Teams, Project Brief

Overview of Software Engineering: Workshop 1

Ruzanna Chitchyan, Jon Bird, Pete Bennett
TAs: Alex Elwood, Alex Cockrean, Casper Wang

Meet the Team

Lecturers

Ruzanna Chitchyan

Pete Bennett

Jon Bird

Lab Support

Alex Elwood

Alex Cockrean

Casper Wang

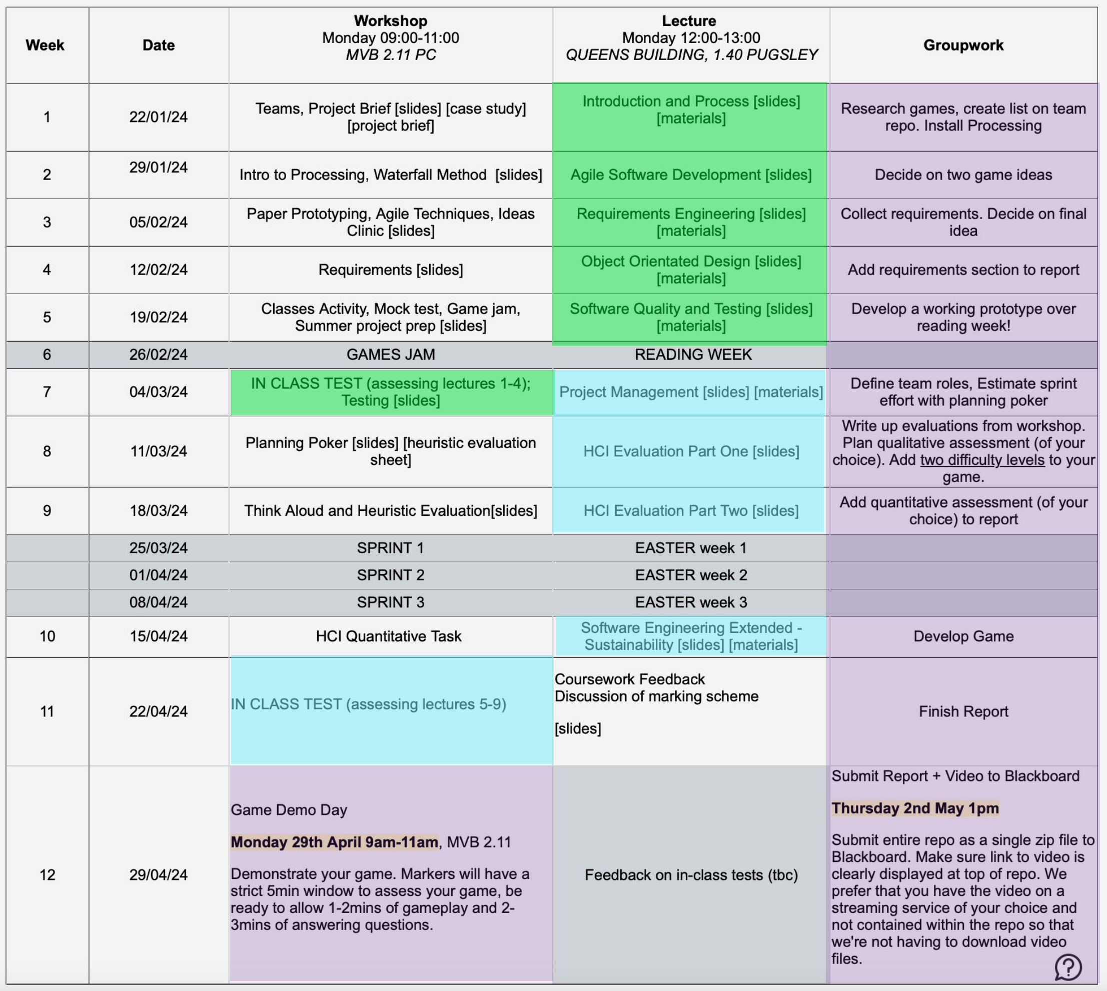
Purpose of Unit

Previous units aim to improve your knowledge as computer scientists

This unit helps package this knowledge as employability skill-sets

This is achieved by targeting the following industry-relevant issues:

- Transferable skills: Management, Collaboration, Communication
- Platform skills: applying your prior programming language knowledge
- Scalability skills: Development “in the large” at industrial scale



Week	Date	**Workshop** Monday 09:00-11:00 *MVB 2.11 PC*	**Lecture** Monday 12:00-13:00 *QUEENS BUILDING, 1.40 PUGSLEY*	**Groupwork**
1	22/01/24	Teams, Project Brief [slides] [case study] [project brief]	Introduction and Process [slides] [materials]	Research games, create list on team repo. Install Processing
2	29/01/24	Intro to Processing, Waterfall Method [slides]	Agile Software Development [slides]	Decide on two game ideas
3	05/02/24	Paper Prototyping, Agile Techniques, Ideas Clinic [slides]	Requirements Engineering [slides] [materials]	Collect requirements. Decide on final idea
4	12/02/24	Requirements [slides]	Object Orientated Design [slides] [materials]	Add requirements section to report
5	19/02/24	Classes Activity, Mock test, Game jam, Summer project prep [slides]	Software Quality and Testing [slides] [materials]	Develop a working prototype over reading week!
6	26/02/24	GAMES JAM	READING WEEK	
7	04/03/24	IN CLASS TEST (assessing lectures 1-4); Testing [slides]	Project Management [slides] [materials]	Define team roles, Estimate sprint effort with planning poker
8	11/03/24	Planning Poker [slides] [heuristic evaluation sheet]	HCI Evaluation Part One [slides]	Write up evaluations from workshop. Plan qualitative assessment (of your choice). Add two difficulty levels to your game.
9	18/03/24	Think Aloud and Heuristic Evaluation[slides]	HCI Evaluation Part Two [slides]	Add quantitative assessment (of your choice) to report
	25/03/24	SPRINT 1	EASTER week 1	
	01/04/24	SPRINT 2	EASTER week 2	
	08/04/24	SPRINT 3	EASTER week 3	
10	15/04/24	HCI Quantitative Task	Software Engineering Extended - Sustainability [slides] [materials]	Develop Game
11	22/04/24	IN CLASS TEST (assessing lectures 5-9)	Coursework Feedback Discussion of marking scheme [slides]	Finish Report
12	29/04/24	Game Demo Day **Monday 29th April 9am-11am**, MVB 2.11 Demonstrate your game. Markers will have a strict 5min window to assess your game, be ready to allow 1-2mins of gameplay and 2-3mins of answering questions.	Feedback on in-class tests (tbc)	Submit Report + Video to Blackboard **Thursday 2nd May 1pm** Submit entire repo as a single zip file to Blackboard. Make sure link to video is clearly displayed at top of repo. We prefer that you have the video on a streaming service of your choice and not contained within the repo so that we're not having to download video files.

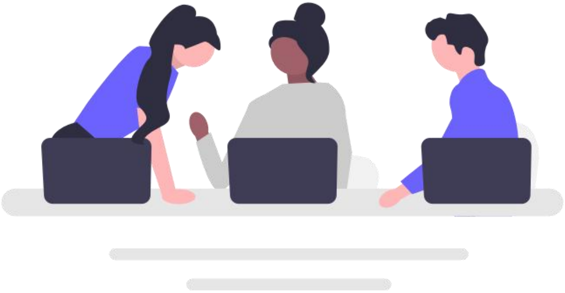
- ○ 25% in class test A
- ○ 25% in class test B
- ○ **25% project report**
- ○ **25% project presentation**

Coursework

- You get to work on one big project – for the whole term

- Apply all of the knowledge you have gained previously…to a significantly sized, real project

- Work on the project is structured, We won't just leave you to get on with it !
- Each week you will be given a structured activity, and support will be provided to complete that work
- You will be working as part of an integrated development team…

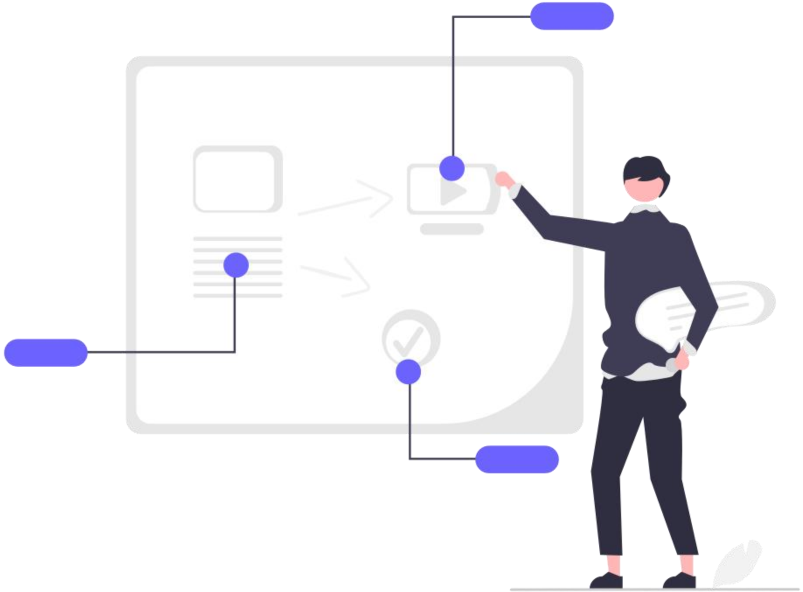


Today’s Workshop

- Introduction (20 min)
- Meet your team (10mins)
- Introduction to Project Brief (20mins)
- Set up team repo on Github (40 mins)

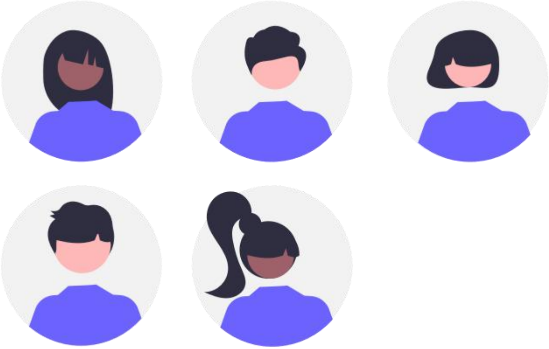
Meet the TAs

- **Alex Elwood**- Github/CI master
- **Alex Cockrean**– expertise from being on course last year
- **Casper Wang** – expertise from being on course last year



Meet your Team

- You have been randomly assigned to a team of five. Look up your team:
  - You can find out which group you belong by checking this list.
  - You can find your group's repository here: https://github.com/orgs/UoB-COMSM0110/repositories
- Get together in your group and introduce yourself (10mins) … *last book you read, favourite music... last game you played.*

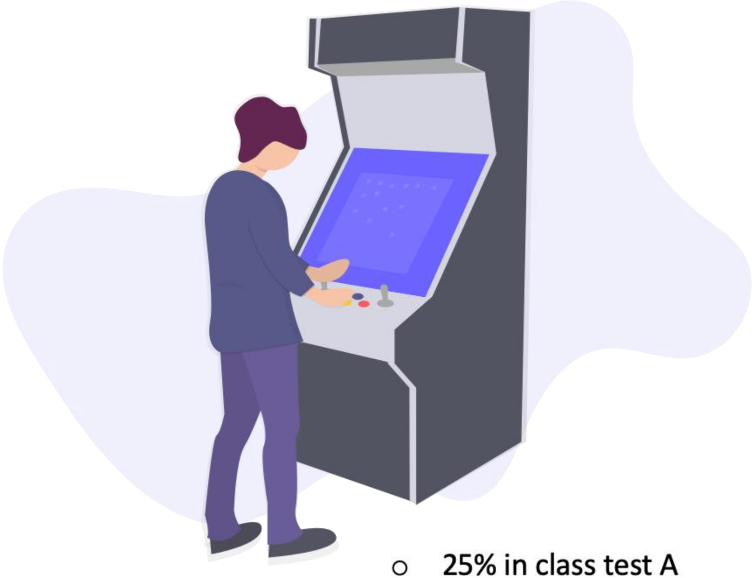


Coursework Brief – Games Project

- The aim of this group coursework is to **design and develop a game**, applying and reflecting on the software engineering methods you have learnt in the taught component of this module.

- Each team of 5 (assigned randomly) will develop a novel computer game based around adding a twist to an existing game or game archetype.

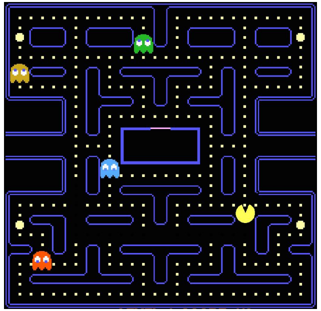
- We would like your team to identify and discuss **three major software development challenges** in developing your game.



  - 25% in class test A
  - 25% in class test B
  - **25% project report**
  - **25% project presentation**

Pac-man Example

- Pac-man, *but* the ghosts can rearrange the map
    - Challenge 1: The ghost's AI
    - Challenge 2: Data structure for storing the map
    - Challenge 3: Global leader-board

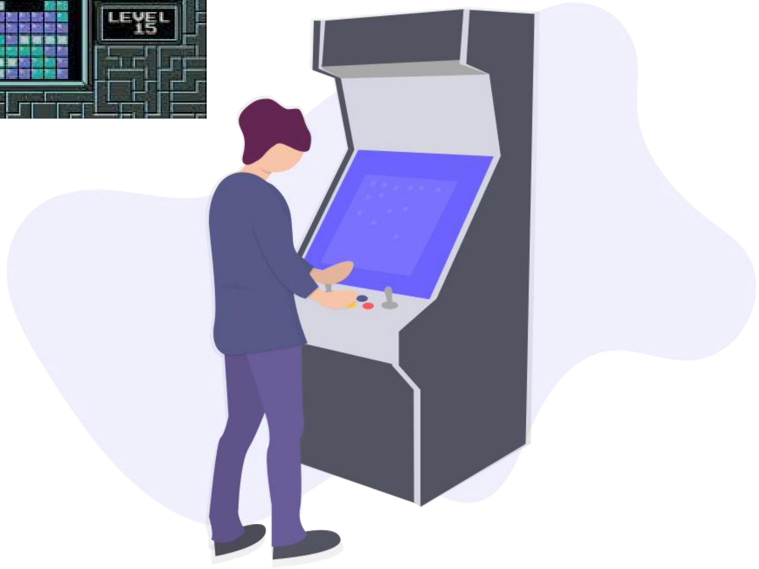


Asteroids Example

- Asteroids, *but* you play as the asteroids rather than the spaceship.
    - Challenge 1: Multiplayer implementation
    - Challenge 2: Asteroids fragmentation
    - Challenge 3: Spaceship AI

Tetris Example

- Tetris, *but* the blocks have different physical properties.
  - Challenge 1: Development of physical simulation
  - Challenge 2: Documentation and onboarding for block properties
  - Challenge 3: Algorithm for collision detection



Any Game Genre? Sure!

- Action
- Platformer
- Shooter
- Beat ‘em up
- Stealth
- Survival
- Rhythm
- Puzzle
- RPG
- Simulation
- Tower Defense
- Arcade
- Sports

- Party Game
- Horror
- Social Deduction
- Serious
- Generative
- Metroidvania
- Roguelike
- Dungeon-crawler
- Deck-builder
- One-button
- Text-based

… and many many more

https://www.theguardian.com/games/2021/oct/11/modern-video-game-genres-explained-metroidvania-dungeon-crawler

https://en.wikipedia.org/wiki/List_of_video_game_genres

That sounds like a lot!

- Remember that the primary aim of this module is to **learn**, **apply** and **reflect** on the principles of **good software engineering**.

- The quality of your process is the primary focus of assessment and not your game idea, though we hope that a quality game will be the outcome of a good process!

- We aim that by following the workshops this should be a straightforward coursework to pass, but with plenty of opportunity to excel and achieve higher grades.

```
                 DEFENDER
              HALL OF FAME

   TODAYS                    ALL TIME
  GREATEST                   GREATEST

1 DRJ   21270            1 DRL 999975
2 SAM   18315            2 DRL 999975
3 LED   15920            3 DRL 999975
4 PGD   14285            4 DRL 999975
5 CRB   12520            5 DRL 999975
6 MRS   11035            6 DRL 999975
7 SSR    8265            7 DRL 999975
8 TMH    6010            8 DRL 999975

          CREDITS: 1
```

But, what language?

- **Processing:** https://processing.org (*not Unity!*)

- It's Java! (with some extras) You'll be learning Java this term.

- Designed for use by artists, designers, creative technologists (designed to be straightforward, excellent documentation)

- Has a simple Integrated Development Environment (IDE)

- Many useful libraries!

- Why? We want to give you the experience of working as a team in learning a new language / environment (useful for summer project)

- But also, we want it to be a simple and enjoyable experience so you can focus on the software development process.

Some tips

- We will not be teaching you Processing, your team will be expected to engage and learn from online resources available (we will link you to these).

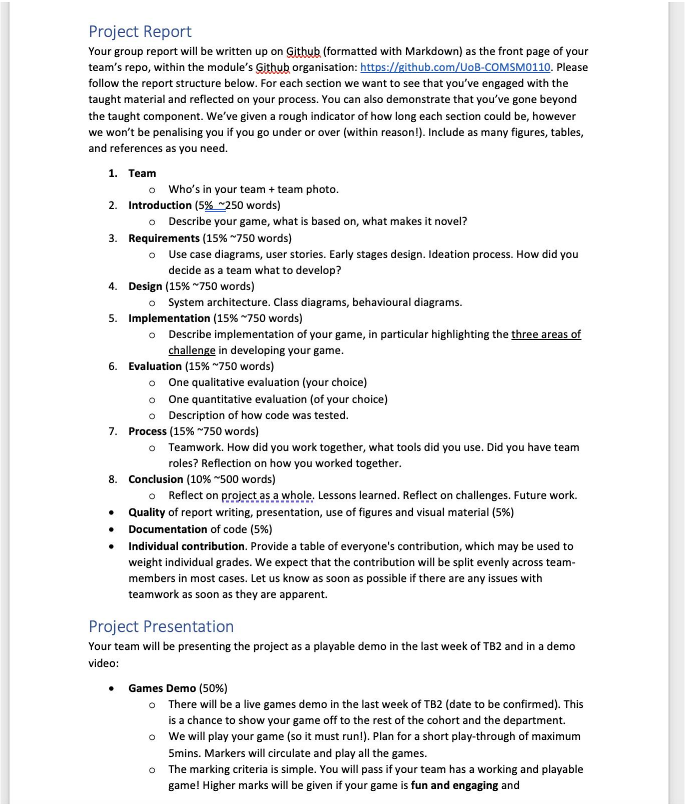
- Processing has many libraries which you are free to use, but remember that you need to identify and overcome three challenges, so if you use a library (for say physical modelling) then this may negate the challenge.

- Please note that this is a software engineering module and not a game design module, so although good narrative/artwork/levels/content will elevate a game, we are ultimately interested in your team's ability to develop software.

- We recommend sticking to 2D games (unless this is one of your challenges!)

Deliverables

- Project Report (25% module)
  - Front page of your Github project repo
- Presentation (25% module)
  - Game Demo in last week (5mins)
  - Video of your game. (3mins)
- Project Brief is available on BlackBoard: (link within table)



Project Report

Your group report will be written up on Github (formatted with Markdown) as the front page of your team's repo, within the module's Github organisation: https://github.com/UoB-COMSM0110. Please follow the report structure below. For each section we want to see that you've engaged with the taught material and reflected on your process. You can also demonstrate that you've gone beyond the taught component. We've given a rough indicator of how long each section could be, however we won't be penalising you if you go under or over (within reason!). Include as many figures, tables, and references as you need.

1. **Team**
   - Who's in your team + team photo.
2. **Introduction** (5% ~250 words)
   - Describe your game, what is based on, what makes it novel?
3. **Requirements** (15% ~750 words)
   - Use case diagrams, user stories. Early stages design. Ideation process. How did you decide as a team what to develop?
4. **Design** (15% ~750 words)
   - System architecture. Class diagrams, behavioural diagrams.
5. **Implementation** (15% ~750 words)
   - Describe implementation of your game, in particular highlighting the three areas of challenge in developing your game.
6. **Evaluation** (15% ~750 words)
   - One qualitative evaluation (your choice)
   - One quantitative evaluation (of your choice)
   - Description of how code was tested.
7. **Process** (15% ~750 words)
   - Teamwork. How did you work together, what tools did you use. Did you have team roles? Reflection on how you worked together.
8. **Conclusion** (10% ~500 words)
   - Reflect on project as a whole. Lessons learned. Reflect on challenges. Future work.

- **Quality** of report writing, presentation, use of figures and visual material (5%)
- **Documentation** of code (5%)
- **Individual contribution**. Provide a table of everyone's contribution, which may be used to weight individual grades. We expect that the contribution will be split evenly across team-members in most cases. Let us know as soon as possible if there are any issues with teamwork as soon as they are apparent.

Project Presentation

Your team will be presenting the project as a playable demo in the last week of TB2 and in a demo video:

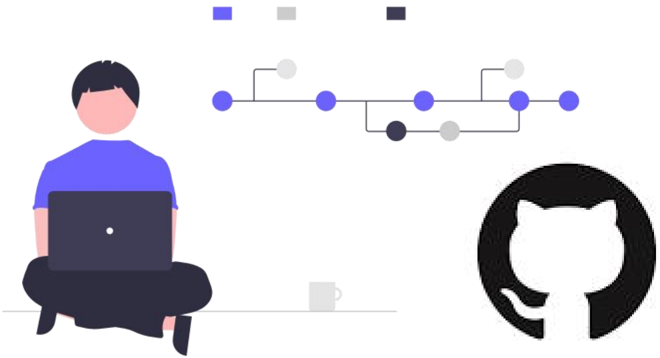
- **Games Demo** (50%)
  - There will be a live games demo in the last week of TB2 (date to be confirmed). This is a chance to show your game off to the rest of the cohort and the department.
  - We will play your game (so it must run!). Plan for a short play-through of maximum 5mins. Markers will circulate and play all the games.
  - The marking criteria is simple. You will pass if your team has a working and playable game! Higher marks will be given if your game is **fun and engaging** and

homework / groupwork

- Make sure you can make a commit. Try adding your name to your team's readme (<u>After Software Tools on Thursday!</u>)

- Then take and upload a team photo!

- Research games, create a list of games inspiration on your team’s repo. Make sure your team has plenty of options. Consider adding notes to each game (what makes it great?) and perhaps even ranking them by suitability and interest.



homework / groupwork

- Also… install Processing on your laptop: https://processing.org/download
  - No need to do any tutorials, we'll be starting from scratch in next week's workshop