CPSC 427
Video Game Programming

Instructor:
Helge Rhodin

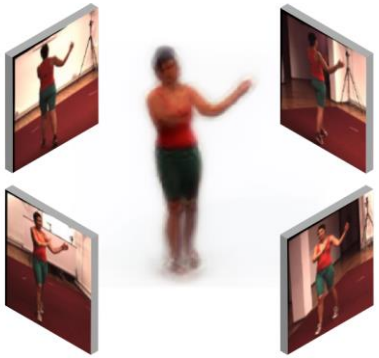
Previous readings
& WT2 section:
Alla Sheffer

Helge Rhodin

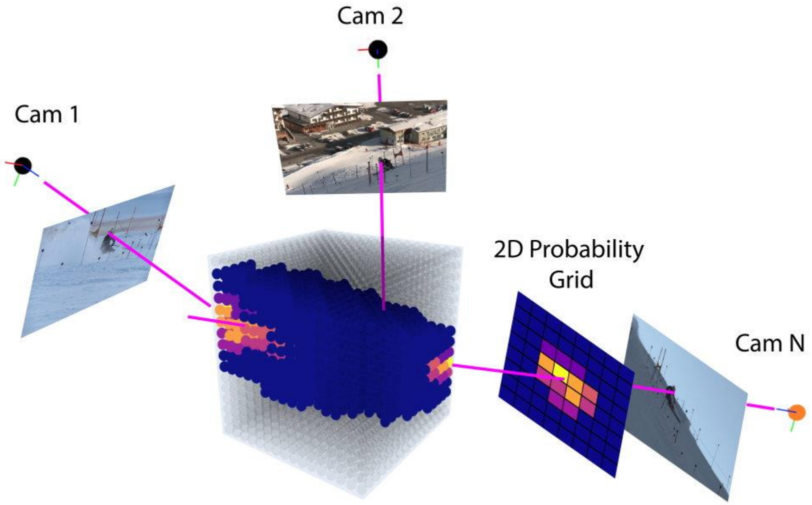
https://www.cs.ubc.ca/~rhodin/

- 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



Computer Vision



Course Staff

Instructor:

- Helge Rhodin
  - *Office hours: Wed, 9:30-10:30 AM, ICSS X653 (or zoom room)*
  - *Email: rhodin@cs.ubc.ca (use Piazza for technical topics)*

TAs:

- Tim Straubinger, Andrew Evans, and Camilo Talero
  - *Contact via Slack*
  - *Office hours TBD*
    (please vote on Piazza poll)

Protect each other!

+

+

KEEP YOUR
DISTANCE

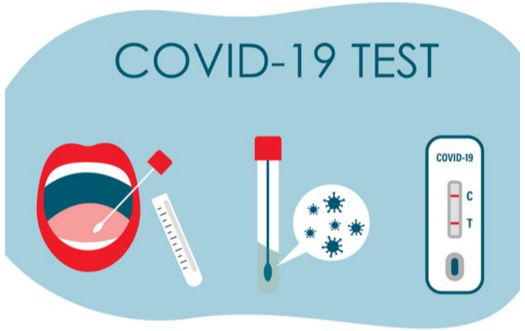
+

Small #contacts

+

Short contact duration

+



+

ventilation

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/2021_2022_CPSC_427/
  - ***Read & know*** *all the course info + policies*
- Piazza discussion forum (link from Canvas and mail)
  - *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
- handin & handback: assignment submission and grading (see https://my.cs.ubc.ca/docs/handin-instructions)
- Canvas: course-internal links and team formation

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., uneven number admitted from waitlist)
- Bi-weekly milestones, dates specified on calendar
  Completed games demoed to peers/expert jury

Theme (for the hybrid situation)

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 in these uncertain circumstances

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 Format

Course Hours:

- Lecture: Mo. 3 – 5 pm; Wed. 3 – 4 pm
  - ***Additional lecture slot*** *(Wed 4-5 pm first three weeks)*
  - *Some asynchronous content*

- Tutorial:
  - *Wed. 4-5 (first three weeks replaced by lecture)*
  - *Fr. 3-4 (unless there is a joint Wednesday event)*

Course Format – Lectures & Tutorials

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

Course Format – Interactive sessions

Progress report meetings

- *One per milestone (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*

Course Format

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 in your schedule, 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 at lecture start (now?!)

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

5%: Assignment 1: online at lecture start

- 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: Individual

4%: Individual Progress Reports

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

Grading System

2%: Classroom Participation

- *Q & A*
- *Zoom chat*
- *Polls / Clickers*

Grading System: Individual/Team

1%: Game Pitches *(listed as team credit, some slides ago)*

- Written pitches due **in the third week**
- Individual or mini-team (up to 6 members)

- **100% Bonus for fully formed teams (exactly 6 members)**

TODOs: Individual

- Assignment 0 (individual)
- Read through course pages
- Register to Slack and Piazza
  - *Vote on office hours*

- Develop game ideas (not just one)
  - *Write game pitch (just one)*
- !!!! Team organizing !!!!
  - *Use slack to find teammates*
  - *Once settled, self-register your team on Canvas -> People -> Groups*
    - Chose a group name and add your own group by hitting “+Group”

TODO: TEAM

- Team organizing (use piazza or slack to connect), seek common game ideas, diversity of experience, *similar working hours*
  - *Initial teams: end of second week*
  - *Finalize by **the third week***
  - **We can help…**
- Game Pitch (storyline + basic technical elements) – individual/mini-team
  - *Informal pitches on Slack, project-pitch channel: ASAP*
  - *Oral pitches: **Wed Sep 15***
    - Plan on ~1 minute: game idea + team
    - Register on Canvas -> People -> Groups
  - *Written pitches: due **Sep 17***

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 break / 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?

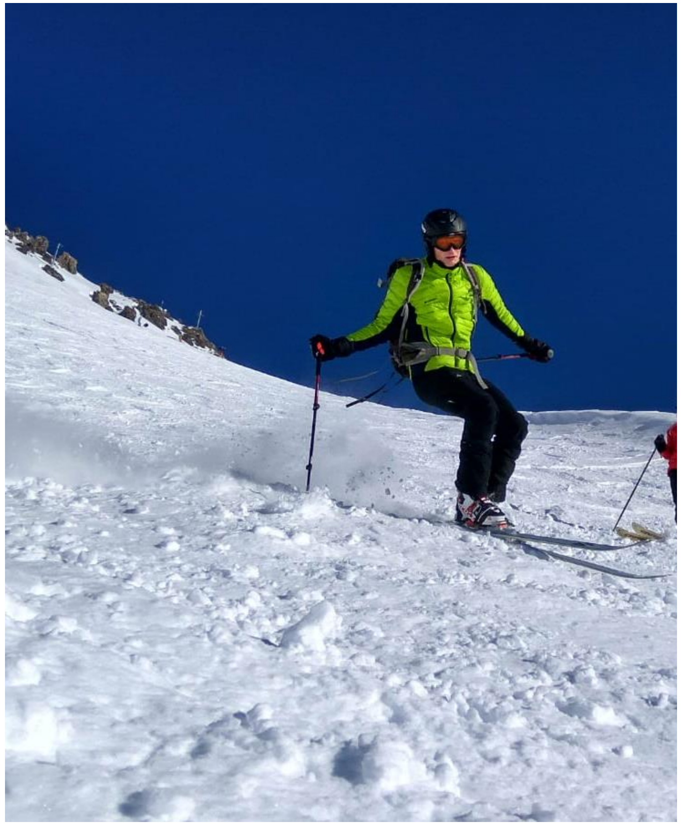
- Designate someone to take notes and report to class

Grounding your expectation

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

not a piece of cake

90 % uphill



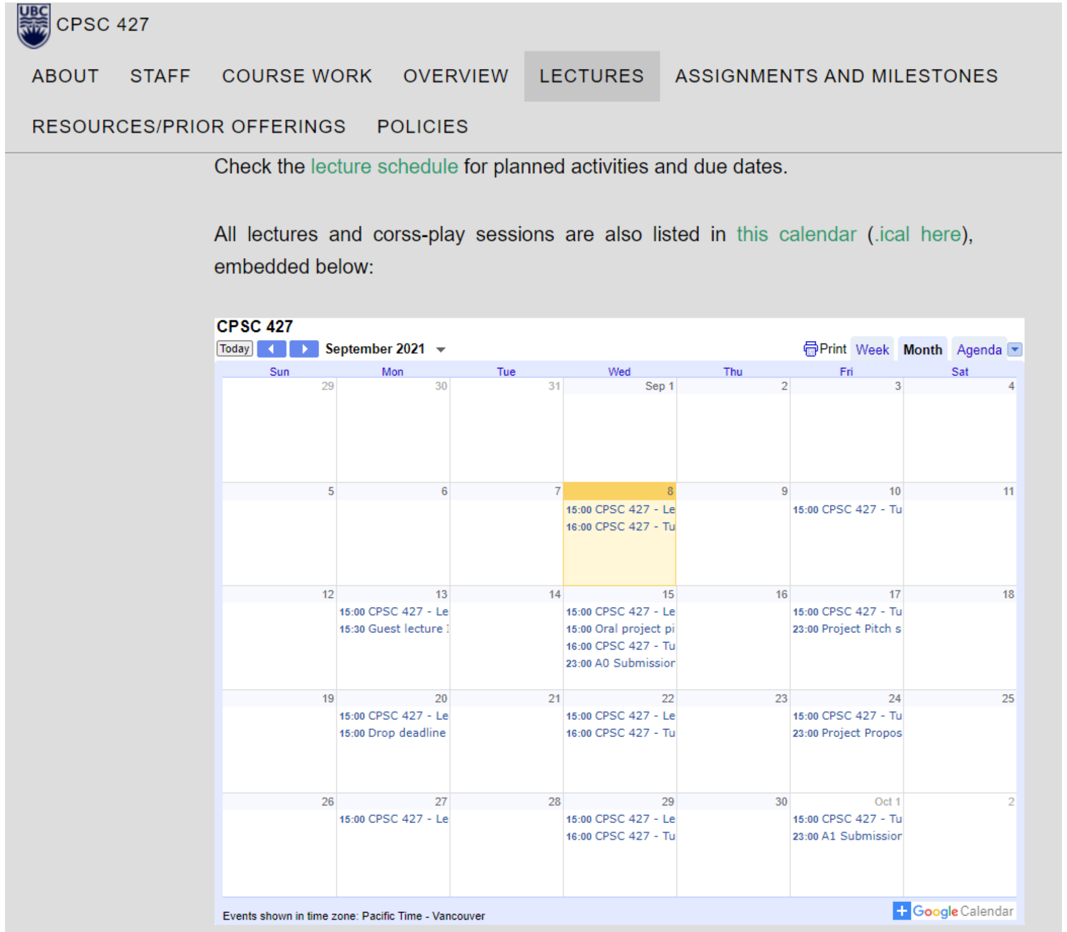
10 % downhill

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.

Schedule and deadlines



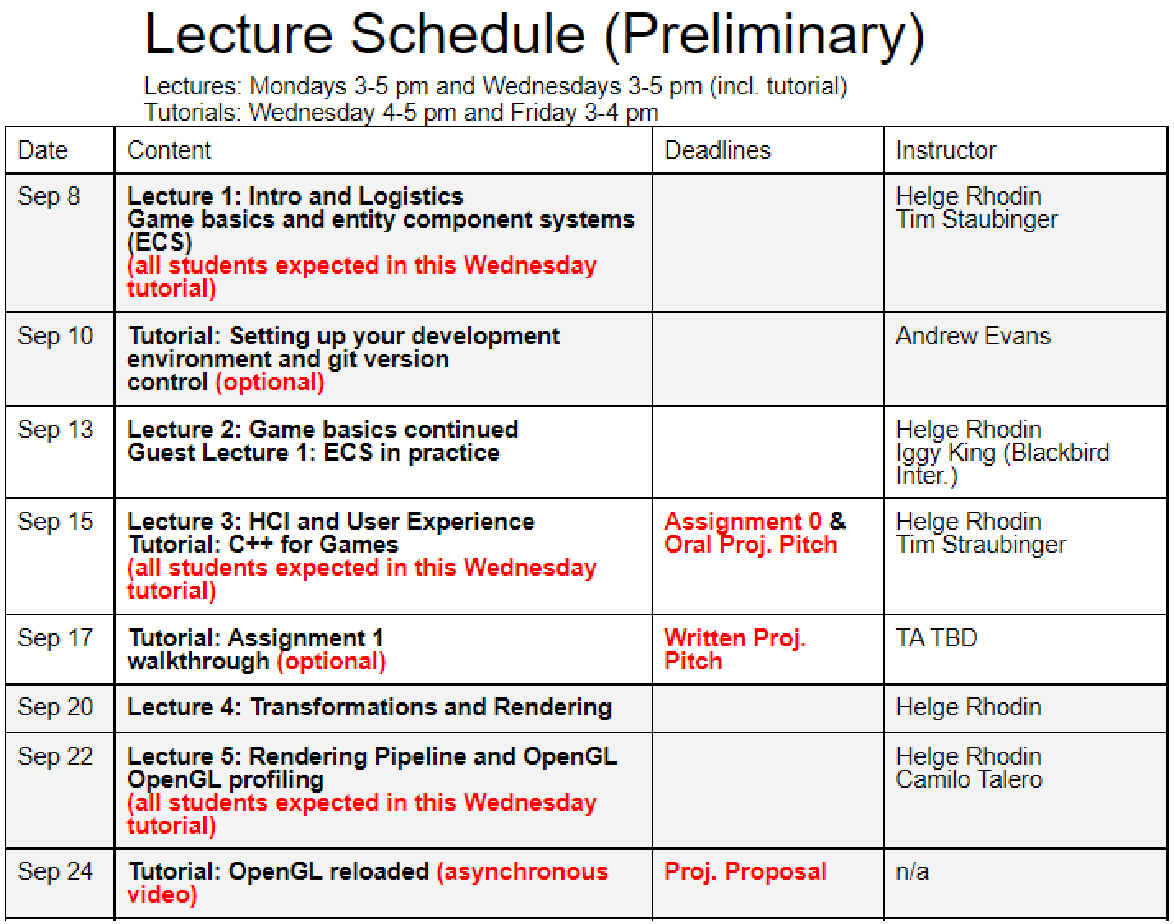
Check the lecture schedule for planned activities and due dates.

All lectures and corss-play sessions are also listed in this calendar (.ical here), embedded below:

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	31	Sep 1	2	3	4
5	6	7	8 15:00 CPSC 427 - Le 16:00 CPSC 427 - Tu	9	10 15:00 CPSC 427 - Tu	11
12	13 15:00 CPSC 427 - Le 15:30 Guest lecture	14	15 15:00 CPSC 427 - Le 15:00 Oral project pi 16:00 CPSC 427 - Tu 23:00 A0 Submissior	16	17 15:00 CPSC 427 - Tu 23:00 Project Pitch s	18
19	20 15:00 CPSC 427 - Le 15:00 Drop deadline	21	22 15:00 CPSC 427 - Le 16:00 CPSC 427 - Tu	23	24 15:00 CPSC 427 - Tu 23:00 Project Propos	25
26	27 15:00 CPSC 427 - Le	28	29 15:00 CPSC 427 - Le 16:00 CPSC 427 - Tu	30	Oct 1 15:00 CPSC 427 - Tu 23:00 A1 Submissior	2

Events shown in time zone: Pacific Time - Vancouver

Google Calendar



Lecture Schedule (Preliminary)

Lectures: Mondays 3-5 pm and Wednesdays 3-5 pm (incl. tutorial)
Tutorials: Wednesday 4-5 pm and Friday 3-4 pm

Date	Content	Deadlines	Instructor
Sep 8	**Lecture 1: Intro and Logistics** **Game basics and entity component systems (ECS)** **(all students expected in this Wednesday tutorial)**		Helge Rhodin Tim Staubinger
Sep 10	**Tutorial: Setting up your development environment and git version control (optional)**		Andrew Evans
Sep 13	**Lecture 2: Game basics continued** **Guest Lecture 1: ECS in practice**		Helge Rhodin Iggy King (Blackbird Inter.)
Sep 15	**Lecture 3: HCI and User Experience** **Tutorial: C++ for Games** **(all students expected in this Wednesday tutorial)**	**Assignment 0 & Oral Proj. Pitch**	Helge Rhodin Tim Straubinger
Sep 17	**Tutorial: Assignment 1 walkthrough (optional)**	**Written Proj. Pitch**	TA TBD
Sep 20	**Lecture 4: Transformations and Rendering**		Helge Rhodin
Sep 22	**Lecture 5: Rendering Pipeline and OpenGL** **OpenGL profiling** **(all students expected in this Wednesday tutorial)**		Helge Rhodin Camilo Talero
Sep 24	**Tutorial: OpenGL reloaded (asynchronous video)**	**Proj. Proposal**	n/a

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
- Game balancing

Syllabus (III)

Game Software Design

- Entity Component Systems
- Observer Pattern

Syllabus (IV)

Gameplay Logic/AI

- 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