

CPSC 427

Video Game Programming



Instructor:
Alla Sheffer



© Alla Sheffer

Instructor: Alla Sheffer (me)



Professor, Computer Science UBC

Research Area: Computer Graphics/Visual Computing

<https://www.cs.ubc.ca/~sheffa/>

- Fellow of the Royal Society of Canada
- Fellow of IEEE
- Fellow of ACM
- Member of SIGGRAPH Academy
- Other Major Awards: Canadian Human Computer Communication Society Achievement Award'18, UBC Killam Research Prize'20



© Alla Sheffer & Helge Rhodin



Course Staff

Instructor:

- Alla Sheffer
 - Office hours: Monday, 1-2PM, ZOOM/X651 (ICICS/CS)
 - Email: sheffa@cs.ubc.ca (use Piazza for all but personal topics)

TAs:

- Camilo Talero, Niloofar Khoshsiyar, Povel Batth, Jinfan Yang
- Contact via Piazza
 - Office hours Mon 5-6PM, Wed 5-6PM, Fri 4-5PM, zoom/X139 (ICICS/CS)

© Alla Sheffer & Helge Rhodin



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

© Alla Sheffer & Helge Rhodin



What is This Course About?

!!!! Writing your own game start to finish !!!!

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

© Alla Sheffer & Helge Rhodin



Topics NOT Covered:

Interesting but no time:

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

© Alla Sheffer & Helge Rhodin



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

© Alla Sheffer & Helge Rhodin



Web Resources

- Course Page: http://www.cs.ubc.ca/~sheffa/games_course/
 - ***Read & know all the course info + policies***
- Piazza discussion forum (link from course page)
 - *Please use for everything **except** private issues*
 - *Use **private mode** for questions to course staff that require **posting code***
 - ***Use private mode or email for personal issues***
- handin & handback: assignment submission and grading (see <https://my.cs.ubc.ca/docs/handin-instructions>)
- Canvas: grade reporting

© Alla Sheffer & Helge Rhodin



Course Project: Video Game

Project

- **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 (+/-)
- 4 core milestones
- Completed games demoed to peers/expert jury

© Alla Sheffer & Helge Rhodin



Cross Play Sessions

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

© Alla Sheffer & Helge Rhodin

Game Type (account for remote play)

Tempo set by user input (not dependent on reaction speed)

- Allow for remote play via zoom
 - Cooperative,
 - Turn-based,
 - Puzzles, or
 - Real-time, but...

Technical implications

- Handle delay and low frame rates



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

© Alla Sheffer & Helge Rhodin

Course Format

Course Hours:

- Lecture: Mon, 3PM-5PM; Wed 3PM-4PM, DMP 110
 - ***Additional lecture hour*** (Wed 4-5 first three weeks)
- Tutorial: Wed 4-5 (first three weeks replaced by lecture), DMP 101

© Alla Sheffer & Helge Rhodin



Course Format

Format:

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

© Alla Sheffer & Helge Rhodin



Course Format – Interactive Sessions

Progress report meetings

- One per milestone (Wednesdays, starting Jan 26)
- Reports from each team (2min) on
 - ▶ Progress, achievements & challenges
- Quick feedback round

Cross-play sessions for milestones 2, 3, 4

- One per milestone (Wednesdays, M2 Mar 2, M3 Mar 23, **TBD for M4**)
- Collect playability feedback
- Feedback impacts bonus component of grade

© Alla Sheffer & Helge Rhodin



Course Format

Tutorials/Office Hours

- *Each team expected to meet with a TA once a week*
 - ▶ Schedule online - contact TAs for any changes
 - ▶ Optional during marking weeks
- *Face-to-face marking*
 - ▶ Schedule online - contact TAs for any changes
 - ▶ During tutorials/office-hours
- *Team members should attend **same** tutorial*
- *Teams 1-8 in DMP 110, Teams 9-16 in DMP 301*

© Alla Sheffer & Helge Rhodin



Grading System: Team Project (78%)

- Proposal (2%)
- 4 Game Progress Milestones (4x19% = 76%)
 - *Marked in face-to-face sessions with TAs*
 - ▶ Includes both demo and Q&A
 - *Includes cross-play feedback for M2-M4*
- **Final exam replaced by juried cross-play session for M4**
 - **Mandatory attendance**
 - *Demo to peers/industry jury (feedback used for grading)*
 - *Extra bonus marks (up to 5%) provided for award winning projects*
 - ▶ based on jury/peer feedback

© Alla Sheffer & Helge Rhodin

Grading System: Team Grade to Individual Grade



All team members *MUST* 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 Q determined based on self reporting, TA interaction, code repo analysis, and peer feedback
 - Students must submit **written reports** documenting theirs & their **team mates** contribution towards each milestone

© Alla Sheffer & Helge Rhodin

Grading System: 4 Individual Assignments



4% Assignment 0

- Introduction to ECS patterns

5%: Assignment 1:

- Basic rendering/event driven programming

5%: Assignment 2:

- Collision processing + Game AI

5%: Assignment 3:

- 2D animation

- **Assignments 2 & 3 marked face-to-face**

© Alla Sheffer & Helge Rhodin



Grading System

2%: Classroom Participation

- Q & A (oral/chat)
- Clickers/Zoom poll questions

© Alla Sheffer & Helge Rhodin



Grading System

1%: Game Pitches

- Written pitches due **Jan 17**
- Individual or mini-team
- **50% Bonus for fully formed teams (6 people)**

© Alla Sheffer & Helge Rhodin



TODOs: Individual

- Assignment 0: (individual)
- Read through course pages
- Register to Piazza
- !!!!Team organizing!!!!
 - *Use piazza (google doc link) to find teammates and self-register your team*
- Develop game ideas (not just one)
 - *Write game pitch (**just one**)*

© Alla Sheffer & Helge Rhodin



TODO: TEAM ORGANIZING

- Team organizing (use piazza to connect), seek common game ideas, diversity of experience
 - *Initial teams: Jan 14*
 - *Finalize by **Jan 21***
 - **We can help...**
- Game Pitch (storyline + basic technical elements) – individual/mini-team
 - *Informal piazza pitches: ASAP*
 - *Oral/Written pitches: **Monday Jan 17***
 - Plan on ~1-2 minutes: game idea+team
 - Register via poll on Piazza

© Alla Sheffer & Helge Rhodin



Your expectation?

4 min get-together break

- 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

© Alla Sheffer & Helge Rhodin



Grounding your expectation

Expect to be challenged: theory, practice, 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

© Alla Sheffer & Helge Rhodin



In person decorum: respect your classmates

Please come on time & stay till end of class

- Coming/leaving disrupts everyone even if done quietly
- Hint: we will have questions near start/end

Please no open screens (unless directly related to the course)

- Very disruptive for folks sitting behind you

Respect your team members

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

© Alla Sheffer & Helge Rhodin



Syllabus (I)

Game Software Design

- Entity Component Systems

© Alla Sheffer & Helge Rhodin



Syllabus (II)

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

© Alla Sheffer & Helge Rhodin



Syllabus (III)

Game UI/UX

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

© Alla Sheffer & Helge Rhodin



Syllabus (IV)

Gameplay Logic/AI

- State Representation
- Decision Trees

- Pathfinding (goal optimization)
- Heuristic Pathfinding/A*/MinMax

© Alla Sheffer & Helge Rhodin



Syllabus (V)

Basic Physics

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

© Alla Sheffer & Helge Rhodin



Syllabus (VI)

Efficiency/Tools

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

© Alla Sheffer & Helge Rhodin