



CS42: Introduction to Game Coding

Section 5363, Spring 2021 Course Syllabus

Instructor: Ethan Wilde (he/him/his), ewilde@santarosa.edu

Course Description

This course introduces students to the design, development, and coding of simple graphical computer-based games. During the course, students will be introduced to various game engines and development environments. Students will gain experience working individually and in a team environment. Emphasis is placed on engaging players through compelling application of game mechanics, dynamics, and aesthetics, as well as on playtesting and iterative development to ensure user-centered design goals are met.

Recommended Preparation: Course completion of CS 110A and/or programming experience

Prerequisites: None

Whether you want to become a member of a professional AAA game development team, or just want to try your hand as an independent game developer, mastery of the game development process, including coding, is essential to those goals. We will work with the ECMA-compliant JavaScript language and real-world tools to develop proficiency in the creation of browser-based, mobile and console games. This course will follow the Transformational Process in development of final game projects.

Student Learning Outcomes

Students will be able to:

1. Understand and apply systems-level thinking and game development methodology best practices to the design and development of simple graphical computer-based games.
2. Develop team-based game creation skills that cover game logic sequencing; storyboarding; artistic and technical coordination; and understanding of technical requirements and limitations.

Upon completion of the course, students will be able to:

1. Explain factors that motivate game players and apply those to the design of games that satisfy the needs of players. This includes the design of a logical sequence of game rules, play flow, and interaction opportunities.
2. Develop game coding skills to implement simple versions of games they design. Desired functions include: graphical, interactive user interface; sprite creation, collision detection, applied simulation of physical forces; classes, object instantiation; and modular implementation.
3. Determine that games are of measurable high quality and error-free through playtesting, iterative development, and an adherence to best practices in quality assurance, including a proper and documented testing process.

4. Work independently and in teams using an iterative process to prepare and create Game Design Documents, storyboards, game assets and scripting to create game simulations.
5. Utilize the concepts of game mechanics, dynamics, and aesthetics to discuss game play of games of different genres on varied platforms.

Topics and Scope:

- I. Game Development Fundamentals
 - A. Introduction to systems thinking
 1. Flow control and diagramming
 2. Visual scripting systems
 - B. Introduction to scripting
 1. Variables and data types
 2. Expressions and operators
 3. Control structures
 4. Functions
 5. Objects and classes
 6. Events and triggers
 - C. Mathematics and physics fundamentals
 1. Cartesian coordinate systems
 2. World space, object space, camera space
 3. Vectors, forces and physics simulations
 - D. Development methodologies
 1. Unified and agile processes
 2. Iterative design
 3. Human-centered design
 4. Design patterns
 5. Project management and quality assurance
- II. Considerations of Game Creation
 - A. Types of games
 - B. Game mechanics, dynamics, and aesthetics
 1. Mechanics of gameplay
 2. Dynamics of gameplay
 3. Aesthetics of gameplay
 - C. Motivational and emotional aspects of playing games
 - D. Generic game design rules
 - E. Technologies, platforms, tools
 - F. Game development frameworks and engines
- III. Game Development Process
 - A. Setting goals for the game
 - B. Developing a storyboard and designing gameplay
 - C. Drafting a Game Design Document (GDD)
 - D. Iterative prototyping
 - E. Stakeholder feedback loop and playtesting
 - F. Features, functions, and program components
 1. Game loop
 2. Assets
 3. User input
 4. Sprites and collisions
 5. Audio representation
 - G. Game rules
 - H. Navigation and wayfinding
 - I. Graphical user interfaces
- IV. Supporting Disciplines

- A. Game playtesting and validation
 - 1. Quality assurance testing plan
 - 2. Bug tracking and resolution
- B. Working as a team
- C. Productization and packaging
- D. Art work

Assignments:

1. Read approximately 25-30 pages a week
2. Prepare 1-2 written Game Design Document(s) (GDD) that closely mirror the documentation process used in the field. (3-7 pages each)
3. Regular group discussion contributions (0 - 12) such as:
 - A. Play logic and navigation design
 - B. Storyboarding, player motivation
 - C. Theories of game development
 - D. Design and implementation of game logic (rules and programmatic navigation)
 - E. Use of design patterns
 - F. Graphical user interface design
 - G. Implementation technologies and validation
4. Game creation and scripting assignments that solve particular technical challenges (6 - 12)
5. Midterm and final examinations that evaluate critical thinking skills such as:
 - A. Game requirements analysis and validation
 - B. Game storyboard design and implementation
 - C. Software games, techniques, and technologies
 - D. Tools and techniques used in the software industry
6. A final class project that includes topics such as:
 - A. Selection of a viable, realistic project
 - B. Interviews with players and project stakeholders to collect requirements and risks
 - C. Creation of a comprehensive software-based game
 - D. Summary of playtesting findings and suggested improvements for future versions

Class Meetings

Spring 2021 Schedule

Class Delivery	Day and Time	Platform
Online Meetings	Weeks start on Wednesdays	Canvas shell
Live weekly Web conference (optional) https://santarosa-edu.zoom.us/j/961817861	Wednesdays, 11:30am - 12:30pm	Zoom

All class materials for each module will be released online in Canvas on Wednesdays throughout the entire semester. A weekly live meeting will be held on Wednesdays, 11:30am - 12:30pm. Attendance at the live web conferences is optional but highly recommended. Every student must either attend the live web conference or watch the entire screencast recording of the web

conference. Use the Zoom link above to join live only on Wednesdays, 11:30am - 12:30pm. To view any weekly lecture's recorded screencast, visit the Screencast page for any week in the Modules section.

Instructor Contact

Ethan Wilde

Email: ewilde@santarosa.edu

Phone: 707-527-4855

Spring 2021 Office Hours
January 25 – May 17, 2021

Day	Time	Location
Mondays (online)	12:00pm - 6:00pm	Online: Email ewilde@santarosa.edu or Skype ethanwilde

» [Reserve a future office hour appointment](#)

I typically respond to emails within 48 hours, weekends excepted. I never respond on Sundays.

Course Web Site

Students will use the Canvas course web site for assignment instructions, submitting assignments, viewing classmates' work, sharing resources, and viewing grades. *The Google Chrome browser is recommended for viewing the Canvas-powered course site. Internet Explorer is not recommended.*

Textbooks

Eloquent JavaScript (3rd)
Marijn Haverbeke
978-1593279509 (ISBN 13)
[Free PDF eBook available](#)

[The Transformational Framework](#) (1st)
Sabrina Culyba
978-1387895274 (ISBN 13)
[Free PDF available](#)

All required textbooks are available online without cost. If you would like a printed copy, you can locate and order textbooks online via the [SRJC Bookstore](#).

Equipment

- A **personal computer**, either at home, work or on the Santa Rosa or Petaluma campuses.

Required Software + Services

- **Internet access**
- **Web browser**
 - [Google Chrome](#) strongly recommended
- **Coding education site**
 - [CodeCombat](#) free account required for all students.
- **Cloud hosting + development services**
 - [Repl.it](#) IDE (Integrated Development Environment) required for all students, starting Week 2, for hosting class assignments. Repl.it will provide a text editor and file transfer support without any additional software needed. *Complete the hosting survey to get your free account.*
 - [PlayCanvas](#) WebGL Game Engine and IDE required for all students, starting Week 10, for hosting 3D-based class assignments. PlayCanvas will provide a text editor and file transfer support without any additional software needed. *You may create a free account at PlayCanvas or use Repl.it to host your PlayCanvas projects.*
 - [MIT Scratch](#) simple coding IDE that is optional for all students. Scratch will provide a code editor without any additional software needed. *You may create a free account at MIT Scratch.*
- **2D Graphics software** such as:
 - Adobe Photoshop, part of a [Creative Cloud](#) subscription
 - [Gimp](#) open source application
 - [Pixlr](#) browser-based image editor
 - [Diagrams.net](#) browser-based drawing application
- **Spritesheet and tilemap editing software** such as:
 - Tiled map editor found online at <http://www.mapeditor.org/>
 - TexturePacker texture/sprite sheet editor found online at <https://www.codeandweb.com/>
- **Phaser.js JavaScript library, version 3.x:**
 - Found online at phaser.io
- **JavaScript code validator:**
 - <http://esprima.org/demo/validate.html>
- **PDF display software** such as:
 - [Adobe Reader](#)

Optional Software

The additional software listed below is often used for game development. Our IDEs – the Repl.it cloud-based IDE for our work with Phaser.js, the integrated PlayCanvas IDE, and the browser-based Scratch software – both provide code editors and file transfer support without any additional software needed.

- **3D modeling software** including:
 - [Blender](#)
 - [Trimble Sketchup Pro](#) (\$49 education license)
- **Enterprise-class game development engine software** including:
 - [Unity3D, Personal Edition](#)
 - [Unreal Engine](#)

- **Code editor** such as:
 - [BBEdit](#) (Mac OS only)
 - [Brackets](#) (Windows, Mac OS)
 - [Microsoft Visual Studio Code](#) (Windows, Mac OS)
 - [Sublime Text](#) (Windows, Mac OS, Linux)
- **Integrated Development Environment** for making native mobile apps:
 - Apple Xcode (Mac OS only)
- **Additional Web browsers** including:
 - [Mozilla Firefox](#)
 - Apple Safari (Mac OS only)
 - Microsoft Edge (Windows 10 only)
- **File Transfer Protocol (FTP) software** such as:
 - [CyberDuck](#) (Mac OS and Windows, free)
 - [Fetch](#) (Mac OS only)
 - [WinSCP](#) (Windows only)

Important Dates

Day Class Begins: Wednesday, January 20, 2021

Day Class Ends: Friday, May 28, 2021

Last Day to Add without instructor's approval: Tuesday, January 26, 2021

Last Day to Drop with refund: Sunday, January 31, 2021

Last Day to Add with instructor's approval: Sunday, February 7, 2021

Last Day to Drop without a 'W' symbol: Sunday, February 7, 2021

Last Day to Opt for Pass/No Pass: Sunday, February 28, 2021

Last Day to Drop with a 'W' symbol: Sunday, April 25, 2021

Dropping the Class

If you decide to discontinue this course, it is your responsibility to officially drop it. A student may be dropped from any class when that student's absences exceed ten percent (10%) of the total hours of class time. It is strongly advised that if you need to miss more than one class/homework deadline in a row that you contact the instructor to avoid being dropped from the class.

Attendance

Students who fail to complete the requirements of the first and second class modules will be dropped by the instructor. **Students must view and participate in online materials released each week in the Modules section of the course Canvas website.**

Pass-NoPass (P/NP)

You may take this class P/NP. You must decide before the deadline, and add the option online within your student portal or file the P/NP form with Admissions and Records. With a grade of C or better, you will get P.

You must file for the P/NP option by February 28, 2021. Once you decide to go for P/NP, you cannot change back to a letter grade. If you are taking this course as part of a certificate program, you can probably still take the class P/NP. Check with a counselor to be sure.

Instructor Announcements

The instructor will post announcements on the “Announcements” page in Canvas throughout the semester. Canvas notifies students according to their preferred Notification Preferences.

Late Policy

Please make a plan before the course starts to allow yourself the necessary time each week to complete the required reading, watching, online discussion posting, and assignments. The official Course Outline of Record for this four-unit semester-length course stipulates that each student is expected to complete 210 hours of learning for the class. This works out to 12 hours per week for each of the seventeen weeks of regular instruction along with 6 hours for Finals Week. If you plan accordingly, you can avoid submitting assignments late.

All assignments are due at 11:59pm Pacific time on the **Tuesday** corresponding to the due date. A late submission will receive a 10% penalty for each week it is late. Submissions more than two weeks late are not accepted without prior written arrangement.

Exams

There will be online midterm and final exams. The material comes from the textbooks, class lectures and supplemental materials. If any exam is missed, a zero will be recorded as the score, unless you have made prior written arrangements with me. It is your responsibility to take the exams by the due date.

Grading Policy

Click the “Grades” link in Canvas to keep track of your grades. I grade once a week and post grades and comments in the Canvas gradebook.

Grades will be assigned as follows:

Letter Grade	Percentage	Points Total
A	90% - 100%	900 points or more
B	80% - 89%	800 to 899 points
C	70% - 79%	700 to 799 points

Letter Grade	Percentage	Points Total
D	60% - 69%	600 to 699 points
F	59% or lower	599 points or less

Grading Breakdown

Percent	Points	Grading Category
36.5%	365 points	Assignments: Problem Solving
30.0%	300 points	Assignments: Skill Demonstration (Final Project)
10.0%	100 points	Assignments: Game Design Documents
9.0%	90 points	Discussions + Participation
7.0%	70 points	Midterm Exam
7.0%	70 points	Final Exam
0.5%	5 points	Syllabus Quiz
100%	1000 points	1000 points possible

Standards of Conduct

Students who register in SRJC classes are required to abide by the SRJC Student Conduct Standards. Violation of the Standards is basis for referral to the Vice President of Student Services or dismissal from class or from the College. See the [Student Code of Conduct page](#).

Collaborating on or copying of tests or homework in whole or in part will be considered an act of academic dishonesty and result in a grade of zero for that test or assignment, except for assignments that allow collaboration. Students are encouraged to share information and ideas, but not their work. See these links on Plagiarism:

[SRJC Writing Center Lessons on Avoiding Plagiarism](#)
[SRJC's Statement on Academic Integrity](#)

I expect each student to maintain high standards of civility and respect when communicating with each other. The following rules of netiquette should be observed in all class discussions and communications:

- Be kind and respectful to others
- Use full sentences
- Avoid jargon and acronyms
- Use language that supports others

Special Needs

Every effort is made to conform to accessibility standards for all instructor-created materials. Students should contact their instructor as soon as possible if they find that they cannot access any course materials. Students with disabilities who believe they need accommodations in this class are encouraged to contact Disability Resources by calling (707) 527-4278 or visit online at drd.santarosa.edu.

Student Health Services

Santa Rosa Junior College offers extensive health services to students. Visit Student Health Services online at shs.santarosa.edu or call them at (707) 527-4445.

Course Outline

Start Date	Canvas Module	Topics	Assignments	Discussions
1/20	Week 1	The World of Game Development / Learn to Code by Playing	<ul style="list-style-type: none"> • #1: JavaScript Self-Assessment • #2: Play a Game, Learn to Code (2 weeks) • Hosting Signup Survey • Syllabus Quiz 	#1: Check-in Discussion (2 weeks)
1/27	Week 2	Game Development Process / JavaScript Basics	<ul style="list-style-type: none"> • #3: Transformational Process 	
2/3	Week 3	JavaScript, Systems Thinking + Browser-Based Games	<ul style="list-style-type: none"> • #4: First Game 	#2: History + Origins of Games (2 weeks)

Start Date	Canvas Module	Topics	Assignments	Discussions
2/10	Week 4	Get Started with Phaser Game Engine / The VR Experience	<ul style="list-style-type: none"> #5: First Phaser Game 	
2/17	Week 5	Sprites, Controls + Basic Physics	<ul style="list-style-type: none"> #6: Working with Sprites + Controls 	#3: Game Typologies (2 weeks)
2/24	Week 6	Spritesheets, Texture Atlases + Animation	<ul style="list-style-type: none"> #7: Using Animation + Spritesheets 	
3/3	Week 7	Tilesets, Tilemaps + Cameras / Defining a Game: GDD	<ul style="list-style-type: none"> #8: Using Level Maps + Tiles #9: Draft GDD (2 weeks) 	#4: Approaches to Game Design (2 weeks)
3/10	Week 8	UI, Scenes + Sound	<ul style="list-style-type: none"> #10: Adding UI + Sound 	
3/17	Week 9	NPC AI + Pathfinding / Project Team Formation	<ul style="list-style-type: none"> #11: Enemy AI Pathfinding 	#5: GDD Presentations (2 weeks)
3/24	No Class	Spring Break		
3/31	Week 10	Advanced Physics / Midterm Exam	<ul style="list-style-type: none"> #12: Use MatterJS Physics Midterm Exam 	

Start Date	Canvas Module	Topics	Assignments	Discussions
4/7	Week 11	NPC AI + Finite State Machines	<ul style="list-style-type: none"> • #13: Enemy AI + Finite State Machines • #14: Final Game Design Document (2 weeks) 	#6: Team Roles + Team Formation (2 weeks)
4/14	Week 12	The World of 3D / Start Your Final Project	<ul style="list-style-type: none"> • #15: First 3D Game 	#7: Working in 3D (2 weeks)
4/21	Week 13	Project Sprint 1: Prototype	<ul style="list-style-type: none"> • #16: Prototype 	#8: Project Playtesting (5 weeks)
4/28	Week 14	Playtest + Project Sprint 2: Rough Draft	<ul style="list-style-type: none"> • #17: Rough Draft 	
5/5	Week 15	Playtest + Project Sprint 3: Alpha	<ul style="list-style-type: none"> • #18: Alpha 	
5/12	Week 16	Playtest + Project Sprint 4: Beta	<ul style="list-style-type: none"> • #19: Beta 	
5/19	Week 17	Native iOS Games / Playtest + Project Sprint 5: Final Build + Documentation	<ul style="list-style-type: none"> • #20: Final Build + Documentation 	
5/24 Mon - 5/28 Fri	Week 18	Final Exam / Final Project Review (no regular class)	<ul style="list-style-type: none"> • Final Exam 	#9: Final Project Sharing

Note to students: the assignments listed above will become available as modules are released in sequence each week. To view course content, go to **Modules**.

All of the original material found on this online course website is the property of the instructor, Ethan Wilde. My lectures and course materials, including slide presentations, online materials, tests, outlines, and similar materials, are protected by U.S. copyright law and by College policy. I am the exclusive owner of the copyright in those materials I create. You may take notes and make copies of course materials for your own use. You may also share those materials with another student who is registered and enrolled in this course. You may not reproduce, distribute or display (post/upload) lecture notes or recordings or course materials in any other way — whether or not a fee is charged — without my express written consent. You also may not allow others to do so.

© 2021 Ethan Wilde.