CPSC 427 Video Game Programming

UBC

Instructor: Helge Rhodin

Previous readings: Alla Sheffer





ECO

Defence

BUILDING: FROTOR





Helge Rhodin

https://www.cs.ubc.ca/~rhodin/

- BSc and MSc at Saarland Univ., Germany
- PhD at the Max Planck Institute for Informatics
- Lecturer and postdoc at EPFL, Switzerland
- At UBC since Sep'19



Computer Graphics







Course Staff

Instructor:

- Helge Rhodin
 - Office hours: Wed, 10-11PM, same zoom room as lecture
 - Email: rhodin@cs.ubc.ca (use Piazza for all but personal topics)

TAs:

- Tim Straubinger, Andrew Evans, Grace Chang, and Dave Pagurek
 - Contact via Piazza
 - Office hours Mon 10-12 am, Thu 4-5 pm (tentatively, please vote on Piazza poll)



What is This Course About?

- Basic Elements of Game Programming
- Content
 - Graphics: Modeling, Rendering, Animation
 - Gameplay: Situational response, User experience
 - . . .
- Implementation
 - Game software design
 - Writing and debugging efficient & robust (runtime/memory) code
- Project management/Teamwork
 - Support software/Best practices



What is This Course About?

! Writing your own game start to finish !

- Learning through experience
 - Programming
 - Teamwork
 - User experience (UX)



Topics NOT Covered:

Interesting but no time:

- Game design
 - Storytelling
 - Game style/look
- Deep dive into graphics, AI, UI, game engines ...
- Asset creation tools



Prerequisites

CS:

• CPSC 221

MATH:

- one of MATH 200, MATH 217, MATH 226, MATH 253
- one of MATH 152, MATH 221, MATH 223

Strong math & programming background is encouraged

No prior graphics knowledge assumed



Web Resources

- Course Page: https://www.cs.ubc.ca/~rhodin/2020_2021_CPSC_427/
 - **Read & know** all the course info + policies
- Piazza discussion forum (link from Canvas)
 - Please use for all technical questions; **no** private issues
 - Use private mode for questions to course staff that require posting code
- Slack for group formation etc.; **no** technical questions
- Canvas: assignment submission and grade reporting



Course Project: Video Game

- 2D Game
- Basic template provided (very basic)
- Mandatory spec requirements (details in milestone documentation)
 - Shaders, 2D transformations, basic animation & gameplay, user experience validation (testing), efficient time/memory management
- Combined with advanced features selected from multiple options (details in milestone documentation)
- Written in teams of 6 (in exceptional cases 5 or 7, e.g., those admitted from waitlist)
- Bi-weekly milestones, dates specified on calendar Completed games demoed to peers/expert jury

UBC

Theme (new)

Social games, played remotely via zoom

- Cooperative,
- Turn-based,
- Puzzles, or
- Real-time, but...

Technical implications

• Handle delay and low frame rates

Why?

- Lets have fun together
- No other way of doing cross-play



https://slate.com/human-interest/2020/03/board-games-video-chat-codenames.html



(Virtual) Cross Play Sessions

- Test / play the games of other teams
- Via zoom or other screen sharing apps with remote control
 - With zoom, the remote user only receives control of the shared window, not your computer!
- Your game must be designed to be playable remotely
- Let us know if your machine / network connection does not permit screen sharing
 - Test this remote play setup on your machine ASAP (with a game of your choice)



Course Hours:

- Lecture: Tue, 5 6:30 pm; 5 6:30 pm, zoom (see Canvas)
 - Additional lecture slot (Wed 4-6 pm first three weeks)
 - Some asynchronous content
- Tutorial: Wed 4-6 (first three weeks replaced by lecture), zoom TBD



Format:

- Lectures:
 - Regular lectures by instructor
 - Guest lectures by industry speakers
 - Team progress report meetings (one per milestone)
 - Cross-play sessions (starting from milestone 2)
- Tutorials:
 - Team meetings with TAs
 - Face-to-Face milestone marking (Overflow during Office Hours)
 - All team members must be present for marking



Progress report meetings

- One per milestone (Thursdays, the week after submission)
- Reports from each team (2min) on
 - Progress, achievements & challenges
- Quick feedback round

Cross-play sessions for milestones 2, 3, 4

- One per milestone (week after each milestone)
- Collect playability feedback
- Feedback impacts bonus component of grade



Tutorials

- Each team expected to meet with a TA once a week
 - Schedule TBD
 - Optional during marking weeks
- Face-to-face marking
 - Schedule TBD
 - During tutorials/office-hours
- Mapping of teams to tutorials TBD

Contact TAs for any changes, 3 days in advance



Grading System: Team Project (78%)

- Game Pitch (1%) and Game Proposal (1%)
- Milestones: M1 19%, M2 19%, M3 19%, M4 19%
 - Marked in face-to-face sessions with TAs
 - Includes both demo and Q&A
 - Includes cross-play feedback for M2-M4
 - Up to 10% bonus based on feedback
- Final exam replaced by juried cross-play session for M4
 - Mandatory attendance
 - Demo to peers/industry jury (feedback used for grading)
 - Extra bonus marks provided for award winning projects
 - based on jury/peer feedback



Grading: Team Project to Individual Grade

We expect all team members to participate in coding for ALL milestones

Individual Project Grade

- Grade computed by multiplying team grade by contribution quotient Q
 - Average contribution: Q=1
 - Below average Q < 1
 - Above average Q > 1
- Quotient determined based on self reporting, TA interaction, code repo analysis, and peer feedback



Grading System: 3 Individual Assignments

1%: Assignment 0: online now

- Entity Component System (ECS) and C++
- small but important, due next Monday (Jan 18), or five days after admission

5%: Assignment 1: online now

- Basic rendering/event driven programming
 - Good for self-assessment before drop deadline

5%: Assignment 2:

• Collision processing + Game AI

5%: Assignment 3:

• 2D animation



Grading System

1%: Game Pitches

- Written pitches due Sep 13
- Individual or mini-team (up to 6 members)
- 100% Bonus for fully formed teams (exactly 6 members)



Grading System

4%: Individual Progress Reports

- Each student *must* submit a progress report for **EACH** milestone
 - Summary of work completed
 - achievements & challenges
 - Feedback on team-member performance



Grading System

2%: Classroom Participation

- Q & A
- Zoom chat
- Polls / Clickers



TODOs: Individual

- Assignment 0 (individual)
- Read through course pages
- Register to Slack and Piazza
 - Vote on office hours
- Game idea
 - Game pitch
- !!!! Team organizing !!!!
 - Self-registration on Canvas > People > Groups



TODO: TEAM ORGANIZING

- Team organizing (use piazza or slack to connect), seek common game ideas, diversity of experience, similar working hours
 - Initial teams: Jan 18
 - Finalize by Jan 22
 - We can help...
- Game Pitch (storyline + basic technical elements) individual/mini-team
 - Informal pitches on Slack, project-pitch channel: ASAP
 - Oral pitches: Tue Jan 19
 - Plan on ~1-2 minutes: game idea+team
 - Register on Canvas > People > Groups
 - Written pitches: due Jan 22



Decorum: respect your classmates

Please come on time

- We will start timely
- Hint: we will have questions near start/end that count to class participation
 - *if you have to attend lectures asynchronously, we will offer other means of participation. Let us know!*

Respect others and their mentalities in groupwork

- Allow equal talking time
- Utilize strengths, compensate weaknesses, and plan ahead

Please no open screens

- Very disruptive for folks sitting around you
- stay focussed for your own benefit!



Your expectation?

4 min get-together in breakout rooms

- Say Hi
- Discuss any questions you may have about the course logistics
- Why do you take this course?
- What game do you want to build?
- Designate someone to take notes and report to class



Grounding your expectation

A course like any other with theory, concepts, assignments, deadlines...



not a piece of cake

90 % uphill

10 % downhill

Decide in the first week! The course is in high demand. No late drop forms



Special requirements?

Let us know about ways we can support you in a private message or office hour.



Syllabus (I)

Graphics: Rendering

- Basic Rendering: Rendering pipeline elements
- OpenGL/Event Driven Programming/Keyboard & Mouse input

Graphics: Geometry

- 2D Transformations
- Curves (in time & space)
- Meshes/Polygons

Graphics: Collision detection



Syllabus (II)

Game UI/UX

- Basics of User Interface Design
- Game interfaces/Game experience
- Testing



Syllabus (III)

Game Software Design

• Entity Component Systems



Syllabus (IV)

Gameplay Logic/Al

- State Representation
- Decision Trees
- Pathfinding (goal optimization)
- Heuristic Pathfinding/A*/MinMax



Syllabus (V)

Basic Physics

- Time stepping
- Euler integration
- Velocity & acceleration
- Particles & springs



Syllabus (VI)

Efficiency/Tools

- Debugging strategies and tools
- Profiling
- (In)efficient coding 101
- Compiler optimization
- Memory allocation
- Version control