

CSCI351: Programming III

Course Description

Now that students have learned the basics of Python, Object-Oriented Programming and Design Patterns, it is time to develop their first full programming project. Students will each build a common game from scratch to its final form. Along the way they will delve into architecture, graphics, user interface, collision, physics, audio, artificial intelligence, file formats, databases, tools development, networking, vector math, and animation. By doing these things they will also learn good development practices.

Course Outcomes

Upon satisfactory completion of this course, you will be able to:

1. Design a full project from scratch
2. Build software systems using top down and test-driven software development techniques
3. Interact with many disciplines (i.e., Graphics, File Formats, and Animation) of programming that will likely be needed in the field:
 - a. Architecture
 - b. Graphics
 - c. User Interface
 - d. Collision
 - e. Physics
 - f. Audio
 - g. Artificial Intelligence
 - h. File Formats
 - i. Databases
 - j. Tools Development
 - k. Networking
 - l. Vector Math
 - m. Animation

Course Materials

Required Material(s)

Students must purchase:

Visual Studio or VS Code installed

All other required materials are provided as PDFs or links in Moodle. See the weekly schedule for more complete information on course readings.

Recommended (but not required) Additional Reading

www.pygame.org

www.pymunk.org

pyopengl.sourceforge.net

Class Policies

You are expected to read all assigned readings, view all lecture videos, screencasts, and access any links posted by the professor. Be prepared to discuss the contents of each.

Attendance and Participation

Because online courses require significant interaction between students, you must upload a current photo of yourself to your Moodle profile. The image should be a headshot with your face clearly visible (no pets, group photos, or cartoons).

You are not required to be online at the same times as your classmates. However, you should check in regularly (to access new materials, submit assignments, and/or participate in ongoing threaded discussions).

Each course week includes a threaded discussion focusing on topics related to the course. The discussions are a great place to ask questions, clarify issues, and share insights. You must check in regularly and contribute to the ongoing conversation, posting on the number of required days.

See the Online Discussion Guidelines for more details.

Any student who has not logged in for course participation during the first week will be administratively dropped along with any subsequent courses in the term.

Note: If you are off campus for any Jessup-sponsored extracurricular activity, you are still required to maintain and follow the due dates outlined in this syllabus. If you have an exceptional instance where internet access is not present either in your transportation and/or accommodations, you will need to have your supervising individual (professor, coach, etc.) inform your instructor to receive additional time on an assignment.

Netiquette

Netiquette, or the rules that surround good communication on the internet, is very important in online courses that are based on high levels of interaction and communication between students and professors at a distance.

Some basic rules to guide you in your online communication (see the Online Student Orientation for a longer, expanded list):

1. **Be thoughtful, kind and courteous in your communication.** Avoid language that may offend others and be cautious when using sarcastic language. In addition, respect your classmates' privacy by not asking them to share more than they would be comfortable doing.

2. **Proofread your writing so it is clear and easy to read.** Avoid acronyms (including text speak), do not use ALL CAPS, and do not overuse exclamation marks (use *italics* for emphasis). Write in short paragraphs and use plenty of white space (extra space between paragraphs) as that makes text easier to read on a webpage.
3. **Engage with your classmates.** Make sure your writing communicates what you intend, ask clear questions of your peers and always be aware of your audience when you are writing in the online classroom.

Written Work Guidelines

Written work is graded for content, organization, style, grammar, and formatting. All papers are to be typed, proofread, spell-checked, double-spaced, and prepared in accordance with APA style and format. Basic formatting should be Times New Roman 12 with 1 inch margins. For help with APA formatting, see the APA Tab of the Course Resources Folder (located on the main page of the course in Moodle).

The Writing Center is available to all Jessup Online students for help with writing papers as well as APA formatting. You can contact them at writingcenter@jessup.edu or schedule a session through the WJU Student Services Scheduler.

Assignments

Submission Format

All assignments must be submitted as an attachment via Moodle no later than 11:59 PM (PST) the day the assignment is due. Unless otherwise specified, you should submit all papers as Microsoft Word documents (.doc or .docx files) via Moodle. If you do not have access to Word, you may also submit Adobe pdf files in Moodle.

Late assignments

Whether instructors accept late work or not is up to their discretion.

In the case that they do, late work may be penalized 10% of the possible points for the assignment for each day, or part thereof, that it is late. *Work may not be submitted more than a week late.*

If you face particular difficulty meeting a deadline, please contact the professor ahead of time to discuss any options.

NOTE: The professor is not obliged to accept any late work after the final day of the class session unless prior arrangements have been made.

Feedback and Grades

You can expect to receive written feedback and grades on each weekly assignment via Moodle within 72 hours of the due date for submission.

For larger assignments (research papers, projects, etc.), you can expect to receive feedback within a week.

Academic Integrity

The University Catalog states:

Academic integrity is an essential component of Christian higher education. Instances of plagiarism will not be treated lightly. If it is a student's first offence, the paper will simply receive a zero. The student may or may not have the option to re-write the assignment for half credit, according to the instructor's discretion. If evidence of plagiarism exists a second time the student will receive an academic dismissal, which can be appealed by the student.

Plagiarism includes:

- The intentional or unintentional representation of another's words or ideas as your own in an academic exercise.
- Using the "copy and paste" method to use text found on a Web site without giving credit to the source.
- Copying information from a source without proper citation and without use of quotation marks or block quotation formatting. If any words or ideas used do not represent your original words or ideas, you must distinguish them with quotation marks or an indented block quotation followed by the appropriate citation.
- Paraphrasing statements or paragraphs without proper citation or using someone else's ideas, data, language, and/or arguments without acknowledgement.
- Presenting work as your own that has been prepared in whole or part by someone other than you.
- Failure to properly cite statistics, data, or other sources of information in your paper.
- Resubmitting a paper that you have already turned in as an assignment for a different course (including a different section of the same course). While the paper may be considered your original work, resubmitting it is considered a form of plagiarism. Your assignments for every class should be unique and original for that course.

Student Complaints

For complete information about WJU and how to file a complaint as a student please see the Consumer Information section of the Jessup website.

If a distance education student who lives outside the state of California believes that the university's internal procedures have not adequately addressed concerns identified under the Program Integrity Rule, there is a link on the Jessup website with Student Complaint Information by State and Agency.

Discussion Forums

Discussion Forums are an integral part of every Jessup Online course. A high percentage of learning in an online environment comes through the dialogue that takes place in Discussion Forums. You should think about the discussion questions in this class as an opportunity for you, your classmates, and your instructor to enter into an interesting conversation about what you are studying. Therefore, you are encouraged to jump into the discussion as often as you'd like. This ensures that everyone will benefit from a variety of opinions and insights on the topics at hand. In other words, your contribution is valuable and important! Since this is a conversation, it's also important that you read the *entire* forum; not only are your contributions important, but you'll find that your classmates' contributions are as well!

Substantive Posts

You must post **at least 3 substantive responses** each week. A substantive post is one that contributes something significant to the academic conversation using academic language (avoid "text speak" or other informal language in your discussion posts). To be substantive and earn full credit, a post should:

1. **Be of appropriate length** (initial = 250-400 words; secondary = 125-225 words).

2. **Engage with the course materials** (lecture, texts, videos, etc.) in such a way that it is evident that you have integrated the course content into your thinking.
3. **Demonstrate critical thinking skills.** In other words, your substantive posts should reflect that you have carefully considered the discussion question and have put effort into writing a response that makes a relevant contribution to the conversation.

Requirements

Since discussion questions are usually given a lot of weight in terms of the final course grade, there are also academic expectations. These are as follows.

You must be active in the discussion forum **at least 3 days per week**. This means that you must post a response on 3 of the 7 days each week of the course in order to receive full credit. Do not write all of your forum posts on one day – that eliminates the opportunity for dialogue with classmates.

For weeks with one discussion question:

1. You must post your **initial response** to the question by **Wednesday @ 11:59 p.m.**
2. By **Sunday @ 11:59PM**, you must post (at minimum) **two secondary posts** (posts responding to your classmates' comments or to your instructor's prompts) for a **total of three posts**. All posts must be substantive to receive full credit.

For weeks with two or more discussion questions:

1. You must post your **initial response to DQ#1** by **Wednesday @ 11:59 p.m.**
2. You must post your **initial response to DQ#2** by **Friday @ 11:59 p.m.**
3. By **Sunday @ 11:59PM**, you must post (at minimum) **four secondary posts** (spread across both questions; responding to your classmates' comments or to your instructor's prompts) for a **total of six posts**. All posts must be substantive to receive full credit.

Grading (Discussion Questions)

You are encouraged to take part in the weekly dialogue as much as you would like. Your instructor will rate your discussion posts according to the following guidelines:

Initial posts = 0 – 4 points each

- Points can be deducted for posting late (after the stated deadline), and/or for your post not meeting the requirements for being substantive (see above).

Secondary posts = 0 – 3 points each

- Points can be deducted for your post not meeting the requirements for being substantive (see above).
- All secondary posts are due each week by Sunday night @ 11:59 p.m. No credit will be given for late discussion posts after this time.

Each discussion question is worth 10 points [4 pts. for your initial post; 3 pts. for each secondary post]. Therefore, for weeks with **one discussion question**, you can earn up to a total of **10 points**. For weeks with **two discussion questions**, you can earn up to **20 points**.

These totals will be accumulated throughout the week in your gradebook as your instructor rates your posts. Your **final grade** [0 – 10 or 20] for the entire week will be reflected in your gradebook **no later than Wednesday of the following week**.

Services for Students with Disabilities

In accordance with Section 504 of the Rehabilitation Act and the Americans with Disabilities Act, WJU Disability Support Services office (DSS) provides eligible students with a variety of individualized, reasonable accommodations. These accommodations are intended to assist college students with disabilities in having equal access to regular college programs and activities. Accommodations are determined individually for each student through an interactive process and are based on functional limitations resulting from a documented disability. Recent (within 3 years), verifiable documentation must be provided by a medical doctor or appropriately licensed professional.

Approved accommodations will be provided for students who present instructor with a copy of their Faculty Notification Letter (issued by DSS).

For more information, please visit the Disability Support Services website.

Disability Support Services Contact Information:

WJU Disability Support Services
(916) 577-2253
dss@jessup.edu

Technology Requirements

Sufficient technology tools and Internet access are required when taking a course through Jessup Online. The following list will help ensure that you are adequately equipped.

Supported Operating Systems

- Windows 8 and Windows 10
- MacOS is supported for most online course materials

It is highly recommended that you have administrative rights to the computer used for your coursework. If you must use a computer over which you do not have administrative rights (such as a workplace computer), you may experience difficulties with needed functions, such as installing plug-ins. Check with your workplace IT department to ensure that you may access course materials from your company's network.

Productivity Tools

Microsoft Office (this software is available to students at deeply discounted pricing through Microsoft or JourneyEd.com.), Google Docs and Apple Productivity Suite are all supported and capable of meeting the needs of your online course.

WJU Email Account

All students are provided with a WJU email address. It should be used for all course communication between you and your instructor. This will avoid issues with Spam blockers and other problems that may prevent you from receiving email from your instructors. Use of this email account will also enable you to participate in special student offers that are available only to students with an "edu" email address. You can access your Jessup e-mail account at my.jessup.edu.

Supported Browsers

- Google Chrome
- Mozilla Firefox

Browser Settings

Please refer to your browser's Help features to check these settings.

- Pop-Up Blocker should be disabled
- JavaScript should be enabled
- Java should be enabled
- Cookies should be enabled

Plug-ins

The most recent version of the following plug-ins is required for many of the resources available in your online courses:

- Adobe Acrobat Reader
- Apple QuickTime Player
- Java SE 8 or higher

All plug-ins needed to participate in components of your online classes are available at no additional cost. It is recommended that you review the list of plug-ins and install them prior to beginning your coursework.

Screen Settings

Screen resolution (size) should be set at minimum 1024 x 768 or higher.

HelpDesk

There is a link on every Moodle page for 24/7 technical support through an outside vendor.

You can also contact the Jessup HelpDesk (which is not 24/7) through WJU. Email helpdesk@jessup.edu or call 916.577.2345.

Course Grading Explanations

Points	Grade
90-100	A
80-89	B
70-79	C
60-69	D
<59	F

A = Excellent performance. Work is truly exemplary and worthy of emulation by others. Student exceeds expectations and constructively contributes to the learning environment.

B = Above average performance. All assignments are complete and on time and exhibit a complete understanding and an ability to effectively apply concepts.

C = Average performance. Student accomplishes only the minimum requirements or does not complete all requirements. Oral and written communication is at an acceptable level for a college student.

D = Work is below acceptable level for a college student. Student shows only a very basic understanding of the material or does not meet all assignment requirements.

F = Work is not passing. Student's work is incomplete or does not apply information and concepts in a satisfactory manner.

Final Grade Calculation

Assignments	Value
Discussion Questions	20%
Weekly Assignments	30%
Midterm Progress Project	20%
Final Projects (2 of them)	30%
TOTAL:	100%

Course Outline

<i>Week 1</i>	<i>Details</i>	<i>Due</i>	<i>Demand Hours</i>	<i>Course Objective</i>
Topics and Learning Objectives	Course Introduction, Overview of making a game, User Input, Data Input By the end of this week, you should be able to: <ul style="list-style-type: none"> • Explain and create a game loop • Draw images and text • Use the PyGame event system • Process mouse and key events • Load data from an XML file • Compile XML data 			
Setup Assignment	Install pygame <ul style="list-style-type: none"> • Ex. python -m pip install -U pygame --user • Read the attached instruction sheet for detailed instructions (including how to turn in your work). 	Complete/ Incomplete Sunday after Week 1 class 11:59PM PT	0.5 hour	
Reading Assignments	Read <ul style="list-style-type: none"> • XML Default https://www.w3schools.com/xml/default.asp • XML Introduction https://www.w3schools.com/xml/xml_what_is.asp • XML Usage https://www.w3schools.com/xml/xml_used_for.asp • XML Tree https://www.w3schools.com/xml/xml_tree.asp • XML Syntax https://www.w3schools.com/xml/xml_syntax.asp • XML Elements https://www.w3schools.com/xml/xml_elements.asp • XML Attributes https://www.w3schools.com/xml/xml_attributes.asp 		2 hours	
Video Resources	View <ul style="list-style-type: none"> • Course Introduction – roadmap View Demo <ul style="list-style-type: none"> • Bouncing Ball Demo • New Project, Game Loop • Add a Character; Moving the Character 		1.5 hours	

	<ul style="list-style-type: none"> Loading from XML 			
Web Resources (in Readings link)	Review <ul style="list-style-type: none"> https://www.pygame.org/docs/ https://docs.python.org/2/library/xml.etree.elementtree.html 		1 hour	
Discussion	Discuss <ul style="list-style-type: none"> DQ #1: The Game Loop <ul style="list-style-type: none"> In the game loop we separate all the rendering from the all the updating. This means that we sometimes will have to compute values in the update loop to store and use later when rendering. Otherwise, we would not have had to store those values in non-temporary places. Evaluate why it is that we should not render things immediately even if we know where they should go? Discuss the various problems that this might cause. DQ #2: Object-oriented coding <ul style="list-style-type: none"> There are various situations in coding wherein it might be a bad idea to use the object-oriented approach. On the other hand, there are various situations that might warrant its use. In the project that we are developing we control the player character entirely inside a class. Assess why you think doing it that way might be important in this project. 	See <i>Discussion Guidelines</i>	6 hours	
Quiz #1 (Pass/Fail)	Complete <ul style="list-style-type: none"> Programmer Academic Integrity Quiz <ul style="list-style-type: none"> 4 - 6 Questions 	Sunday after Week 1 class 11:59PM PT	0.5 hour	
Weekly Assignment #1	Complete <ul style="list-style-type: none"> Fill the world with grass. Read the attached instruction sheet for detailed instructions. 	Sunday after Week 1 class 11:59PM PT	1.5 hours	
Weekly Assignment #2	Complete <ul style="list-style-type: none"> Add an image that represents the character. Read the attached instruction sheet for detailed instructions. 	Sunday after Week 1 class 11:59PM PT	3 hours	
Weekly Assignment #3	Complete	Sunday after Week 1 class	4 hours	

	<ul style="list-style-type: none"> Devise a way for the character image to run around on screen in response to keyboard input. Read the attached instruction sheet for detailed instructions. 	11:59PM PT		
		TOTAL HOURS FOR THE WEEK:	20	

<i>Week 2</i>	<i>Details</i>	<i>Due</i>	<i>Demand Hours</i>	<i>Course Objective</i>
Topics and Learning Objectives	Physics Collision & Forces, Game AI By the end of this week, you should be able to: <ul style="list-style-type: none"> Use PyMunk for physics Show understanding of physics object types Make physics shapes Detect collisions Apply physics forces Implement a state machine to create game AI Explain the Decision verses the Action Produce designer configurable game AI 			
Setup Assignment	Install pymunk <ul style="list-style-type: none"> Ex. python -m pip install pymunk Read the attached instruction sheet for detailed instructions (including how to turn in your work). 	Sunday after Week 2 class 11:59PM PT	0.5 hour	
Reading Assignments	Read <ul style="list-style-type: none"> Pymunk overview http://www.pymunk.org/en/latest/overview.html 		0.5 hour	
Video Resources (in Course Materials Folder)	View <ul style="list-style-type: none"> Overview of PyMunk View Demo <ul style="list-style-type: none"> Attach collision to world objects Throw a rock Create Enemy from XML file State Machine and Game AI Apply AI to the Enemy 		1.5 hours	
Web Resources (in	Review		0.5 hour	

Course Materials Folder)	<ul style="list-style-type: none"> http://www.pymunk.org/en/latest/ 			
Discussion	<p>Discuss</p> <ul style="list-style-type: none"> DQ #1: Collision <ul style="list-style-type: none"> Suppose you were to write your own physics engine. In your engine you have 3 moving boxes and one stationary box. What sort of problems do you think might come up that would make it more difficult to see if the moving boxes collided and where they collided? DQ #2: Game AI <ul style="list-style-type: none"> Game AI is different from traditional AI. Someone who is developing traditional AI is trying to achieve actual intelligence, but in game AI one is going for what is mostly only the appearance of intelligence. Why do you think games are designed this way? What are the pros and cons? 	See Discussion Guidelines	6 hours	
Weekly Assignment #4	<p>Complete</p> <ul style="list-style-type: none"> Create Oval and Capsule custom collision types for our 2D game. Read the attached instruction sheet for detailed instructions. 	Sunday after Week 2 class 11:59PM PT	6 hours	
Weekly Assignment #5	<p>Complete</p> <ul style="list-style-type: none"> Allow the player to "shoot" the enemy Read the attached instruction sheet for detailed instructions. 	Sunday after Week 2 class 11:59PM PT	4.5 hours	
		TOTAL HOURS FOR THE WEEK:	19.5	

Week 3	Details	Due	Demand Hours	Course Objective
--------	---------	-----	--------------	------------------

Topics and Learning Objectives	Animation, Camera, Audio, User Interface Screen Elements By the end of this week, you should be able to: <ul style="list-style-type: none"> • Animate a character • Move the in-game camera • Play music and sound effects in-game • Build a UI engine • Demonstrate UI anchors • Display UI Images, Text, and Buttons • Handle a 9-slice image • Process UI Button clicks 			
Video Resources (in Course Materials Folder)	View Demo <ul style="list-style-type: none"> • Animate a Character; Walking Animation • Walking Sound • Camera Movement • Displaying a UI Image • Displaying a UI Text element • Process a button click 		2 hours	
Web Resources (in Course Materials Folder)	Review <ul style="list-style-type: none"> • Pygame.mixer https://www.pygame.org/docs/ref/mixer.html • Pygame.mixer.music https://www.pygame.org/docs/ref/music.html • 9-slice scaling https://en.wikipedia.org/wiki/9-slice_scaling 		1.5 hours	
Discussion	Discuss <ul style="list-style-type: none"> • DQ #1: Audio <ul style="list-style-type: none"> ◦ There are many different approaches to triggering audio in response to things that happen in the game. You can simply start the sound effect right in the code where the action is started, or you can use an event system, or a mediator pattern, or an observer pattern. Critique each of these for what is best in what situations? • DQ #2: UI Engine <ul style="list-style-type: none"> ◦ When making a UI engine you may want to use the composite pattern by adding container groups as an element option. What use would this have and how would you approach doing it in code? 	See <i>Discussion Guidelines</i>	6 hours	
Weekly Assignment #6	Complete <ul style="list-style-type: none"> • Add the attack animation when the character throws the rock. • Read the attached instruction sheet for detailed instructions. 	Sunday after Week 3 class 11:59PM PT	3 hours	

Weekly Assignment #7	Complete <ul style="list-style-type: none"> Make the player health bar work. Read the attached instruction sheet for detailed instructions. 	Sunday after Week 3 class 11:59PM PT	5.5 hours	
		TOTAL HOURS FOR THE WEEK:	18	

<i>Week 4</i>	<i>Details</i>	<i>Due</i>	<i>Demand Hours</i>	<i>Course Objective</i>
Topics and Learning Objectives	Load/Save Game Data, Create a Game Server By the end of this week, you should be able to: <ul style="list-style-type: none"> Load/Save game to file Set up a database Show knowledge of databases and SQL Modify the database with SQL from Python Create a server with REST API Allow the client to connect to the server 			
Setup	Install MySQL Community Server <ul style="list-style-type: none"> https://dev.mysql.com/downloads/mysql/ Read the attached instruction sheet for detailed instructions (including extra steps that you need to take after installation). Install Python mysql <ul style="list-style-type: none"> Ex. pip install mysql-connector-python Read the attached instruction sheet for detailed instructions (including how to turn in your work). 	Sunday after Week 4 class 11:59PM PT	2 hours	
Video Resources (in Course Materials Folder)	View Demo <ul style="list-style-type: none"> Pause-game mechanics Streaming of Music; Load/Save the Volume Create a server project; Database from Python REST API Introduction; Add REST API to the server 		1.5 hours	
Web Resources (in Course Materials Folder)	Review <ul style="list-style-type: none"> Input and Output https://docs.python.org/3/tutorial/inputoutput.html Data Types https://dev.mysql.com/doc/refman/8.0/en/data-types.html Python MySQL Create Database https://www.w3schools.com/python/python_mysql_create_db.asp 		2 hours	

	<ul style="list-style-type: none"> Designing a RESTful API with Python and Flask https://blog.miguelgrinberg.com/post/designing-a-restful-api-with-python-and-flask 			
Discussion	<p>Discuss</p> <ul style="list-style-type: none"> DQ #1: Saving the Game <ul style="list-style-type: none"> In the game that we are making, the designer can add enemies through the XML document. However we want to save the positions of those enemies in the database. If the designer regularly adds, removes and moves these enemies, this means what is saved in the database must change as well. Discuss (Devise, critique) techniques for keeping them in-synch and discuss the pros and cons for the various techniques. DQ #2: REST API <ul style="list-style-type: none"> One of the rules for REST API is that it must be employ a “stateless protocol” (look it up). Discuss what you think are the pros and cons of this rule and how it will impact how you use REST API. 	See <i>Discussion Guidelines</i>	6 hours	
Weekly Assignment #8	<p>Complete</p> <ul style="list-style-type: none"> Add an ability for the user to start a new game or load an old game. Read the attached instruction sheet for detailed instructions. 	Sunday after Week 4 class 11:59PM PT	7 hours	
Midterm 2D Game Progress Project Turn-in	<p>Turn-in</p> <ul style="list-style-type: none"> Turn in your 2D game client completed to the point just before adding the server connection. That is, do not turn in the server that we just started working on or any code in the client to connect to the server. Submit your entire project folder (minus the server) to Moodle as a zip file. 	Sunday after Week 4 class 11:59PM PT		
		TOTAL HOURS FOR THE WEEK:	18.5	

<i>Week 5</i>	<i>Details</i>	<i>Due</i>	<i>Demand Hours</i>	<i>Course Objective</i>
---------------	----------------	------------	---------------------	-------------------------

Topics and Learning Objectives	TCP-IP Connection, Messaging, Server Spawning, Moving of Characters/Objects, Chat By the end of this week, you should be able to: <ul style="list-style-type: none"> • Set up a TCP-IP connection • Set up a client-server messaging system • Have the client and server share major sections of code • Communicate movement of objects between client and server • Spawn and move projectiles in a client-server setup • Set up server-client chat 			
Reading Assignments	Read <ul style="list-style-type: none"> • Socket Programming HOWTO https://docs.python.org/3/howto/sockets.html 		1 hour	
Video Resources	View Demo <ul style="list-style-type: none"> • Set up a TCP-IP connection • Create a messaging system • Have the server run the client World code • Add the UI for chat to use for the assignment 		1.5 hours	
Web Resources (in Course Materials Folder)	Review <ul style="list-style-type: none"> • Higher-level threading interface https://docs.python.org/2/library/threading.html 		0.5 hour	
Discussion	Discuss <ul style="list-style-type: none"> • DQ #1: Common Code <ul style="list-style-type: none"> ◦ Studios will often choose to have the client and server share large sections of code and sometimes even all of the code. This is to reduce duplicating code and to reduce bugs relating to the client and server code getting out of sync. Discuss the kinds of problems that you think might be caused by sharing code in this way and how you might address it. • DQ #2: Race Condition <ul style="list-style-type: none"> ◦ When working with threads or network communication there is the possibility that you can end up with a bug called a race condition. This is a bug that happens randomly only sometimes because two independent events or messages will not always happen in the same order. They are racing to be the first to happen. However the code needs them to happen in the right order. What might you do to try to debug and fix this 	See <i>Discussion Guidelines</i>	6 hours	

	kind of problem? (Note that in one of the demos we are doing something to avoid a race condition when trying to connect the client and server.)			
Weekly Assignment #9	Complete <ul style="list-style-type: none"> Transmit player and enemy movement between the server and client. Read the attached instruction sheet for detailed instructions. 	Sunday after Week 5 class 11:59PM PT	8 hours	
Weekly Assignment #10	Complete <ul style="list-style-type: none"> Send chat messages to the server. Read the attached instruction sheet for detailed instructions. 	Sunday after Week 5 class 11:59PM PT	3 hours	
		TOTAL HOURS FOR THE WEEK:	20	

<i>Week 6</i>	<i>Details</i>	<i>Due</i>	<i>Demand Hours</i>	<i>Course Objective</i>
Topics and Learning Objectives	Multiplayer game, Using OpenGL By the end of this week, you should be able to: <ul style="list-style-type: none"> Have two players on different clients play at once on the same server in the same world. Use Github Set up a game using OpenGL Display a 3D rotating cube Code vertices, surfaces in 3D graphics Create Diffuse lighting Use a Quaternion 			
Organize Assignment	Form Teams <ul style="list-style-type: none"> Forms groups of 2 or 3. (No you may not form a group of 1 or 4) Read the attached instruction sheet for detailed instructions (including how to turn in your work). 	Complete/ Incomplete Sunday 11:59 PM PT	1 hour	
Reading Assignments	Read <ul style="list-style-type: none"> Understanding the GitHub flow https://guides.github.com/introduction/flow/ Hello World https://guides.github.com/activities/hello-world/ Git Handbook https://guides.github.com/introduction/git-handbo 		2 hours	

	ok/			
Video Resources (in Course Materials Folder)	View Demo <ul style="list-style-type: none"> Set up the server to handle multiple player connections; Make the client allow a second player Set up a new project for a 3D game using OpenGL; Show a rotating cube Ambient, Diffuse & Specular lighting; Show usage of diffuse lighting in code 		1.5 hours	
Discussion	Discuss <ul style="list-style-type: none"> DQ #1: Team Coding <ul style="list-style-type: none"> In the field you will be working on the same coding project as other people because the project is too big for one person to meet the deadline. Discuss logistical problems with this and offer solutions. 	<i>See Discussion Guidelines</i>	3 hours	
Project Completion	Complete <ul style="list-style-type: none"> Complete the 2D game Get the two player game fully working Players must see each other moving Enemies must be able to choose either player to chase 	Turn in at the end of the course as the required 2D project	7 hours	
Weekly Assignment #11	Complete <ul style="list-style-type: none"> First group assignment for the 3D game. Read the attached instruction sheet for detailed instructions. 	Sunday after Week 6 class 11:59PM PT	5.5 hours	
		TOTAL HOURS FOR THE WEEK:	20	

<i>Week 7</i>	<i>Details</i>	<i>Due</i>	<i>Demand Hours</i>	<i>Course Objective</i>
Topics and Learning Objectives	Hardware Acceleration and Shaders, 3D Collision Detection By the end of this week, you should be able to: <ul style="list-style-type: none"> Use Hardware Acceleration with OpenGL Decipher the basics of Shaders in 3D graphics Detect collisions in 3D 			
Video Resources	View Demo		1 hour	

(in Course Materials Folder)	<ul style="list-style-type: none"> Change the cube rendering to use hardware acceleration Create a border out of OpenGL line drawing 			
Discussion	<p>Discuss</p> <ul style="list-style-type: none"> DQ #1: 3D Game Development <ul style="list-style-type: none"> Building a game that is in 3D is exponentially more difficult than building one in 2D, although 2D games can sometimes have challenges that 3D games don't have. Thinking about camera movement, physics, animation, models, textures, rendering, gameplay mechanics, etc. discuss two specific development problems that 3D games can have that you wouldn't have to deal with in making a 2D game. 	See <i>Discussion Guidelines</i>	3 hours	
Weekly Assignment #12	<p>Complete</p> <ul style="list-style-type: none"> Second group assignment for the 3D game Read the attached instruction sheet for detailed instructions. 	Sunday after Week 7 class 11:59PM PT	7.5 hours	
Weekly Assignment #13	<p>Complete</p> <ul style="list-style-type: none"> Third group assignment for the 3D game Read the attached instruction sheet for detailed instructions. 	Sunday after Week 7 class 11:59PM PT	7.5 hours	
2D Game Project	<p>Submit</p> <ul style="list-style-type: none"> Submit your completed 2D Game Project in Moodle 	Sunday after Week 7 class 11:59PM PT		
3D Game Project	<p>Submit</p> <ul style="list-style-type: none"> Submit your completed 2D Game Project in Moodle 	Sunday after Week 7 class 11:59PM PT		
		TOTAL HOURS FOR THE WEEK:	19	