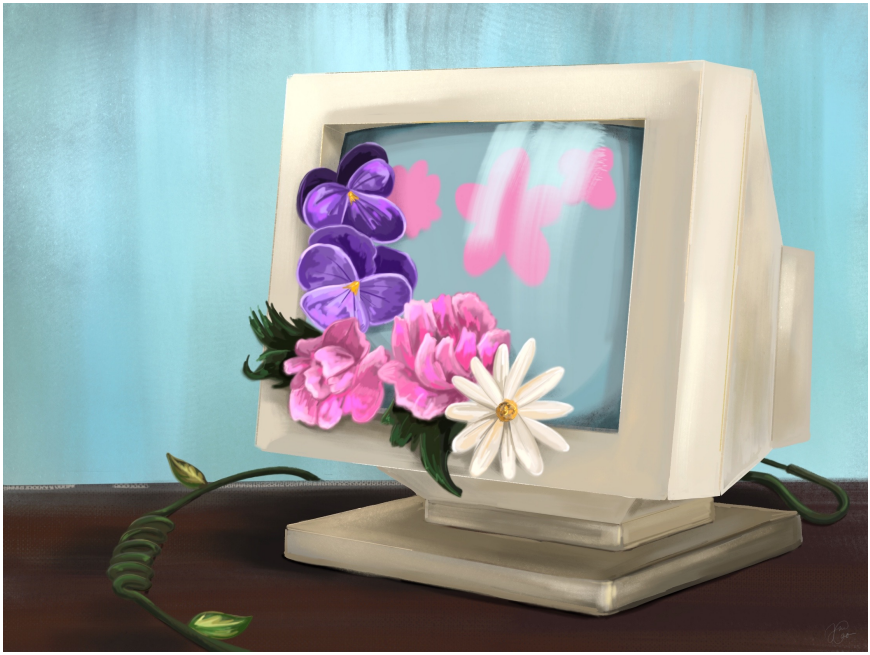


UNIVERSITY OF
WATERLOO

Software Engineering Capstone Design Projects 2023

March 28, 2023

Teams

Natural Language Processing	1	[REDACTED]
	2	Rizzume
Research	3	Jazzberry Jam
	4	Wombat
	5	RapidDash
	6	MAAPZ
Music	7	Angostura
	8	Studio Echos
	9	ZAMS
	10	JATY
Games	11	Peg.len()
	12	Rat Team Supreme
	13	Bonk
Graphics	14	OBJecTive
	15	HRTZ
	16	Pacific Time
Networks	17	Mirage
	18	Beam
Productivity	19	Aloha
	20	JVM
Education	21	Flash
	22	Runtime Terror
	23	SHSN
Community	24	Khalsa Tech
	25	Zephyr
	26	Power

Projects

AI-Powered Redaction for Refugee Documents
Rizzume: Grammarly for Resumes
Is scale all you need in neural networks?
Wombat SymX: A Node-Based Symbolic Executor
RapidDash: Compile Dash Into Executable Code
Ape: The Time Travel Programming Language
Code Generation for Alda, a music programming language
Pocket Ensembles: At-Home Music Practice
Musync: Location based music sharing platform
AQUA: A System-Wide Parametric Equalizer
Veloren: Chat and Procedurally Generated Structures
Kill the Lich: a multiplayer dungeon crawler
OURvoxel: A 3D voxel game
Adding OpenGL Pixel History to RenderDoc
Bezier-rs: A Bézier curve library for 2D graphics
Procedural Generation Toolkit
Mirage: A Lightweight Hi-Fi Honeypot Network
Beam: A Secure File Transfer App
Enabling engineers to be productive on day 1
Restocked: a notification system for consumers
Flash: Modernized Flashcards
Kid Investor: A Stock Market Game for Kids
CapstoneGatsby: New and Improved SE Capstone Project Website
Akalbook: A Booking Platform for Gurdwaras
Halo: A Street Harassment Prevention App
Reducing Carbon Emissions Through Data

1 AI-Powered Redaction for Refugee Documents Team [REDACTED]

We are working with the Refugee Law Lab (RLL) at Osgoode Hall Law School to create a privacy-respecting machine learning system for automatically redacting refugee applications. This proof of concept will form the basis of a human-AI system for quickly redacting refugee applications and similar legal texts.

Nancy Shen, Kate Granstrom, Benn McGregor, Juliana Zadarko, Serena Hacker

Refugee applications being released to the public must be redacted. This is done manually by the Immigration and Refugee Board of Canada (IRB). Manual redaction is used to ensure accuracy and because the IRB conflates the use of technology to enhance redaction with privacy risks for refugee applicants. Manual redaction is expensive and time-consuming, leading to only a small fraction of refugee applications being publicly released, hindering scrutiny of the refugee application process by researchers and the public. Our system demonstrates to the IRB that ML can enhance both transparency and privacy for refugee applicants.

Our system features a large language model that takes in applications and suggests to a human collaborator which phrases should be redacted to anonymize the refugee applicant. We use one million paragraphs of real, redacted refugee applications to do unsupervised "pretraining" on a transformer model, followed by supervised learning with a smaller synthetic dataset of documents before and after being redacted.

The state of the art (SOTA) uses regular expressions and named entity recognition to find text that should be redacted. However, this approach is unable to detect more vague contextual information that should still be redacted, while our ML-powered approach can. For success, we aim to show that our approach has better performance and reduced bias against the SOTA, and maximal privacy-protection of refugee applicants' data throughout the development process.

2 Rizzume: Grammarly for Resumes

Team Rizzume

Tyler Pinto, Mann Patel, Eric Feng, Dhvani Patel, Curtis Chong

According to the Wall Street Journal, 75% of resumes never make it to the hiring manager. This indicates that most do not know how to polish their resume. Although some online courses teach people how to write better resumes, these courses do not give specific feedback on your current resume. Unfortunately, most people never watch these lengthy courses because they would rather spend their time applying for jobs or preparing for interviews. Moreover, since people apply to jobs only once every couple of years, resume writing is a skill that is rarely exercised and polished.

Capturing this opportunity, Rizzume aims to help undergraduate students improve their resume through a Grammarly-like web app that surfaces specific feedback on how students can enhance their resume.

Unlike traditional online courses, which offer generic advice, our service uses natural language processing to understand a user's resume to provide specific feedback. An example of such a rule would be to eliminate first-person pronouns for consistency (e.g. "our product" would become "the product"). Like Grammarly, Rizzume will highlight sentences that users can improve and provide a suggestion next to them for users to accept.

Another problem students currently face is that waiting for peer feedback could take hours or days. Reviewers are also distracted by small mistakes such as grammatical ones. By providing immediate and tailored feedback for these common mistakes, Rizzume enables human reviewers to focus on more meaningful feedback, such as the content.

We soft-launched Rizzume during the winter term of 2023, allowing students to improve their resumes with our app. We released it to a focused group of 20 users who we were critiquing at the SE Soc resume critique session.

3 Is scale all you need in neural networks? Team Jazzberry Jam

Bradley Brown, Jordan Juravsky, Atif Mahmud, Wais Shahbaz

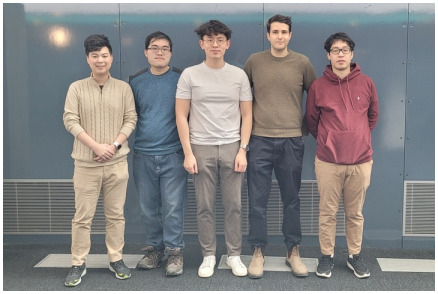
Recent progress in deep learning has demonstrated that model capabilities can be dramatically improved by simply scaling up the size of the models and the datasets that they were trained on. One prominent example of this phenomenon is large language models (LLMs), neural networks trained to predict the next word in a partially-completed passage of text. Models optimized with this simple objective, paired with sufficiently large computational and data budgets, have shown a remarkable ability to complete complex tasks such as code synthesis, question answering, and coherent dialogue generation.

As language model sizes and datasets continue to scale, an important question to ask is whether this increased scale is beneficial for all problems. Are large language models uniformly better than smaller models across all domains, or are there areas where increased scale is detrimental to model performance? This is the question specifically addressed by the Inverse Scaling Competition, a machine learning research competition which tasks participants with finding problems where the performance of LLMs decreases with increasing scale. In our FYDP, we design and evaluate a broad set of natural language tasks. Our experiments show that some of these tasks show pronounced inverse scaling, which we submit to the competition.

The tasks that we developed investigate a variety of hypothesized failure modes for language models. We explore whether models are able to assign new semantics to previously seen notation and concepts, whether models can be misled by correct, but misrepresentative examples, and whether models are susceptible to many of the cognitive biases present in humans. We evaluate these tasks on a diverse set of models whose sizes span three orders of magnitude, finding that some tasks' performance correlates positively with scale, some tasks scale inversely, while others demonstrate almost no correlation.

As models continue to scale and LLMs become increasingly prevalent in daily life, it is critical that we understand not only the advantages but also the shortcomings of these models. We hope that our study contributes to a growing body of literature attempting to better characterize the suitability of LLMs for practical applications.

4 Wombat SymX: A Node-Based Symbolic Executor
Team Wombat



Benjamin Wu, Bjon Li, Jerry Yu, Justin Reiter, Kevin Zhu

Symbolic execution is a powerful approach to formally verifying program safety. It is more comprehensive than common approaches such as running individual test cases. Symbolic execution can verify the safety of certain programs by checking that there does not exist any input in the domain of a function causing the program to panic.

This project aims to create a new tool for verifying Rust program safety that is more performant under large numbers of conditional code execution branches than current leading solutions such as KLEE by analyzing nodes instead of paths.

Writing safe software is hard and proving the safety of software is even harder. Symbolic executors can analyze some programs to either certify the absence of any inputs leading to a panic state or find an input leading to a panic state.

Symbolic executors historically had performance issues resulting from path explosion. Path explosion is the exponential growth of the number of execution paths relative to the number of conditional jumps in a program. This research project addresses path explosion by considering nodes of the program control flow graph instead of paths—relying on optimizations of modern SMT solvers to handle the additional variables this approach generates.

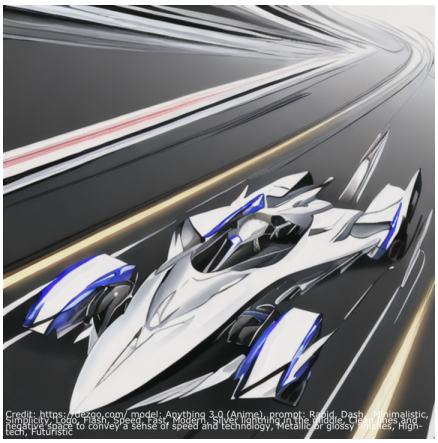
The team has identified that the LLVM IR of Rust code should be analyzed by the symbolic executor. The LLVM IR can be interpreted as a control flow graph (CFG) that symbolic executors operate on. Furthermore, CFGs can be transformed into the required dynamic single assignment form via existing libraries. The team has also written code demonstrating the validation of safety on simple Rust programs supporting basic arithmetic, logical operators, comparators, conditional branching, and assertions.

https://github.com/JustinReiter/wombat-symx

5 RapidDash: Compile Dash Into Executable Code

Team RapidDash

RapidDash explores the possibility of program synthesis from the Dash modeling language. Dash provides a state transition framework as well as an expression framework that can be used to build guards and actions for the transitions. RapidDash is a command-line tool that takes in a Dash model and generates a Python program. Our objective is to be able to translate all Dash state transition features and simple Dash expressions into Python, and the translation should satisfy the constraints specified by the input Dash model.



Jianyan Simon Li, Ruomei Yan, Aidan Joseph Campbell, Kent Wang

The design of RapidDash has two parts: the state transition system and the expression system. Our state transition system design is inspired by existing research on UML state charts, where states are translated into a hierarchy of state objects in the output program. For the expression system, existing research proposed direct translation using syntactical rules, or testing different program instances and selecting the ones that satisfy the expression. RapidDash adopts a combination of both approaches, where we translate expressions into Python predicates at compile time and choose possible variable values that satisfy the predicates at runtime.

There are several existing code synthesis techniques in research literature: UML state charts rely on user-provided operational details; Sketching can only generate a portion of the program, or fail at non-deterministic input; Other projects resort to calling the solver at runtime. RapidDash is able to mitigate these problems using our novel expression-translation technique.

Following the footsteps of existing research, we verify the correctness of RapidDash by doing case studies with Dash models that utilize different features of Dash. We feed the models to RapidDash and inspect the output Python program to verify that the programs satisfy the specification.

6 Ape: The Time Travel Programming Language
Team MAAPZ

Zac Joffe, Aliqyan Tapia, Alex Bakker, Paul Cento, Mio Yamanaka

Programming languages are often created to improve functionality from an already existing language, or to venture into a space that introduces a new way of programming. One area that has not been explored much in the field is reverse execution of code. More specifically, the notion of undoing previously executed instructions and reverting global program state. Such functionality eliminates the need for explicit cleanup in programs where there may be multiple complex phases that are prone to exceptional behaviour. Transaction processing, for instance, benefits from this implicit state restoration. Our team aspires to create a programming language that employs reverse execution functionality and use it to demonstrate this new way of programming.

Ape is a statically typed programming language with functionality to programmatically reverse program execution. Ape is implemented as a tree-walk interpreter, which incorporates execution state memory. The interpreter implements the execution state store and the associated state saving and restoration functionality. Decoupling the implementation of the code generator and the runtime allows for a simpler implementation of the state store and minimizes regressions for new features in the Ape language.

Developers do not interact with the state store explicitly, but rather leverage Ape's exception handling mechanism to implicitly manage global program state. The novel exception handling constructs allow for blocks of exception-prone code to be run, rerun, and skipped. Exceptions unwind the stack — similar to C++/Java exceptions — and rewind the effect of instructions executed prior to finding a corresponding handler. Through Ape's exception handling, a well-written program can eliminate the need for manual state cleanup upon exceptional code.

Compared to the traditional stack unwinding-based exception handling mechanisms found in languages like C++ and Java, Ape provides more powerful control flow. This approach builds on top of something developers are already familiar with and prevents the potential misuses that come with the power of explicitly interacting with the runtime's state store.

7 Code Generation for Alda, a music programming language
Team Angostura

Music notation is notoriously complicated, and existing software such as Finale and Sibelius are both rigid and difficult to get started with. Alda is an open source programming language for text-based music notation focusing on flexibility and ease of use. In an Alda score, standard musical elements such as notes and chords are combined with programmatic features such as variables and labels.

David Lu, Alan Ma

Our contribution is to implement code generation for Alda, supporting the transformation from internal representation into idiomatic Alda code. The code generation will be split into two distinct phases. First, a generator maps the internal representation back into Alda's parsed AST representation. Second, a formatter cleverly formats AST nodes into idiomatic code. Both changes involve significant updates and refactors to the codebase, and are discussed in-depth with the creator and owner of Alda: Dave Yarwood.

Code generation is an important addition to the Alda ecosystem, and will be highly impactful to all Alda users. The primary purpose of a code generator is to support importing from MusicXML. MusicXML is the industry-standard representation for musical scores, and all popular software is capable of exporting to it. In combination with a previously written tool (by our team) that parses MusicXML into Alda internal representation, the code generator will support direct translation of scores from third party music software to Alda text files. This will greatly increase overall language interoperability and lower the barrier of entry to writing Alda. A secondary purpose is to provide a standalone Alda code formatting utility. Various underlying language changes are necessary to make this happen, but the result will be a significant quality-of-life improvement to the development process.

A next step is to continue with improving interoperability by implementing Alda export to MusicXML. This is a much more challenging project, as it's debatable how to represent some of Alda's programming constructs in plain sheet music.

https://alda.io/

8 Pocket Ensembles: At-Home Music Practice
Team Studio Echos

Bruce Wang, Ariel Liao, Yifei Zhang, Kevin He, William Wen

Listening has always been an important skill for musicians, from beginners to professionals, and working with ensemble members is an important skill to develop. However, musicians often have limited time in a week to practice together, with other time spent practicing by themselves. Professional musicians play with their orchestra an average of 20 hours a week, while they spend another 28 hours a week practicing alone. Pocket Ensembles seeks to make this practice time more effective, while also making it fun for beginners.

Pocket Ensembles is a web application where you can practice ensemble and group music right at home. Music teachers and ensemble leaders can upload pieces for practice. Users then have access to music sheets, and can select which instruments that they want in the audio, and vary the tempo of the piece that they are practising. They can choose to use computer generated audio, or to use audio clips uploaded by fellow ensemble members.

For musicians, it is very difficult to practice ensemble music remotely. While recordings of pieces may exist online, they also include your part, and the tempo is at a performance level, which may be too fast for practising and learning the piece. In addition to this, manually scrolling back and forth on a recording can be cumbersome and annoying, and adjusting tempo for your own practice is virtually impossible. The main benefit that this application brings is the ability to seamlessly assemble different instrument recordings and to play it back in order to make the orchestra come alive.

Our objective is to make an application that caters to all musicians. It will help make practice both fun, and also give tools for professional musicians to emulate full orchestral rehearsals at the comfort of their own home. As the final product, Pocket Ensembles will be a practice tool that will be both fun for amateur musicians, while providing quality practice tools for more professional musicians.

9 Musync: Location based music sharing platform Team ZAMS

Ethan Zohar, Abdullah Bin Asad,
Lukman Mohamed, Lucas Budz

We live in an era where content-sharing apps are dominating the internet. These apps tend to specialize in one specific type of media such as photos, videos, and music. As new platforms are constantly developed and released, there has yet to be a successful app that focuses on live music sharing. While mainstream music streaming apps, such as Spotify and Apple Music, have billions of active users, they only support limited methods to share songs with a live audience. With a large number of music streaming users and no dominant competition, Musync will capitalize on the current market conditions and become the leader in live music sharing.

More specifically, Musync is a music streaming platform where users can share music with a live audience. Leveraging Spotify's API, we allow streamers to sync their actions (changing songs, pausing, playing, seeking, etc.) to any listeners following them in real-time. For our users searching for an audience, they can become streamers. To start streaming, they only need to be playing a Spotify song or playlist before clicking the "Go Live" button. For our users trying to find new and exciting music, they can become listeners. To start listening, they can press on any streamer within their area on a map.

Musync uses a streamer's location for discoverability rather than an algorithm like most content-sharing apps. The users who discover a streamer are the people within their community, the music lovers around them. Musync allows people to connect not by name or appearance, but by the music they listen to. This provides an all-new way of exploring and finding new music!

https://linktr.ee/zams_musync

10 AQUA: A System-Wide Parametric Equalizer
Team JATY

Frank Wu, Darren Li, Hemit Shah, Selina Li

Audio equalizers allow users to adjust the volumes of different frequency bands in an audio signal, in real-time. Software-based equalizers have become a popular tool for audiophiles and enthusiasts to fine-tune their headphones' unique sound signatures.

Windows users have very limited options for system-wide parametric equalizers (pEQs), the most powerful of these tools. Peace is a free pEQ on Windows that averages 100K downloads per month, but has had the same slow, cluttered, and unintuitive interface for 9 years. With little to no competition, there is an opportunity to provide Windows users a modern, user-friendly alternative to Peace.

We have designed and developed a long-awaited alternative pEQ with an intuitive and responsive interface for Windows users. Our product, AQUA (Audio eQUAlizer), utilizes the same underlying hardware library as Peace for system-wide audio equalization. We have nearly achieved feature-parity with Peace, and have also implemented functions highly valued by the audiophile community such as auto equalization to achieve standardized sound signatures with over 4700 different headphone measurements.

AQUA's current interface is already more modern, responsive, and user-friendly than Peace. It has been received very enthusiastically by the community, with over 100 downloads of our latest release in 24 hours and 150+ individuals interacting with our GitHub, Reddit, and Discord posts. Many of the comments and replies to our posts have already stated that we are providing a better experience than Peace, and one that is more accessible to new users.

Having established our minimum viable product that works across all Windows versions, our next steps are to improve our app by prioritizing user feature requests as we develop new releases. We will continue to engage with audiophile and audio-enthusiast communities to promote new versions of our app and gather feedback to iteratively improve the user experience on AQUA.

https://github.com/h39s/AQUA#readme

11 Veloren: Chat and Procedurally Generated Structures

Team Peg.len()

Peggy Li, Ellen Sun

The critically acclaimed open-world game Breath of the Wild has sold over 27 million copies and set a new standard for its genre. Minecraft is the best-selling video game in history, with over 238 million copies sold and nearly 140 million monthly active players as of 2021. Players are insatiable when it comes to open-world RPG gaming experiences.

Veloren is a multiplayer voxel RPG written in Rust. It's inspired by games such as Cube World, Legend of Zelda: Breath of the Wild, Dwarf Fortress and Minecraft. Veloren strives to synthesize the strongest elements from its award-winning influences to create an immersive, novel, and exciting gameplay experience. It is fully open-source, currently in pre-alpha, and has builds for Windows, Linux, and MacOS. We have made multiple contributions to different areas of Veloren.

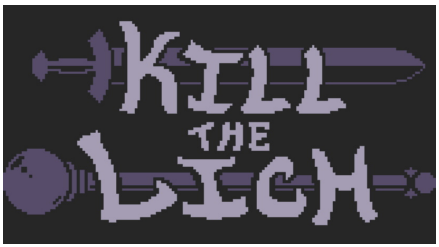
In multiplayer games, a robust chat feature is an absolute must. Within Veloren's chat system, players can DM other players, communicate with the world, and run dozens of commands (including the creation of groups, teleportation, and dropping explosives). We noticed that chat was missing some common features and submitted a feature which suggest similar commands when an invalid one is entered, implements /help command, cleans up code for server side and client side commands, and adds unit tests to the chat feature.

Many of Veloren's assets are directly imported, but the eventual goal is for Veloren's lush and beautiful world to be completely procedurally generated. Currently, some directly imported and generated assets appear repetitive or look unnatural which does not make for the best possible gameplay experience.

We also contributed the creation and spawning of procedurally generated realistic structures including natural arches, rauks, and hoodoo rocks. These are important characteristics of Veloren's biomes, which all have distinct geological features.

https://veloren.net

12 Kill the Lich: a multiplayer dungeon crawler
Team Rat Team Supreme



Ahmed El Shatshat, Eve Guo,
Brendan Engelman, Ishan Amlekar,
Nicholas Hachmer

The last few years has seen an explosion in the popularity of asymmetric multiplayer games. Games like Among Us and Phasmophobia have become viral sensations, driven in part by the pandemic motivating players to find games that are easy to pick up and play with friends.

Given this increased interest in this field, we seek to innovate on genres we feel have had some stagnation. Kill the Lich is a new multiplayer party game that combines the elements of popular online multiplayer games with the hectic fun of roguelikes and the strategy of tower defense games. By choosing these genres to be the basis of our design, we hope to bring a new take to the immensely popular roguelike genre and long-due innovation to tower defense, and compete in the online multiplayer party game space.

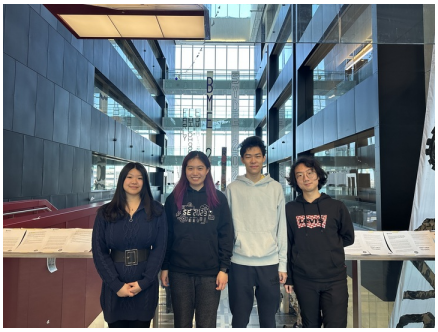
Three hero players attempt to progress through a series of rooms, competing against a Lich player who places enemies and traps to stop them. The heroes receive rewards for clearing rooms quickly, while the Lich receives rewards for stalling and killing heroes. This culminates in the final room with a fight-to-the-death between the heroes and Lich.

Compared to traditional tower defense games which feature slow, drawn-out gameplay, Kill the Lich is fast-paced and self-contained, ideal for quickly picking up and playing with friends. Despite its compact package, the game offers diverse gameplay choices to both teams, who must adapt their tactics and upgrades to respond to the other team. This helps the game stay fresh when played repeatedly, especially when compared to existing asymmetric multiplayer games which provide a more unchanging game-to-game experience.

Throughout the development cycle, we have held playtests to assess the game's state, gather feedback, and help prioritize development. Using this feedback, we fine tuned the game mechanics and systems to achieve a gameplay loop that is fun for all players. Player satisfaction reported from surveys, and metrics such as game download count and active user playtime, help support our claim of Kill the Lich being an enjoyable party game.

https://discord.gg/BjCqpEQXbR

13 OURvoxel: A 3D voxel game
Team Bonk



Isabel Zhang, Cindy Wang,
Jacky Liao, Cosmo Zhao

Most games these days use polygons and distinct objects that cannot be broken apart. This can be seen in many modern games where the Player may clip into a wall and see that the wall is actually just a hollow rectangular prism and does not have actual substance.

A voxel is a volumetric pixel or a three dimensional pixel. It is a pixel that has been given depth and position in three dimensional space. As opposed to every object being rendered by a series of polygons creating its surface and being hollow on the inside like in a traditional video game. A voxel in a game can be thought of as being similar to an atom in real life. Just as everything in real life is made up of a collection of atoms. Every object in a video game is made up of a collection of voxels filling its entire volume. For example, rendering flowing water would comprise of thousands of voxels all traveling in the same general direction.

This allows for a very different game play experience. In traditional games, a player can only interact with certain objects which the game developers have designated. In our voxel game, a player can interact with any object. This freedom is afforded by the fact that every object actually has volume as in real life. So if the player chooses to destroy a surface, they will still see all the voxels underneath and not a hollow void.

Polygons are the primary choice for video games as they take up less memory and the modern graphics rendering pipeline is optimized specifically for them. Using voxels will be much more inefficient in comparison. Our goal is to explore methods for making a well performing voxel game. We want to leverage hardware acceleration to help improve performance and to explore the possibility of using the GPU to perform the simulation of the movement of the voxels.

The game is both a technical demo and a sandbox, where the player is provided with a collection of different materials and can build anything they can imagine. It supports different material properties like fluids and gasses. The game also allows the user to create new material types through the user interface by specifying a set of properties about the material.

14 Adding OpenGL Pixel History to RenderDoc
Team OBJecTive

Ting Cai, Zi Ming He, Orson Baines, John Kattukudiyil, Tony Tascioglu

Whether you are new to graphics programming and trying to draw your first triangle in OpenGL or an experienced Vulkan programmer working on a complex pipeline, it can be extremely frustrating to figure out why your frame does not look the way it is supposed to. Much configuration is required and many components are working together to create a graphical program and even the most innocuous errors can result in a black screen, making it very difficult to pinpoint the cause of the bug. RenderDoc is a cross-platform open-source graphics debugger that helps developers identify and fix bugs in Vulkan, Direct3D and OpenGL programs. It offers a number of features for frame capture and detailed inspection of applications. It has a sizeable user base of around 6700 users per month.

Our team contributed to RenderDoc by implementing Pixel History for the OpenGL API. Previously, Pixel History was only supported for programs using Vulkan or D3D11. Several requests for this feature have been made by users, and this contribution was recommended to us by the primary maintainer of RenderDoc. Since OpenGL is the second-most used API on RenderDoc, being used by 15% of the user base, this new feature benefits many users.

The Pixel History view shows how the value of a pixel in a render target changes throughout the course of a frame. Each frame in a graphical application consists of many API calls that modify the render target incrementally. Pixel History focuses on a single pixel in a render target and shows the user which events in the frame modify that pixel and how they modify it. To implement Pixel History in OpenGL, we replay all of the draw calls in multiple passes to obtain information about how each draw call modified the value of the pixel.

We submitted the OpenGL Pixel History feature in incremental pull requests while keeping it hidden from end users using a feature flag. After completing the feature, we deleted the feature flag in PR #2865, making the feature available to all users by default. OpenGL Pixel History is now automatically enabled in the RenderDoc nightly builds, and will be part of the upcoming v1.26 release.

https://renderdoc.org/

15 Bezier-rs: A Bézier curve library for 2D graphics
Team HRTZ

Thomas Cheng, Linda Zheng,
Rob Nadal, Hannah Li, Jackie Chen

Graphite is an open-source web application written in Rust for editing 2D rasters and vectors. Many of Graphite's planned features require a significant number of Bézier curve manipulations, and though libraries exist in other languages that provide this functionality (such as bezier.js), there are no comparable Rust counterparts. Kurbo, an existing Rust library for simple 2D geometry, is missing several key functions critical to Graphite's development plan, such as computing the outline of a Bézier. In addition, both libraries use lists of Béziers to represent paths, while Graphite represents such data using their custom data structure: `ManipulatorGroup`.

We are creating a standalone open-source Rust crate. This library defines concise and efficient data structures to represent Bézier curves, as well as paths composed of Bézier curves. The library further provides computational geometry algorithms useful in the context of 2D graphics, such as functions to compute path intersections, bounding boxes, and outlines.

With Graphite being our main users, creating a custom Rust crate allows us to collaborate with the Graphite team and make design decisions that easily integrate with their codebase. Moreover, we've designed our data structure to store the curve data as points rather than as a segment, which is idiomatic to how beziers in SVG paths are stored and processed. We further prioritized the developer usability of our crate by creating a page of interactive function demos, which we have integrated with the comprehensive RustDocs we created alongside our crate. These demos have already been proven useful when doing code reviews as the Graphite creator said "it's a very effective way to understand [the functions] compared to just reading the function signature and comments".

https://crates.io/crates/bezier-rs

16 Procedural Generation Toolkit
Team Pacific Time

Tony Jung, Danny Ou,
Sherwin Varkiani, Kevin Li,
Yiming Tan, Haydn Keung

Procedural generation is an industry-leading method of algorithmically generating game assets that have been applied by critically acclaimed game studios since the 1970s. Several award-winning games such as Minecraft and No Man's Sky ascribe a substantial portion of their success to procedural generation.

However, advanced procedural generation algorithms are mathematically complex. Around 31% of all game developers do not have a bachelor's degree, which may result in them lacking the mathematical background required to implement advanced procedural generation algorithms.

We aspire to introduce a solution that lowers the barrier of entry to advanced procedural generation algorithms. We propose a solution that abstracts the implementation details of mathematically complex algorithms with an intuitive interface. We propose a toolkit of abstracted building blocks for procedurally generating game content. Most notably, this toolkit will consist of a novel way of visualizing the recursive nature of formal grammar rules, coherent encapsulations of computer algorithms and statistical models, and configurable 3D terrain generation support.

Aspiring game developers that use Unity as their game engine of choice are our targeted audience. Unity has ever-evolving APIs that we use to accomplish our objective of abstracting away the complexities of the algorithms.

While alternatives exist, our solution introduces support for handpicked advanced algorithms, such as formal grammars and terrain generation, that are not supported in existing solutions. Our solution is also free of charge, making it accessible for novice developers. Metrics such as number of views, number of downloads, and user feedback for our toolkit will be collected through Unity's asset store and will serve as the primary indicators of our toolkit's success.

17 Mirage: A Lightweight Hi-Fi Honeypot Network
Team Mirage

Jin Yang (Jason) Liu

Cyberattacks are becoming a common occurance for many companies, many that are impacted may have little to nothing to do with technology and as such rely on technological solutions to defend against these events. The demand for proper defenses have grown and the cybersecurity market as a whole is expected to grow to 262 billion dollars in 2027 from 143 billion in 2023. Network security solutions continue to be in demand due to the rise in remote work and the continued risk of compromised or lost credentials.

The users of network security solutions aim to detect and disrupt attackers on the network. For a honeypot, this largely plays into the role of detection by alerting the user when an attacker attempts to interact with a honeypot. In addition, honeypots can also disrupt an attacker as they are used to imitate existing devices on the network, forcing an attacker into a choice where they would either have to attack all the devices and risk raising an alert or only attack a subset which disrupts their ability to cause a damage on the network.

Decoy systems such as honeypots largely suffer from a tradeoff between realism or fidelity and scale. Systems that scale well are largely constrained to providing only a thin facade to attackers and systems that replciate their targets to a high degree of fidelity require more powerful hardware and will result in difficulty in scaling. Mirage aims to provide an alternative to this tradeoff by leveraging the economy of scale for server hardware to run a series of high fidelity virtual machines. In addition, by introducing a lightweight and cheap hardware proxy, this allows for rapid deployment of decoy devices on networks with minimal changes to the actual network layer.

Initial deployment of Mirage in a controlled environment shows the feasibility of this design. However, a 4 month deployment demonstrates the stability of the system as the proxy did not experience any disconnects or instability. Experiments were done to simulate an attacker attempting to detect a Mirage instance, configuration of the hardware proxy to return any ICMP traffic allows for the system to appear indistinguishable from both inspection of a traceroute command as well as inspection of ping round trip times. In addition user validation was conducted with discussions with the lab manager of the ESG lab where Mirage is currently deployed. It was noted in the discussions that Mirage was quite easy to administer as all the routing logic is defined in software on a layer that overlays the network meaning there is no disruption caused by changes in the deployment.

18 Beam: A Secure File Transfer App

Team Beam

Damian Reiter, Marcel O'Neil,
Michael Jiang, Mufeez Amjad

Imagine a file transfer app with the simplicity of Airdrop, the convenience of cloud storage, and the security of state-of-the-art encryption. The best current solutions often suffer from limited platform support, slow multi-step workflows, leave file artifacts in the cloud, or treat security and privacy as an afterthought. Beam aims to offer a fast, end-to-end encrypted one-click transfer of files between devices, without cluttering cloud storage or inboxes.

Over the last few years, the issue of how companies collect and manage private data of the general public has been a growing concern. A survey conducted in 2019 found that "Roughly eight-in-ten or more U.S. adults say they have very little or no control over the data that government (84%) or companies (81%) collect about them." We believe that this concern will drive the design of the software applications of the future, and that there is a market for a secure and private file transfer application. Through the use of modern cryptographic protocols, Beam aims to provide provably secure and private data transfer services to users without compromising on convenience.

Our target user base includes people who are privacy-conscious and work with digital files every day. This can include professionals who need to transfer sensitive documents between devices, and students who often find themselves transferring school-related documents. These people want an easy, secure, and fast means of transferring their documents and streamlining their workflows.

The existing solutions can be a source of pain for users due to their lack of security, privacy, and tendency to leave file artifacts behind. Additionally, some solutions like AirDrop are platform-locked, making it inconvenient for users. Our solution is to develop a secure, ephemeral, and cross-platform file sharing app that provides a suitable alternative to the existing solutions, effectively addressing the pain points of the best current solutions.

Our team has developed a functional desktop and mobile application, which are supported by a deployed production server. We are currently engaged in an iterative process to polish and refine the product, with the aim of enhancing its overall quality.

19 Enabling engineers to be productive on day 1
Team Aloha

Technology companies of all sizes spend thousands of hours every year onboarding new engineers to their technical stacks and enterprise tools. Onboarding is a repetitive process and drains valuable engineering time that could be otherwise used to deliver business value.

Shahbaz Momi, Shehryar Assad, Advitya Chhabra, Rafit Jamil

Scaling technical onboarding is expensive and as a result is often solved with setup scripts and recorded video calls which are difficult to keep up to date as a company scales. Outdated setup scripts lead to incorrect developer setups which results in frustration and counterproductive engineering efforts. Managers are additionally tasked with the lengthy process of giving new engineers access to applications, documentation and other resources.

Aloha aims to automate and simplify developer environment management for both managers and engineers. For managers, Aloha provides a central platform to design, monitor and manage all tools required for an engineer's developer environment. For developers, Aloha completely automates the initial setup so there is no need to manually download, configure, and install any tool, framework, script or application.

Aloha offers a web and desktop platform that is pre-installed on the devices of managers and their engineers. Aloha has permissions to manage codebase environments and software libraries. Moreover, Aloha allows managers to configure and mutate developer environments remotely.

Aloha's primary differentiator is that it is a no-code tool that requires little technical expertise or orchestration to start. Our device installation is also bare-metal which provides added security and performance compared to existing container-based solutions. We also support configuration of low -level tools to enterprise applications and everything in between. This allows us to provide additional functionality such as subscription-managed software auditing, audit trails in the event of a security issue, and more.

https://tryaloha.io/

20 Restocked: a notification system for consumers
Team JVM

Jin Huang, Vikram Subramanian, Michael Qin

Online retail sales worldwide are projected to reach $6.5 trillion in 2023, with 22.3% of retail sales being taken up by ecommerce websites. People are increasingly buying things online, creating a huge market opportunity for our application.

A frustration for shoppers is finding out that a desired product is out of stock. Other times, one might be anticipating the purchase of a new product, only to find that it immediately sells out. To secure a desired item, avid shoppers might frequently check product pages, or for new products, wait on a webpage, constantly refreshing or joining a queue for the chance to buy it. Restocked is a website-agnostic system for consumers to be notified when a product comes back in stock.

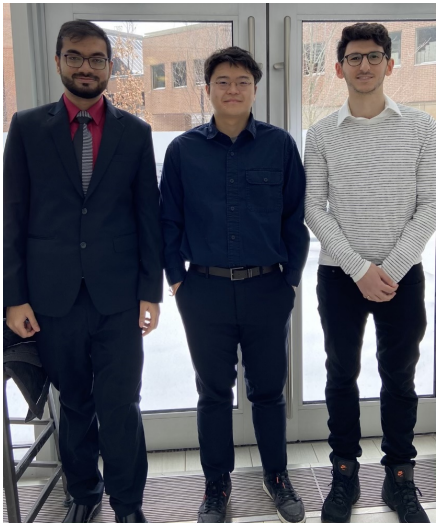
The current version of Restocked allows users to sign up easily, then input a link to the page of the product they want to buy. Then, Restocked monitors that page for any changes to the user-selected parts of the page, notifying users through text when one is detected so they can purchase the item as soon as possible. We are in the process of adding functionality for users to select specific attributes of product pages they want to detect changes for, such as to only send notifications when a specific size or colour of a product is updated, thus making the application usable on any website.

Key design challenges are creating a streamlined, robust user interface, scaling Restocked to handle a large number of products to monitor and handling rate limits, and sending notifications in a timely manner – if notifications are sent too slowly, then there is no use for the application. Claimed advantages for Restocked are being website-agnostic and better marketing than competitors.

We have performed user interviews which show interest in the product. We plan on doing more marketing towards a wider audience and will measure success of the application by the number of users, number of websites being monitored, and frequency of use. Currently, Restocked works on Amazon and Nike products, and we are working to provide a more customized experience to allow users to select specific elements they want to monitor on even more websites.

https://restocked.info

21 Flash: Modernized Flashcards
Team Flash



SK Sadman Sayeed, Seong Min Park, Eric Weitzman

Students are required to learn vast amounts of information concurrently for multiple courses. For instance, at UWaterloo, students are expected to take five courses per term, with all the content needing to be learned in just over three months. Regrettably, a significant part of learning still involves memorization. Flashcards are a timeless tool used by students to aid in memorization, which have now transitioned into the online realm.

The Flash app aims to enhance the user experience and efficiency of flashcard-based memorization through the use of spaced repetition, collaboration, friendly competition and AI generated cards. This creates a collaborative environment that encourages students to learn together. Our design approach consists of spaced repetition, shared flashcard decks, a comparison method between users and an integration with OpenAI's chatGPT API.

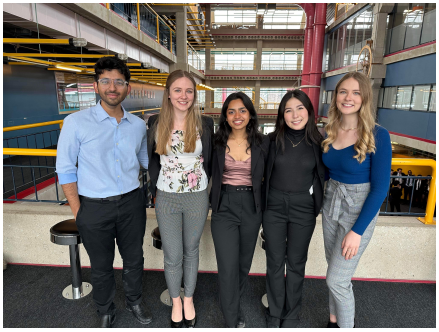
Spaced repetition has been demonstrated to improve long-term recall of large amounts of information. Several studies on the effectiveness of spaced repetition have been conducted and conclude that it positively affects long-term retention and efficient learning. A few examples of such studies are Rawson & Kintsch in 2005 and Toppino & Gerbier in 2015, both of which studied spaced learning and testing in long-term retention.

The primary advantage of our design is providing users with an accessible, effective, and free way to learn through digital flashcards. Existing alternatives are either paid services, freemium models cluttered with advertisements, lack the combination of features offered by Flash or do not have AI generated flashcards.

Since Flash is user-facing, its success metrics can only be determined by the users. We will conduct user surveys and track activity metrics to validate our results.

https://flashapp.ca/

22 Kid Investor: A Stock Market Game for Kids Team Runtime Terror



Akshay Saxena, Kara Dietz, Chandana Sathish, Sara Hosseini, Angela Dietz

Build a Biz Kids is a non-profit organization which looks to equip kids with the skills and confidence needed to enter the "real-world". Many children grow up without proper exposure to finance management or investing knowledge. The client has identified this gap and has expanded their existing programs to fill it in. The team built an educational finance/investing game for this organization which will be promoted to children who participate in their existing programs, as well as through a curriculum in classrooms by licensing it out to schools. Build a Biz already had a bare-bones application which replicates investing stocks in the real-world, however it was not in a production-ready state, was difficult to navigate, and was likely to have low retention.

There were two ways that we worked to improve this app for Build a Biz. First, we have worked with Build a Biz to make the app easier to use, more informative, and more exciting for kids. Second, we designed and implemented a dashboard for teachers to use which aggregates data from their students' accounts, allowing for classmates to learn about the stock market collaboratively while giving the teachers the opportunity to oversee the students' progress. The game is an online web application so that it can be accessed from any device, and it will be licensed out to schools and teachers across Canada.

The expected impact is that kids will have a better experience learning about finances, while gaining skills required to excel in the real-world. It will also help Build a Biz provide more programs to support children. To prove that the objective has been accomplished, we have maintained a fast communication stream with stakeholders from Build a Biz and can conduct trials with the application with groups of kids to determine areas of improvement.

23 CapstoneGatsby: New and Improved SE Capstone Project Website
Team SHSN

Hussain Abdi, Sterling Finn, Nicholas Johnston, Chen Chai

The Software Engineering fourth year design project is the culmination of an undergraduate degree that involves working in a group on a self-selected project over the course of a student's fourth year. Relevant resources are stored on a website that students frequently visit to access the course handbook, research past projects, and navigate to other relevant links for the course.

The existing SE Capstone website organizes past student project by category and year, but video links and other information are scattered, and search capabilities are limited. Capstone Gatsby improves on this by adding search, and combining project data with video links in one place.

Instead of copying and pasting links and trying to cross reference different sites to find information on past capstone projects, Capstone Gatsby will allow students to be able to click on a project link and see all associated information, including the project's video, abstract, themes, and related documents.

Future SE390 students will be able to use the website to find examples of past projects and view presentation videos using a searchable, clean user interface. New projects will continue to be added using gitlab commits, but will be converted into web pages using an easy-to-run script.

Capstone Gatsby is developed in collaboration with SE Director Derek Rayside. The website was deployed for the SE 2024 SE 390 class via email to the SE 390 professor. The website is in the process of being deployed to the ECE server for future SE students to access.

24 Akalbook: A Booking Platform for Gurdwaras

Team Khalsa Tech

Saman Singh Sandhu

Punjabi Sikh communities in first-world countries do not enjoy the advantages of technology. This is often the case with most individuals that emigrate from third-world countries. As a first-generation Canadian, I see immense opportunity for the Punjabi Sikh community to upgrade the tech infrastructure that is used within the tasks they lead.

For example, all program bookings at a Gurdwara (Sikh Place of Worship) are done manually using ink and paper. Individuals interested in booking a program must contact the Gurdwara via phone or visit in person to determine the best available date.

These phone calls and visits waste a lot of time since individuals don't have access to the schedule containing the existing program bookings. This is especially the case for couples that must contact or visit numerous nearby Gurdwaras to find a venue that aligns with their preference and timeline for their wedding. Further, the designated committee member, Gurdwara Secretary, like other members is a volunteer with a full-time job and family, so they may not be available to attend phone calls at most times and are typically only present at the Gurdwara on weekend mornings and afternoons. Hence, this service is not available 24/7, and first come, first serve is not always respected.

All in all, Gurdwara Secretaries wish to minimize the time they spend managing program booking requests and responding to user inquiries so they can focus on other tasks. And community members wish for a more transparent, fair and efficient way to book and check programs at their local Gurdwaras.

We are currently working with the Guelph and Kitchener Gurdwaras to publish a mobile app for program bookings that serves both locations. The Gurdwara Secretaries will be empowered to enter bookings for community members who prefer the traditional ways.

https://khalsatech.xyz/

25 Halo: A Street Harassment Prevention App

Team Zephyr

Han Hao, Emily Tao, Mira Yadav

From walking home alone at night to entering a secluded parking garage, seemingly normal activities are associated with stress for many individuals due to street harassment. A survey taken in 2014 found that 65% of all women in Canada have experienced street harassment at least once and 20% have been followed throughout their lifetime. Despite street harassment being a significant problem in today's society, there are currently very few widespread and effective technological tools to help potential victims. We want to change this with our mobile app, Halo.

To deter potential attackers in these situations, individuals commonly pretend to talk to a family member or friend on the phone. This helps to convince potential assailants that the individual is able to ask for help immediately and/or has someone monitoring their wellbeing. Fake safety phone call audios went viral on TikTok for this reason. If the situation escalates, people also want to contact help by sharing their location with trusted contacts and/or emergency services as quickly as possible.

There exist apps that help individuals plan safer travel routes like 'Hollaback!' and 'WalkSafe'. However, for many people it is impossible to avoid certain areas out of necessity, especially in denser cities. These users need a tool that potentially allows them to reduce the risk of being attacked and, in the case that there is imminent danger, contact help quickly. Our app, Halo, gives users the ability to quickly initiate a realistic, fake phone call with an AI voice bot from their phone's lock screen. During the call, the app will listen for pre-programmed safety keywords and, if detected, will contact emergency services or share the user's location with trusted contacts. In this way, users are given a subtle and simple method to ask for help.

https://play.google.com/store/apps/details?id=com.zephyr.halo

26 Reducing Carbon Emissions Through Data Team Power

Elisa Luan, Mei Yi Niu, Cynthia Ding, Julia Du, Nathan Au

At the 2015 UN Climate Change Conference, worldwide political leaders committed to the Paris Agreement: a pledge to reduce global greenhouse emissions and limit the global temperature increase to 2°C this century. This transition to a low-carbon electricity system will require every organization to reduce their energy footprint by making informed decisions drawn from real-world data. ElectricityMaps is a leading resource in providing actionable data that quantifies historical, real-time, and forecasted electricity data. Major clients like Google leverage ElectricityMaps to reduce their climate impact by shifting computationally heavy tasks in their hyperscale data centers to times when low-carbon power sources, like wind and solar, are most plentiful.

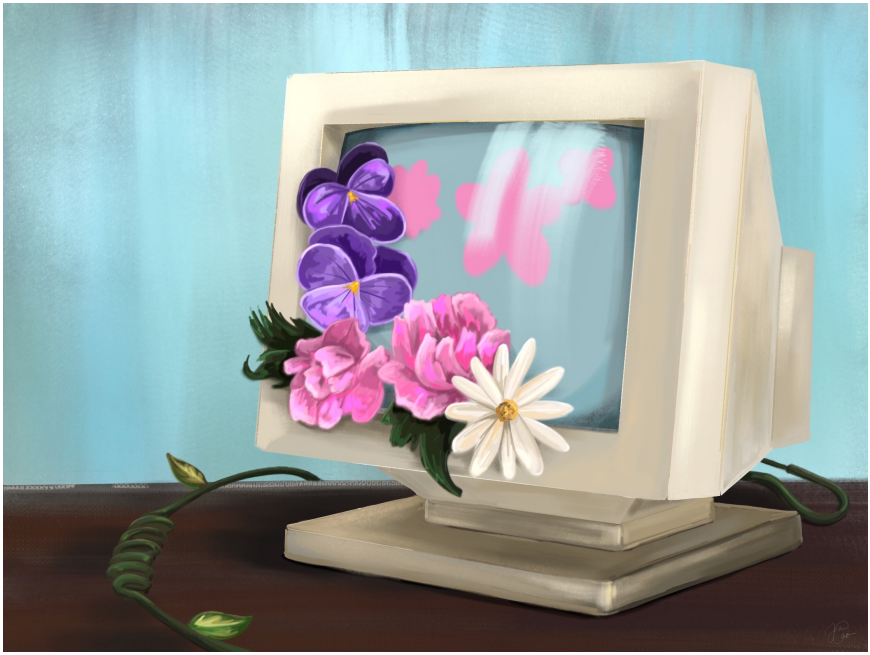
As part of the Paris Agreement, Canada has pledged to reduce its greenhouse gas emissions by 40–45% by 2030 as compared to levels in 2005. However, Canadian data is notably sparse in ElectricityMaps, despite repeated requests and attempts to populate this data, making it difficult for researchers and concerned parties to monitor and analyze progress towards this commitment. Team Power plans to address this need by working with research engineers at Natural Resources and Statistics Canada to expose the data that the organization has collected via an API and make it easily accessible. This will produce a valuable resource that can serve as the groundwork to populating Canadian data in ElectricityMaps. Filling in these gaps in data completeness will benefit users globally as a tool for analysis, comparison, and education.

To date, Team Power has merged multiple PRs to the ElectricityMaps project to make data available for other areas, namely the French regions and Argentina. This work established Team Power as a reliable partner for ElectricityMaps, in preparation for the larger challenge of integrating Canadian data. Moving forward, Team Power will continue to work with both Statistics Canada and ElectricityMaps to provide actionable Canadian emissions data and empower smart decisions for electricity usage for the future.

https://www.electricitymaps.com

Cover art by Tina Gao

Student Index



In memory of Grace Chi Hung Leung 1932–2020

UNIVERSITY OF
WATERLOO

FACULTY OF MATHEMATICS
DAVID R. CHERITON SCHOOL OF COMPUTER SCIENCE

FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING