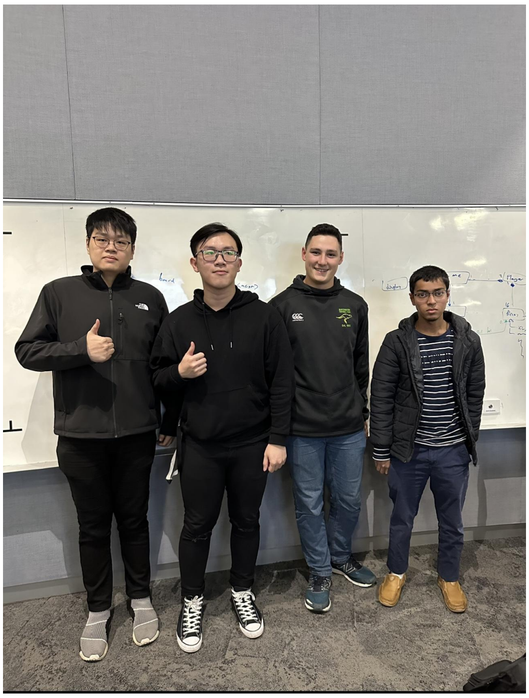
Awesome Games Group (AGG)



Team Members

Name	Email	Discord Handle	Strengths	Fun Fact
Huynh Nguyen	hngu0112@student.monash.edu	Nera#4749	Fullstack, confident with using Java	Own a part of Tesla shares
Luke Phillips	lphi0011@student.monash.edu	Lionsluke#4181	UI/UX design	Won a B&F in Footy
Yu Xuan Yio	yyio0001@student.monash.edu	roasty #6664	Backend, Usability in mobile applications.	Survived cancer in 2017
Zareef Masud	zmas0005@student.monash.edu	Zareef#9118	Design	Used to play cricket

Team Schedule

Team Meeting Availability

	MON	TUE	WED	THU	FRI	SAT	SUN
Huynh Nguyen	NONE	AFTER 4PM	AFTER 6PM	BEFORE 2PM	WHOLE DAY	AFTER 4PM	AFTER 4PM
Luke Phillips	BEFORE 4PM	NONE	AFTER 4PM	AFTER 10AM	NONE	NONE	AFTER 7PM
Yu Xuan Yio	Before 2pm & After 8pm	After 10am	After 5pm	None	Before 12pm	Anytime (Notify 1 day in advance)	Anytime (Notify 1 day in advance)
Zareef Masud	Null	3pm-6pm	5pm-6pm	4pm-6pm	3pm-6pm	3-pm-6pm	3pm-6pm

Team Scheduled Weekly Meeting

After a team discussion, we came to a consensus that Friday 8 am would be the best time/day for us to meet as a team.

However, this scheduled meeting is subject to change at any time. The change in the team's scheduled weekly meeting may be because of several reasons. For example, A team member falling ill. In this case, the scheduled meeting will be postponed unless the content of the meeting is of urgent concern. This is why we listed our availability throughout the week above, in case of a need to reschedule.

Communication

The team will communicate primarily through Discord. Instant messaging is preferred as it is 'asynchronous' and all members don't need to be present at the same time for communication.

The secondary communication method would be via email.

Work Distribution

Work distribution will be done during our scheduled weekly meetings or before/after workshops. If a team member has a certain preference for the work they want to do, they must voice out so that the piece of work can be their main responsibility. If more than one member wishes to work on a certain piece of work, they can work together. However, they would also have to take up another responsibility for another piece of work so that the work distribution is fair.

All work done should be appropriately logged in the wiki page found in the GitLab repository. If a member is met with setbacks, the team shall collectively help complete the work.

In the case of Sprint 1, we have decided that everyone should contribute equally to the user stories, domain model and justifications, and wireframes. As a result, we want to have an equal understanding of the baseline which will help each member of the team to their personal task later.

Technology Stack

Integrated Development Environment

The team has decided that we will be using IntelliJ IDEA as our IDE. IntelliJ is an IDE built for Java (JDK version 15.0). It is very well supported and very user friendly. Since all members have experience with it, it will be used as the IDE.

Version Control System

The version control system is the Monash GitLab server. We have agreed on a simple git policy that we are going to follow. The main project will be stored in the main branch. Whenever someone wishes to add a feature, a branch will be created for that feature. When the feature is deemed complete, teammates will review the feature before approving the merge to main.

In order to avoid conflicts in git, we will try to not have two people working on the same feature at once. However, this is probably impossible, meaning conflicts are bound to happen. Hence, whenever a merge conflict happens, it is of utmost importance that the work in the main branch is preserved and not overridden. The person trying to merge will need to try and resolve the conflict before trying to merge again.

Technical Stack

The team evaluated the project and found that the common languages that could be used are Java, Python, C++, C#, and TypeScript.

Some members preferred not to use Python because of past unfavourable experiences with Python GUI packages. C++ is a lower level language than the others and programming with it may take more time and be more error prone (e.g., faults with pointers). GUI development with C++ is also cumbersome as low level system calls need to be used which is not portable.

Although some members may have used C#, the others have no experience with C#. Taking into account the short timeline that we have to complete the project, the general preference is to use Java as all members have past experience with it. Therefore, Java will be used as the programming language for this project. There are several Java frameworks for building GUIs. Swing is an older and more complex one. There is also Java AWT which is simpler and some members have used it before. Since some members have used the Java AWT framework before it will be used in this project. For members that have never used Java AWT before, spikes will be written for them and they will need to complete the spike as soon as possible.