

# INFO1113 Assignment 2 Report

Waka Waka

Anthony Yip ayip2151

## Concrete Classes:

Separate concrete classes were used to divide up the game logic and functionality. Each class handles a separate component of the game including setup and execution.

- App
  - App handles the game window and draws the game to the PApplet window.
- Game
  - Game handles the game logic, and stores all game objects and settings.
- Config
  - Config stores the game settings and map filename from reading and parsing the config file.
- MapGrid
  - MapGrid reads and parses the map file and stores all map data including game object positions.

## Abstract Classes:

Abstract classes and inheritance were used to reduce repetition of code and allow multiple classes to access mutual attributes and methods.

- Actor
  - Ghost and Waka both inherit from Actor as they both share common attributes regarding position, speed, and a movement method. They also both inherit the getter and setter methods regarding positioning and also the tick method which applies the movement to the characters.
  - However, Ghost and Waka look and behave differently and thus separate classes were created with different methods. Waka has additional attributes and methods for animation and the ability to eat fruit.
- Tile
  - Fruit and Wall both inherit from Tile, they share the same attributes and methods but Fruit allows collision while Wall does not.

## Interfaces:

Interfaces were used to specify required methods and allow Ghost and Waka to apply any implementation of its subclasses.

- Movement
  - PlayerMovement and TargetMovement both implement Movement as they both can move Actors. However, PlayerMovement is controlled by user input while TargetMovement provides movement based on the position of a Target.
- Target
  - RandomTarget, ChaserTarget and Scatter all implement Target as they will guide movement by using different targets.
  - RandomTarget is used by the Ghost in Frightened mode, moving it randomly.
  - ChaserTarget is used by Ghost (Chaser) and moves it towards Waka.
  - Each Ghost has a different Scatter implementation and target different corners of the map when they enter scatter mode.

## Enumerations:

Enumerations were used to reduce code repetition, avoid clutter from the creation of additional classes and avoid additional conditional code logic. They were also used when a group of objects had predefined values.

- Direction
  - Direction was used as there are only five possible directions of Actor movement (up, down, left, right, none) as Actors can only move in either the horizontal or vertical direction.
- GhostMode
  - GhostMode was used as the Ghosts have three behaviours (Chase, Scatter, Frightened).
- GhostType
  - GhostType was used as there are four different types of Ghost (Ambusher, Chaser, Ignorant, Whim). Each Ghost has their own image file and scatter mode.
- ScatterMode
  - ScatterMode was used as there the four different corners of the map are targeted by the Ghosts during the Scatter behaviour.
- WallType
  - WallType was used as there are six different orientations of the wall, each having its own image.

# Waka Waka UML Diagram





