

# COSC345 Assignment 1

## Our Group

Team name: DACA

Team members

1. Daniel Lee (8198050)
2. Adam Aizal (2570831)
3. Callum Cooper (4199323)
4. Andrew John (5413936)

Daniel Lee

- Help with flowchart / project planning
- Punctual with deadlines

Adam Aizal

- UI Design
- Artwork

Callum Cooper

- C++ Experience
- Project structure/design
- GUI Experience

Andrew John

- Organised with given task
- Punctual with deadlines
- Hard worker

Our group composition is ideal for the development of a game as we have a UI designer/artist along with multiple people that are able to implement the suggestions. We also have people experienced in developing C/C++ applications and thus will be able to guide others who are not as experienced.

# What we will build

## Game Idea

We aim to build a turn based survival game that uses real life datasets to provide an entertaining simulation for real survival situations. The game will allow you to choose a region in which you will attempt to survive for as long as possible. Initially our plan is to do the Mt Aspiring region in the South Island, however there are plans to add more once we complete this region. Each game day there are multiple events that will take place, some of these are related to a user's actions that they have chosen, other events will be taken from a random list related to the region and the collected data for that region.

**Events** are categorised as:

- Weather event; Rains, drought etc
- Region event; Interaction with native population, avalanche, etc
- Animal event; Native wildlife may interact in a good or bad way
- Personal event; Forage for resources, etc

Players will start out with four crucial **bars** which are

- Hunger
- Thirst
- Temperature
- Energy

These four bars will be crucial to maintain, if any are to fall too low, the player may die and the game will end. For each action taken, or event generated, the result will impact these bars and therefore it is important for the user to make appropriate and correct decisions. However, there will always be some sense of randomness in the result of actions to ensure that the game has enough variability. Furthermore, the higher these bars are, the more likely the player will succeed in the actions they take - and vice versa - thus creating a positive - or negative - feedback loop which will be balanced by the variability and randomness of the game.

## The datasets we've chosen for our game:

We chose a series of datasets that represent what occurs in our game regions. For Mt Aspiring we chose the following datasets: Weather & Climate, Animals & Plants, and Regional information. These datasets, when combined, will allow us to incorporate realism into our game to make the experience more authentic and entertaining. Furthermore, the datasets encompass the majority of information that occurs within a region and thus give us a full picture of the area to better implement into the game.

### Weather & Climate Data

[Mount Aspiring National Park Weather In January 2020](#) through to December 2020

We are taking the climate data for the Mt Aspiring region and compiling this data into a JSON format to use within our game. We aim to use this data to determine the climate within the game so that the user will have to respond e.g making a fire, finding shelter.

## Animals & Plants Data

### [Mt Aspiring Animals & Plants](#)

This dataset will help us create our list of random events and encounters that can happen when a user plays the game. Using this dataset, we will be able to add realistic encounters into the game such as animal encounters or eating dangerous fauna.

## Regional Data

### [Avalanche Data](#)

### [Mt Aspiring Geographical Map](#)

Regional data will allow us to make the game more authentic by making use of real features, such as rivers and mountains, in the region to better create a list of events and actions for the user to interact with.

# How we will build the game

## Design

We will begin by designing the core principles of this game, some of which we have already discussed. This will ensure we have a clear vision for the game and thus we will all be working towards the same goal.

This phase includes defining how the game systems will work:

- The turn based mechanics
- The actions/events the player experiences
- The player's bars

## Planning

The planning phase involves figuring out how we will begin to build the game and defining the implementation methods for our game systems e.g how we will display the game to the user and how they will interact.

## Subgroups

Our group will be split into two subgroups to improve the workflow of the development process. This ensures that we are working on the part of the game we are more comfortable with and thus better utilising our abilities.

## GUI / GAME ART

This will include how the game will look like, artstyle of the game, and how the game will be displayed to the user.

### Members:

- Adam Aizal
- Callum Cooper

## GAME DESIGN

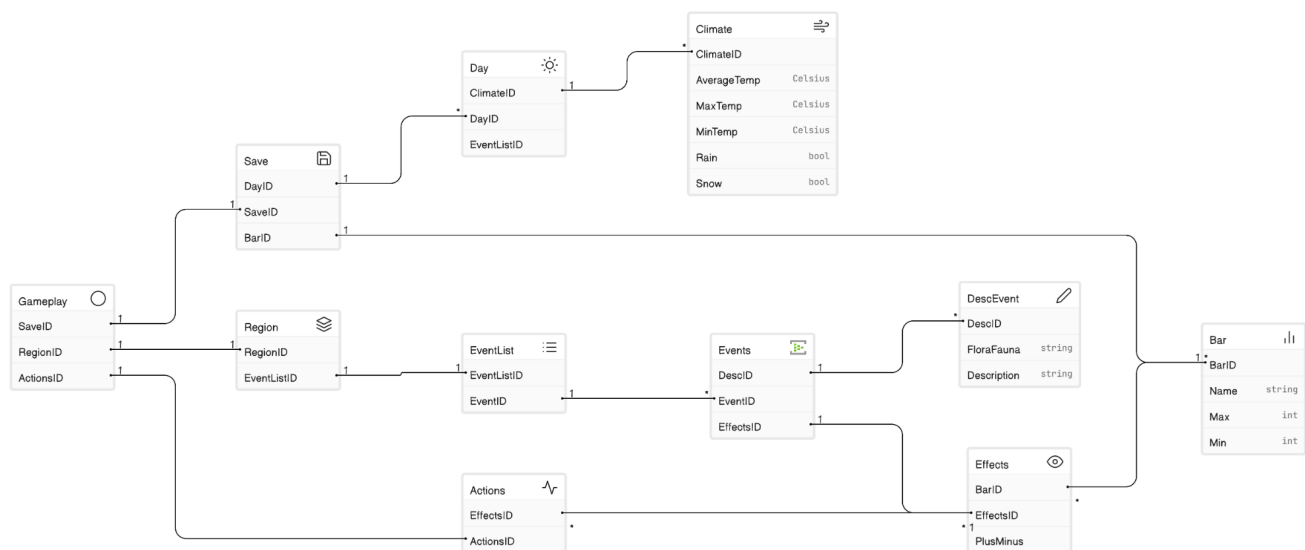
This will include game mechanics, implementation of game events, and structure of the game.

### Members:

- Andrew John
- Daniel Lee

## Game Design

After hours of searching for an engine, we decided to use a GUI rather than a game engine. This is mainly due to the fact that the game is UI based rather than allowing for interaction with character movement. Many of the engines that allow for making a UI are overly complicated and complex and therefore would have a steep learning curve before we could begin development.

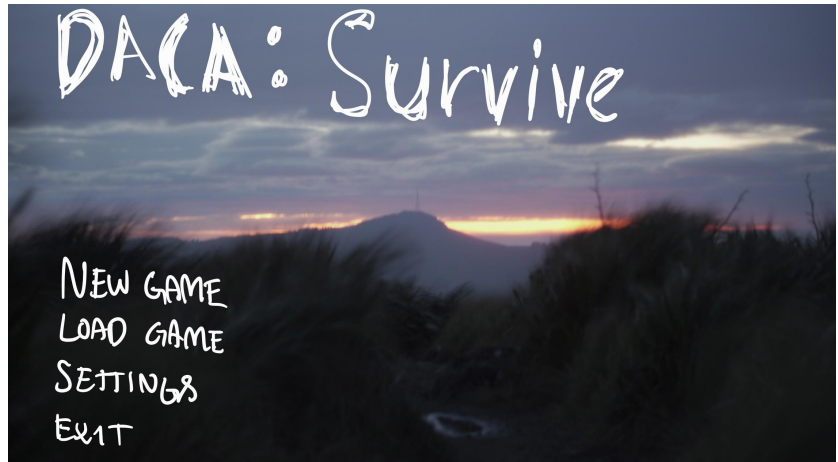


## GUI

### Main Menu

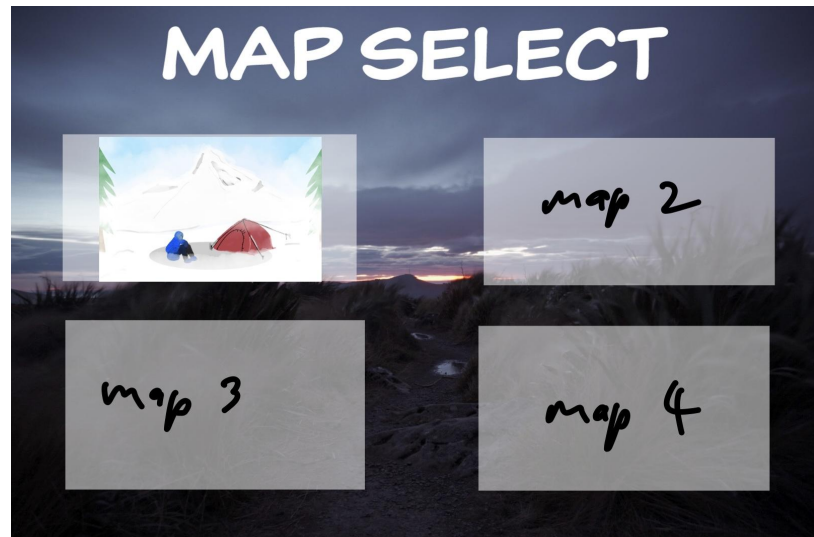
The main menu is a standard menu layout with necessary buttons. The background will be a slideshow with images of our maps, with “hopeful” style background music.

- New Game - Start a new game.  
Load Game - Load a saved game.
- Settings - Change settings such as mouse sensitivity, volume, fullscreen, etc.
- Exit - Exit the application.



### Map Select

The map select screen is a 2x2 grid, showing images of our maps. When hovered, the name of the map will pop up.



## Game Art

### Set



The set is static throughout the whole game, with the exception of changes to show the effects of in-game events; such as extreme snow, heavy rain. Additionally, the scene will change when the player goes to explore, or any other category.

### User Interface

Day count on top left. Crucial bars on top right. Turn category on bottom. Players ought to choose 1 of 4 categories, which will lead to events. The events will have outcomes which will increase/decrease a player's crucial bars.

### Art Style

DACA: Survive will follow a cartoon/drawing art style. This style is very cost effective and does not use heavy computing resources.



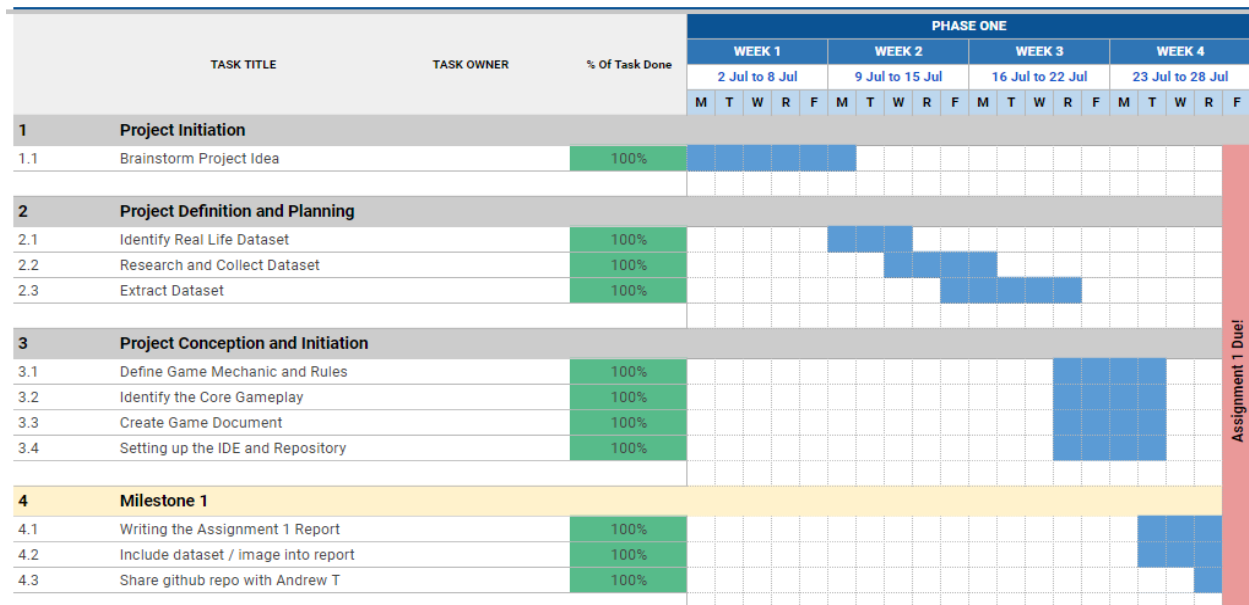
## Development

Once the game has been designed and the project has been planned, we will need to begin building the GUI and implementing our game features. For building the GUI we have identified [Qt](#) as an optimal choice, this is due to Qt's ease of use and vast set of features. Using this library, we will be able to build an interface that both looks good and is easy to use. On the development side, Qt provides tools for designing UIs and integrating with useful tools like CMake. It also provides a variety of code features that we will be able to make use of such as its internal QString and QList.

In order to develop the game efficiently, we have identified several tools to help with this process. Our game will be able to be built using [CMake](#) which allows us to easily build the program across all of our devices without worrying about Visual Studio solution files needing to be managed across devices. CMake also allows for easy scalability and integration support for utilities such as [Google tests](#) for verifying our code and [Doxygen](#) extraction to allow documentation to be easily built.

## Project planning using Gantt Chart

### DACA Gantt Chart

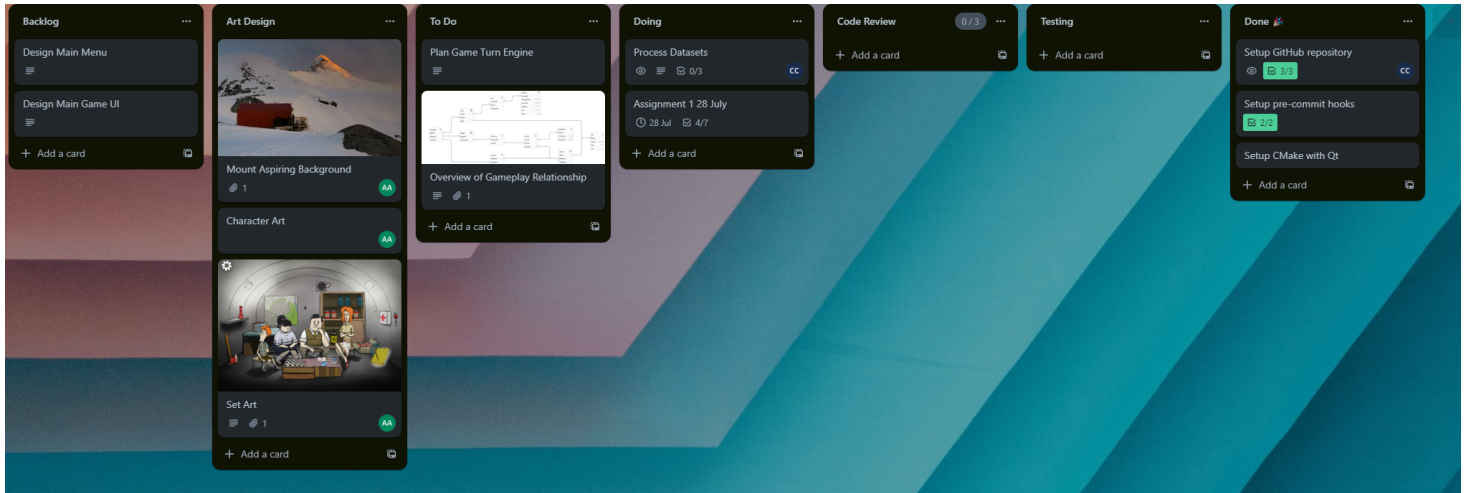


[Link to the full Gantt chart](#)

We are also using a Gantt Chart to visualise our project timeline. The image above shows our project timeline from the start of this paper to the due date of assignment 1. There are a few tasks that overlap in between (eg. Project Conception and Initiation section) because there are times where a few of us will work on different tasks at the same given time to maximise the time usage.

## Trello

We aim to use a [trello board](#) - as seen below - to organise our tasks and track progress. This will ensure that each developer knows what they are doing and the expected deadline for completion. By utilising Trello we will be able to see whether we are on schedule according to our Gantt chart and our assignment deadlines.



## What exists that is similar to our app

Our main inspiration for the project is the game [60 Seconds](#). This game is similar to ours in that it is a turn based game where each day there are actions and events that the player can respond to. 60 Seconds became extremely popular due to its simplicity and ease to play. We hope to take the best parts of this game and incorporate them into our game to try and produce a game that is just as enjoyable and interesting.

## How our game will differ

While survival games are extremely common and popular, we believe our game will add some unique features to the genre. Firstly, the regions of our game are real places. There are no turn based survival games that utilise real world data and real world places to create a scenario. Secondly, our game is purposefully simple, we make use of a simple GUI and aim to reduce the barrier to entry to make the game easy to play but hard to master. This will make the game entertaining as it will be relaxing to play while still being extremely interesting.

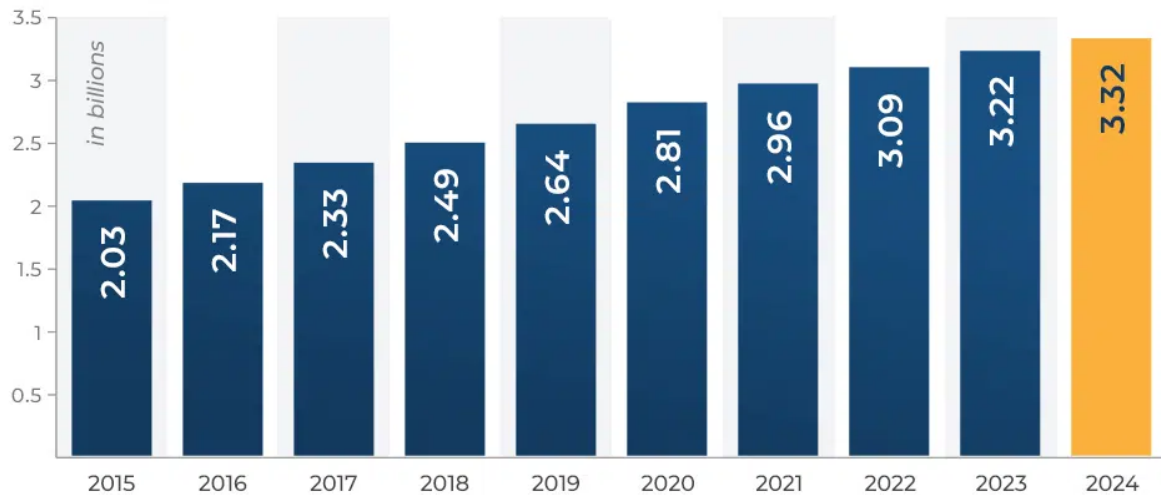
## Proof there is interest in our app

Games have gotten increasingly popular ever since the covid outbreak with people having more and more time to spend playing, and purchasing, games. While a slight regression is likely as





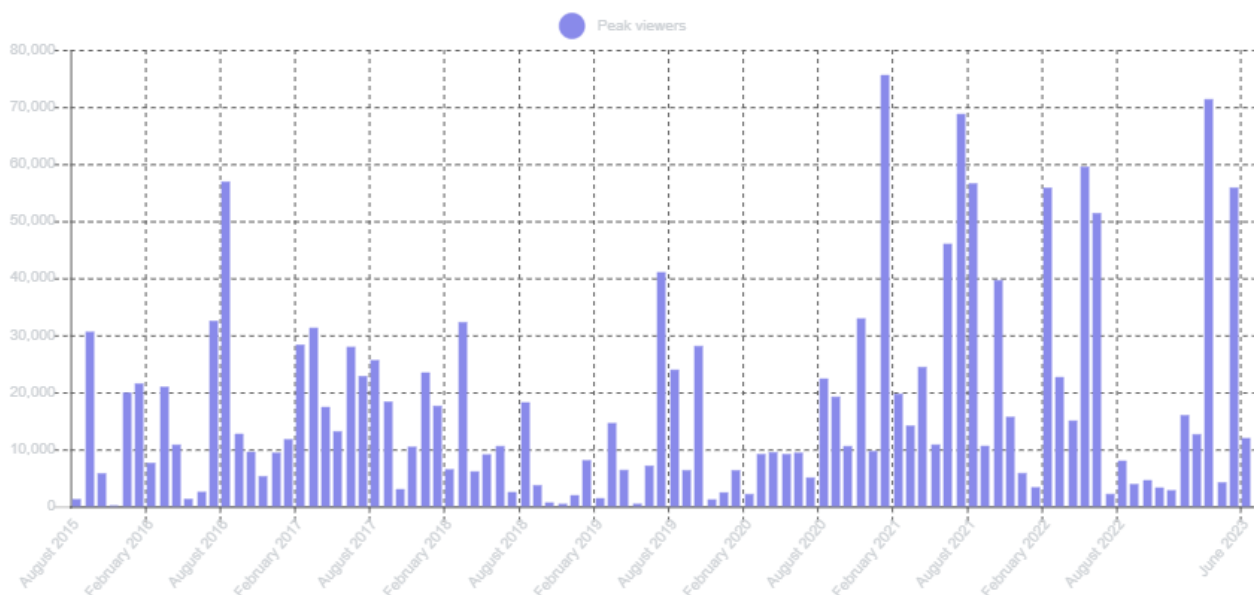
## Number of Video Gamers in the World From 2015 to 2024



Source: Marketer

the world has begun to return back to normal, the effect is likely to remain. Therefore, there is clearly a large market for games and thus interest in new ones.

Furthermore, the success of our inspiration game 60 Seconds provides proof that this type of game and genre are appealing to users. Below is a [graphic](#) showing the peak viewers for the game 60 Seconds on the popular live streaming site [Twitch](#). With multiple months having over 30,000 peak active viewers, it's clear that this type of game interests people and provides entertainment. Furthermore, a simple [youtube search](#) provides evidence that there is a large number of people who enjoy this sort of game as there are many videos with millions of views.



## Risk Analysis/Management

During the development of our game there will likely be some problems that arise that could complicate the process. These risks need to be anticipated to ensure that the disruption they occur is minimised, or that we have an appropriate and measured response to the problem. While these risks are all unlikely, there is still a possibility that they occur.

### Risk: Someone drops the paper

The likelihood of this risk is doubtful. There are a number of reasons that this might occur, some are quite serious such as family emergencies or unseen health complications, others are more simple such as stress. In the event that this occurs the outcome will likely not be too serious. Fortunately, we are a group of four and thus one less developer would not slow development too much as we would be able to pick up the resulting workload. However, if Adam were to drop this paper the result would be more noticeable as he is our sole artist. While this would not prevent us from finishing the project it may result in a visual quality decrease and we may have to look for outside help for our art.

### Risk: Poor project management

The likelihood of this risk is slightly unlikely. This project is, for most of the group, the first big software development project and as such there may be mistakes in the project planning and expectations. Project oversights, or too ambitious plans, may result in an unfinished project or a subpar result. We aim to reduce this likelihood by aiming for achievable goals, such as just one region, to ensure that we are able to deliver what we set out to do. From here we can then begin to expand and look to improve upon the game by adding more features. However, should this risk occur we will need to look to resolve it by holding a team meeting and discussing what we can do to solve it.