Computer Science, Maths & Computer Science, Computer Science & Philosophy Part A

Group Design Practical

Briefing Document Hilary Term 2022

All second-year undergraduates reading Computer Science, Mathematics and Computer Science, or Computer Science and Philosophy take part in a group design exercise in their second year.

Goal

The goal of the group design practical is to give you, the student, a chance to understand and learn the skills required to succeed in delivering a real-world project, not just writing a program.

In addition to using the skills learnt during the four core programming courses to actually code, to succeed in this project you'll need to

- engage with your customer to understand what their needs are
- agree with the customer what your solution will and won't do in order to solve their needs (and update/negotiate changes with them as necessary)
- self-organise with your team to distribute work *and* make sure the whole solution works together
- plan your time to deliver what the customer needs when they need it
- provide the solution to the customer in a form
    - they can easily deploy (installation, configuration)
    - they can easily use (documentation, briefings, maintainability, good diagnostics)
    - they won't break (careful testing).
    - and considers the ethical and societal impacts of the solution.

These are all solid basics that any tech company or research group expects of any good contributor – this is your chance to develop these skills as part of your degree.

Key Dates

Briefing Meeting	Wednesday 26th January, 2pm	Week 2
Deadline to fix team members	Monday 31st January, 12pm	Week 3
Responsible Innovation Seminar	Wednesday 2nd February, 2pm	Week 3
Deadline to fix project choice	Friday 4th February,12pm	Week 3
Briefing on specific projects	Wednesday 9th February, 2pm	Week 4
Project Co-ordination Seminar	Wednesday 16th February, 2pm	Week 5
Project Management Seminar	Wednesday 23rd February, 2pm	Week 6
First progress meeting with external (and internal supervisor) – agree customer requirements and project specification and plan	Week 6 - Time and day to be confirmed by team and supervisor	Week 6
Software version control seminar	Wednesday 2nd March, 2pm	Week 7
Software testing seminar	Wednesday 9th March, 2pm	Week 8
Second progress meeting with external (and internal) supervisor –	Week 8 - Time and day to be confirmed by team and	Week 8

Progress report, initial implementation and test results	supervisor	
Third progress meeting with external (and internal) supervisor – delivery and final report	Week 2 - Time and day to be confirmed by team and supervisor	Week 2
Presentation event	Wednesday 11th May	Week 3

Organisation

You will be allocated to a team of around 6 people before the Briefing Meeting. Swaps between teams are allowed until the deadline to fix team members – to confirm a swap both students involved should email student-projects@cs.ox.ac.uk.

Depending on government guidelines, you might be required to conduct your work on the project remotely. You can use tools such as Teams, Slack and Zoom for communication, and Bitbucket and Github for code sharing.

Each team will then need to choose 3 projects from the list of available projects and email their first, second and third choices to student-projects@cs.ox.ac.uk, with a copy to all team members. Each team will be notified which project they have been allocated, and who is supervising that project. Further information about each project, and the materials and equipment available, will then be given to the relevant teams.

You will have an internal and an external supervisor.

The external supervisor will act as a customer with whom the team engage to understand any detailed requirements for the project and to whom the team deliver their finished product.

The internal supervisor will act as a mentor, ready to help the team deliver their project.

It is the team's responsibility to drive the project, requesting help from the external and internal supervisor as necessary.

As part of the project, each team *must* do the following.

- Arrange an initial planning meeting, with their external (and internal) supervisor in Week 6 of Hilary Term. At this meeting the team should agree what they will be delivering, when, to the customer and agree with their mentor a plan for how they'll achieve that.

- Arrange a progress report meeting with their external (and internal) supervisor in Week 8 of Hilary Term. At this meeting, they'll present a progress report to the customer on their module implementations, test plans and test results. If the team and their internal and external supervisor agree, the team may instead provide a brief written report on their progress by the end of Hilary Term.

This meeting is also a good point to negotiate changes to the scope or timeline for what they are delivering (if for example it turns out that their project is significantly harder than originally expected, they can negotiate to reduce the scope). It is absolutely fine for teams to agree with the customer to change the scope of what they're delivering as long as the customer has reasonable warning (while it is absolutely not fine for example for the team to propose a reduction in scope 24 hours before the final delivery date!)

- Arrange a delivery meeting with their external (and internal) supervisor in Week 2 of Trinity Term. At this meeting they deliver the agreed project to the customer, including the product, installation instructions, documentation, etc. as previously agreed.

- Arrange separate progress meetings with their internal supervisor unless the internal supervisor is able to attend progress meeting with the external supervisor. Each group should meet their internal supervisor at least three times.

In Week 2 of Trinity Term, each group will deliver to the internal supervisor a final project report and each student will also deliver to the internal supervisor a one-page summary of their individual contribution.

Teams are encouraged to engage with their external and internal supervisors more frequently than these three mandatory meetings.

On Wednesday 11th May there will be an event to celebrate you and your hard work. Each group will be able to present their work and give a 5-minute talk, and there will be cash-prizes, too!
You'll receive detailed information nearer the time.

Overview and Marking

The group practical exercise is designed to take 20-30 hours of student time, mainly during Hilary Term (so a team of 6 delivers a project of 120-180 hours of work). The contribution of each student will be based on the group report, demonstration, and individual contribution sheet. This mark will count for approximately one third of the practical marks for the second year. Students must pass the group practical in order to pass the practical component of the course. The group design exercise will be marked on a scale of S-, S(pass), S, S+. These marks should be converted to a numerical mark using the following scale:

S+	100
S	60
S (pass)	40
S-	20

Computer Science, Maths & Computer Science, Computer Science & Philosophy Part A

This exercise gives students experience of working against rigid deadlines, with a team of colleagues not of their own choosing, using externally supplied tools to undertake a fixed project. This will give some idea of the problems encountered in normal professional practice. Groups will be expected to exhibit professional skills in design, quality and management. Specifically, they will have to show that the work has been carefully planned, that components and systems have been properly tested, and that members of the group have cooperated effectively. The review meetings with sponsors provide an opportunity to monitor group progress and for general discussion, but sponsors will not be expected to provide technical advice or resolve technical issues. The responsibility for organising and completing the work lies squarely with the group members.

Our intention is that all group projects should be successful, and all students are expected to contribute to their own project in accordance with the timetable. Groups often encounter problems, and students are expected to manage these as they arise. However, if serious problems are encountered, such that members of the group are unable to resolve them, students should contact the academic admin team (academic.administrator@cs.ox.ac.uk).

Computer Science, Maths & Computer Science, Computer Science & Philosophy Part A

Group Design Practical

Deliverables

Each part of the project gives rise to a set of deliverables *which must be given to the internal supervisor at the review meetings.* The internal supervisors will forward all group and personal reports to the examiners for inclusion in the practical work portfolio of all group members.

Specification and plan

Project topics are presented in the form of an outline design brief. Part of the work is to undertake a proper requirements analysis for the chosen project. However, it is important not to develop an over-elaborate specification which commits the group to more work than is necessary. The first major task is therefore to turn a relatively open and informal design brief into a more detailed project specification, setting out the major components of the system you will produce, what documentation you'll deliver, what diagnostics you'll provide, etc. Once this is agreed with your internal supervisor, you must produce a project plan which sets out who will do what, the time needed for developing and testing each module, dependencies between modules, etc.

When planning and executing the project, it is important to work to a budget. The entire project should take no more than 40 hours per team member, and records should be kept of time invested. Groups should set realistic targets and achieve them; there will be no additional credit for over-elaborate projects, or for individuals who offer or do more than is required. Each member of the group is expected to gain experience of programming in the course of the project – this may possibly involve test harnesses or scripts, data conversion utilities, a tutorial system, external interfaces, demonstration examples, installation packages, or other code and associated materials as appropriate to the project and the individuals in the team.

Module implementation and testing

The next task is to implement the components and test them. This is likely to require the construction of special test harnesses for separate classes. The deliverable for this task is a written *progress report* describing the testing procedures and results.

Project delivery and report

The final task is to piece the whole system together, test it and ensure that it is adequately documented. At the final meeting, the group must:

- Demonstrate the solution to the external supervisor.

- Deliver the solution and associated documentation in a format that the external supervisor can use.

- Submit a brief (approximately 4 pages long) group report describing the project aims, outcomes, successes and failures, and any lessons learnt. This report should include a short (approximately half a page) discussion of responsible innovation, exploring (i) the ethical issues you needed to consider during your design and development activities and (ii) the potential societal impacts (both positive and negative, intended and unintended) of the solution that you have developed.

Each team member must also submit a one-page summary of their own individual contribution.

Project A: Tradeteq: Adversarial attacks and defences for credit scoring systems
[Project in collaboration with Tradeteq]

Many financial decision making systems are increasingly reliant on machine learning. Credit scoring for companies and individuals in one of the common applications. Ample incentives for unscrupulous market participants to game those systems create the need for robust credit scoring resilient to realistic attacks.
In this project, you will be playing the role of a “white hat” hacker of a financial machine learning system. We will provide you with a large pre-processed UK limited company dataset. You will calibrate a simple machine learning model to predict company status transitions for these companies and will then develop adversarial attacks on this model. Some company features are much more prone to manipulation than others. The attacks will seek to improve the credit score by changing those features. The attacks may be developed while using the target model as an oracle or with full access to target model coefficients. After creating a successful attack, you will need to investigate possible defences against this attack

Project B: Web-based Editors simulation of C/C++ code and Python
[Project in collaboration with Micro:bit Educational Foundation]

Teaching people to code with blocks is great, but we ultimately want them to learn to code with text-based languages. When moving to text from block-based coding, where syntax errors and typos are impossible, giving very fast feedback to students when they make these errors is essential. Sadly, we currently do not do this in our Python editor! Therefore when the micro:bit is running MicroPython, we want to be able to run their code immediately in the browser, without needing to run it on the device to find out it’s broken.

We’ve got a prototype “Python Simulator” that builds the MicroPython C/C++ code using ‘emscripten’ so that it can run in the browser. This has resulted in a proof of concept for simulating Python… we need a team that can understand the requirements of users to and work across many levels of the software stack: C/C++, compilers, javascript, UI/UX in order to deliver an awesome simulator. There’s scope to turn this project into a more generic simulator for C/C++ projects on the micro:bit, perhaps by attaching a compilation backend or running firmata on ht micro:bit.

Project C: Creating and recognising unique and playful sounds on the BBC micro:bit

[Project in collaboration with Micro:bit Educational Foundation]

Version 2 of the BBC Micro:bit launched in 2020 with an added speaker and microphone. It can now make and sense sound, and we've worked to create a unique and playful set of sounds (giggle, twinkle, happy, sad, soaring, etc) that really engage kids. The sounds are created using a synthesiser that runs on the embedded device, sweeping and interpolating between frequencies, using effects as specified by the particular sound. We'd like to take this experience further in two ways

1. Create a way for the micro:bit to also respond to these built-in sounds. Whether that's a task for structured audio analysis (frequency fingerprinting, analytical recognition), machine learning (perhaps the constraints of the environment make it too hard to do analytically), our audio fingerprinting (can you encode a fingerprint in near-ultrasound signal as part of the audio without distorting it?)
2. Build an interface for students to experiment with making their own sounds. We have a prototype 'sound maker' tool that lets an advanced user specify a bunch of parameters that was used to create the exiting sounds. We want completely redesign this to be something students will enjoy using and find intuitive - this side of things should run in a browser.

This will be a mixture of low-level optimised C/C++ coding and UI/UX experimentation. There is also scope to turn this project into a 'transmitting data using sound" problem, where the micro:bits in a room can talk to each other, for example using Morse Code that students can audibly hear, or some other sound encoding with sensible audio encoding and error correction.

Project D: Using machine learning to interact with the BBC micro:bit and MicroPython

[Project in collaboration with Micro:bit Educational Foundation]

The V2 of the BBC micro:bit, launched at the end of 2020, has a much more capable process than the original device, including the ability to run TensorFlow models and perform recognition of patterns on the device. There's huge potential for using the BBC micro:bit to teach AI and ML to students, but understanding the way to plug together the various pieces of the puzzle is tricky. We would like to investigate using MicroPython (and embedded, stripped down Python Interpreter) on the BBC micro:bit as a way to store and run tensor flow models, as well as extending our online Python Editor to include tools for training the models in the front end. Can we build an experience where a student claps a custom pattern 5 times to the computer's microphone and we train a model to recognise this pattern, and then deploy it to the micro:bit device using the online editor? What about recognising a custom keyword the user says? Can we use model retargeting or transfer learning to make this process tractable in the front-end in real time? What about using the built-in motion sensors instead of the microphone as input?

We already have a proof-of-concept example of the micro:bit recognising the word micro:bit being said, some of the time. It's a good start but it isn't interactive, in that students can't train the device, so it's not an adequate demonstration of the actual real-world use of ML. We need to improve it and make and end-to-end experience for learners aged 8-14 to enjoy.

This will likely involve a mixture of C/C++ for MicroPython, Python, Javascript (likely on the Python Editor and some Tensor-flow JS) and possibly some WebAssembly or Emscripten work. There's a lot of flexibility to modify the scope as you find interesting, as long as it's around creative ways to teach students AI & ML with the micro:bit.

Project E: Automated Real-time Global Events Data Collator and Persister

[Project in collaboration with Apex:E3]

APEX:E3 has created a global event based financial instrument backtesting framework, where users can answer questions like:

What impact does a hurricane hitting the American Eastern coastline have on insurance stocks? Or How did the victory of Boris Johnson affect FX rates and Banking stocks?

The events are stored as time lines and our users can backtest trading strategies around these timelines e.g. Buy GBP/EUR and GBP/USD 1 month before the election and sell 2 months after the election.

The idea is to create an automated real-time global events data collator and persister, where the following type of events are consumed from publicly available sources on the internet then classified, tagged by sentiment and persisted to a database or index for future querying:

1. Geo Political - e.g. US Trade war timeline, Brexit timeline, Oil tanker issues, Trump tweets
2. Financial - e.g. Companies earnings report timeline, key company announcements, IPOs, mergers & acquisitions
3. Sports - e.g. European / US / Asian soccer/baseball/cricket/Formula 1 teams that are listed on stock exchanges or have sponsors that are listed on stock exchanges. Example events include premier league match results, Formula 1 race wins/losses
4. Extreme Environmental Events - e.g. hurricanes, earthquakes, tsunamis, droughts, landslides
5. Entertainment - Film releases, Game launches, Stadium events like music concerts and boxing matches, Music releases

The outcome of this project is to have a functioning prototype which automatically collates and persists events as described above, with means for further extension. Bonus points will be awarded if the prototype can also detect fake news/tweets/content.

Project F: Earth Trust

[Project in collaboration with Earth Trust]

Trees are the lungs of the planet absorbing carbon dioxide and sequestering the carbon. However, the extent and timescale of predicted climate change will impact trees and woodlands extensively.

Earth Trust, an environmental learning charity in South Oxfordshire, owns Paradise Wood, a national research woodland. Paradise Wood contains the largest genetic collection of hardwood forestry trials in Britain. They have raw data on the growth of different tree species in the woods and what to use this data to show the importance of the woodland and the impact of climate change of different trees.

The task for this project is to build a prototype system to correlate this data with metadata from the Met Office to visualise which trees have grown best despite the more extreme weather events we've had in the last 20 years. They are looking for an output with a wow factor - the system will need to be able to produce an appealing front-end visualisation that engages people in the importance of the Paradise Wood and why it is vital to build reliance into our woodlands for the future.

Project G: Finding Technical Talent

[Project in collaboration with Microsoft]

A software telecoms company are looking for ways to attract and identify new technical talent. They're looking for some kind of game or puzzle that will run in a web browser, with twin underlying goals of (a) get lots of students interested in and applying to the company and (b) automatically flagging any particularly good looking candidates to the company to fast-track to interview. The company has some ideas for puzzles and games that might work, but are open to ideas and suggestions from the consultants (you!) – they're really looking for something with wow factor. There's no restriction on the set of languages/packages/back-end that you use.

Project H: Automating Contact Tracing

[Project in collaboration with Bloomberg]

One of the things we've collectively learnt over the past year is that when global health emergencies such as the Coronavirus Pandemic happen, being able to effectively test and contact-trace large proportions of a country's population is critical in the fight against the disease. It is therefore of important value to have a unified software that can be employed by professionals involved in this effort to make their workflows as efficient as possible and to maximize throughput.

Users of this system would include: testing laboratories, contact tracers and government officials. Testing labs and contact tracers would act as data sources for the system, and government officials would consume the information within (who tested positive, when, where, what contacts?) to generate reports and analytics.

An example workflow would start with a person getting a Covid-19 Test. Once the test has been processed, the testing facility would add the result into the system, and, if positive, the case would automatically be dispatched to a human tracer. That person would then contact the individual and perform contact tracing, notifying all close contacts that they were exposed. Government officials responsible for the country's response would be able to use the data to extract trends as they happen and generate reports and maps to disseminate to the public and to use in guidance for decisions.

The project will involve designing, building and testing such an application using artificial data for proof of concept. You should largely consider the system a pub-sub system where publication is done by test centres, and the information is then consumed by contact tracers who then manually do the work. You should give special considerations to how this system can guarantee security, as the information it contains is highly sensitive. You should also consider what kinds of analytics of the data is enabled by having a unified test & trace system.

Project I: Real world, real time social media sentiment analysis
[Project in collaboration with Amazon Web Services]

AWS is a data driven company and therefore gathering real time data about key world events is critical to our business. Whether it’s trends in technology, reactions to financial news or importantly, data about AWS related events and launches. Increasingly social media data is a crucial source to not only gather hard metrics, but the sentiment around that data. Is the news positive or negative, what do people feel about a technology trend, about an AWS launch? Our customers are our number one priority and knowing how they feel about something is invaluable data to us.

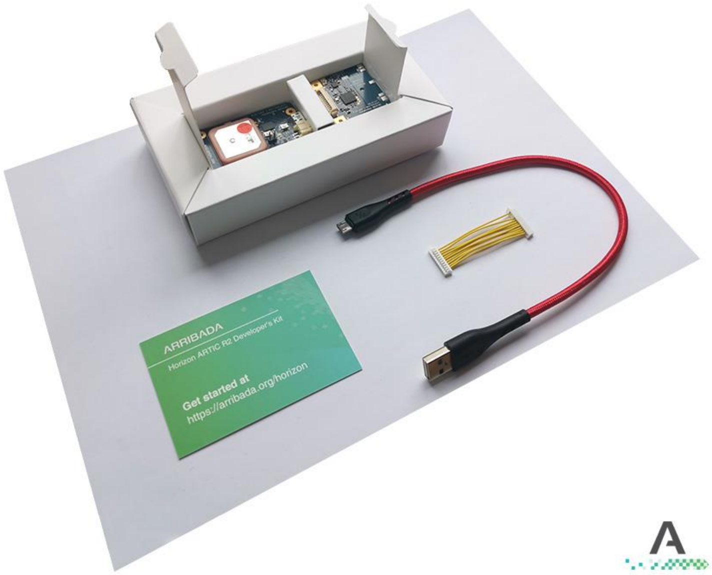
Your task is to create an application that marketing can use to analyse the sentiment of any topic in real time from Twitter, and provide demographic information, such as gender, location, age range etc of the people the data is about. The application needs to be user friendly to a non-technical marketing audience so that they can input any news topic and receive and output of data in a user friendly presented format. You can either write your own ML model to abstract sentiment or you can use the AWS AI managed services such as Amazon Comprehend.

Project J: Visualisation and management of satellite tag tracks
[Project in collaboration with Arribada Initiative]

The Arribada Initiative (https://arribada.org) co-develops open, customisable and impact-driven conservation technologies for conservation organisations across the globe. One such project is the open-source Arribada Horizon tag; a low-power open-source GPS tracker that transmits near real-time position updates over the Argos satellite constellation. This device has been used in applications as diverse as tracking sea turtles to understand their feeding behaviour, and tracking the path of plastic waste discarded in the oceans.

In this project, you will build a data analysis backend for this system. The solution will allow users to manage a deployed fleet of Horizon tags, automatically downloading position data from the Argos API, and presenting the raw data in a usable form. In particular, the system should allow integration of associated weather (and other relevant streams), and allow users to produce compelling visualisations of tracks in real-time.

Horizon developer kits will be provided to allow you to transmit data to the Argos satellite constellation and test your backend system.


ARRIBADA
Horizon ARTIC R2 Developer's Kit
Get started at
https://arribada.org/horizon

Project K: Idea Atlas
[Project in collaboration with Idea Atlas]

Idea Atlas is an Oxford University Innovation supported (phase 2) startup focusing on public thinking and idea monetization. We are working with several Oxford/UK, EU, and US based organizations to develop our new solution for constructive collective thinking. Our core platform combines the best aspects of online fora, social media, and encyclopaedia formats to deliver an intuitive, intriguing, and highly rewarding ideation experience.

We would like to explore novel use cases or verticals around our core technology. For this project you will assist with creating specific UI/UX solutions and higher quality visualizations than currently operational through the following suggested routes. You can choose to tackle one of these suggestions;

1. Integrated academia-to-policymaking problem-solving and discussion tool. The objective here is to co-develop (i.e., iterate) policy proposals with features that enable research-supported arguments and selective referencing. Our engine and networking solutions would be augmented by the team’s UI/UX vision and database management proposals. Codename: ‘Polycratic'

2. Integrated blog-to-basket or forum-to-basket solution for commercial use (e.g., wine/delicacy shopping, clothing, etc.). Our solution would require context specific tweaking

and we would help choose the starting retail focus together with the team. Otherwise: same as above. Codename: ‘Scalefeed’

3. Our on-going project with the Oxford Hub (codename: 'Oxolve’) requires a team of dedicated students to help build a local community oriented problem-solving platform. This would be a special version of our core platform which would be entirely focused on solving ESG/SDG/welfare challenges in and around Oxford. (NB: This project will certainly continue after Hilary Term.)