

COMP 4482A/9511A, Fall 2019

Game Engine Development

Syllabus

Course Information

Overview

This course will provide a high-level look at the design, implementation, and usage of video game engines. The primary goal of this course is to offer a basic understanding of the systems required to create a usable and reusable foundation for game development. Extra time will be spent examining and working in existing game engines in order to demonstrate the effectiveness of good engine design (or perhaps bad design).

Some focus will be given to the graphical nature of game engines and will therefore include an introduction to shaders (expanding on CS 3388), animation techniques, and optimization algorithms. Depth will be given in selected areas, but the student will be expected to do a reasonable amount of independent reading and learning outside of the lecture hours.

Calendar Description

Core concepts and techniques of real-time rendering and physical simulation as applied to the development of interactive game and simulation software. Topics from: real-time programming, indoor and outdoor rendering algorithms, character animation, vertex and pixel shaders, shading models, real-time shadows, procedural methods, simulation of classical mechanics, numerical integration, threaded programming.

Prerequisite Requirements

- CS 3307 (Basic Software Engineering), or equivalent.
- CS 3305 (Operating Systems), or equivalent.
- CS 3340 (Algorithms I), or equivalent.
- CS 3388 (Computer Graphics), or equivalent.

Useful Background Knowledge

- Working knowledge of multivariate calculus and linear algebra.
- Ability to code well in the C-like languages (C#, C++, Java).
- Experience coding with the OpenGL graphics API.

Unless you have either the requisites for this course or written special permission from your Dean to enroll in it, you may be removed from this course and it will be deleted from your record. This decision may not be appealed. You will receive no adjustment to your fees in the event that you are dropped from a course for failing to have the necessary prerequisites.

Teaching Staff + Office Hours

Alex Brandt

abrandt5@uwo.ca

MC 365

Mondays, 15:30 - 17:00, or by appointment.

Teaching Assistants

TBA

Office Hours to be posted on OWL.

Email Contact

Students should contact the instructor and teaching assistants using their university provided email accounts only. Include "CS4482" in the subject line or else your email may be missed by the teaching staff.

Course Materials and Website

There is no required textbook for this course. Course notes, supplementary materials, and all other course content will be posted on OWL. However, a suggested reading material is *Game Engine Architecture*, Second or Third Edition, by Jason Gregory.

Class Schedule

There are 3 lectures hours per week.

Mondays, 13:30 - 15:30, TC-205

Thursdays, 13:30 - 14:30, NCB-117

Course Topics

This is a suggested a list of topics and we will likely cover a subset based on the interests of the class and time constraints. Topics will not necessarily be presented in the order listed here.

- History of Game Development
- Programming for the *Nintendo DS* console.
- Game Engine Development - Common Systems & Pitfalls
- Unity 3D - A Game Engine
- Real-time Rendering
 - The scene-graph model
 - Indoor real-time rendering: BSP, portal rendering
 - Outdoor real-time rendering: ROAM, Geomipmapping, Geoclipmaps, GPU raycasting
 - Character Animation: Explicit and Implicit

- Shading: Lighting models, NPR, Shadows, Full-screen effects, HDR, Spherical Harmonic Lighting
- Physically based rendering
- Physics
 - Basic physical concepts.
 - Basic properties of bodies: mass, centre of mass, moment of inertia.
 - Newton's laws.
 - Kinematics for particles and rigid bodies.
 - Kinetics for particles and rigid bodies.
 - Collision and conservation of momentum.
 - Pulling it all together in an engine.
 - Current Physics Engines

As there simply is not enough class time to teach the above topics in detail, the course will proceed in a “breadth-first” manner. For many of the topics, we will provide references to material you already know from your basic CS education, along with instruction on how to apply that knowledge to the domain of video games. The best we can do is provide a starting point and a helpful push; to get the most out of these topics, you will have to do much study on your own time.

Student Evaluation

The bulk of the student's mark will come from applying what is learned in class in a progressive series of assignments. There are three primary ‘streams’ of assignments: *Game Programmer*, *Engine Developer* and *Tools Programmer*. The student can pick and chose which assignments they wish to do, according to their interests and career aspirations. The only constraint is that the student must have completed any specified pre-requisite assignments prior to submitting an assignment (assignments may *not* be completed concurrently). A graphical representation of the pre-requisite structure is given in figure 1. You may notice that it looks like a skill tree from a video game; that's no accident.

Each assignment has an “experience point value” (XP for short) associated with it. Completing the assignment results in the student being awarded the associated amount of XP. *Note that there are no specific grades given for assignments, they are strictly pass/fail.* If you meet the specified requirements for the assignment, you pass and are awarded the associated XP. If you don't meet the requirements, you will be informed which requirements were not met and receive no XP. You can keep resubmitting the same assignment until you meet the requirements, there is no submission limit. However, **no re-submissions will be accepted after Friday, November 29th, at 23:55.** Occasionally a particularly awesome assignment may be awarded “bonus XP”.

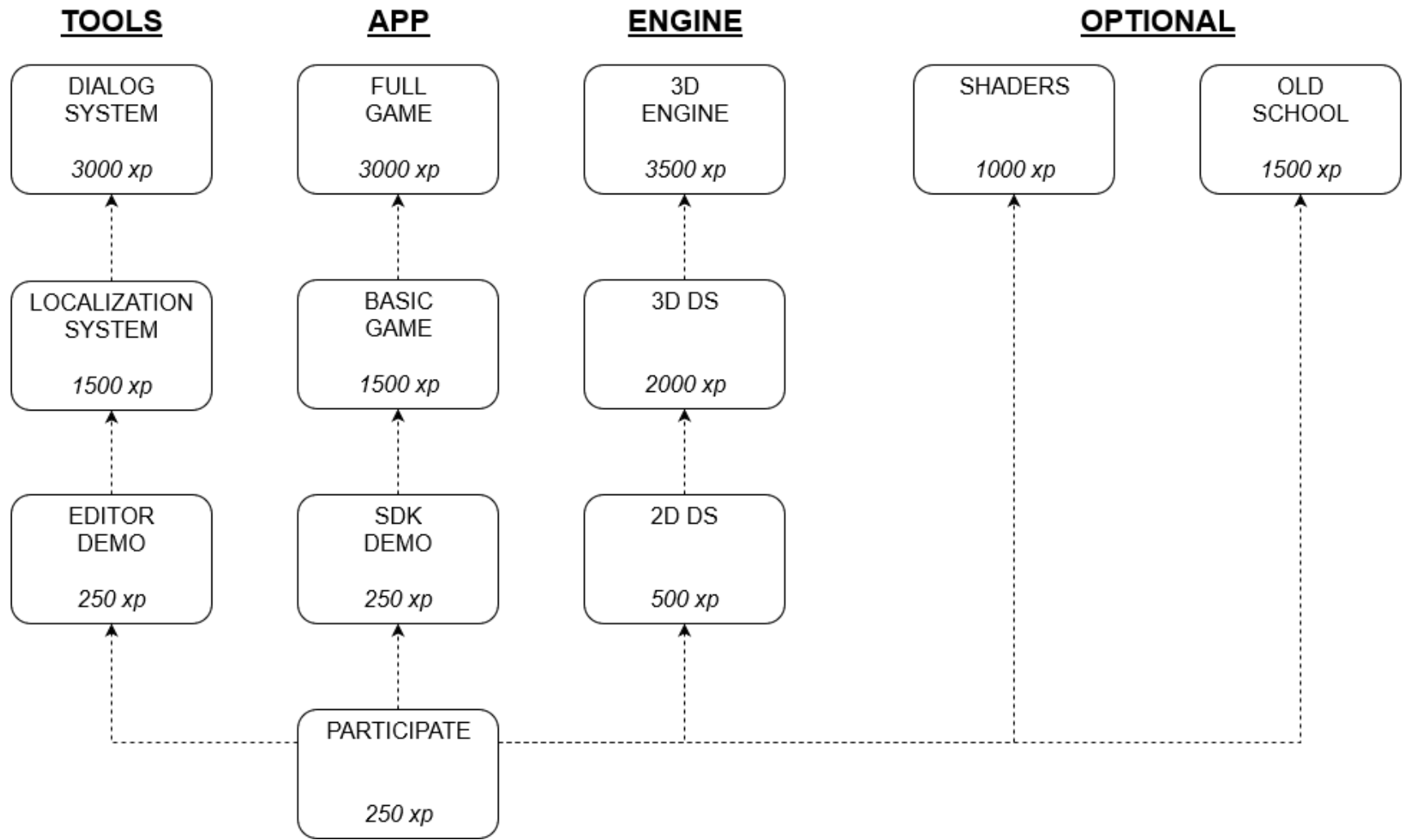


Figure 1: CS 4482 Skill Tree.

Computing your final grade

Every good RPG has an "XP curve" and this course is no different. We use here a very simple XP curve (to make computing your grade simple):

$$\text{Current Grade} = \sqrt{XP}.$$

That's it. That's your mark in the course. You add up the XP you've earned so far and take the square root. There is no final exam and the assignments are all pass/fail. This means there is *no* nondeterminism in your grade. You can decide, right now, which assignments you're going to do and, if you put in the work, you know *exactly* what your final grade will be.

WARNING: READ THIS

Note that because your grade is computed as the *square root* of accumulated XP, the accumulation of 100XP at the beginning of the course will increase your final grade much more than 100XP at the end. As your total XP grows, you need ever larger amounts of XP to increase your grade by a fixed amount. All of this should be completely familiar to anyone who has played an RPG in the past 20 years.

The important consequence is this: Make sure you plan to do one of the high-XP assignments at the end of a particular skill tree and proceed accordingly. The goal of the skill tree system is to provide flexibility for students with different interests. Under no circumstance should you plan to complete every assignment in the skill tree... unless you are independently wealthy, don't work and have absolutely no other classes.

This non-traditional grading scheme is frightening to me. What can I do?

Ignore everything above and come see me. We can decide together which assignments are best for your interests and which will give you an overall great mark in the course.

Assignment Schedule

There is a great deal of flexibility in assignment choice in this course and with that flexibility comes great responsibility. Much like the real world, you must manage your time accordingly for the different tasks involved in completing your projects.

Due to the mix-and-match aspect of this evaluation approach there are no firm due dates... However, it is expected that you submit some assignments throughout the term to show progression and engagement with the assignments. In particular, you must meet the following deadlines:

- Submit at least one “Tier 1” assignment (Editor Demo, SDK Demo, or 2D DS) by September 27, 2019, at 23:55. Additional Tier 1 assignments may be submitted after this date without late penalty.
- Submit at least one “Tier 2” assignment (Localization System, Basic Game, 3D DS) by October 18, 2019 at 23:55. Additional Tier 2 assignments may be submitted after this date without late penalty.
- **All assignment submissions and re-submissions submitted by Friday, November 29th at 23:55.** No assignments may be submitted after this date. If your last submission before this time is returned as a “fail” then this is its final status.

Late Assignments, Medical and Non-Medical Accommodations

Based on the above schedule, late submissions will be penalized 20% of their potential XP for each day late. That is, 0-23 hours late: -20%, 24-47 hours late: -40%, 48-71 hours late: -60%, etc. Given the flexible nature of this course and evaluation scheme, accommodations for missing these deadlines must be requested sufficiently early. For example, it is *not* feasible to complain about an illness on September 27, 11:52pm, and asking for an extension. Contact the instructor **48 hours** before any deadline, except in truly exceptional circumstances, to discuss your options for extensions or otherwise.

Due to the weight of Tier 2 and Tier 3 assignments, students seeking accommodations are required to request such accommodations from their home faculty’s academic counselling office. These requests may fall into one of three categories: 1. self-reported absence, 2. medical absence, 3. non-medical absence. More information on this policy can be found in Western University’s Policy on Academic Consideration for Student Absences and in Western University’s policy on Accommodation for Religious Holidays.

Policies, Accommodation, Accessibility

Mental Health

Mental and emotional well-being is highly important and should not be treated lightly. Students who are in emotional/mental distress should refer to Mental Health@Western at <http://www.uwo.ca/uwocom/mentalhealth/> for a complete list of options about how to obtain help.

Accessibility

Please contact the course instructor if you require material in an alternate format or if you require any other arrangements to make this course more accessible to you. You may also wish to contact Student Accessibility Services (SAS) at 661-2147 if you have any questions regarding accommodations. More information can be found in Western University's Policy on Academic Accommodation for Students with Disabilities

Ethical Conduct

Scholastic offences are taken seriously and students are directed to read the appropriate policy, specifically, the definition of what constitutes a Scholastic Offence, at this web site.

All required papers may be subject to submission for textual similarity review to the commercial plagiarism detection software under license to the University for the detection of plagiarism. All papers submitted for such checking will be included as source documents in the reference database for the purpose of detecting plagiarism of papers subsequently submitted to the system. Use of the service is subject to the licensing agreement, currently between The University of Western Ontario and Turnitin.com

Computer-marked multiple-choice tests and/or exams may be subject to submission for similarity review by software that will check for unusual coincidences in answer patterns that may indicate cheating.

Student Support

The Student Development Centre provides learning skills services for students. Other services are also provided by the University Students' Council

Registration Services

Students should refer to the Registrar's website or information and services involving registration.