MEMORIAL UNIVERSITY OF NEWFOUNDLAND
Department of Computer Science

Computer Science 4300
Introduction to Game Programming

Instructor:	David Churchill	Phone:	864-6140
Office:	ER-6030	Email:	dave.churchill@gmail.com
Office Hours:	TBA	Website:	https://www.cs.mun.ca/~dchurchill/

Course Website: https://www.cs.mun.ca/~dchurchill/teaching.shtml
(most course activity will take place on D2L)

Course Objectives:

This is a course for students interested in learning the fundamentals of game programming and game engine architecture. Topics include an introduction to: vector math for games, rendering, animation, and artificial intelligence, collision detection, game physics, and user-interfaces. Students will be writing fully functional games using the C++ programming language and the SFML graphics library.

Course Outline: (not in order of instruction)

- Introduction to C++ / SFML
  - C++11 Syntax, Semantics, Standard Template Library (STL)
  - C++ Compiling, Linking, Makefiles
  - C++ / SFML Basics, Graphics Primitives, Drawing, Input Handling
- Low-Level 2D Game Engine Design / Implementation
  - Game Engine Layout / Architecture
  - Basic Game Engine Main Loop Structures / Tick Rate
  - Game States / State Machine Architecture
  - Asset Loading / Memory Management
  - Sprite & Animation Rendering, Basic Shaders
  - User Input Handling / Event Systems / Replay Files
  - Data-Oriented Design / Configuration Files
  - Basic Window / Menu / Drag & Drop Systems
  - World View: Camera / Viewports
- ECS Game Engine Architecture
  - Entities, Components, Systems (ECS) Overview, Architecture, Design
  - ECS Classes, Structure, and Memory Management
  - Systems for Implementing Gameplay Mechanics
  - Memory Pooling Strategies / Implementation
- Physics / Math for Games
  - Vector Math / Class Implementation
  - Kinematics (Position, Velocity, Rotation, Acceleration, Projectiles, Gravity)
  - Collision Detection / Resolution (Basic Geometric Shapes)
  - Line Segment Intersection / Ray Casting / Visibility

- Gameplay Programming
  - Artificial Intelligence: Basic NPC Behavior / Steering / Path-Finding
  - Entity Interaction / Dialogue System
  - Difficulty Settings / Game Configurations
  - Game Event Triggers
  - Item / Inventory / Weapon Systems
  - Saving / Loading Game States
- Game Programming Tools / Level Editor
- Additional Topics to be Decided in Class

Textbook: Game Programming Patterns http://gameprogrammingpatterns.com/
(optional) SFML Game Development By Example http://a.co/d/5Y3uoLL
Beginning C++ Through Game Programming http://a.co/d/2Ts1N4P

Format: 2 lectures per week on Tuesday / Thursday (80 minutes each)

Evaluation:

The evaluation structure of the course is as follows:

- Assignments 50% (≤ 2 Per Group)
  - Intro to C++ / SFML (Programming)
  - First Game with ECS (Programming)
  - Game: Mega Mario (Programming)
  - Game: Not Zelda (Programming)
- Final Game Project (Programming) 50% (≤ 4 Per Group)

Note: Due the unique marking structure of this course, you must pass the final project to pass the course. If your grade on the final project is less than 50%, then your overall course grade will be equal to the mark that you received on the final project. If your final project grade is greater than or equal to 50%, your course grade is determined by the scheme above. This is to ensure that students can't skip working on the project and still pass the course.

COVID Notice:

The Midterm and Final exams for this course will be held in-person on campus. If any COVID-related issues arise during the term which requires MUN to close campus to students, the exams in this course will be moved online. Since lectures are being delivered remotely, they will not be affected by any COVID related issues that may arise.

Academic Misconduct:

I take academic misconduct very seriously, especially for remotely delivered courses. Anyone found cheating in this course will received the harshest possible academic penalties. Academic misconduct for this course includes (but is not limited to) the following:

- Handing in any material for evaluation that was done outside you /your group
- Obtaining solutions from ANY non-class source, such anyone outside of your group, previous course offerings, stack overflow, etc (unless specifically stated otherwise)
- Sharing assignment or exam questions outside of the course for any reason, including assignment sharing websites or online repos such as GitHub
- Reverse engineering any obfuscated solution code that may be given to you

Memorial University Policies:

Memorial University of Newfoundland is committed to supporting inclusive education based on the principles of equity, accessibility and collaboration. Accommodations are provided within the scope of the University Policies for the Accommodations for Students with Disabilities (www.mun.ca/policy/site/policy.php?id=239). Students who may need an academic accommodation are asked to initiate the request with the Glenn Roy Blundon Centre at the earliest opportunity (www.mun.ca/blundon).

Students are expected to adhere to those principles which constitute proper academic conduct. A student has the responsibility to know which actions, as described under Academic Offences in the University Regulations, could be construed as dishonest or improper. Students found guilty of an academic offence may be subject to a number of penalties commensurate with the offence including reprimand, reduction of grade, probation, suspension or expulsion from the University. For more information regarding this policy, students should refer to the University Regulations for Academic Misconduct (Section 6.12) in the University Calendar.