

mouseandthebillionaire /
Cart315

<> Code

Issues

Pull requests

Actions

Projects

Security

Insights



main



Cart315 / course-information / outline.md



mouseandthebillionaire Initial commit

last week



91 lines (53 loc) · 6.17 KB

Preview

Code

Blame

Raw

[CART315](#) | [Schedule](#) | [Projects](#) | [Resources](#)

CART 315: Game Prototyping – Winter 2024 – 3 credits

Thursdays, 17:45 – 21:45 in EV 5.615 Instructor: Matthew Bethancourt |
matthew.bethancourt@concordia.ca

Territorial acknowledgment

We acknowledge that Concordia University is located on unceded Indigenous lands. The Kanien'kehá:ka Nation is recognized as the custodians of the lands and waters on which we gather today. Tiohtiá:ke/Montreal is historically known as a gathering place for many First Nations. Today, it is home to a diverse population of Indigenous and other peoples. We respect the continued connections with the past, present and future in our ongoing relationships with Indigenous and other peoples within the Montreal community.

Description

Students study specialized game technology, create a series of digital game prototypes, and are introduced to higher level programming concepts pertaining to interactive applications. Efficient approaches to the design and development of complex interactive software, such as iterative development and rapid prototyping, are explored.

This class will take you through the pre-production and development of a series of game prototypes building up to a final game project, using the Unity game engine and the C# language. This will require continuous iterative development and testing, and therefore students will be evaluated on their progress every week.

Prerequisite: CART 263 or COMP 218 or COMP 248; or written permission of the Department.

Learning Objectives

After completing this course, students should be able to:

- Use rapid prototyping and iterative design methods to design and develop a game from beginning to end.
- Learn basic programming skills and game development concepts using C# and the Unity Engine
- Run effective playtests and incorporate playtesting feedback into your design process.
- Plan development to mitigate risks and meet deadlines autonomously

Course Materials

- [Introduction to Game Design, Prototyping, and Development](#) by [Jeremy Gibson Bond](#)
- [Unity Version 2020.3.xx LTS](#)
- [Rider](#) (which you can get a free student license for) or another code editor
- [Github Desktop](#) or a git client you are familiar with
- The [Discord](#) Client

Assignments and Class Structure

This course will be an organic mix of lecture and practice with the professor working to explain the concepts, give examples, and also oversee how well students grasp the concepts discussed as they work on assigned projects. There will be multiple game programming exercises, a mid-semester practicum to test this understanding, and two major projects with corresponding design statements, presentations and critiques which will be spread over the course of the allotted 15 weeks.

Prep Materials

Each week I will assign readings/videos/activities that must be done *before* the class sessions. Not all material will be discussed in class every week, but it lays the foundation for what we will accomplish during our time together.

Class Sessions

Each class will begin with a discussion period in the classroom. I will answer questions about the week's material and build example prototypes based on what we have covered so far. Additional topics for discussion will be introduced as necessary.

Studio time

The second half of some sessions will be studio time for you to work on exercises/projects and get any assistance and/or feedback as needed.

Attendance of the studio period is technically optional, but students are *very heavily encouraged* to take the time to connect with the instructor and to work in the group setting.

Projects

There will be three graded projects during the course. The first project is a quick turnaround to introduce proper scoping of small projects, and will be graded pass/fail. The second and third projects will scaffold on work done in class and on each other, increasing in complexity, and will each go through a complete design cycle of ideation, prototyping, testing, and refinement.

GitHub

You will need to create your own repository on GitHub to save, track, and present coursework. We will be using the [MDM Framework](#) for tracking progress and changes in both the design thinking and execution of your work.

Course Website

You're looking at it! I will post most things here (including details on lecture schedules, readings, examples, assignments, etc.). I will be posting grades to you via the course [Moodle](#).

Evaluation

Specific evaluation requirements will be provided when assigned.

- Limited Scope Introduction Project @ 5% (pass/fail)
- Scaffolded Shmup Project @ 15%
- Debugging Practicum @ 10%
- Exploration Prototypes (2) @ 5% each / 10% total
- Iterative Prototypes (2) @ 5% each / 10% total
- Playtests (4) @ 5% each / 20% total
- Final Game and Presentation @ 15%
- Weekly Design Journal @ 15%

Accessibility & Inclusion

Please do not hesitate to contact me to bring up any issue related to accessibility, or if you require any accommodations in order to participate - whether that is because of disability, interactions with other students, work or child-care responsibilities, trauma, or any other reason.

Academic Integrity

Game development frequently involves a lot of code reuse. Whenever you reuse or build on someone else's code, you are expected to appropriately credit and reference its authors. The same is true of imagery, sounds, video, excerpts of literature, art works, and contributions from other students. Failure to do so may be regarded as plagiarism. Importantly, the university doesn't care about intentions when it comes to plagiarism.

Additional Course Polocies:

- [Design and Computation Arts Syllabus](#)