaking
- Subtitle system; will rely on GUI.Label

Sorta-Difficult Tasks

- Camera/clip management
- Audio management

Difficult Tasks

- Timeline functions / clip editing
- Scene Preview
- Making Cutscene Editor easy-to-use for non-developer