The University of Western Ontario

London, Ontario, Canada

Department of Computer Science

Computer Science 4482a – Game Engine Development: Real-time Rendering and Physical Simulation

Course Outline – Fall Term 2017

1 Course Description

This course will provide a high-level survey of the core techniques and algorithms required for video game, and other real-time graphical interactive, software. The course will consider applications in the context of video game development but the course is primarily concerned with concepts, algorithms and techniques that are applicable in any real-time rendering environment (simulation, computer-aided design, computer-assisted surgery, etc.) .

Some focus will be on game engine development as a whole, including some examples from real-world implementations.

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.

2 Prerequisites

- CS 3388 (Computer Graphics), or equivalent.
- CS 3307 (Basic Software Engineering), or equivalent.
- CS 3305 (Operating Systems), or equivalent.
- CS 3340 (Algorithms I), or equivalent.
- Working knowledge of multivariate calculus and linear algebra.
- Ability to code well in the C and C++ languages.
- Experience coding with the OpenGL graphics API.

3 Instructor

- Name: Kailey Joanette
- Office: MC4
- Office Hours: By appointment.
- E-Mail: kjoanett@uwo.ca

4 Textbook and Lecture Notes

There is no required textbook for this course. However, some material covered can be further investigated in *Game Engine Architecture: Second Edition* by Jason Gregory. Course notes, assignments, and other materials will be made available on the course OWL group.

5 Course Website

Course content will be distributed through OWL `https://owl.uwo.ca`. Log in using your UWO credentials.

Course announcements will appear on the OWL group and students are responsible for checking this regularly. Contributing towards classroom discussion groups is encouraged.

6 Lecture 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 ray-casting
  - 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.

    - Numerical integration
    - 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.

7 Lecture Dates

- September 8th, 2017
- September 15th, 2017
- September 22nd, 2017
- September 29th, 2017
- October 6th, 2017
- October 20th, 2017
- October 27st, 2017
- November 3rd, 2017
- November 10th, 2017
- November 17th, 2017
- November 24th, 2017
- December 1st, 2017
- December 8th, 2017

8 Course Conduct

The course will meet for regularly scheduled lectures on Fridays from 8:30am-11:30am in MC105 during the fall/winter term. Students are also encouraged to participate in the class OWL site, offering links, comments, thoughts and stories of interest.

9 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. 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. Occasionally a particularly awesome assignment may be awarded "bonus XP".

9.1 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.

9.2 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.

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

Ignore everything above and just do the assignments marked with a * on the chart. These are similar to the standard assignments used in past versions of this course and will earn you a very solid grade when completed.

10 Assignment Schedule

There is a great deal of flexibility in assignment choice in this course and with that flexibility comes 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 approach, there are no firm due dates. Instead, I expect you to submit a minimum of one assignment per month for September, October, November, and December. Note that this is a *bare minimum* requirement and should you choose to do this by moving straight up a single skill tree you will only get a barely passing mark. You should aim to submit more frequently than this.

If any students are unable to achieve this bare minimum requirement on due dates, I will reach out and arrange a meeting to discuss options and alternatives.

All assignments must be submitted by **Friday, December 2nd**.

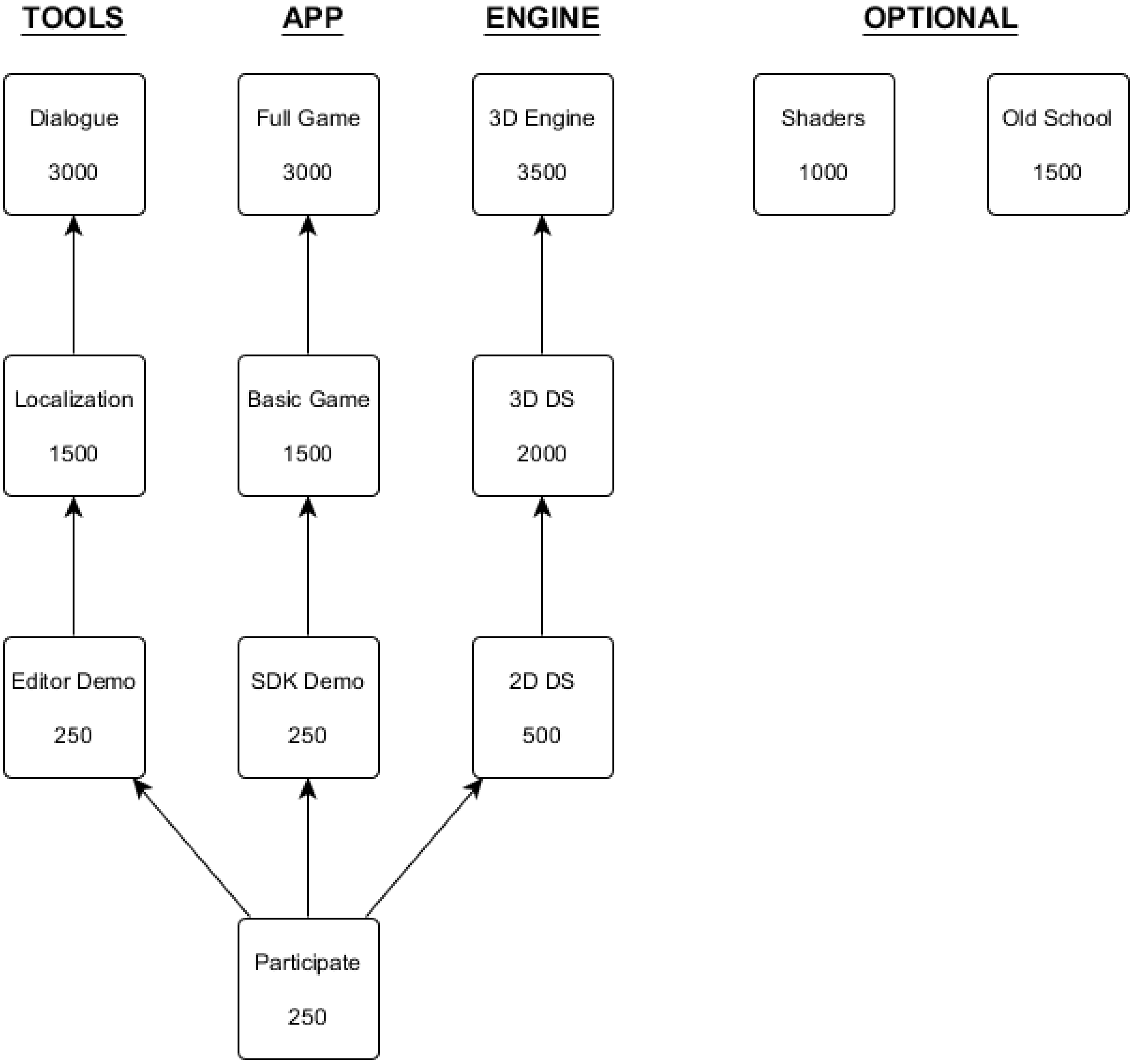

Figure 1: CS 4482 Skill Tree.

11 Ethical Conduct

All assignments are individual assignments. You may discuss approaches to problems among yourselves; however, the actual details of the work (assignment coding, answers to concept questions, etc.) must be an individual effort. Assignments that are judged to be the result of academic dishonesty will, for the student's first offence, be given a mark of zero with an additional penalty equal to the weight of the assignment also being applied. You are responsible for reading and respecting the Computer Science Department's policy on

Scholastic Offences:

`http://www.csd.uwo.ca/current_students/undergraduate_students/scholastic_offences.html`

Rules of Ethical Conduct:

`http://www.csd.uwo.ca/current_students/undergraduate_students/rules_of_ethical_conduct.html`

12 Tutoring

The role of tutoring is to help students understand course material. Tutors should not write assignments or take-home tests for the students who hire them. Having employed the same tutor as another student is not a legitimate defense against an accusation of collusion, should two students hand in assignments judged similar beyond the possibility of coincidence.

13 Academic Accommodation for Medical Illness

For work representing 10% or more of the overall grade in the course: If you are unable to meet a course requirement due to illness or other serious circumstances, you must provide valid medical or other supporting documentation to your Dean's office as soon as possible and contact your instructor immediately. It is the student's responsibility to make alternative arrangements with their instructor once the accommodation has been approved and the instructor has been informed. In the event of a missed final exam, a "Recommendation of Special Examination" form must be obtained from the Dean's Office immediately. For further information please see:

`http://www.uwo.ca/univsec/handbook/appeals/medical.pdf`

A student requiring academic accommodation due to illness should use the Student Medical Certificate when visiting an off-campus medical facility or request a Record's Release Form (located in the Dean's Office) for visits to Student Health Services. The form can be found here:

`https://studentservices.uwo.ca/secure/medical\_document.pdf`.

14 Support for Mental Health

Students who are in emotional/mental distress should refer to Mental Health@Western `http://www.uwo.ca/uwocom/mentalhealth/` for a complete list of options about how to obtain help.

15 Accessibility Statement

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 Services for Students with Disabilities (SSD) at 661-2111 x 82147 for any specific question regarding an accommodation.

16 Addendum

The UWO Senate Academic Handbook has specified that the following points should be added to all course outlines:

Plagiarism: Students must write their essays and assignments in their own words. Whenever students take an idea, or a passage from another author, they must acknowledge their debt both by using quotation marks where appropriate and by proper referencing such as footnotes or citations. Plagiarism is a major academic offence (see Scholastic Offence Policy in the Western Academic Calendar).

Plagiarism Checking: The University of Western Ontario uses software for plagiarism checking. Students may be required to submit their written work and programs in electronic form for plagiarism checking.

Prerequisites for a course: Unless you have either the requisites for this course or written special permission from your Dean to enroll in it, you will 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.

Exam checking: Use may be made of software to check for unusual coincidences in answer patterns that may indicate cheating.