TEAM SQUAAD

Table of Contents

Team Introduction

Team Photo

Team Biography

We are team Squaad, a team of highly motivated and ambitious computer science students possessing a wide array of experiences, skills, and interests. We aim to improve ourselves both individually and as a team, and hope to contribute to an open source project to enrich our development experience as well as our portfolios for future employers. Every individual on our team is fluent in English, proficient in coding, and has many combined years of Python experience under their belt - we have confidence and pride in our work, and hope that our contribution will help the Matplotlib community and users.

Together as a team we are synergetic, team oriented, and adaptive, and we aim to make make new friends, expand our network throughout the progression of this semester together. Not only that, but we aim to help each other improve, heightening both our technical skills (coding, testing, to documenting) and our interpersonal skills (communication, presentation, organization). Both as a team and as individuals, we strive to work with professionally with integrity, honesty, and most of all, with a smile on our faces while we enjoy the experience!

Team Biographies

Qi Cui (GitHub: QiCuiHub)

My name is Qi Cui. Building software has always been something I enjoyed, and as such I have a deep interest in all aspects of the computing field. I have explored computing through numerous personal projects ranging from 3D computer graphics and machine learning, to reverse engineering and web/game development.

I have over eight years of programming experience and I am fluent in the languages C, Java, Python, and JavaScript. In addition to this, I am an expert at debugging code and implementing core concepts such as object oriented programming and unit testing. Beyond the software side, I also have interest in the electronics engineering component of computer science. My work with embedded systems have allowed me to understand how computers function on a fundamental level.

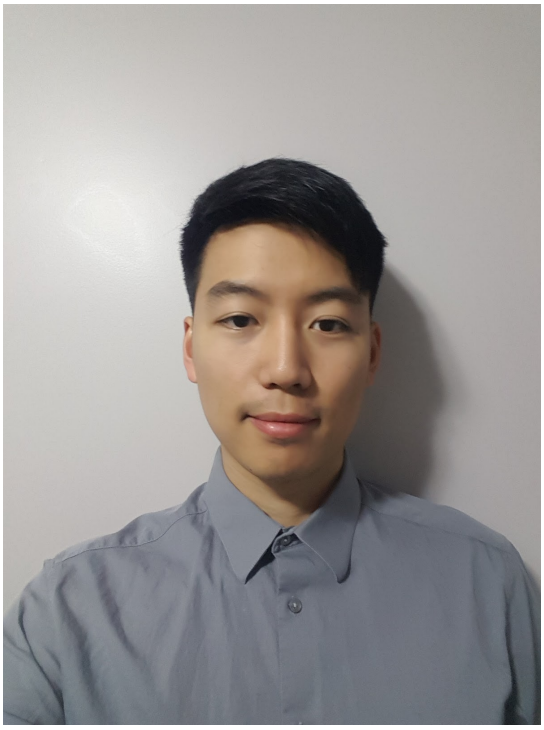
My education background is that of a Computer Science student; I am currently in my fourth year of study at the University of Toronto. I worked for 8 months as an applications developer for the Government of Canada, where I gained experience on how to work well with a team and even a little on how to manage projects.

Amy Huang (GitHub: yunjinh)

My name is Amy Huang. I am a fourth year Computer Science student who has an inquisitive nature with an interest in learning about technology. Through academic experience, I obtained high proficiency in Java, Python, C, and UNIX, as well as mobile application development. Moreover, I am also experienced in working with Photoshop, Flash, Corel Draw, Git, CMVC, and db2 through personal interest and internship opportunities.

In 2014, I had an eight month internship as an Automated Test Developer at CaseWare International, for which I learned the concepts of Page Objects, and using them to automate GUI interactions with desktop applications using Java and SilkTest. In 2015, I worked at IBM as a test developer in the WebSphere Commerce department for sixteen months in a fast paced, Agile environment. I was a part of the Functional Verification Test team in charge of the execution of functional, regression, accessibility, and localization test cases, as well as taking part in test planning, manual testing, and test automation using Java and Selenium for the Commerce Insights and Customer Service Representative components. In addition, I took part in customer review meetings to present and demonstrate new functionalities. Furthermore, I was also a member of the Customization Verification Test team for the Commerce On Cloud component; mainly involved in preparing, deploying, and testing customization assets, and installation testing.

Joosub Lee (GitHub: Leejoosub)



My name is Joosub Lee and I am enrolled at the University of Toronto in the computer science program. I first started coding in high school, and through my academic career, I have experienced programming in languages such as Java, Python, C, assembly, MySQL, and Android Studio. I've worked on projects such as creating simple games, phone applications, encryption/decryptions programs, and other applications for school, and have worked in group settings. Aside from school, I also worked as an intern at Real Matters Inc. where I learned about how to work with databases and developed tests for their code, as well as suggest fixes for their bugs. Aside from computer science, I am also in school for economics, to understand a part of the decision-making process a business does, as well as media studies, to analyze why people consume the media they do.

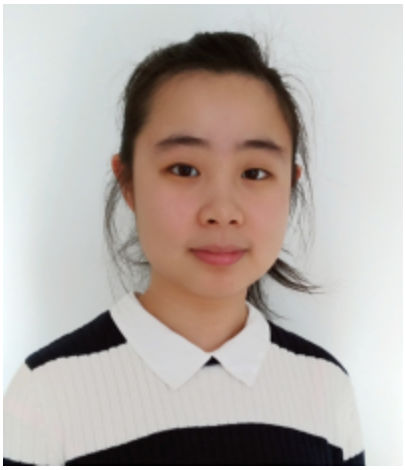
Benson Tang (GitHub: btang02)

My name is Benson Tang. A fourth-year computer science student in the software engineering specialist stream (Non-coop) at UTSC. Experienced in Python, Java, C, Assembly, Git, Unix, Shell, Bash, Android studio, database [Relation Algebra (RA), SQL], web programming [HTML, JavaScript, CSS, react, node.js] through academic experiences; I.e. Computer Science courses. In those courses, projects have been worked on such as creating android app (backend, and frontend), making system calls for kernels (operating systems), making websites (full stack, MVC design pattern, REST), and etc.

Also, have done organizing, delegating, and solving times between workers when there existed a conflict, as well as having worked a long shift of ~16 hours; average being around 13 hours in a single day (Part time job). Finally, have some experience in writing a formal technical lab reports; i.e. Technical documents, for science (Biology, Physics, Chemistry) through academic experiences (University level).

Interests include web development, game development, software engineering, building computers, biology (Human physiology and anatomy, evolution, and genetics), martial arts, fitness, and bowling.

Eugenia Zhang (GitHub: zhangeugenia)



My name is Eugenia, and I am a fourth year Computer Science co-op student at the University of Toronto. I have previously worked at TELUS and the Ontario Ministry of Education for my co-op work terms.

At TELUS, I worked as a Technology Architect Assistant in the Enterprise Architecture department and polished my documentation, SQL, and presentation skills, helping create a synchronization process of company assets between a database and an internal dashboard. I also created and updated many charts and visuals relating to the company's growth and performance for an internal dashboard used by company directors.

Working as an Application Programmer at the Ministry of Education, I helped develop and launch the first version of the Transfer Payment Common Registration system, a registration system for all grant applicants working with Grants Ontario. During my time as an Application Programmer, I made use of my Web Development skills, learned techniques to make an application accessible all users and AODA compliant, and also helped with requirement gathering sessions with ServiceOntario. Not only that, but I also attended training for automation testing, and learned how to create scripts which are able to test applications with a GUI interface.

Outside of the classroom, I enjoy drawing and occasionally work as a freelance character artist. I also enjoy drinking tea, and I hope to be able to sample as many different kinds of tea as possible.

Nathan Sim (GitHub: natesgithub)

My name is Nathan Sim. I am a fourth-year computer science student at the University of Toronto Scarborough campus. During my enrollment time here, I have obtained much experience coding and studying in the languages of Python, Java, C, and Assembly, (with much more versatility and proficiency in Java development due high school exposure). During my own time, I've also learned to develop using the language of C#; working with a C# based online game and its servers, employing a good understanding of C# server-side programming and client updates/management. I also have knowledge in the fundamentals of cryptography and system and network security. My primary interests within this field of study include securities, operating systems, web development, and human-computer interaction. In addition to computer science, I am also taking classes in financial economics to better understand the business/marketing world, and music development to further hone my skills as a hobbyist.

Chengli Yang (GitHub: chengliyang)

My name is Chengli Yang and I am currently a third year undergraduate student studying at the University of Toronto Scarborough campus majoring in Computer Science and taking minors in Statistics and Economics. In the field of computer programming, I am proficient in the languages of Python, Java, and C.

My work experiences include working for the University of Toronto Scarborough Athletics and Recreation department as a swim instructor and aquatics program monitor, which involves overseeing and assisting students in the university ran aquatic programs including learn to swim classes, Inner tube water polo, and water volleyball. He is the president of both the UTSC Swim Club and University of Toronto Scarborough Underwater Club, the latter of which is engaged in the sport of underwater rugby, which is a three-way cross between basketball, rugby, and scuba diving.

Nathan Jaremko (GitHub: njaremko)

My name is Nathan Jaremko and I'm a fourth year Computer Science student. I've enjoyed solving puzzles since I was very young, as I got older I discovered computers and became fascinated by the problems they could solve. When I was 8 years old, I started using linux and programming in C to try and develop a deeper understanding of how computers worked. Over the years I've spent thousands of hours devouring computer science literature, and I'm constantly researching the latest trends. I've worked professionally as a full stack developer at PointClickCare for over a year.

The majority of my programming experience has been in systems languages (C, C++, Rust), Lisp, Java, and Go. Since starting school I've written a relatively small amount of Python, but I am proficient. While working at PointClickCare as a full stack developer I've solidified my knowledge of computer security, optimization, relational and non-relational databases, web development, user interface and API design, testing, and understanding customer/business needs. I have experience with the Agile development process and have led many sprint reviews. In my spare time, I work on open source hobby projects including a command line podcast manager, software to control hue lighting systems, and a concurrent ray tracer.

Definition Of Done (DoD)

For each contribution/commit, we define it as done when it has:

- Completed patch/fix/implementation of the feature/bug
- Cleaned code/directories such that irrelevant temporary/debugging items are moved or removed
- Completed documentation according to the style guide of Matplotlib (?)
    - Docstrings are present, and internal comments are present when needed
- Been unit tested by at least 2 members of the team and passes all unit tests (i.e. no more bugs are found)

For each deliverable, we define it as done when it has:

- All the needed contributions (documents, code, etc.) making up the deliverable, and all contributions are done (see above for definition of done for contributions)
- A general team consensus that the contributions making up the deliverable are up to the standards and expectation of the team and project owner
- Confirmation that the deliverable has been proofread and/or documented by all members of the team, and the confirmation that each team member is satisfied with submitting the associated contributions
- Been submitted to the correct location (Git, Emailed, etc.)

TEAM SQUAAD Working Agreement

Methods of Communication:

- Official method of communication is:
  - Slack
- Optional methods of communication are:
  - Discord
  - Facebook

Communication Response Time:

- Must respond/reply on Slack (or on the method of communication which message was received, if optional communication method) within:
  - 1 hour
- Preferably respond/reply within:
  - 30 mins

Running Meetings:

- In-person weekly meetings
  - Mondays from 6:00PM to 7:00PM
  - Held in IC406 or online
  - If someone is late, then they must bring Timbits; unless Timbits are the reason they are late. In that case, **sharing is caring**.
- Daily stand-up meetings:
  - Online as needed
  - Online meeting held at 6:00 PM to 6:15 PM (maximum 15 minutes)

Meeting Attendance:

- Mandatory, unless an emergency occurs
  - Must give reasonable explanation for missing a meeting
- If meeting is/will be missed, must update team members through Slack

Version Control:

- Using GitHub
- Each bugfix and feature must have its own branch(es)
  - When bugfix or feature is completed, branches with corresponding bugfix or completed feature will be merged with master branch
  - Testing for each bugfix or feature will occur on the branch in which the bugfix or feature was implemented in.
- When merging branches to master, person must get the approval of all team members.
- No silly log messages, each committed message should be contain clear and concise explanation on what has changed, or been implemented

Submitting Work:

- Everyone will submit their own, or partnered work on tasks they were assigned to for each sprint
    - During each sprint, each team member is expected to have put in about 8 hours of work per week
- All members will familiarize themselves with the style guide of Matplotlib, as well as review the general good practices for coding, testing, and documentation
    - Submitted work will be reviewed by everyone during the weekly meetings, in order to provide feedback for improvement for future sprints
    - Members will sit in a circle around each with their own copy of the code being reviewed, and will each provide feedback to the code/docs in terms of its efficiency, clarity, and how Pythonic it is (as well as how to improve it)
    - Together, the team will improve the submission as both review and a learning experience
- Pull requests submitted to Mabplot must be approved by team members in terms of coding, testing, and documentation

Procrastination Resolution:

- If a member of the team chooses to be unproductive (be it slacking off, or not getting help when roadblock persists in hindering their progress) and the majority of team members agree that said member is unproductive, then:
    - The team will have a meeting discussing the problem, giving a warning at first (or make them get Timbits)
- If problem continues, depending on severity (amount of work undone, and how long this has gone on for) of unproductivity of the team member, then:
    - The team will come up with a creative and/or fitting punishment during a second intervention meeting for the team member
- If there is a reason for not helping/contributing to the team, a **valid** reason must be given and accepted by all members of the team

Conflict Resolution:

- A conflict is identified by:
    - Two or more parties (each party consisting of one or more team members) having a non-productive disagreement with each other lasting over 12 hours (i.e. the views/stances of the opposing party act as a roadblock)
    - Having parties involved in the conflict who are unable to agree on a set solution which will clear the roadblock produced from the conflict between parties

Conflict Resolution (continued):

- Should there be a conflict, involved parties shall:
    - Inform all uninvolved group members of the conflict, as well as provide their stance on the subject in private
        - Stance includes their views, as well as how they hope the conflict will be resolved in a manner which will benefit the group
    - Listen to the feedback and suggestions of the uninvolved members
    - Hold a group meeting between the conflicting parties and uninvolved parties, in an attempt to mediate the situation
- Should holding a group meeting fail and/or the conflict has lasted longer than 36 hours:
    - Parties involved in the conflict will create a text document containing:
        - Their views and feelings
        - What roadblocks are stopping them from feeling like the conflict is resolved
        - What they suggest should happen
    - And present their case to a uninvolved party in a debate/trial
    - The uninvolved party will then settle on a solution, be it favourable to either conflicting party or not
    - All parties are forced to oblige with the solution in a professional manner
- All parties must follow the verdict solution from the trial for at least 24 hours
    - After 24 hours, all parties involved with the trial shall reconvene and discuss any further steps, or if there are any ways to modify the solution such that the conflict works towards being resolved
    - If after 48 hours and two meetings the case is still unresolved, the team shall escalate it to the course staff
- Should the situation become unsafe/life threatening due to a conflict, the course staff must and will be notified

Contingency Plan:

- In the case that a group member leaves the group, the team shall:
    - Reduce expectations and features
    - Redistribute responsibilities
- Team members must get together to determine what other actions are needed

Tools/Standards:

- Git/Github:
    - Version control
- Trello:
    - Retrospective meetings
    - Task assignments
    - Organizing lists of incomplete/completed tasks

Tools/Standards (continued):

- Slack:
  - Main method of communication
  - Optional meeting location if necessary
  - Updates on tasks such as roadblocks, progress, completion of task, or help if required
  - Approval requests for merging branches to master

Name	Signature
Qi Cui	Qi Cui
Benson Tang	
Joosub Lee	
Nathan Sim	
Eugenia Zhang (Yujia)	
Yunjin (Amy) Huang	
Nathan Jaremko	
Chengji Yang	