

**LMS**  
**Swift**  
**Training Course**  
**Syllabus**  
**Automation with Grass Valley**

DATE	07/02/2019
PRINCIPAL AUTHOR	Justin Avery
VERSION	2.1; 2.2; 2.3
UPDATE	15/10/18; 07/02/2019;03/06/2020



Unit 6 Hurlingham Business Park, Sullivan Road, London SW6 3DU

# Overview

The purpose of this document is to define the Advanced Graphics training course, syllabus and materials for those personnel tasked with authoring graphics templates for automation purposes, utilising scheduling and automation solutions such as Grass Valley ICE and Morpheus.

## About the Advanced Graphics Options

The Advanced Graphics editor options available for automation graphics template creation are RT Software's Swift and Edit-3d.

### About Swift

Swift is a GUI based graphics editor and assembly environment for RT Software's OEM render engine and provides an advanced toolset that enables the templatisation of branded graphical content for multiple use-case solutions and scenarios. Swift utilises a 2d workspace x and y axis paradigm that allows designers to create templates in any pixel resolution.

#### Who uses Swift

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 Adobe Creative tools such as Photoshop, Premiere or similar
- Image file formats and graphical assets in general
- Typography and the importance of branding
- Video editing or animation in a post production environment
- Live operation of graphics systems

## About Playout and Automation graphics branding

Swift & Edit-3d provide the interfaces that enable template designers to author graphical templates. Typically, networks provide a "kit of parts" and this imagery and graphics content define the on-air brand. The template authoring process begins by importing branded graphical assets for a given channel.

Dynamic, updatable content, such as text, bitmap, or video clips are assigned during the template creation process from within the editors.

In the playout environment, the templates are accessed, driven, updated and automated by control applications - such as Grass Valley Morpheus and ICE. The RT DLL render engine sits transparently on the playout solution (such as ICE) delivering the graphic templates to air as they are scheduled.

#### What types of templates are used in automation?

The types of graphics that are authored for automation are typically channel bugs, lower thirds, snipes, tickers, squeezebacks, menus, ECE's and IPP items that form the bedrock of branded promotional content for channel networks.

# Course primary objectives

At the end of the training period, delegates will:

- Have acquired the skills to use the Swift graphics editor to create graphic templates
- Have explored the wide potential of Swift editor features including:
  - Animation toolsets
  - External asset compatibility
  - Adding functionality
  - Applying user-code within templates
- Author, install and test a project with particular attention to critical:
  - Nomenclature
  - Files, paths and locations
  - Data sources
- Develop an understanding of the functionality that scheduling and infrastructural automation tools require:
  - Advance the importance of gathering a functional and creative specification prior to template creation

# Session 1: Swift Introduction

## Description

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

**Materials:** Syllabus document; Swift Manual

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

## Session specific objectives

### Gaining familiarity with the Swift interface, features and toolsets

#### Chapter 1: Creating a project and a graphic

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

#### Chapter 2: Preferences menu options

- Preferences menu
- Formats, aspect ratios and workspace definitions
- Locale and defaults

#### Chapter 3: Tabs

- Graphics, Images, Fonts, Primitives
- Dragging into the preview (rectangle)
- Preview window
- Safe frames, grids and snaps
- Transparency and key options

#### Chapter 4: Scenegraph nodes

- Scenegraph nodes
- Selecting to edit
- Right click menu

#### Chapter 5: Preview navigation and input

- Coordinate system
- Navigation and interaction
- Scales and positions, resets

#### Chapter 6: SHDR and DGEOM tabs

- Colours and Diffuse colour editor
- Dgeom tab interaction

# Session 2: Lower third strap

## Description

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

**Materials:** None

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

## Session specific objectives

### Applying Swift tools to create a simple lower third

#### Chapter 1: Text format tools

- 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 2: Zooming, panning and reset camera

- ALT + right mouse drag Preview window zooming
- ALT + mouse wheel scroll Preview window incremental zooming
- ALT + mouse wheel depress Preview window panning

#### Chapter 3: Develop experience with the scenegraph L1

- Grouping
- Parent child hierarchy
- Moving the branch
- Ordering
- Display toggle
- Alpha nodes
- Alpha multiply

#### Chapter 4: Edit the Strap

- Edit the DGEOM for rounded corners L1
- RectangleRadius
- Scaling in the DGEOM versus the TRFM
- Edit the SHDR
- Change the colour

# Session 3: Lower third animation

## Description

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

**Materials:** None

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

## Session specific objectives

### Familiarisation with animation principles in Swift.

#### Chapter 1: Animation concepts and Timeline Editor tab L1

- Animator icons and animators
- Methods
- bringOn
- takeOff
- Timeline animator colours - red dot and bar
- Default duration
- Timebase and timecode readout - frames
- Icon options and animator ordering
- Sliding and changing in and out points
- Widen interface

#### Chapter 2: 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
- Zooming
- Saving at bringOn
- Refresh with Reload Current Script
- Animating alpha multiply values

# Session 4: Lower third inputs

## Description

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

**Materials:** None

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

## Session specific objectives

### Adding dynamic inputs to the simple lower third to expose values for control application

#### Chapter 1: Input features L1

- Adding inputs on the bringOn to text nodes
- Blue dot node icon
- Inputs tab
- Source types
- Destination types
- Node and field parameters
- Default values
- Input names and nomenclature
- Icon shortcuts

# Session 5: Lower third Playout

## Description

This video tutorial guides users in the use of the Playout Editor for testing.

**Materials:** None

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

## Session specific objectives

**Using the Playout interface for testing.**

### Chapter 1: Playout overview

- 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
- Testing inputs
- Changing dynamic content
- Stack population
- Moving Stack items up and down
- Deleting Stack items
- Saving and loading Stacks
- Abort and Clear Stacks



# Session 6: Importing assets L1

## Description

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

**Materials:** Session06.zip;

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

## Session specific objectives

### Changing the font on the lower third and creating a new bug template

#### Chapter 1: Importing fonts

- Dragging into the Preview window
- Changing existing font types from the text editor
- Compatible font files for import (ttf;otf)

#### Chapter 2: Importing and placing still images

- Pre-multiplication and alpha channel transparency
- When stills are imported - GMDData/images
- When stills are previewed - GMDData/Shaders
- Using view transparent toggle
- New Item menu when dragging
- Scenegraph ordering with transparent images

#### Chapter 3: Adding inputs on still images

- Adding inputs on the Texture0 slot
- Passing the correct name and file extension
- Testing in Playout

#### Chapter 4: Shaders, Materials, Textures and States L1

- Creating a new Shader

# Session 7: Controlling movies L1

## Description

This video tutorial details the features for imported movie clips.

**Materials:** Session07.zip; RT\_ Supported Formats & Filetypes.pdf

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

## Session specific objectives

### Movie clip controls and Advanced Shader Editor features L2

#### Chapter 1: Importing movie clips and the Advanced Shader Editor

- Compatible and recommended file types
- Encoded movie properties and the Quicktime Movie Inspector
- Texture tab
- Texture Image bar options - image path
- Format bar options - Transparency and alpha channels
- Animation bar options - Number of Loops
- Saving Shader changes

#### Chapter 2: CueGraphic Method

- Rationale
- Step animators for Start
- Movie Pause on load
- Reset step animators

#### Chapter 3: Adding inputs to clips

- Importing another bug clip
- Adding an input
- Testing in Payout

# Session 8: DVE templates

## Description

In this video tutorial the DVE functionality of Swift 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

**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 9: Using audio

## Description

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

### 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 controls

# Session 10: Persistence

## Description

This tutorial examines basic transition logic, persistence and layer ordering of multiple templates.

**Materials:** None

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

## Session specific objectives

**Using the object node order number, and understanding the nature of persistent templates.**

### Chapter 1: Examining the nature of transition logic and Persistent graphics

- What and how does Persistence work
- Making a proof-of-concept runorder
- Testing in the Playout interface

### Chapter 1: Setting the object node order

- Testing in the Playout interface

### Chapter 3: Saving icons for templates

- How to generate an icon as a template placeholder

# Session 11: Blocking and multiple Methods

## Description

This tutorial examines blocking and non-blocking in templates as well as templates with more than 3 Methods.

**Materials:**                      **None**

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

## Session specific objectives

**Using blocking features and multiple methods.**

### Chapter 1: Setting a template to non-blocking

- Blocking and non-blocking rationale
- Testing in Playout

### Chapter 2: Adding new blocks

- Creating a new block
- Reasons to use blocks: Using Inputs on Methods with more than 1 block

### Chapter 3: Adding new Methods

- Creating a new Method
- Using the same Method for updating information

# Session 12: Updating manifests

## Description

This tutorial examines the Project Update Manifests menu item, it's uses and requirements.

**Materials:** None

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

## Session specific objectives

**Updating the project manifest files.**

### Chapter 1: How and why to Update Project Manifests

- Looking at the file with a text editor
- Rationale

# Session 13: Creating IPP's & snipes

## Description

IPP's or "in programme promotions" and "snipes" are a common form of channel branding. These types of promotional mechanisms are normally scheduled during a programme.

**Materials:** Session13.zip

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

## Session specific objectives

### Creating 2 IPP (snipe) variations

#### Chapter 1: Using supplied movie clip as bed

- Importing assets
- Snipe.mov
- Setting movie control animators
- Testing in playout

#### Chapter 2: Constructing a lower third from still assets

- Importing assets
- grad1~2~3;

#### Chapter 3: Adding inputs for movie and text nodes

- Importing assets
- gradText1~2~3



# Session 14: Ingest for TMV

## Description

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

**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 15: Using duplicate nodes

## Description

Duplicate nodes are a powerful feature in Swift 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:** None

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

## Session specific objectives

### Using duplicate nodes and introducing Clip Plane nodes

#### Chapter 1: Example duplicate

- A 10 row leaderboard example
- Constructing a row of information

#### Chapter 2: Adding and editing Clip Plane nodes

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

#### Chapter 3: Adding duplicate nodes

- Setting the numbers of duplicates for Max and Number
- Arranging duplicates in x or y
- Using Vertical Offsets
- Using Animation Delays
- Refreshing the preview window

#### Chapter 4: Copying elements from one template to another

- What to look out for

#### Chapter 5: Adding inputs to Duplicate nodes

- Adding a separator character

# Session 16: Using Links L1, L2

## Description

Links are a powerful feature in Swift 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:** None

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

## Session specific objectives

**Creating a plinth that scales according to the length of a text string; Links editor.**

### Chapter 1: Creating a plinth and link. L1

- Editing the pivot point
- Linking the plinth to text width
- Source and Destination Fields
- Using Scale with text size
- Group Function: Average
- Adding margins
- Using Bias

### Chapter 2: Using the Group Function for multiple links

- Adding Max-X size to Xsize links
- Group Function: Min

### Chapter 2: Add two dynamic text strings and centre them on screen. L2

- Setting the correct text ranges - left and right
- Group function: Expressions
- \$1, \$2 syntax

# Session 17: Tickers

## Description

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

**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 18: Standard clocks

## Description

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

**Materials:** None

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

## Session specific objectives

### 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 19: Mattes & masters

## Description

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

**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 20: Exporting a graphic

## Description

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

**Exporting a graphic from one project to another.**

### Chapter 1: Exporting a template

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

# Session 21: User code & scripting

## Description

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

**Materials:** Session21.zip

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

## Session specific objectives

### Adding simple ruby scripts to understand the principles behind its use: L1

#### 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>")`
  - `_text.gsub!(/pm/, "<sv 30.0>PM</sv>")`
  - `_text.gsub!(/PM/, "<sv 30.0>PM</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 22: External data L1

## Description

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

### Understanding differing external data usage and requirements in Swift 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 23: External data L2

## Description

This tutorial examines the use of mysql databases.

**Materials:** Session23.zip

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

## Session specific objectives

### Understanding differing external data usage and requirements in Swift 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 24: Validation

## Description

This tutorial examines the use of the IsGraphicValid Method in order for automation to validate data held externally.

**Materials:** None

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

## Session specific objectives

### Implementing the IsGraphicValid Method

#### Chapter 1: Adding an IsGraphicValid Method to a template

- Rationale

# Session 25: Transition Maps

## Description

This tutorial examines the use of Transition Maps to automate template bringOn and takeOff methods without the use of scheduling or control.

**Materials:** None

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

## Session specific objectives

**Applying a Transition Map and using its inherent logic to streamline scheduling.**

### Chapter 1: Creating a Transition Map

- Project Settings, Transition Map editor

# Session 26: Performance monitor

## Description

This tutorial examines the use of Performance Monitoring for evaluating differing aspects of performance issues.

**Materials:** None

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

## Session specific objectives

### Using the Performance Monitor.

#### Chapter 1: Evaluating a template's performance

- Render and Max render
- Shader sets
- Memory

# Session 27: GSD

## Description

This session details the necessary information for graphics presentation team managers and lead designers to author and publish a **Graphics Specification Document** in line with their broadcaster's workflows, scheduling and external data requirements. This session helps define how Swift channel branded graphics for automation can be managed from concept to transmission.

**Materials:** RT\_Specification Document Sample.pdf

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

## Session specific objectives

### Formulating a GSD (Graphics Specification Document)

#### Chapter 1: Asset acquisition

- Asset origins; Files and filetypes; Formats, codecs and clips

#### Chapter 2: Defining functionality

- Engineering requirements; Nomenclature; External data

#### Chapter 3: Integration considerations

- Partner and hardware solutions
- Video channels
- Downstream and upstream
- Dual channel configurations

#### Chapter 4: Graphics Specification Document

- Collating scheduling information, brand guidelines and functional requirements
- Publishing a Graphics Specification Document
- Sample Graphics Specification Document

# Session option 1: Matchday clocks

## Description

The focus of this tutorial is to build a standard football (soccer) league clock: Premier League 90 minute football match.

**Materials:** MatchdayClocks.zip

**Target level:** Advanced: Only course completed (L1 & L2) users

## Session specific objectives

**Using game clocks and added time clocks for specific control by WebControl.**

### Chapter 1: Creating and editing a game clock

- Add a clock node
- Add an ALPHA node
- Add a step animator
- Add an ALPHA node animator
- Create method: startClock
- Create method: setClock
- Add new inputs
- Create method: setClockEnd

### Chapter 2: Creating an added time clock

- Add a clock node
- Create method: addedTimeOn
- Add new inputs
- Set the following user code:
- Add an ALPHA node
- Add an ALPHA node animator
- Create method: addAddedTimeEvent
- Set the following user code:
- Create method: clearAddedTimeEvent
- Set the following user code:
- Create method: setAddedTime
- Set the following user code
- Create method: setAddedTimeClock
- Add new Block
- Add new inputs
- Create method: startAddedTimeClock
- Create method: stopAddedTimeClock

# Session option 2: Histograms

## Description

This tutorial examines the assembly of a proof-of-concept histogram chart.

**Materials:** None

**Target level:** Advanced: Only course completed (L1 & L2) users

## Session specific objectives

**Creating a histogram and implementing logic that enables the update and display of new data.**

### Chapter 1: Creating a histogram

- Add a clock node
- Use of the duplicate node
- Offsets
- Translate and frame
- Linking parameters
- Inputs



# Session option 3: Presets

## Description

The focus of this tutorial is understand Preset Methods, their uses in WebControl and the rationale behind them. This tutorial requires the use of the WebControl, Chrome, and internet connections.

**Preparation:** **SwiftSessionOption03.mp4; WebControl and DataServer installation**  
**Materials:** **Presets.zip**  
**Target level:** **Advanced: Only course completed (L1 & L2) users**

## Session specific objectives

**Understand Preset Methods, their uses in WebControl and the rationale behind them**

### Chapter 1: Adding a Preset Method

- Develop understanding and overview of base control project
- Develop knowledge of \*preset\* methods and uses for presets
- Deploy test graphics to WebControl application and test
- Testing Preset in Playout
- Testing Preset in WebControl

# Session option 4: Countdown clocks

## Description

The focus of this tutorial is to build a multi-day countdown clock for the promotion of upcoming events.

**Materials:** **CountdownClocks.zip**

**Target level:** **Advanced: Only course completed (L1 & L2) users**

## Session specific objectives

**Using game multiple clocks and a control clock node to create countdown clocks for durations of days, weeks or months.**

### Chapter 1: Creating a multi-day countdown clock

- Adding multiple clocks for days, hours, minutes and seconds
- Using a control clock node
- Applying appropriate user code
- Driving clock times with inputs for event times

# Session option 5: Texture fonts

## Description

The focus of this tutorial is to import a font as a set of textures that can form the basis of a drop shadow effect.

<b>Preparation:</b>	<b>SwiftSessionOption05.mp4; Photoshop</b>
<b>Materials:</b>	<b>TextureFonts.zip</b>
<b>Target level:</b>	<b>Advanced: Only course completed (L1 &amp; L2) users</b>

## Session specific objectives

### Creating drop shadows for fonts and applying images to text

#### Chapter 1: Importing a font as a Texture Font

- Setting the name and sizes
- Batch processing in Photoshop
- Replacing in Swift
- Linking and scenegraph ordering in Swift

#### Chapter 2: Applying images to fonts

- Create shader on import
- Descender characters and measurements
- TextureS:0 tiling and editing
- TextureT:0 tiling and editing