ANIMATOR

Integration with Rendering Software

FREDO6 – V1.0 – 31 MAR 2016

1. Introduction

Animator is a script dedicated to the animation of Sketchup models. It provides a parametric, interactive framework to control movements of objects and cameras along a timeline.

Animator includes a feature to generate videos from the animation, at any given frame rate and image dimensions, in various formats (sequence of images, MP4, MOV, etc….). The mechanism is based on moving through each frame of the animation and capturing the image (via *view.write_image*). The stitching of individual images into a video file is then performed by **ffmpeg**, which is a free open source program.

For rendering an animation, the simple idea is that **Animator would request the Rendering Software to generate the image for each frame**. So instead of calling *view.write_image*, Animator would call a method of the Rendering software that would generate the image. The stitching of image and generation of the video would remain the same and done by Animator.

The Animator framework is flexible enough, so it can accommodate specific namings and level of functionality in the Rendering software.

The present document is a **guideline** for this type of integration, which requires that the Rendering software exposes some methods to be called by Animator:

- Mandatory
  - `render_viewport`
- (strongly) Advised
  - `render_stop`
  - `render_set_dimensions`

2. Method to Render the Viewport into an Image File

This method will be called by Animator for each frame of the animation. When it is called, the viewport is updated for the view and object position. Its task is to **generate a rendered image of the viewport into a specified image file** (***png*** preferably or ***jpg***).

The name of the method is left to the Rendering Software. Animator can adapt. We use *'render_viewport'* for illustration.

2.1. Minimum specification

The minimum specification of this method should be:

```
def render_viewport(image_file_path, &finish_proc)
```

where

- `image_file_path` is the absolute path to the image file to be created.
- `finish_proc` is a callback to tell Animator when the rendering of the viewport is terminated and the image is ready, so that Animator can go to the next frame. The callback has no argument.

The generation of the image can take long. So it is assumed that the rendering is processed asynchronously in a thread or so. This is why a callback is important to indicate completion. Please advise if not the case.

2.2. Variants and Other Methods

There may be some flexibility and improvements in this method, as suggested below:

1) **Stop rendering**: Animator allows to interrupt the process for several purposes:
    a. **Abort** the process
    b. **Suspend** the process and then resume it
    c. **Preview** the video based on frames processed so far, and then either abandon or resume generation

   In all cases, it is helpful that Animator can tell the rendering software to **stop the current rendering**, especially for case c. So a method `render_stop` would be welcome.

2) **Image dimensions and other parameters**: Animator can handle the two cases where:
    a. *dimensions are set in Animator and passed to the rendering method*. In this case, the rendering method should take a 2nd argument as a Hash array:

       ```
       render_viewport( image_file_path, hparams, &finish_proc)
       ```

       with hparams containing the specified dimensions as **:dimX** and **:dimY**

       With a Hash array, it is possible to pass extra information that would be useful to the rendering software, for instance to ensure the homogeneity of rendering for all frames of an animation.

       Alternatively, dimensions could be set by a separate call, since they are basically the same for the whole rendering process.

       ```
       render_set_dimensions(dimX, dimY)
       ```

    b. *dimensions are not set in Animator and determined in the Rendering software GUI*. This is the minimum requirements.

       VERY IMPORTANT: due to a limitation in **ffmpeg**, the images must be generated with an **EVEN** resolution in X and Y. If one of the dimension is an odd number of pixels, ffmpeg will fail to generate the MP4 video file.

3) **Error and Progression indicator**. The minimum requirement is to know when the image generation has finished. However the same callback could be used to give:
    a. An indication of error. This is useful, because very likely, the whole video generation process should stop.
    b. An indication of progress (as a % of completion or any useful metrics)

   It is therefore useful to have the callback take one argument, which could be for instance:

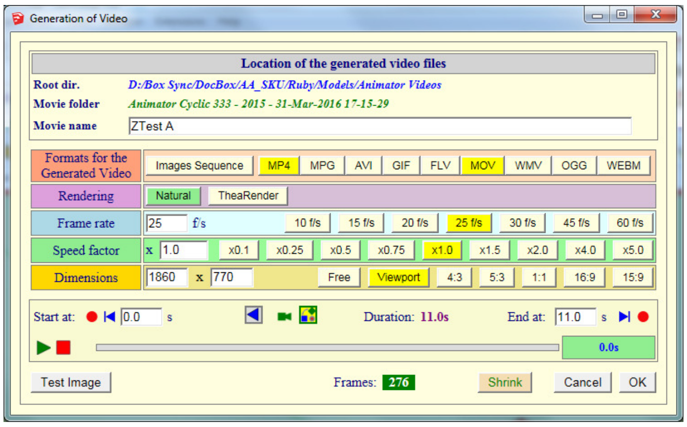
   – **:finished**, for completion
   – **:error**, when error happened (rendering should terminate too)
   – **A string message** that could be displayed to the user by Animator

3. Appendix: Workflow for Generating Videos in Animator

In Animator you can generate videos for pieces of animation and the whole animations. This is indicated by a small button in the palettes.

The dialog box for generating a video is shown below:

1) **In Natural rendering** (i.e. NO rendering)



Generation of Video

Location of the generated video files

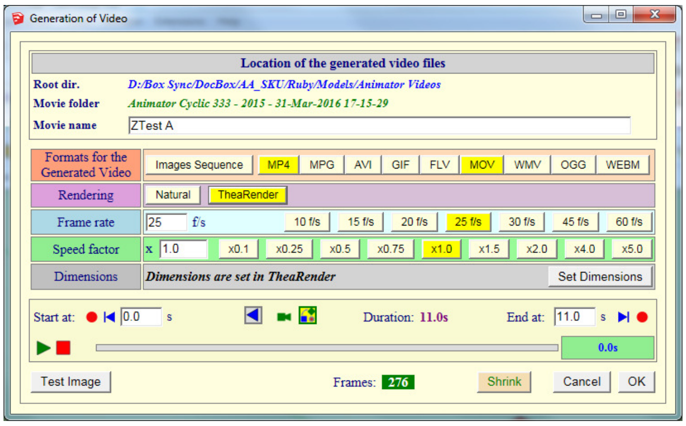
Root dir.	*D:/Box Sync/DocBox/AA_SKU/Ruby/Models/Animator Videos*
Movie folder	*Animator Cyclic 333 - 2015 - 31-Mar-2016 17-15-29*
Movie name	ZTest A
Formats for the Generated Video	Images Sequence, MP4, MPG, AVI, GIF, FLV, MOV, WMV, OGG, WEBM
Rendering	Natural, TheaRender
Frame rate	25 f/s — 10 f/s, 15 f/s, 20 f/s, 25 f/s, 30 f/s, 45 f/s, 60 f/s
Speed factor	x 1.0 — x0.1, x0.25, x0.5, x0.75, x1.0, x1.5, x2.0, x4.0, x5.0
Dimensions	1860 x 770 — Free, Viewport, 4:3, 5:3, 1:1, 16:9, 15:9

Start at: 0.0 s Duration: 11.0s End at: 11.0 s

0.0s

Frames: 276

2) **With Rendering** (here TheaRender), dimensions inputs are replaced by a message to indicate that dimensions are set directly in TheaRender.



Generation of Video

Location of the generated video files

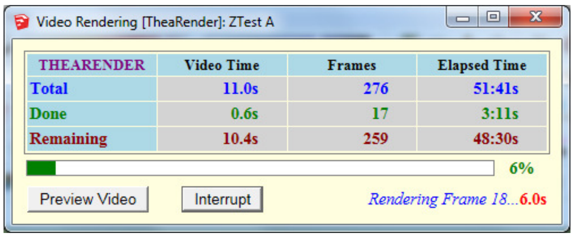
Root dir.	*D:/Box Sync/DocBox/AA_SKU/Ruby/Models/Animator Videos*
Movie folder	*Animator Cyclic 333 - 2015 - 31-Mar-2016 17-15-29*
Movie name	ZTest A
Formats for the Generated Video	Images Sequence, MP4, MPG, AVI, GIF, FLV, MOV, WMV, OGG, WEBM
Rendering	Natural, TheaRender
Frame rate	25 f/s — 10 f/s, 15 f/s, 20 f/s, 25 f/s, 30 f/s, 45 f/s, 60 f/s
Speed factor	x 1.0 — x0.1, x0.25, x0.5, x0.75, x1.0, x1.5, x2.0, x4.0, x5.0
Dimensions	*Dimensions are set in TheaRender* — Set Dimensions

Start at: 0.0 s Duration: 11.0s End at: 11.0 s

0.0s

Frames: 276

The generation is then launched. A floating dialog box shows the **progress**:



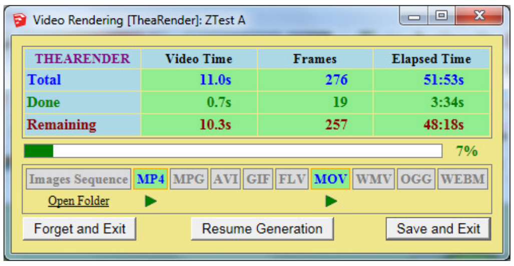
Video Rendering [TheaRender]: ZTest A

THEARENDER	Video Time	Frames	Elapsed Time
Total	11.0s	276	51:41s
Done	0.6s	17	3:11s
Remaining	10.4s	259	48:30s

6%

Rendering Frame 18...6.0s

At the end of the process, or when you chose to **preview the video** during the process, you get a dialog box allowing to add / remove extra video formats and play selected video files.



Video Rendering [TheaRender]: ZTest A

THEARENDER	Video Time	Frames	Elapsed Time
Total	11.0s	276	51:53s
Done	0.7s	19	3:34s
Remaining	10.3s	257	48:18s

7%