



# INFO1113

## Assignment 2 – Stage 2

Due: Friday 20 November, 11:59PM AEST

*This assignment is worth 15% of your final assessment*

### Task Description

By now you should have developed a prototype of **Waka Waka** meeting at least the minimum requirements of the milestone submission. **ArcadeRetro** has found the prototype to be a promising product and wishes to try out more features. In addition to the base requirements and the extension laid out in the last set of specifications, they want you to add the features outlined in this document.

### Additional Requirements

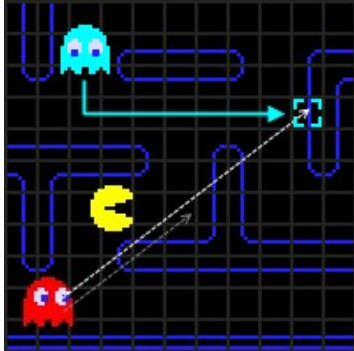
#### More Ghost Types



There are now 4 different Ghost types, each with their own unique behaviour. The Ghost types are Ambusher (pink), Chaser (red), Ignorant (orange) and Whim (blue). Their map file representations are as follows:

Ghost	Character
Ambusher	a
Chaser	c
Ignorant	i
Whim	w

Their movement patterns are the same as that of the Milestone Ghost, in that they alternate between SCATTER, where they move to one of the corners of the map, and CHASE, where they find their target location and then turn in the direction closest to that location. The adjustments to these behaviours are as follows:

Ghost	CHASE – Target location	SCATTER – Target location
Ambusher	Four grid spaces ahead of Waka (based on Waka's current direction)	Top right corner
Chaser	Waka's position	Top left corner
Ignorant	If more than 8 units away from Waka (straight line distance), target location is Waka. Otherwise, target location is bottom left corner	Bottom left corner
Whim	Double the vector from Chaser to 2 grid spaces ahead of Waka. (E.g. if Waka is moving right at (0,1) and Chaser is at (0,0), Whim's target location would be (4, 2))  	Bottom right corner
Source: <a href="https://gameinternals.com/understanding-pac-man-ghost-behavior">https://gameinternals.com/understanding-pac-man-ghost-behavior</a>		

In cases where the target location is outside the bounds of the grid, the closest point is used. For example, if a target location is (-3, 4), the Ghost would target (0, 4).

Note that similar to Milestone, there is no specified number or type of Ghosts each map must have.

## Super Fruit

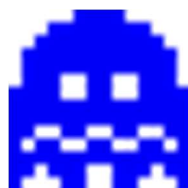
There is an additional Fruit type, known as the SuperFruit. It is twice the size of normal Fruit and is represented in the map file with the character '8'.

When Waka collects a SuperFruit, all Ghosts enter FRIGHTENED mode. While in FRIGHTENED mode, Ghosts decide which way to turn at an intersection at random, only turning back on themselves if they have no other choice.

If Waka hits a Ghost while it is in FRIGHTENED mode, the Ghost is removed from the map and Waka does not lose a life. The Ghost only returns if Waka is later hit by a Ghost not in FRIGHTENED mode.

The duration of FRIGHTENED mode is specified in seconds in the config file under the attribute "frightenedLength". When FRIGHTENED mode ends, the Ghost returns to whichever mode it was in before FRIGHTENED mode began.

When in FRIGHTENED mode, Ghosts use an alternate sprite:



## Result Screens

When the game is over, a result screen is displayed to the player. The two results possible are Game Over and Win, and have the following screens:



The Game Over screen is rendered once Waka has run out of lives. The Win screen is rendered once Waka has collected all Fruit and SuperFruit from the map. After 10 seconds, the game should restart, with all Fruit, Ghosts, SuperFruit and lives restored.

You have been provided with “PressStart2P-Regular.ttf” font file to write text to the screen.

## Marking Criteria (15%)

Your final submission is due on Friday 20 November at 11:59PM. To submit, you must submit your *build.gradle* file and *src* folder to Ed (by pressing MARK) and your report to Canvas. You have made a successful submission if you are then able to download your submission, add a *config.json* and map file to the project, and run *gradle run* without issue.

## Final Code Submission (5%)

You will need to have implemented and satisfied requirements listed in this assignment. Make sure you have addressed the following and any other requirements outlined previously.

- Window launches and shows black background
- Configuration file is correctly read in
- Map loads and walls are displayed
- Ghosts, Fruit, SuperFruit and Waka all start in their correct locations
- Waka is controlled by input as described in Milestone
- Waka cannot move through walls
- Waka’s sprite alternates between closed-mouth and open-mouth
- Waka’s sprite has the correct orientation
- When in CHASE mode, Chaser targets Waka’s position
- When in CHASE mode, Ambusher targets four grid spaces ahead of Waka (based on Waka’s current direction)
- When in CHASE mode, Ignorant targets Waka if more than 8 units away from Waka (straight line distance), otherwise, targets bottom left corner
- When in CHASE mode, Whim targets double the vector from Chaser to 2 grid spaces ahead of Waka
- Ghosts move towards their target locations as outlined in Milestone specifications
- Pressing space activates/deactivates DEBUG mode
- Waka can collect fruit by passing through them
- Waka loses a life when he is hit by a Ghost

- Waka's lives are shown at the bottom of the screen
- When Waka loses all of his lives, the Game Over Screen is shown
- When Waka collects all the fruit in the map, the Win screen is shown
- After 10 seconds, result screens are cleared and the game restarts
- When Waka collects a SuperFruit, all Ghosts enter FRIGHTENED mode
- When in FRIGHTENED mode, all Ghosts have the frightened sprite
- When in FRIGHTENED mode, Ghosts move at random
- When in FRIGHTENED mode, Waka can remove Ghosts by hitting them
- When Waka is hit by a Ghost, Waka and all Ghosts are reset to their starting positions
- Ensure that your application does not repeat large sections of logic
- Ensure that your application code exhibits good Object Oriented principles

### Milestone (1%)

To ensure progress has been made within on your project, you will need to submit a working submission of your project by Monday 2 November by 11:59pm AEST. You should achieve the following as a minimum:

- Map and config file are read in
- Map is correctly rendered (including fruit)
- Waka is rendered and movable
- At least one Ghost is rendered

Note that marks for this component will be returned after the final due date. However, if you attend your scheduled lab in week 10, your tutor will be able to provide feedback on your current progress.

### Test Cases (4%)

During development of your code, add test cases to your project and test as much functionality as possible. You will need to construct unit test cases within the `src/test` folder. To test the state of your entities without drawing, implement a simple loop that will update the state of each object but not draw the entity.

Ensure your test cases cover over 90% of execution paths (Use *jacoco* in your *gradle build*) Ensure your test cases cover common cases. Ensure your test cases cover edge cases. Each test case must contain a brief comment explaining what it is testing.

### Design, Report and Javadoc (4%)

You will need to submit a report that elaborates on your design (2%). This will include an explanation of any object-oriented design decisions made (such as reasons for interfaces, class hierarchy, etc) and an explanation of how the extension has been implemented. This should be no longer than 500 words. This report will be submitted through Canvas.

As an appendix to your report, you need to include a complete UML diagram of your design (1%). Markers will refer to both your codebase and this diagram when judging your use of Object Oriented Design.

Your code should be clear, well commented and concise. Try to utilise OOP constructs within your application and limit repetitive code. The code should follow the conventions set out by the [Google Java Style Guide](#). As part of your comments, you will need to create a Javadoc for your program (1%).

**Extension (1%)**

You have the choice of either adding one of the listed features to the program or creating your own feature (must first be approved by TA).

- Collectable soda-can that frightens ghosts and turns them invisible for a period of time
- 3D arcade machine, where the player can pause the game and walk around the arcade
- Be the ghost mode (requires AI for Waka)
- Online mode, where two players can play as Waka and Ms. Waka from different computers

**Warning**

Any attempts to deceive or disrupt the marking system will result in an immediate zero for the entire assignment. Negative marks can be assigned if you do not properly follow the assignment specification, or your code is unnecessarily or deliberately obfuscated.

## Academic declaration

By submitting this assignment you declare the following:

*I declare that I have read and understood the University of Sydney Student Plagiarism: Coursework Policy and Procedure, and except where specifically acknowledged, the work contained in this assignment/project is my own work, and has not been copied from other sources or been previously submitted for award or assessment.*

*I understand that failure to comply with the Student Plagiarism: Coursework Policy and Procedure can lead to severe penalties as outlined under Chapter 8 of the University of Sydney By-Law 1999 (as amended). These penalties may be imposed in cases where any significant portion of my submitted work has been copied without proper acknowledgment from other sources, including published works, the Internet, existing programs, the work of other students, or work previously submitted for other awards or assessments.*

*I realise that I may be asked to identify those portions of the work contributed by me and required to demonstrate my knowledge of the relevant material by answering oral questions or by undertaking supplementary work, either written or in the laboratory, in order to arrive at the final assessment mark.*

*I acknowledge that the School of Computer Science, in assessing this assignment, may reproduce it entirely, may provide a copy to another member of faculty, and/or communicate a copy of this assignment to a plagiarism checking service or in-house computer program, and that a copy of the assignment may be maintained by the service or the School of Computer Science for the purpose of future plagiarism checking.*