COMP30540
2018-2019

Game Development

Lecturer: Dr. Arthur Cater

Teaching Assistant: Seán Behan

Admin issues

- 22-23 lectures, 2 assignments, 10 practical sessions – starting Thursday

Bring your own earphones to practical sessions

- Your attendance at lectures and at practicals will be recorded and used.
- Enrol yourself on Moodle at http://csmoodle.ucd.ie/moodle/course/view.php?id=362
- The book ***Introduction to Game Development*** by Steve Rabin has been used as the source for several figures and diagrams, and for much course content.
  - UCD Library has it
  - UCD Bookshop will obtain it if you request it – roughly one week needed

Assessment issues

- 1st assignment set Wednesday 23 Jan, due ***by*** 11am Monday 4 March, worth 25%
- 2nd assignment set Wednesday 30 Jan, due ***by*** 11am Monday 15 April, worth 25%
- 2-hour written exam in RDS – worth 50% – will occur after semester end.
  - *note there are likely to be other assignments due at similar times*
  - *note that assignments contribute 50%, a very important fraction of total marks*
  - *note that final year projects need time to finish, write up, prepare presentation*
  - assignments will need to be demonstrated in action, individually, in person
  - you should provide small short recordings of your games being played
- The demonstration of your assignments is not like presentation of final year projects: much less preparation is required, much less important (less is at stake)

More assessment issues

UCD lateness penalties apply: one week 10%, two weeks 20%, then all is lost!

– Lateness penalties are 10 or 20 marks from the 100 marks available, not 10% or 20% of the marks you earned. eg 72% drops to 62% then to 52% then to 0%.

– This makes it theoretically possible to get a negative mark for a late assignment, but that will not occur. It is always better to submit something (provided it is your own work), even incomplete and late by up to two weeks, than to submit nothing at all.

Beware that developing realistic-looking animations is extremely time consuming, not very interesting from a Computer Science point of view, and not well rewarded in marking schemes. Relatively awful animation is quite good enough for assignments.

This is a Computer Science module, therefore the computing aspects (such as conformance to specifications in matters of game play) are more important than artistic aspects (such as choice of colours or realism of figures, animation, landscape).

Admin issues 2: Plagiarism

Assignment work you submit must be your own work.

Plagiarism is cheating and is highly likely to be detected and punished.

School of Computer Science policy on plagiarism is on main page of CS Moodle.

You can and should *discuss problems* with each other, but

- you must not copy the work of others except with explicit acknowledgement,
  - so e.g. you may use asset-store artwork etc. but say clearly where you got it
- you must not let others copy your work, helping them to cheat,
- you must not submit work developed in close collaboration with others.

Resits

Any student failing the module will have a resit opportunity.

It is hard to pass without doing both assignments.
It is almost impossible to pass without doing either of the assignments.

If necessary (and let's hope it isn't)

- a resit written exam will occur in-school in first week of 2019-2020 Semester 1
- submissions/resubmissions of assignments will be examined that same week

There can be absolutely no delaying beyond that week.

Plagiarism & UCD Computer Science

- **Plagiarism is a serious academic offence**
  - [Student Code, sections 6.2 & 6.3] or [UCD Registry Plagiarism Policy] or [CS Plagiarism policy and procedures]
- Our staff and demonstrators are **proactive** in looking for possible plagiarism in all submitted work
- Suspected plagiarism is investigated by the CS Plagiarism subcommittee
  - Usually includes an interview with student(s) involved
  - 1st offence: **usually** 0 or NG in the affected components
  - 2nd offence: may be referred to the **University disciplinary committee**
- Student who enables plagiarism is equally responsible

http://www.ucd.ie/students/guide/academicregs.html

http://libguides.ucd.ie/academicintegrity

Please Please Please …

Please Participate in class!

Join in discussions

Share your knowledge and experience

Please share your ignorance too

Please show good manners.

Don't chatter, tweet, browse or social-network during lectures

Please be considerate. Bring your own headphones to practicals

See examples of previous assignment submissions

- 1st: Picnic Protector
- 1st: Penguins and Piranhas
- 1st: Ursa Piscatorius
- 1st: Magic Grenade Shield
- 1st: Tennis To Die For
- *1st: Egg Collector*

- 2nd: Punt Jouster
- 2nd: Water Pistols At Dawn
- 2nd: Fairground Fracas
- 2nd: Spacewalk Paintball
- 2nd: Skateboard Eggfight
- *2nd: Drone Destroyer*

Major topics

- Different sorts of computer games (genres)
- The computer game industry (possibly with visiting lecturer)
- Resources, especially free toolsets for developing games and/or assets
  - gamedevelopers.ie and other national websites; wikipedia and others
  - **GameMaker**, **Blender**, Unity, Unreal, CryEngine, …
- Techniques game developers use
  - 2d & 3d geometry, Asset Stores, collision detection and response, data structures for spatial organisation, AI for NPCs, Particle systems for SFX, , ,
  - Tricks for getting acceptable performance (60 fps or better)
- Techniques game engines use

Game Engines

The so-called AAA publishers tend to develop their blockbuster games from scratch, often using C++, with teams of hundreds of people with various specialised skills.

Sometimes, after a game or a few of a series, they identify useful generalisations that can be leveraged to make subsequent developments easier. And sometimes they share.

These "game engines" facilitate development in various ways, e.g.:

- easy porting to different platforms eg PC & Mac, XBox, NES, HTML5, …
- easy publishing eg on Steam
- easy access to event-driven programming for mouse clicks, key strokes, idling
- easy exploitation of sound cards, graphics cards using e.g. OpenAL, OpenGL
- easy ways of handling collisions (in 2D, in 3D)
- easy import of assets (models, algorithms, …) provided in a variety of formats
- easy use of animation, of special effects for e.g. water, smoke, fire
- easy simulation of realistic physics for forces, masses, momentum, energy
- easy ways of coding NPC AI behaviours using e.g. FSTNs

(Some) Development tools with free versions

- Game Maker: https://www.yoyogames.com/gamemaker
- Unity: http://unity3d.com/pages/create-games
- Unreal: http://www.unrealengine.com/udk/
- Construct 2: https://www.scirra.com/construct2
- CryEngine: https://www.cryengine.com

- Blender: http://www.blender.org
- Maya: http://www.autodesk.com/products/autodesk-maya/free-trial
- Ogre: http://www.ogre3d.org
- Photoshop: https://creative.adobe.com/products/download/photoshop
- SketchUp: http://www.sketchup.com
- 3ds-Max: http://www.autodesk.com/products/autodesk-3ds-max/overview
- 3dgamestudio: http://www.3dgamestudio.com/

For Game Development. GameMaker (trial version) required for Assignment 1

Mainly for Images, Modelling, Animation, and/or Rendering

Types of games

Ask yourself, rather than look up an answer in Wikipedia or whatever:

- What motivates a player to play a game? If "fun", what makes it fun?
- What sorts of player objectives or fantasies do games encourage?
- What sorts of game-play mechanics are available?
- How important is the technology (console, controller) to particular games?

See
http://www.makeuseof.com/tag/10-video-games-that-changed-the-world/