User Manual - Quick Cutscene Creator

Welcome to the Quick Cutscene Creator, a plugin by Gadget Games. By choosing QCC, you have the power to create detailed, high-performance cutscenes and camera paths quickly and easily.

The Quick Cutscene Creator is designed to integrate intuitively into your Unity workflow, allowing for fast iteration and flexible authoring options.

Contact links

support@gadget-games.com

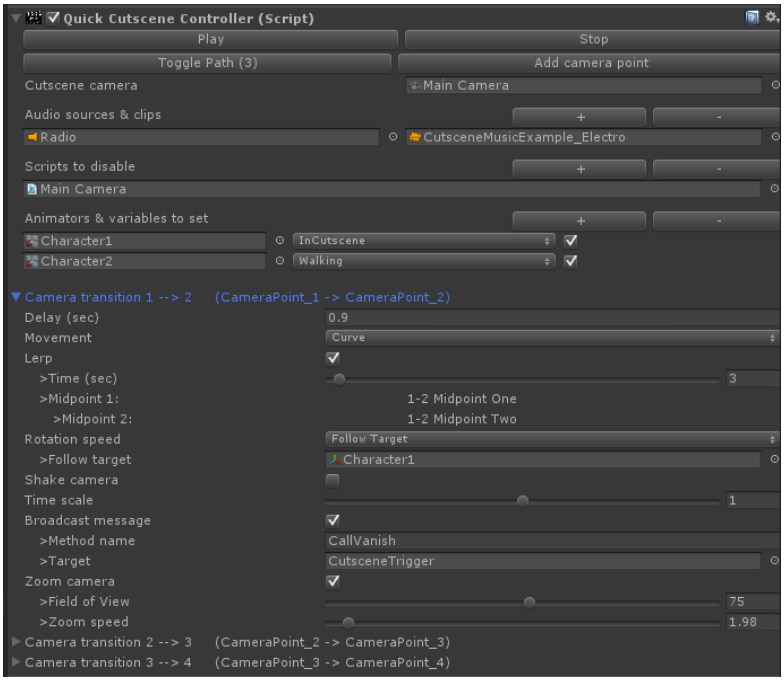
Forum thread

Gadget Games on Twitter

Quick start guide

1. Import the Quick Cutscene Creator asset to your project
2. Drag the Example Cutscene prefab into your scene
3. Adjust the locations of the child transforms of the Example Cutscene prefab to your desired camera points, add or remove transforms according to how many camera points you need
4. In the Quick Cutscene Controller inspector, fill in any extra options if desired
5. In the Quick Cutscene Controller inspector, modify each camera transition according to your needs
6. Run your game, and test out the cutscene with the “Play” button on the Quick Cutscene Controller inspector

Inspector GUI overview



Buttons:

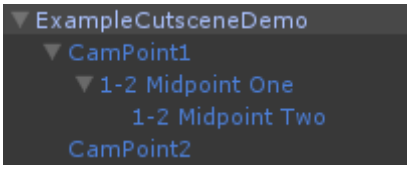
- *"Play" Button*
  Plays the cutscene. Game must be running
- *"Stop" Button*
  Stops the cutscene if it is playing
- *"Toggle Path" Button*
  Toggles between *no path*, *positional path*, and *full path*. The path is drawn in the editor
- *"Add Camera Point" Button*
  Creates a valid camera point in the cutscene hierarchy.

Start modifiers:

- *Cutscene camera*
  The camera that will be manipulated during the cutscene Defaults to the Main Camera if given no input.
- *Audio sources & clips*
  Audio clips to be played from corresponding audio sources at the beginning of the cutscene
- *Scripts to disable*
  The Monobehaviours to be disabled at the start of the cutscene, and re-enabled at the end of the cutscene
- *Animators & variables to set*
  The Animator Controller to modify, the boolean variable to be modified, the value to be set (at the start of the cutscene, and reset at the end)

Transition modifiers:

- *Delay (sec)*
  Delay (in seconds) before starting this transition
- *Movement*
  Camera movement options.
  - *Slow*, *medium*, *fast* and *faster* are pre-set movement speeds for linear movement
  - *Custom* lets you tweak the linear movement speed
  - *Instant* moves the camera instantly to the next camera point
  - *Mobile curve* is a mobile-friendly bezier curve option, which requires a bezier curve mid-point (a child transform of the first transition point)
  - *Curve* is a cubic bezier curve, which requires two bezier curve mid-points (example hierarchy shown to the right)
  - *Lerp* moves to the target position over a guaranteed period of time. **Recommended** for games without a set framerate.



Correct transform hierarchy setup of a cubic curve between two camera points

- *Rotation speed*
  Camera rotation options.
  - *Super slow* through to *super fast* are pre-set rotational speeds
  - *Custom* lets you tweak the rotational speed of the camera
  - *Follow target* lets you set a target transform to look at rotationally during that transition
  - *Lerp* rotates to the target rotation over a guaranteed period of time. **Recommended** for games without a set framerate.

- *Shake camera*
  If ticked, applies camera shake during that transition. *Shake intensity* slider increases intensity of camera shake.
- *Time scale*
  Slows down or speeds up time during that transition. Audio is also sped up/slowed down.
- *Broadcast message*
  Calls the method in "*Method name*" on the *target* GameObject.
- *Zoom camera*
  Moves the cameras FoV to the desired value over time

Public methods

- **ActivateCutscene()**
  - Call ActivateCutscene() on a Quick Cutscene Controller script to start that cutscene.

- **EndCutscene()**
  - Call EndCutscene() on a Quick Cutscene Controller script to end that cutscene.

Event extensions

- **OnCutsceneStart()**
  - Sent to the local GameObject when the cutscene has been started

- **OnCutsceneEnterTransition(int transitionNumber)**
  - Sent to the local GameObject when the cutscene enters a new transition (transitionNumber)
- **OnCutsceneEnd()**
  - Sent to the local GameObject when the cutscene ends

Common issues

- I can't select the correct script to disable in the inspector
  - Follow the instructions here: http://answers.unity3d.com/questions/36131/editor-multiple-inspectors.html to set up multiple inspectors, then drag the desired scripts to disable from the GameObject into the Quick Cutscene Controller.
- My camera is not moving/stuck
  - Make sure all camera points are child transforms of your Quick Cutscene Controller game object
  - Make sure you have set other camera-controlling scripts to be disabled via the Quick Cutscene Controller

Addons

Example addon scripts are included with the Quick Cutscene Creator, which use the extension methods provided to add extra functionality to your cutscene and workflow if needed.

- **QCNavigatorAddon**
  Adds support for NavMeshAgents, to allow for Moving to a destination, Warping, and Stopping at key points in the cutscene.
- **QCAnimationAddon**
  Adds support for more Animation actions at events in the cutscene.
- **QCLegacyAnimationAddon**
  Adds legacy animation support options, to allow for playing & crossfading of legacy animations at key points in the cutscene. **Note:** The QCLegacyAnimationAddon script does work as intended, but because of the way Unity handles legacy animation states & clips, Unity cannot guarantee legacy animation clip order at runtime, so they cannot be generically accessed via their index. This is why the visible clip values in the inspector GUI will change at runtime.
- **QCSceneViewCameraFollow**
  Moves a scene view camera to the target object position. Useful for debugging.

Resources

Thanks to Props Animations (https://www.assetstore.unity3d.com/#/content/10214) for the breakdancing animations included in the demo scene, be sure to check them out on the Asset Store!