Mac Development for Media Arts
AME 430

School of Arts, Media and Engineering
Herberger Institute for Design and the Arts
Arizona State University
Fall 2019

Tuesday/Thursday, 10:30am-11:45am, Stauffer B123

Loren Olson
loren.olson@asu.edu

FALL 2019 OFFICE HOURS
Location: Stauffer B260
Time: M/T/W/Th 2:00 - 3:00pm
During office hours you are welcome to walk in, no appointment is necessary.
If you need to see me, and you can't make my office hours, please send me an email to make an appointment at another time.

This syllabus is subject to change. I will let you know when it does, via class and online annoucements.

OVERVIEW
This course explores native application development for macOS (Macintosh) for use in the domain of Media Arts. The course is project based, and covers the native Xcode development environment, and the Swift programming language. Native applications are able to achieve the highest levels of performance, they leverage the unique technologies built for a platform, and they are able to participate in the native user experience (look & feel) that all users of a popular platform expect. To understand native apps for macOS, the course will examine a number of code frameworks provided for developers by Apple (the core of which are often called Cocoa), including (but not limited to) Foundation, AppKit, AV Foundation, Core Animation and SpriteKit.

There is one thing I can confidently forecast for your future as you enter a career that involves using and creating digital products. The platforms and development tools you use will evolve and change. There will be new and exciting systems. Where does that leave you today - what should you learn? You do need to learn tools today in order to making things today and tomorrow. However, you shouldn't get too hung up on particular tools. Suppose you are a photographer and the camera you own is a Sony A7 III. It is necessary to know how to expertly use that Sony camera in order to be good at taking pictures. Still, you don't want to forget that your goal isn't to be a Sony camera expert - your goal is to be a great photographer. Considering this forecast for the future and our own goals. Let's take some time as we learn to think about larger issues that impact the programs we might create, how we write programs, and even how we learn to use platform or environment frameworks.

Course Objectives

Students are able to write native macOS applications of significant complexity.
There are a wide range of deep frameworks available for creating applications, we won’t have time to explore them all. So, our goal will be to cover the important fundamentals, and look carefully at some exemplar frameworks. Students should then be ready to learn additional more specialized frameworks on their own initiative.
Upon successful completion of this course, students will be able to:

- Use the Xcode development environment to develop native cocoa apps for macOS.
- Use standard system frameworks and apis, and demonstrate understanding of programming fundamentals.
- Create applications using the Swift programming language.
- Use Foundation and AppKit frameworks in the development of native apps.
- Understand the strengths and weaknesses of native applications, and be able to make good judgements about when native applications provide the best solution in a particular context.
- Read and understand framework documentation. Students will develop the ability to understand new frameworks and apis to use in programs they are creating.

TEXTBOOK

There is no required textbook. If you want or need an additional helpful reference for learning Swift and writing macOS programs, I am recommending Hacking with macOS by Paul Hudson. It is available electronically for $40 at https://www.hackingwithswift.com/store/hacking-with-macos.

ASU Fall 2019 Important Dates

Session Dates and Deadlines	Session C: 15 weeks (Aug 22 – Dec 6, Finals week is Dec 9-14)
Classes Begin	August 22, 2019
Drop/Add Deadline	August 28, 2019
HIDA Extended add/swap date	September 11, 2019
Tuition and Fees 100% Refund Deadline	September 4, 2019
Course Withdrawal Deadline	November 6, 2019
Complete Session Withdrawal Deadline	December 6, 2019
Fall Break	October 12-15, 2019
Final Exam	Dec 10 (Tuesday), 9:50-11:40

For additional university deadlines and important dates for the fall 2019 term, please visit: students.asu.edu/academic-calendar.

EVALUATION

70% Exercises, Assignments, Projects
Assignments will involve writing a short program, assignments are turned in using Critviz. You should always check Critviz for updated Assignment information and due dates. I expect there to be 6 to 8 significant projects during the semester. The final number could vary depending on how fast we get through the material.

20% Final Project
Final projects will be demoed on Tuesday December 4th, during the final exam time. The final exam is not a test, it is you demonstrating your final project.

10% Class Participation, Attendance
There will be a class roster to sign for attendance, it is your responsibility to sign the roster every class. If you don't sign the roster, you don't get attendance credit.

We will use an online system called Critviz to turn in assignments. Critviz can be found at http://critviz.com. I will show you how to use the system in class, and we will do an example project before a "real" assignment is due.

Not all exercises, assignments, or projects will be weighted equally, I may change the weights during the semester.

Grading Scale

A+	97-100%
A	93-96%
A-	90-92%
B+	87-89%
B	83-86%
B-	80-82%
C+	77-79%
C	70-76%
D	60-69%
E	0-59%

LATE ASSIGNMENT POLICY

The concepts, techniques and ideas you learn in this class will be cumulative. They build on each other as the semester progresses, and you will use all of these things together throughout the semester as you learn more about programming. Therefore, it is important for your success to do all assignments in a timely fashion, in the order that they are given. The important concepts from one assignment will become foundations for subsequent coursework. If you miss an assignment, or for some reason you struggle with a particular assignment and do poorly - we want to make sure you don't miss those concepts and therefore struggle in future assignments. For this reason, we will accept late assignment submissions or resubmissions with a late penalty. Work submitted the week after due date is marked down 15 points. Each further week late means an additional 10 point penalty. After 4 weeks, late submissions will no longer be accepted.

CLASSROOM CITIZENSHIP

Attend class, and pay attention. Participate in class discussion. Be curious. Be open to working hard and trying new things. Speak up. If you have a burning questions that you think is too simple to ask, it probably means several other people have the same question.

Be courteous to your fellow classmates, teaching assistant and teacher. Help us to create a positive and constructive learning environment that encourages everyone. Any disruptive behavior will be dealt with according to ASU policy. Please see the Student Services Manual for more details.

Policy on Sexual Discrimination

Policy on sexual discrimination as described in ACD 401, "Prohibition Against Discrimination, Harassment, and Retaliation", including the fact that the instructor is a mandated reporter and therefore obligated to report any information regarding alleged acts of sexual discrimination. Arizona State University is committed to providing an environment free of discrimination, harassment, or retaliation for the entire university community, including all students, faculty members, staff employees, and guests. ASU expressly prohibits discrimination, harassment, and retaliation by employees, students, contractors, or agents of the university based on any protected status: race, color, religion, sex, national origin, age, disability, veteran status, sexual orientation, gender identity, and genetic information. As an employee of ASU, I am a mandated reporter and obligated to report instances of reported or suspected incidences of sexual harassment.

ACADEMIC INTEGRITY

Is writing code like solving an equation in Calculus or is it more like writing an essay in English? If two students in English turn in essays that are nearly identical, it will be flagged as plagiarism. When two students have identical solutions in Calculus, not a second thought would be given - but everyone understands that copying answers from a neighbor's paper is cheating. In fact, I think that our projects should be more closely related to the English class essay, than the Calculus example. When you work on your project, please keep in mind the essay example - you must write your own unique essay.

What about code you found to help you? (Code at developer.apple.com, or stackoverflow, or github, etc) It can be ok to use other code in projects. It is critical that you acknowledge that code. You must cite where the code came from, and what code you have used. Also, I should add that you should never add code to a project that you don't understand. It is critical that you understand code that you add, it should not be a "black box." Always cite code in comments, where you use it. Clearly mark what code is involved, and include a URL, and explanatory text. Failure to cite code in an assignment will result in an automatic 0 grade.

How much code, and what code is ok to use is also context dependent. Again, think about an English essay. You would not turn in an essay that consists of two sentences you wrote, then simply quote another writer for two entire pages of content. That wouldn't be your own essay. In this class, you cannot copy and paste many lines of code, change a few variable names, then submit the result as your own work. That would be a coding example of plagiarism.

What about study groups? A problem has arisen in the past when students work together, then turn in identical code. You should not be turning in identical code, since it becomes impossible for us to tell what is innocent coincidence due to a study group versus blatant unethical copying. I think study groups are very helpful, and do not want to discourage that community. Experience so far shows you can avoid this problem with a simple rule of thumb - always type all of your own code. Never copy a file from a friend, or allow a friend to copy one of your files.

University policy regarding copyright
Students must refrain from uploading to any course shell, discussion board, or website used by the course instructor or other course forum, material that is not the student's original work, unless the students first comply with all applicable copyright laws; faculty members reserve the right to delete materials on the grounds of suspected copyright infringement.

Attendance and Participation

Students are expected to attend all classes. In the case of absence, please inform the instructor before the class if possible, and/or after the missed class. Classroom attendance and participation is 10% of the overall grade.

Excused absences related to religious observances/practices in accord with ACD 304–04, "Accommodation for Religious Practices." Students may be excused for the observance of religious holidays. Students should notify the instructor at the beginning of the semester about the need to be absent from class due to religious observances. Students will be responsible for materials covered during their absence and should consult with the instructor to arrange reasonable accommodation for missed exams or other required assignments.

Excused absences related to university sanctioned activities in accord with ACD 304–02, "Missed Classes Due to University-Sanctioned Activities." Students required to miss classes due to university sanctioned activities will not be counted absent. However, absence from class or examinations due to university-sanctioned activities does not relieve students from responsibility for any part of the course work required during the period of the absence. Students should inform the instructor early in the semester of upcoming scheduled absences and immediately upon learning of unscheduled required class absences. Reasonable accommodation to make up missed exams or other required assignments will be made. Consult the instructor BEFORE the absence to arrange for this accommodation.

Line-of-duty absence and missed assignment policy:
A student who is a member of the National Guard, Reserve, or other U.S. Armed Forces branch who misses classes, assignments or examininations due to line-of-duty responsibilitites, shall have the opportunity to make up the coursework in accordance with SSM 20-18 Accommodating Active Duty Military Personnel. This accommodation also applies to spouses who are the guardian of minor children during line-of -duty activities. This policy does not excuse students from course responsibilities during their absence. Students should first notify the Pat Tillman Veterans Center of their activation and then the instructor to discuss options.

Special Accommodations

To request academic accommodations due to a disability, please contact the ASU Disability Resource Center (http://www.asu.edu/studentaffairs/ed/drc/# ; Phone: (480) 965-1234; TDD: (480) 965-9000). This is a very important step as accommodations may be difficult to make retroactively. If you have a letter from their office indicating that you have a disability which requires academic accommodations, in order to assure that you receive your accommodations in a timely manner, please present this documentation to me no later than the end of the first week of the semester so that your needs can be addressed effectively.

Stauffer Media Lab

The Stauffer Media Lab - Stauffer B135 - is available for all students in this class, weekdays 8am to 8pm. All the computers in the lab have Xcode installed. There may be weekend hours this semester, check the lab to find current hours.

REFERENCE

Apple Developer Web Site.
https://developer.apple.com/

The Swift Programming Language Book. This book is also available as a free ebook.
Swift_Programming_Language [https://docs.swift.org/swift-book/]

There are many other books about Xcode and Swift available, some of them are very good. There are enough that keeping a current list here isn’t practical.

If you have an iPad, Apple has released an iPad app called Swift Playgrounds. See this (marketing) web page, and take a look - its free. The app includes links to educational material. There is a Learn to Code iTunes U course that uses the Playgrounds app.

OUTLINE

1. Introduction
    1. Native apps?
    2. Macintosh
        1. Apple Human Interface Guidelines - macOS
    3. Xcode
        1. Xcode Help
    4. Swift
        1. https://swift.org
        2. Swift Language Guide
2. Project 1. Hello World
    1. Swift and Xcode workflow
    2. Intro to IB and connecting UI and code
    3. IBOutlet
    4. IBAction
3. MVC design pattern
    1. Model-view-controller article at wikipedia
    2. Model-view-controller in Cocoa core competencies
    3. Controller object in Cocoa core competencies
4. Project 2. Unit Converter
    1. Swift standard library
        1. Standard library
    2. Foundation
        1. Foundation
    3. AppKit (Cocoa)
        1. AppKit
    4. NSTextField
        1. NSTextField
    5. Formatter
        1. Formatter
    6. NumberFormatter
        1. NumberFormatter
5. Project 3. Stopwatch
    1. Cocoa
        1. Cocoa Application Layer
    2. Timers

        1. Timer
    3. NSView
        1. NSView
6. Project 4. Media Player
    1. AVFoundation
        1. avfoundation index page (Overview) at developer.apple.com
        2. AVFoundation documentation
    2. AVKit
        1. AVKit framework
        2. AVPlayerView
        3. AVPlayerViewController
        4. AVPlayer
        5. AVAsset
7. Project 5. 2D Game
    1. SpriteKit
        1. spritekit
        2. skview
        3. skscene
        4. sknode
        5. skspritenode
        6. skaction
        7. skphysicsbody
    2. GameplayKit
        1. gameplaykit
        2. gkrandom protocol
        3. gkarc4randomsource
8. Project 6. 3D environment
    1. SceneKit
        1. scenekit
        2. scnview
        3. scnscene
        4. scnnode
        5. scnlight
        6. scncamera
        7. scngeometry
        8. scnmaterial