Course syllabus

DATE	24/07/2021
PRINCIPAL AUTHOR	Justin Avery
OTHER AMENDS	
VERSION	1.0

Overview

The purpose of this document is to define the Swift CG+ training course, syllabus and materials for those personnel tasked with authoring graphics templates for delivery to tx from RT standalone workstations/render heads.

About Swift CG+

Swift CG+ is deep-level GUI based 3d graphics editor and assembly environment. This tool is used for both virtual and augmented reality studios as well as automated and operator-driven overlay graphics. Swift CG+ provides a high-end, sophisticated toolset that enables the templatisation of branded graphical content for multiple use-case scenarios. Swift CG+ utilises a lens-based, xyz workspace paradigm that allows designers to create templates using a real world coordinate system.

Who uses Swift CG+

This template designer will need an apt range of skills and experience. Individuals that are tasked with and undergo training with this software option are generally those with the following skills and experience:

- Familiarity with or experience of Autodesk, Newtek or Maxon 3d products such as 3dsMax, Maya, Cinema4d or Lightwave
- Live operation of 3d render engines such as Viz, Chyron, Brainstorm
- Visual fx editing or animation in a post production environment

About the course

This course is aimed at broadcast operational staff and oriented towards developing essential skills, knowledge and expertise for graphics template creation *specifically* for live broadcast operations - in particular sports events and programming.

The types of graphics that are authored with Swift CG+ are typically overlay graphics, augmented reality and virtual studio sets.

Course primary objectives

At the end of the training period, delegates will:

- Have acquired the skills to use the Swift CG+ graphics editor to create graphic templates
- Have explored the wide potential of Swift CG+ editor features including:
    - Animation toolsets
    - External asset compatibility
    - Adding functionality
    - Applying user-code within templates
- Understand import processes for 3d geometry from a range of content creation applications
    - File formats
    - Compatibility
    - Textures and coordinates
    - 3d spacial data
- Author, install and test a project with particular attention to critical:
    - Nomenclature
    - Files, paths and locations
    - Data sources
- Develop an understanding of real space versus virtual space concerns in the design process:
    - Working with real space coordinate systems
    - Appreciate the importance of tracking systems

Session 1: Swift CG+ Intro

This video tutorial provides an overview of the Swift CG+ interface for first-time users.

- Materials: Syllabus; Swift CG+ Manual; Swift CG+ application
- Target level: All new and L1 (Level 1) users

Session specific objectives

1. Gaining familiarity with the Swift CG+ interface, layout and toolsets

Chapter 1: Creating projects and templates

- Creating, Saving and Opening projects and scripts
- Understand project directory and folder structures
- Project footprint, path and filetypes

Chapter 2: Layout and Preferences overview

- Interface areas
    - Preview; Scenegraph; Editor; Timeline
- Top row Icons, Browsers and Menu options
    - Project related; Undo redo; Safe frames and transparency settings
- Preferences
    - Overview

Chapter 3: Primitives browser

- Drag and drop
- Add primitives

Chapter 4: Preview navigation and input

- Coordinate system
- Navigation and interaction
    - Viewports and global view
    - ALT + mouse option
- Scales and positions, resets

Session 2: Scenegraph & shader basics

This video tutorial provides an overview of the Swift CG+ scenegraph tree.

- Materials: No external assets required
- Target level: All new and L1 (Level 1) users

Session specific objectives

1. Gain familiarity with basic scenegraph nodes: camera and light
2. Shader basics
3. Per pixel lighting - basic overview

Chapter 1: Scenegraph camera and light nodes

- Scenegraph nodes
- A quick look at camera properties
- Changing light types
- Activating a light

Chapter 2: Scenegraph right click menu

- Right click menu
    - Rename
    - All items overview

Chapter 3: Shaders - basic overview

- Create a new shader
- Shader Materials
- Shader States
- Shader Texture Slots

Chapter 4: Material properties

- Diffuse, Ambient, Specular, Emission and Shininess

Chapter 5: Per pixel lighting basics

- What per pixel lighting involves
    - Advantages and disadvantages

Session 3: Lower third part 1: Set up

This video tutorial guides users in the creation of a simple lower third graphic.

- Materials: No external assets required
- Target level: All new and L1 (Level 1) users

Session specific objectives

1. Using a cube and text node to create a simple lower third

Chapter 1: Editing a cube

- Editing the cube from the TRFM node

Chapter 2: Text format tools - basic

- Adding a text node
- Text strings and font
    - Text sizes
    - Alignment and justification
    - Character, word and line spacing
    - Defining Max-X sizes
    - Y scale fixing
    - Defining Auto Wrap sizes

Chapter 3: Drops shadows, grouping and parenting

- Setting the drop shadow
- Adding Groups and Transforms
- Moving branches
- Scenegraph managing

Chapter 4: Position the Lower Third

- Position the cube parent for the strap

Session 4: Lower third part 2: Animation

This video tutorial explores basic animation features for use in a simple lower third graphic.

- Materials: No external assets required
- Target level: All new and L1 (Level 1) users

Session specific objectives

1. Familiarisation with animation principles in Swift CG+. Using the Curves Editor.

Chapter 1: Animation concepts

- Animators
- Methods
    - bringOn
    - takeOff

Chapter 2: Animation concepts and Timeline Editor tab L1

- Adding an animator
    - Timeline animator colours - red dot and bar
- Timebase and timecode readout - frames
- Icon options and animator ordering
- Sliding and changing in and out points
- Widen interface

Chapter 3: Curves Editor tab L1

- Path keyframe defaults
- Keyframe fields - CurrField and CurrValue
- Bezier handle editing
- Class types - Path and Linear
- Transport controls
- Scrubbing
- Copying and flipping animators
- Saving at bringOn
- Refresh with Reload Current Script

Session 5: Lower third part 3: Inputs

This video tutorial looks at the how template designers expose values to control applications.

- Materials: No external assets required
- Target level: All new and L1 (Level 1) users

Session specific objectives

1. Adding dynamic inputs to the simple lower third.
2. Generating Interfaces to expose values for control applications.
3. Updating the project manifest files.

Chapter 1: Input features L1

- Adding inputs on the bringOn to text nodes
- Source types
- Destination types
- Node and field parameters
- Default values
- Input names and nomenclature

Chapter 2: Generating an Interface

- How and why

Chapter 3: How and why to Update Project Manifests

- Looking at the file with a text editor
- Rationale

Session 6: Lower third part 4: Playout

This video tutorial provides an overview of the Playout Editor.

- Materials: No external assets required
- Target level: All new and L1 (Level 1) users

Session specific objectives

1. Using the Playout interface for testing.

Chapter 1: Playout overview

- What is Playout?
    - Why is it required?
- Interface areas
    - Graphics tray
    - Methods tray
    - Script Stack
- bringOn > takeOff
- Dynamic GUI tray
- Take - cycling through the Stack

Chapter 2: Stack creation, updating and Playout

- Template population on the method with inputs
- Changing dynamic content
- Stack population
- Moving Stack items up and down
- Deleting Stack items
- Saving and loading Stacks
- Abort and Clear Stacks

Session 7: Blocking and multiple Methods

This tutorial examines blocking and non-blocking in templates as well as templates with more than 2 Methods.

- Materials: No external assets required
- Target level: All new and L1 (Level 1) users

Session specific objectives

1. Using blocking features and multiple methods.
2. Using the object node order number

Chapter 1: Using multiple methods

- Multiple methods: Rationale

Chapter 2: Adding new blocks

- Creating a new block
- Multiple blocks: Rationale

Chapter 3: Blocking versus non-blocking

- When and why

Session 8: Transition Logic

This tutorial examines Transfer Logic with Default and Persistent Object Node states.

- Materials: No external assets required
- Target level: All new and L1 (Level 1) users

Session specific objectives

1. Using the object node order number
2. Persistent templates.

Chapter 1: Object node

- Renaming
- What and how does Persistence work

Chapter 2: The F9 Playout view

- Invoking the scenegraph view in Playout
- How does Persistence work

Chapter 3: Adding step animators to reset Persistence

- Why, when and where

Session 9: Importing 2d assets

This video tutorial details the Import feature for still images and font assets.

1. Materials: Session09.zip;
2. Target level: All new and L1 (Level 1) users

Session specific objectives

1. Importing 2d assets
2. Importing 2d fonts as geometry and extruded

Chapter 1: Importing and placing still images

- Standard import
- Creating a new Shader from the Default Shader
- Assigning to a rectangle

Chapter 2: Texture tab settings:

- Blending mode sub menus
- Position sub menus
- Filter and Mapping sub menus
- Format sub menus

Chapter 3: Mixed resolution images

- From 720 to 8k images

Chapter 4: Images with alpha and transparencies

- Importing with “Create Shader for Each Image” setting
- Interpreting and setting alpha channels

Chapter 5: Managing images in 3d space

- Z fighting
- Scenegraph ordering
- Pre multiplication

Chapter 6: Importing fonts

- Compatible font files for import (ttf;otf)
- Using and wrapping images onto text
- Importing fonts as Extruded

Session 10: Controlling movies

This video tutorial details the features for imported movie clips.

- Materials: Session10.zip
- RT_ Supported Formats & Filetypes.pdf

Session specific objectives

1. Movie clip controls and Advanced Shader Editor features L2

Chapter 1: Importing movie clips

- Importing clips

Chapter 2: Movie Pause on Load

- Why and how to set

Chapter 3: Step Animator curve types

- Adding a StartMovie step animator

Chapter 4: Looping movies

- Setting the number of loops in the Animation Sub-Menu

Chapter 5: Reset and Start values

- Making a StartMovie animator reset at the same time

Chapter 6: Reset Movie animators

- Using the cueGraphic for automation

Chapter 7: Movies with alpha channels and transparencies

- Identifying and setting the correct transparency settings

Chapter 8: Pausing movies

- Using a StopMovie step animator

Session 11: Importing 3d assets

This video tutorial details the Import feature for 3d geometric assets.

Materials: Session11.zip (cube.fbx)

Target level: All new and L1 (Level 1) users

Session specific objectives

1. Importing fbx files
2. Geometry nodes
3. Automated nomenclature
4. Images as textures

Chapter 1: Importing fbx files

- Into a new graphic
- Origin
- Scaling and parenting

Chapter 2: Naming and automated nomenclature

- Object node name
- Shader, material, texture and state names

Chapter 3: Updating material and texture properties

- Changing diffuse colours and adding specularity
- Changing textures for new images

Chapter 4: Basic reflection maps

- Using the Additive transfer mode
- Spherical Image Mapping

Chapter 5: Normals and Wireframe views

- Using the Normals view
- Using the Wireframe view

Session 12: Duplicate nodes

Duplicate nodes are a powerful feature in Swift CG+ that enable information to be copied in a variety of ways. Typical graphics that utilise Duplicate Nodes are Full Form tabular graphics such as a league table, scoreboard or listing.

- Materials: Not required

Session specific objectives

1. Using duplicate nodes and introducing Clip Plane nodes

Chapter 1: Example duplicate

- Prep for a tabular template
- Parenting
- Naming

Chapter 2: Editing text

- Positions and maxX sizes

Chapter 3: Clip Plane nodes

- Using Clip Planes as masking tools
- Editing and viewing Clip Planes
- Translating and rotating Clip Planes

Chapter4: Adding inputs to Duplicate nodes

- Set numbers
- Vertical offsets

Chapter5: Animation

- Animation delays

Chapter6: Inputs

- Tilde separation

Chapter7: A detailed example

Session 13: Using Links L1, L2

Links are a powerful feature in Swift CG+ that enable the linking of parameters of one node to any set of parameters on another node. A typical link would be one that sets the size of a rectangle - that is placed beneath some text - to always be the same width as the text that is entered above it.

- Materials: As below `($1+$2)/2`

Session specific objectives

1. Text editing: Styles
2. An overview of Linking features
3. Using links for 3 different use cases

Chapter 1: Text Styles: Plinths

- Offsets and margins
- Shaders
- Drop shadows
- Z depth issues

Chapter 2: Linking 1

- Rectangles linking to text widths
- Toggle Basic Editor
- Source and destination
- Bias

Chapter 3: Linking 2

- Two dynamic text strings linked and centered
- Group Function: Expressions

Chapter 4 Linking 3

- Linking a plinth to two dynamic text strings
- Group Function: SUM

Session 14: Tickers

Tickers form an ever more significant part in broadcast information - especially social media related content.

- Materials: None
- Target level: All new and L1 (Level 1) users

Session specific objectives

1. Creating a ticker, populated in the contents, fed by a manual text string.

Chapter 1: Creating a ticker

- Ticker nodes tabs
- Changing fonts
- Number of Cycles parameter

Chapter 2: Editing the ticker

- Spacing, Limits and Speed options

Chapter 3: Adding inputs to the Contents

- Where to add the input

Chapter 4: Adding a Restart Ticker animator

- Using a New Block

Chapter 5: Adding images between slugs

- Using icons between slugs

Session 15: Standard clocks

Standard clocks in Swift CG+ automatically take system time from the render machine.

- Materials: None
- Target level: All new and L1 (Level 1) users

Session specific objectives

1. Using clocks

Chapter 1: Adding a clock node

- Setting the font
- Using clock formats to format the clock time

Chapter 2: Adding offsets for different regions

- Adding an offset using an input

Chapter 2: Countdown and count-up clocks for less than 24 hours

- Absolute clocks
- Supplied time calculations
- Start time and end times

Session 16: Exporting a graphic

Establishing the workflow rationale behind exporting templates from one project to another.

- Materials: None
- Target level: All new and L1 (Level 1) users

Session specific objectives

1. Exporting a graphic from one project to another.

Chapter 1: Exporting a template

- Rationale - multiple designers working separately on 1 project
- Export menu options
- Evalu ating the Folder contents
- Importing the template
- Conflicts on import
- All from project or import

Session 17: User code

This tutorial examines the use of user code to extend the capabilities of templates in Swift CG+.

- Materials: Session16.zip
- Target level: All new and L1 (Level 1) users

Session specific objectives

1. Adding simple ruby user code to extend logic in templates

Chapter 1: Forcing the case of text strings to either upper or lowercase

- User code tab
  - Ruby scripting
  - Session materials
- Example
  - _textUpper = _textUpper.upcase
  - _textLower = _textLower.downcase

Chapter 2: Resizing a word on a text string

- Example
  - _text.gsub!(/am/,"<sv 30.0>am</sv>")

Chapter 3: Resizing a word on a text string and forcing upper case

- Example
  - _text.gsub!(/am/,"<sv 30.0>AM</sv>")
  - _text.gsub!(/AM/,"<sv 30.0>AM</sv>")

Chapter 4: Concatenating two text strings with different fonts

- Example
  - _userName = "<f FrutigerNext_LT_MediumCn_Bold>" + _userName + "</f> " + _userID + "<f FrutigerNext_LT_MediumCn_Regular>"

Session 18: External data L1

This tutorial examines the use of external files: .txt; .xml; .xls

- Materials: Session22.zip
- Target level: All new and L1 (Level 1) users

Session specific objectives

1. Understanding differing external data usage and requirements in Swift CG+ templates: L1

Chapter 1: Populating a template from a standard text file

- Paths and locations, subfolders and directory structures
- Common uses and example

Chapter 2: Populating a template from a .xml file

- Common uses and example
- Paths and locations, subfolders and directory structures
- Creating the query string

Chapter 3: Populating a template from a .xls file

- Installing ruby
- Installing gems
- Paths and locations, subfolders and directory structures
- Common uses and example
- Creating the query string

Session 19: External data L2

This tutorial examines the use of mysql databases.

- Materials: Session23.zip
- Target level: All new and L1 (Level 1) users

Session specific objectives

1. Understanding differing external data usage and requirements in Swift CG+ templates: L2

Chapter 1: Populating a template from a mysql database

- Installing mysql
- Adding the database to the project
- Basic query statements
- Tables
- Rows and columns

Chapter 2: Creating and using a libs script

- Common uses and example
- Paths and locations, subfolders and directory structures
- Creating the query string

Session 20: Preset Methods

This tutorial examines the use of mysql databases.

- Materials: Session23.zip
- Target level: All new and L1 (Level 1) users

Session specific objectives

1. Understanding differing external data usage and requirements in Swift CG+ templates: L2

Chapter 1: Populating a template from a mysql database

- Installing mysql
- Adding the database to the project
- Basic query statements
- Tables
- Rows and columns

Chapter 2: Creating and using a libs script

- Common uses and example
- Paths and locations, subfolders and directory structures
- Creating the query string

Session options:

AR Player Profile deep-dive

Creating a player profile AR template.

- Materials: Session12.zip; scene.fbx
- Target level: All new and L1 (Level 1) users

Session specific objectives

1. Using all course set objectives to construct an AR working template

Chapter 1: Preparation

- Importing assets
- Workflow
    a. Creating parents
    b. Arranging scenegraph
    c. Editing lights and shaders
    d. Adding reflection maps
    e. Adding text
    f. Adding methods
    g. Adding inputs
    h. Creating animations
    i. Preset method setup
    j. Testing in Playout

Session options:

Mattes & masters

This tutorial defines the methods behind the creation of - and the rationale behind - creating a matte and master to streamline performance for clips that contain transparency and high bitrates.

Materials: Session19.zip

Target level: All new and L1 (Level 1) users

Session specific objectives

1. This tutorial defines the methods of reducing data rates for clips that contain transparency.

Chapter 1: Matte and master rationale

- A look at an example
- Reduce performance bottlenecks
- Reduce data rates
- Optimise graphical performance

Chapter 2: Creating a matte and master from prepared assets

- Using the Matte Shader option on preview drag

Chapter 3: Creating a matte with FFMPEG

- Creating a matte using ffmpeg (gray 256 col)

Session options:

DVE templates

In this video tutorial the DVE functionality of Swift CG+ is explored by the creation of a squeezeback template. These templates are one of the typical mechanisms used in automation for channel branding and promotion of upcoming programmes in a schedule. Squeezebacks generally occur at the end of programmes when the end credits roll. The duration of DVE moves vary between 10" and 30".

- Materials: Session08.zip
- Target level: All new and L1 (Level 1) users

Session specific objectives

1. Understanding Video Screen Nodes and their integration hardware i/o options.

Chapter 1: Assembling a squeezeback from supplied visuals

- Importing assets
- Using position guides

Chapter 2: Editing Video Screen Nodes

- Setting Channel numbers for your hardware

Chapter 3: Setting loops on a video clip to infinite

- Advanced Shader editor > Texture > Animation

Chapter 4: Animating Video Screen Nodes

- Adjusting start points
- Testing in the Playout interface

Session options:

Using audio

Using audio nodes for controlling audio associated with incoming video and audio spot fx from project asset files.

- Materials: Session09.zip
- Target level: All new and L1 (Level 1) users

Session specific objectives

1. Audio node features and settings.

Chapter 1: Adding a .wav file to the project

- Importing assets
    - audio48kTest.wav
- Audio features
- Audio associated with incoming video channels

Chapter 2: Audio Input Groups for video channels

- Cross fades

Chapter 3: Controlling .wav files

- Setting the file path
- Using the volume control

Session options:

Image sequences for TMV’s

TMV’s are native movie files for the RT render engine. TMV’s are generated during the Import process when used with image sequences such as a TGA or PNG stream.

- Materials: Session14.zip
- Target level: All new and L1 (Level 1) users

Session specific objectives

1. Ingesting image sequences as TMV’s; When and when not to use 3TMV’s.

Chapter 1: Importing an image sequence

- Importing assets
- Setting the Field/Frame versus Value correctly

Chapter 2: Determining the optimum times to use TMV files

- Decision matrix: Please select the Materials tab above for further details

Chapter 3: Scrubbing movie clips

- ToggleClipScrub

Session options:

Using clock nodes to schedule methods

TMV's are native movie files for the RT render engine. TMV's are generated during the Import process when used with image sequences such as a TGA or PNG stream.

- Materials: Session14.zip
- Target level: All new and L1 (Level 1) users

Session specific objectives

1. Ingesting image sequences as TMV's; When and when not to use 3TMV's.

Chapter 1: Importing an image sequence

- Importing assets
- Setting the Field/Frame versus Value correctly

Chapter 2: Determining the optimum times to use TMV files

- Decision matrix: Please select the Materials tab above for further details

Chapter 3: Scrubbing movie clips

- ToggleClipScrub

Session options:

Using Texture Fonts

Importing a font as a texture font enables the use of drop shadows and glows.

- Materials: tbc
- Target level: Advanced users

Session specific objectives

1. Importing, renaming, processing and overwriting texture fonts

Chapter 1: Importing and renaming texture fonts

- Character sizes
- Character images

Chapter 2: Processing fonts

- Using external applications to blur font character images

Chapter 3: Overwriting existing fonts

- Linking string to string
- Linking required parameters

Session options:

Setting a matte on a geometry or still

Defining a matte for a rectangle using an image as a matte

- Materials: tbc
- Target level: Advanced users

Session specific objectives

1. Applying an image to use for a matte

Chapter 1: Creating an image to use for a matte

- Black and white images defining transparencies

Chapter 2: Adding new textures

- Additive transfer mode
- Alpha format type
- Texture0 and texture1 slots

Session options:

Creating nulls

How to use transform nodes as null objects that can stores values, become helpers and define durations

- Materials: tbc
- Target level: Advanced users

Session specific objectives

1. Why create nulls?

Chapter 1: Creating a null duration

- Using a null to define durations

Chapter 2: Creating a null helper

- Storing values

Session options:

Using FX nodes with PBR

This tutorial examines the use of physically based rendering techniques using the native Swift CG + FX node.

- Materials: tbc
- Target level: Advanced users

Session specific objectives

1. Applying FX nodes

Chapter 1: Adding an FX node

- Choosing the effect

Chapter 2: Applying FX

- HDR lighting
- Normal mapping
- Ambient occlusion

Chapter 3: External applications

- Substance Painter and Designer
- Workflows and alternative tutorials