

DIG 6836: Design and Development of Texts and Technology

Section 0M01, Course # 90458, 3 credit hours
Fall 2019, Mondays, 6:00-7:15, Location TCH-116

Professor: Dr. Rudy McDaniel (“Rudy”)
Phone: 407-823-0728
*Email: rudy@ucf.edu

Office: VAB-117
Office Hours: Mondays 5:00-6:00
and virtually by appointment

Overview

Fundamental to a deep understanding of new media theory and practice is knowledge about the underpinnings (code layer) of digital computing and interactive design. This course will introduce you to some digital design and development techniques to be used in the interdisciplinary scholarship of Texts and Technology. Specifically, you will learn how to use various scripting languages to build interactive projects useful to your own research interests and to the broader domains of digital media and the digital humanities. As you work toward these goals, you will learn and apply foundational techniques in computation such as iteration, conditional logic, randomness, and algorithm design. You will learn how to use data structures to organize your thinking and implement more efficient, programmatic methodologies for digital inquiry. We will also study how to apply programming techniques for functions such as text parsing, data analysis, repetition, and interactive design. These skills will further aid you as you consider applied digital components for your dissertation or other scholarly projects.

The bulk of your “writing” in this course will be done in digital form through interactive projects. The three main languages we will use in this course are JavaScript, Python, and Processing. You will also maintain a Developer’s Journal (DJ) that includes some online, reflective writing prompts throughout the semester. For your digital work, you will design four interactive projects this semester. By the end of the course, you should have the knowledge to a) plan and design a workflow for an interactive digital project, b) select the appropriate data structures, functions, and/or objects to employ in your project, c) determine the best technical and rhetorical means by which to deploy your project to an audience, and d) document and assess your project within a scholarly context.

We will use a course website to extend our inquiry online, share work with others, and track our work. This website will contain copies of our major assignments, example code and tutorials, and links to student projects. The official link is below.

M-Model Course

You will notice that our course is only scheduled to meet in person for one hour and fifteen minutes each week. This is not very much time! Our face-to-face time is shortened on purpose in order to provide you extra time each week to work through the programming materials at your own pace. You should be sure to schedule your time appropriately so that you have plenty of time to work through the programming materials and DJ postings. The discussion boards will be deployed on our official Canvas web site (<https://webcourses.ucf.edu/courses/1333312>). Please bookmark this site as you will be using it throughout the semester.

Webcourses is additionally available from the “Online Course Tools” of your MyUCF panel. Students are encouraged to make use of the “Ask A Question” forum thread for any questions about assignments or course materials. They can also use the “General Student Discussion” forum thread to interact with classmates. For general help with Webcourses or logging into your account, see <http://learn.ucf.edu/>.

We will also be using video tutorials throughout the course from Lynda/LinkedIn Learning. The web site for this resource can be found here: <https://digitallearning.ucf.edu/lynda/>. Please register an account using your UCF credentials and ensure you can login and access this content.

Late Work Policy

Late work will not be accepted unless there is a documented medical emergency. Assignments can be submitted early if necessary; be sure to speak with me if you need to take advantage of this.

Course Learning Objectives

- Students will understand and apply media programming techniques and algorithmic design to develop scholarly textual and interactive projects for the World Wide Web and their scholarly portfolios.
- Students will learn how to solve problems using interactive programming languages.
- Students will explore ideas and juxtapositions of critical theory and technology for use in their dissertations, future publications, or future project coursework.
- Students will reflect critically on the design and development process through presentations and writing in a developer's journal.

Required Texts

- Montfort, Nick. *Exploratory Programming for the Arts and Humanities*. Cambridge: The MIT Press.
- Sayers, Jentery (Ed.) *Making Things and Drawing Boundaries*. Minneapolis: U of Minnesota Press.

Recommended Text

It is recommended that those students without much/any programming experience pick up a beginner's guide to programming on each of our languages: JavaScript, Python, and Processing. For Python, I recommend *Python Crash Course* by Eric Matthes (2nd Ed.). If you already have experience with another programming language or feel comfortable with web scripting, then an additional book is probably not necessary. As a class, we will also read additional selections (available on our Web site) and follow tutorials from LinkedIn Learning each week.

Assignments and Grading

<u>Major Assignments</u>	<u>Percentage of Overall Grade</u>
1) Developer Journal Reflection Posts x 5	20%
2) Project 1 (Interactive Poetry in JavaScript)	15%
3) Project 2 (Simple Widget in Python)	15%
4) Project 3 (Data Parser in Python)	20%
5) Project 4 (Interactive Argument in Processing)	20%
6) Class Participation and Online Interaction	10%

Financial Aid Reporting Policy

All faculty members are required to document students' academic activity at the beginning of each course. In order to document that you began this course, please complete the first developer response activity (DJ#1) no later than August 29th. Failure to do so will result in a delay in the disbursement of your financial aid.

Technology Policy

Outside class, students are required to have access to word processing software and a Mac or PC computer with access to the Internet and a text editor capable of saving programming scripts and executing code. Classroom labs and the T&T server can also execute Python scripts. During class time, students may use technology to take notes, experiment (during appropriate class discussions), show examples, etc. It is expected that these technologies will not be used during class for purposes outside the scope of discussion, including for purposes such as instant messaging classmates, texting, e-mailing, accessing social media, or playing games (outside of their use as examples to support particular arguments). Please feel free to use any device that makes your participation in class discussions easier. Please do not leave your cell phones on audible ring, and barring emergencies, do not take or make phone calls during class. In other words, be courteous to your instructor and your peers.

Other Course Policies

- I am always happy to meet with you about the course or your larger T&T program of study. If my office hours are not convenient for you, I can schedule alternative times to meet in person or virtually via Skype or phone.
- We will mostly follow the syllabus and schedule, but they are subject to minor changes, about which I will apprise you ASAP during normal class meetings or by email/Webcourses.
- In order for the class to be a success, you must be well prepared for and actively engaged in all class meetings.
- Because this is a discussion-oriented class, attendance and punctuality are crucial. Beyond affecting your participation grade, missing more than one class will result in your overall course grade being lowered. Missing more than two classes will likely cause you to fail the course.
- All UCF students are responsible for upholding standards of academic integrity as explained by The Golden Rule (<http://www.ucf.edu/goldenrule>). When it amounts to academic dishonesty, plagiarism (of computer code or written text) will have dire consequences (e.g., failing the course and being reported to Student Conduct for an ethical and student conduct violation). Any code that is reused/extended/modified or libraries that are used in projects should be fully referenced and documented as such. In other words, credit must be given where due.
- Students with customized learning needs will be accommodated in this course if registered through the university with Student Accessibility Services (SAS).

Tentative Schedule (Always check Webcourses for any changes or updates to syllabus or deadlines). In addition, tutorial videos will be assigned from LinkedIn Learning each week – check Webcourses for specifics. Excepting the first DJ, each additional DJ will be due Friday at 11:55pm on the week it is assigned.

Week	Date	Topics and Activities	Readings and Assignments Due By Next Meeting
1	August 26	Welcome and Introduction to the Course; Review of Syllabus; Student Introductions JavaScript I	<input type="checkbox"/> Read: Montfort: Appendix A: Why Program? <input type="checkbox"/> Read: Sayers: Introduction: I Don't Know All the Circuitry. <input type="checkbox"/> DJ Post #1: Introduction and Technology Autobiography (Due by 11:55pm on Thursday, Aug 29th)
*2	Sept. 2 (Labor Day)	No face-to-face meeting – See Webcourses JavaScript II	<input type="checkbox"/> Read: Kirschenbaum (Hello Worlds via Webcourses) <input type="checkbox"/> Read: Montfort pp. 1-33 <input type="checkbox"/> Read: Sayers pp. 21-56 <input type="checkbox"/> Work on Project #1
3	Sept. 9	JavaScript III	<input type="checkbox"/> Read: Montfort pp. 35-44 <input type="checkbox"/> Read: Sayers pp. 57-94 <input type="checkbox"/> Project 1 due next week
4	Sept. 16	Project 1 Due by Class Time Project 1 Demonstrations in Class	<input type="checkbox"/> Read: Montfort pp. 45-78 <input type="checkbox"/> Read: Sayers pp. 95-129 <input type="checkbox"/> DJ Post #2: Project 1 Reflection (due by 11:55pm on Friday, September 20th).

5	Sept. 23	Python I	<input type="checkbox"/> Read: Montfort pp. 79-123 <input type="checkbox"/> Read: Sayers pp. 130-145
6	Sept. 30	Python II	<input type="checkbox"/> Read: Montfort pp. 125-145 <input type="checkbox"/> Read: Sayers pp. 149-176 <input type="checkbox"/> Project 2 due next week
7	Oct. 7	Project 2 Due by Class Time Project 2 Demonstrations in Class	<input type="checkbox"/> Read: Montfort pp. 147-168 <input type="checkbox"/> Read: Sayers pp. 177-205 <input type="checkbox"/> DJ Post #3: Project 2 Reflection (due by 11:55pm on <u>Friday, October 11th</u>).
8	Oct. 14	Python III	<input type="checkbox"/> Read: Montfort pp. 169-184 <input type="checkbox"/> Read: Sayers pp. 206-218
9	Oct. 21	No face-to-face meeting – See Webcourses Python IV	<input type="checkbox"/> Read: Montfort pp. 185-213 <input type="checkbox"/> Read: Sayers pp. 221-248
10	Oct. 28	Python V	<input type="checkbox"/> Read: Sayers pp. 249-270 <input type="checkbox"/> Project 3 due next week
11	Nov. 4	Project 3 Due by Class Time Project 3 Demonstrations in Class	<input type="checkbox"/> Read: Sayers pp. 271-287 <input type="checkbox"/> DJ Post #4: Project 3 Reflection (due by 11:55pm on <u>Friday, November 8th</u>).
*12	Nov. 11 (Veteran's Day)	No face-to-face meeting – See Webcourses Processing I	<input type="checkbox"/> Read: Montfort pp. 215-240 <input type="checkbox"/> Read: Sayers pp. 288-300 <input type="checkbox"/> Work on final project
13	Nov. 18	Processing II	<input type="checkbox"/> Read: Montfort pp. 241-248 <input type="checkbox"/> Read: Sayers pp. 301-316
14	Nov. 25	Processing III	<input type="checkbox"/> Read: Montfort pp. 249-256 <input type="checkbox"/> Read: Sayers pp. 319-341
15	Dec. 2	Processing IV	<input type="checkbox"/> Read: Montfort pp. 257-266 <input type="checkbox"/> DJ Post #5: Final Course Reflection (due by 11:55pm on <u>Friday, December 6th</u>). <input type="checkbox"/> Final Project Due Next Class
16	Dec. 9	Project 4 Due by Class Time Project Presentations in Class	Congratulations on completing the course! Have a great Winter Break.