Course Syllabus

COMP 323/488 - Game Design and Development

Dr Nick Hayward

Semester - Spring 2019

Overview

- Lecturer: Dr Nick Hayward
- 2.5 hours weekly taught class
- Units: 3
- Graduate course

Sample course website is currently available at the following URL,

- [http://csteach488.github.io/](http://csteach488.github.io/)

Sample course GitHub repositories can be found at the following URL,

- [https://github.com/csteach488](https://github.com/csteach488)

Table of Contents

- Intro
- Description
- Goals
- Outcomes
- Course Assessment
- Course Sections
- Supporting Material

Intro

This semester long course will offer CS department students a practical introduction to *Game Design and Development*. Underlying concepts will be introduced and demonstrated throughout the course with a particular focus on working examples and *playtesting*.

Students will gain practical experience of design and development, which is applied as part of the project-based assessment for the course.

Throughout the course, students will also be exposed to the many and various types of games and gaming environments now available across multiple genres.

The course has been structured to provide logical groupings of common technologies, which complement each other in the development of games and games-based applications.
As such, this course will offer a combination of technical concepts and development, awareness of aesthetic requirements, and cultural perspectives inherently necessary to create well-rounded, performant games and gaming environments.

**Description**

This course studies design, development, and publication of games and game-based applications. This includes example games and designers, industry practices, and team-based project development. Students will acquire an awareness of different game design and development methods, technologies, and techniques suitable for the development of a variety of game based environments. Students will gain experience with various game development options including Python, Pygame, and Unity.

**Goals**

The study of such technologies provides the students with the opportunity to consider and develop two project-based games. The first project may be developed using a combination of Python and Pygame. However, student groups may also select another suitable programming language, subject to instructor approval. This initial game design and concept is then modified and extended using Python and Pygame, or re-imagined using Unity or another approved language and game engine. The final goal of the course is to develop a demonstrable, working game with a combination of the above technologies.

**Outcomes**

Students will develop fundamental skills and knowledge in game design, including formal structures, narrative, and playtesting. Advanced development of games will include many class examples including Python, Pygame, and Unity.

**Course Assessment**

Course assessment will include a combination of group exercises throughout the semester, a development (or DEV) week project, and a final project demonstration and report. Projects may be completed in groups or individually.

**Exercises & Discussions - 20%**

Exercises and discussions will constitute twenty percent of the overall grade, and provide a test of knowledge acquired and understood for each defined section within the course.

Group exercises are used to help develop the course project, whilst also helping to test acquired knowledge. They are scheduled, and organised, to complement course material per week.

An additional weekly component, introduced from approximately week 5 onwards, is an occasional discussion of a chosen game or game-based application. The class is asked to discuss different design, structural, or playtesting considerations and options per week.
Project outline & mockup - 15%

The project outline and mockup is an opportunity to prepare an initial concept and design for the ongoing course project. This design and mockup allows each group to test and begin design of their game.

This is an opportunity to discuss current game outlines, demonstrate any paper prototypes, and receive appropriate feedback for your group’s game idea and concept.

Each project group may also show initial and ongoing concept designs for this application. Such designs should, at a minimum, reflect design notes and material covered in the course’s extra notes.

This project work will be presented to the class in week 5 of the semester.

DEV week project - 25%

The DEV week project is scheduled for week 10 of the semester with group presentations of results. This project includes the development of an initial game or game-based application, which needs to be built from scratch by each project group. It is built upon the technologies discussed and outlined in the first half of the semester, including Python, Pygame, and Unity.

One of the many goals of this initial development is the successful demonstration of considered understanding of the above technologies, and their application for game design and development. Each project group is also required to outline the following in their class presentations,

- choose a game genre
  - why did they choose this genre?
- outline game theme and story
  - what is it about?
  - what is the purpose of the game?
  - outline script for the game
- outline mechanics of game play
  - detail options, logic…
- show storyboards and initial designs
  - storyboards, mockups, prototypes…
- show working example
  - where applicable…
  - detail what does and does not work…

This initial project concept should be playable, for example, by at least one person.

DEV week presentations also include anonymised, collated feedback from peer reviews during class. This helps each project group consider and respond to feedback and suggestions for modifications and improvements, which can then be applied, where appropriate, towards the final project assessment.

Final project - 40%

The final project assessment is a continuation of the group work developed for the DEV week assessment. The primary goal is the modification of the initial game or game-based application to leverage, where appropriate, the benefits of the advanced technologies introduced in the second half of the course. These include additional
concepts for custom game engines, rendering, and general performance improvements with Python, Pygame, and Unity. Additional material is also provided to introduce advanced development with Unity.

Each group is also required to respond to the collated feedback received following the DEV week project presentations, and then demonstrate how and where they have incorporated suitable updates and modifications in the final game. For example,

- where and why did a project group update their game?
  - how did playtesting influence their updates and designs?
- what are the perceived and tested benefits of these updates?

As part of the final presentation, and final published project report, each project group is also required to carefully explain their design choices, including a description of design patterns for UI and interaction. They should also detail layout and data presentation patterns chosen for their given project.

Further details on coursework requirements may be found at the following URL,

- https://csteach488.github.io/coursework/

**Course sections**

The semester long course is divided into the following contiguous sections, which naturally complement each other in the overall development of a game or game-based application.

**Section 1**

We begin with an overview and appreciation of fundamental concepts that unify how games work. This will include, for example, gaming theory related to rules, objectives, and procedures.

As such, a student should be able to clearly define the underlying nature of a game, and, therefore, what makes a game compelling for players.

The first part of the course will begin with an appreciation of game design fundamentals.

For example,

- the underlying structure and role of games
  - what makes a game? i.e. a whole from the sum of its parts
  - various gaming definitions
- the fundamental types of games initially considered relative to gaming theory, e.g.
  - cooperative and non-cooperative
  - simultaneous and sequential move games
  - symmetry and asymmetry in games
  - ... 
- analysis of gaming frameworks, e.g.
  - mechanics, dynamics, and aesthetics (better known as MDA)
- a consideration of formal gaming elements
  - player roles and engagement
  - gaming objectives, rules, and procedures
  - conflicts, resolution, and potential boundaries
• crafting stories and concepts

Section 2

As we transition to designing more formal games and gaming environments, we may now start to consider the following requirements for game design.

• concepts and outlines
• general functionality and goals
• how to define and design fun elements within a game
  o i.e. factors and concepts that encourage play...
• prototypes
• methods, patterns, and stages of development
  o defining required stages for a game outline
  o development patterns suitable for game types
• playtesting games
  o its role in design and development
  o basic concepts and techniques
  o metrics and feedback
  o hands-on application and practice

Section 3

The third part of this course will offer practical guidance and examples for developing games. This will focus upon a series of practical games and gaming solutions, which will help provide a clear reference for game development and design.

This course will use Python as the primary programming language for introductory concepts and patterns. This will be complementary to Pygame modules, and game development using the Simple DirectMedia layer.

Students will learn to develop games and gaming environments using Pygame, and associated modules and technologies. This includes standard and advanced character and environment motion, interaction, and visualisation.

However, students will also see examples of working with 3D game development, including the latest version of Unity, and complementary options for 2D rendering and visualisation. This presents opportunities to develop not only different game genres and environments, but interesting combinations of 3D gaming levels and worlds with 2D visuals and user interface elements.

n.b. Whilst the above sections are logical groupings for concepts and technologies, their structure within the course itself is more dynamic. In particular, many aspects of these technologies are introduced in the context of ongoing demo application development. Therefore, these concepts and technologies will be covered throughout the semester.

Supporting Material

The following includes samples of ongoing materials, which form a core part of each semester’s course. Each student has access to these materials on the course’s website and GitHub account, and these resources are updated on a weekly basis.
Sample Course Notes

Example weekly notes can be found on the course's GitHub repository, which is available at the following URL,

- https://github.com/csteach488

and the course's website,

- course notes - http://csteach488.github.io/notes/

Sample Bibliography

A sample bibliography can be found on the course website, which is available at this following URL,

- course bibliography - http://csteach488.github.io/bibliography/

Sample Links & Resources

Sample links and references used within the course can be found at the following URL,

- links & resources - http://csteach488.github.io/links/