

**Assessment 1: Lab 1**

**Initial documentation**

Author: Bertocco Thomas-Killian - #13758322

Lecturer: Da Sheng Liu

CO5609 Intelligent Agents and Simulation | SP53 | 2019

8 December 2019

Table of Contents

[Glossary of terms 3](#_Toc26396279)

[Stakeholders 4](#_Toc26396280)

[Bandai Namco 4](#_Toc26396281)

[The players 4](#_Toc26396282)

[The video game industry 4](#_Toc26396283)

[The platform of distribution 4](#_Toc26396284)

[Developers of the product 5](#_Toc26396285)

[Functional requirements 5](#_Toc26396286)

[Technologies 5](#_Toc26396287)

[Game design 5](#_Toc26396288)

[architectural design 7](#_Toc26396289)

# Glossary of terms

**The ghost house**: Spawn of the four enemies

**Tile**: in this context refers to an 8x8 pixel square

**Chase mode**: Enemies run after Pacman

**Scatter mode**: The four enemies disperse to the corners

**Frightened mode**: enemies pseudorandomly decide in which direction they will advance

**Shadow**: red ghost

**Speedy**: pink ghost

**Bashful**: blue ghost

**Pokey**: orange ghost

**Pac-dots**: Items that Pacman must eat to advance to the next level.

**Power pellets**: If eaten, gives to Pacman the ability to eat the ghosts.

**Dijkstra’s algorithm**: Algorithm used to determine the shortest path between nodes in a graph.

**C++**: One of the most powerful programing language for object-oriented programming.

**Model View Controller**: Architectural pattern that separates an application into 3 logical components.

**Object-oriented programming**: A programming language model in which programs are organized around data or objects.

**OpenGL**: An application programming interface used to render 2D or 3D graphics

**SFML**: A software development library that provides access to various multimedia components in computers.

# Stakeholders

## Bandai Namco

The Bandai Namco company has been found in 1955 by Masaya Nakamura and is now directed by Toru Iwatani, the creator of Pacman. It is the publisher of the original Pacman game (in 1980), and still retains all the rights of the franchise.

Throughout its existence, the company has created dozens of Pacman games, which have brought in more than $14 billion.

Their last original Pacman game, Pacman Championship Edition 2, had been released on multiple platforms (PC & console) in 2016 and has been sold over 1 million copies.

To ensure the continued success of the franchise, and to increase its annual turnover, the company would like to publish a new game.

As the customer who ordered the development of this product, the role of this stakeholder is to define the constraints, rules, and design of this new game. It must also ensure that it provides the necessary resources to make the game happen. He will then have to ensure the quality of the product and take care of the publication as well as the communication around the game.

## The players

For this type of game, the definition of "Player" remains vague.

Pacman is one of the few games that has the potential to target everyone. It can be played by both girls and boys, adults and children, regular and casual players.

The player's role is to judge. It is he who will be able to determine whether or not the game is of good quality. He can also participate in its improvement by sending feedback or bug reports, as well as in communication by deciding to share his gaming experience on social networks.

## The video game industry

The video game industry can be advantaged by the release of a new game from a legendary historical license. Firstly because our product can bring back old players into video games, but also because it can attract new players, people that are usually not attracted by video games. It will also continue to promote the current trend of playing retro games style.

## The platform of distribution

Platforms of distribution, such as Playstation 4, Android or PC, also have a lot to win from a new Pacman game. If we release the game as exclusivity for a specific platform, it can increase sales of the console (e.g. some people buy the Xbox instead of the PlayStation only because of Halo or Forza). Besides, Platforms take a significant percentage of sales in their online store. The platform we chose is Steam. It is the widely used software on PC that has a vast library of games. With good communication and good visibility, we can reach a good part of its 90 million active users. Steam online store collects 30% of the sales.

The role of a platform of distribution is to participate in the communication around the game, assure that it has excellent visibility on their store. They can also help developers if they have problems to adapt the game on the platform.

## Developers of the product

The developers' role is to create the product corresponding to the client criteria. They usually receive a fixed amount of money. We will send the game to the publisher (Bandai Namco) and produce updates or bug fixes after the release if necessary.

# Functional requirements

## Technologies

**Programming language**

* The game must be implemented in C++

**Libraries / Engines**

* The game must use OpenGL to render graphics
* The game must use SFML

**Integrated development environment**

* The game must be developed with Visual Studio

**Code Architecture**

* The code must use the object-oriented programming model
* The code must use the Model View Controller pattern

**Operating system**

* The game must be played on Microsoft Windows 7 and newer

## Game design

**Display, Control, and Audio**

* The game must be controlled with the arrow keys of the keyboard
* The game must be played with a proper resolution
* The game must have music and sound effects

**Levels**

* The game must have at least 1 level

**Maze arcade game**

* The game must be a maze arcade game, equal or similar to the original Pacman game

**Characters**

* The game must contain 4 enemies
* The game must contain 1 playable character

**Scoring system**

* The game must have a scoring system
* The game must have a leaderboard

**Menus**

* The game must have the main menu
* The main menu must have a start and a leaderboards buttons

**Map**

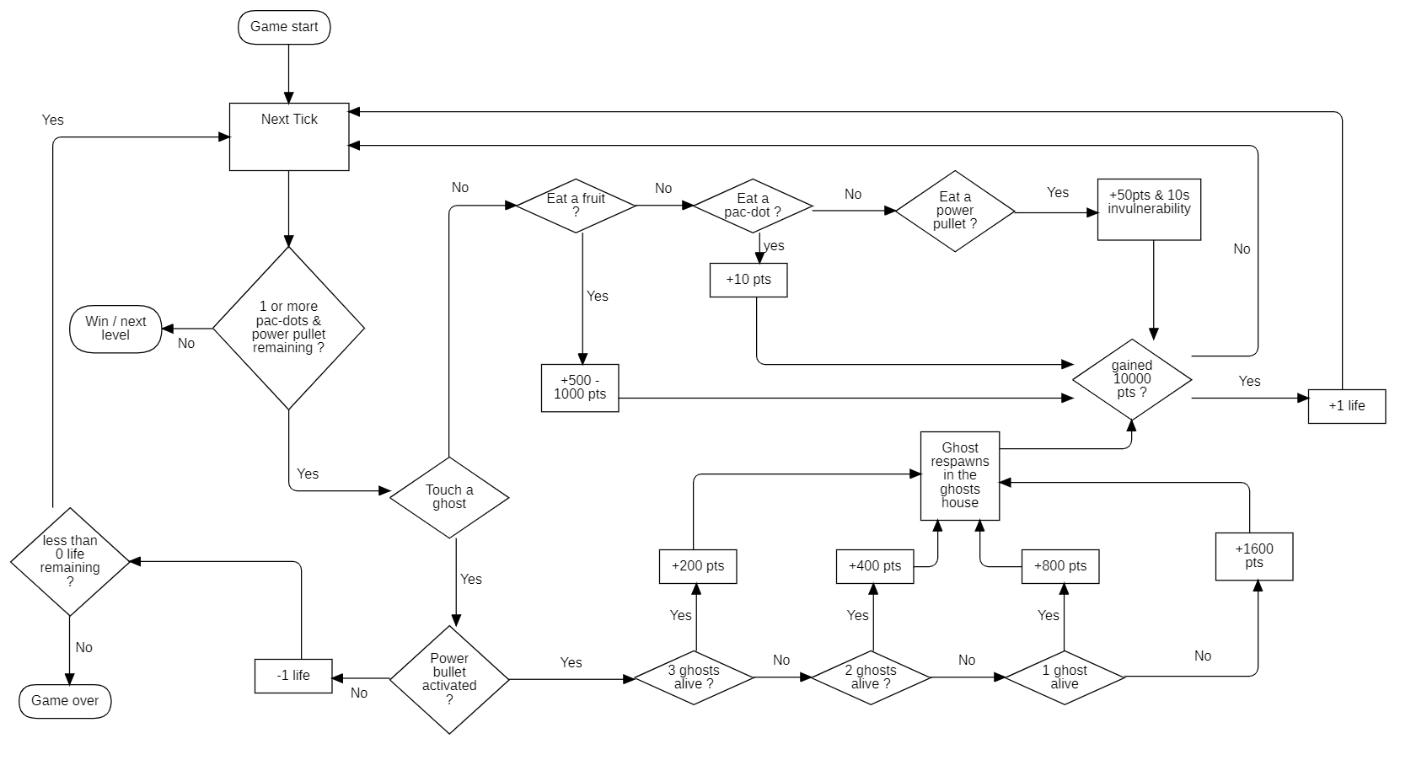
* The map must contain pac-dots on every non-wall tiles
* The map must contain power pellets at its corners
* The map must contain items to increase the score
* The map must contain walls, and teleportation exits (e.g. Pacman exit by the right and appears on the left of the map)

**Artificial Intelligence**

* The AI of the enemies must be balanced to make the game not too easy, and not too hard
* The enemies must have a chase mode
* The enemies must have a scatter mode
* The enemies must have a frightened mode

# Architectural design

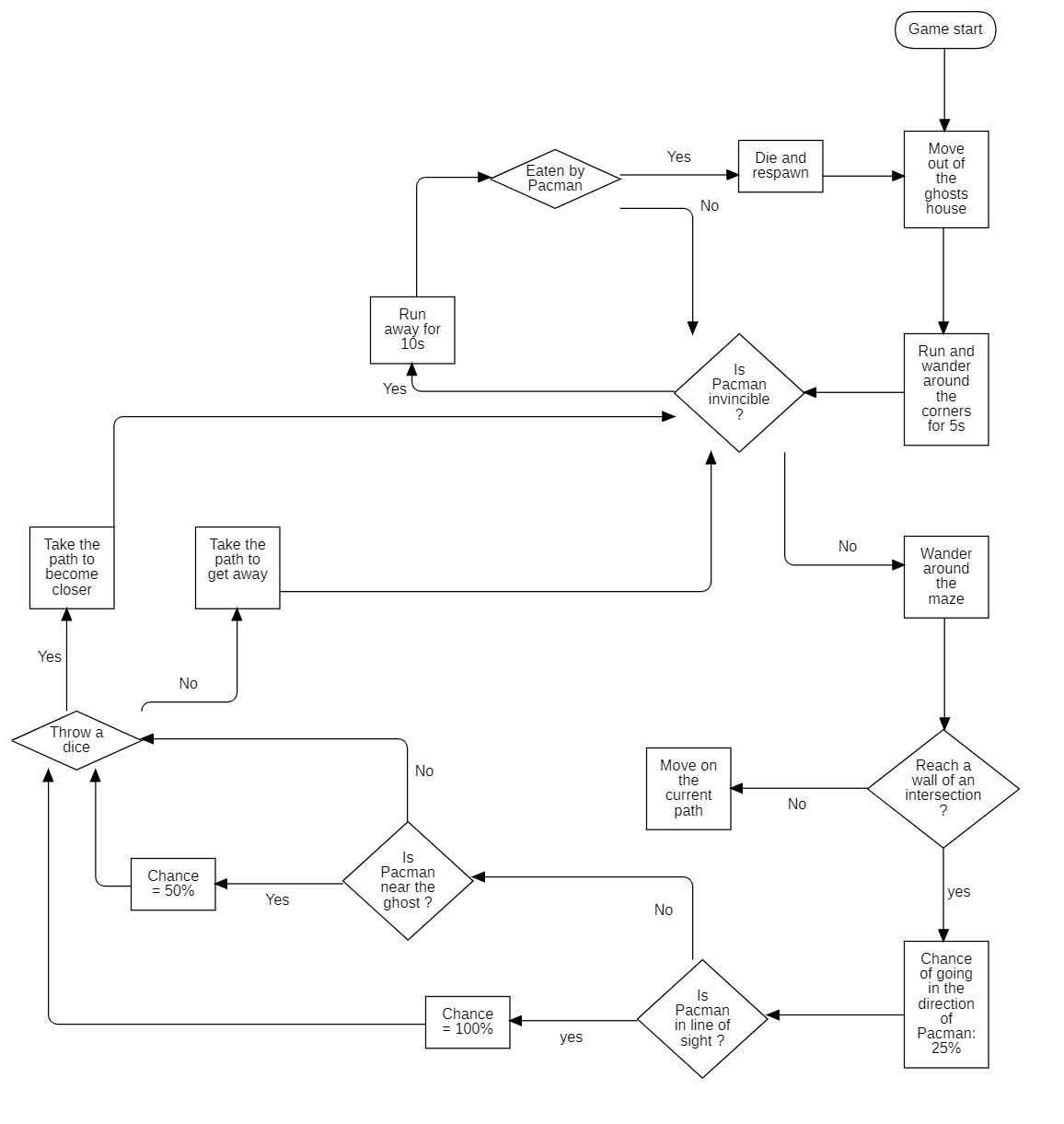
## The game logic



The code will check the position of each elements of the game at every tick.

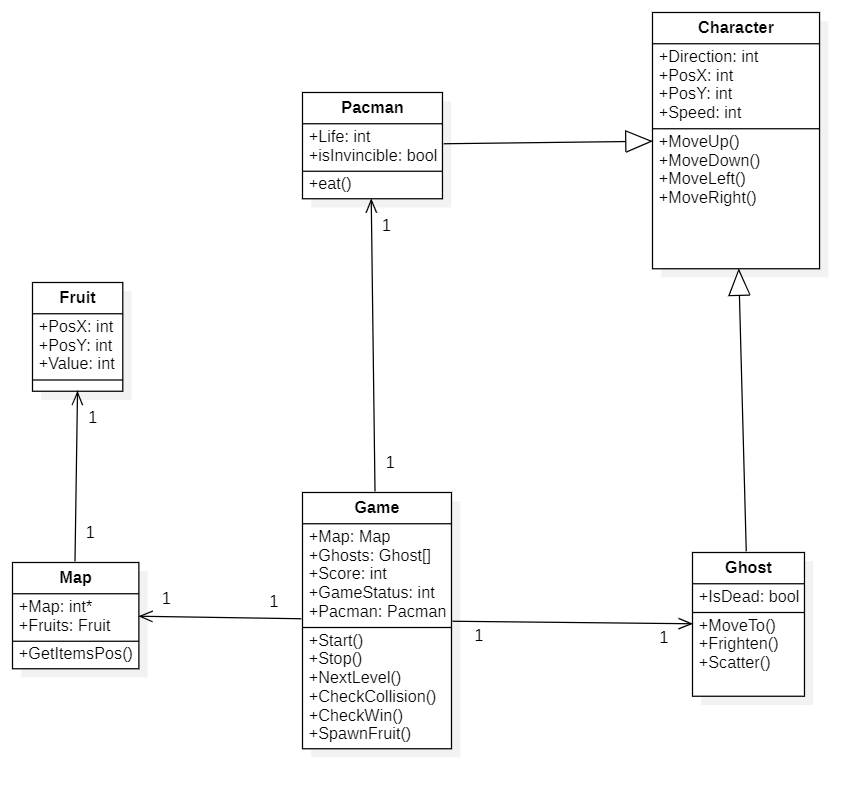
As long as there is one or more pac-dot/power pullets in the game, and Pacman is still alive, the level continues. If Pacman touches a ghost, he lost one life, and after losing his three lives, it is game over. However, he can still gain a life thanks to the scoring system. Every time he collects 10000 points, he earns one life. To reach 10k, he has to eat different items on the map or kill the ghosts. The bigger your kill combo, the more points you earn. After eating every pac-dots and power pullets, the game loads the next level.

## The Artificial Intelligence



Every ghost got the same behaviors. To make the game balanced, we use probabilities. Ghosts will first run to a corner at the start of the game to spread. Then, they will check if Pacman is invincible. If he is, they will run away, if he is not, they will advance. When arrived at an intersection or a wall, they throw dice. They have 25% chance to use the path that leads to Pacman, and 75% chance to go in the opposite direction. However, If Pacman is in the line of sight of the ghost, he has 100% to chase him. If he is not in the line of sight, but near him (3 cases around), the ghost will have 50% chance to chase him. If the ghost is touched by Pacman while he is in frightened mode, he dies and goes back to the ghost house.

## The UML diagram



This is a very basic UML example for a Pacman game. The Game object is our Main. It contains the map, an array of ghosts, the score, the Pacman and the status of the game. Since Ghosts and Pacman are similar, they share some functions and attributes from the abstract class Character. The map object loads a map stored in a file, and store in it an integer array. As the fruits appear later in the game, they are not included in the map file, so we need to create an object for them. The Game object will spawn a fruit on a random free tile of the map. At every tick, the Game object checks if Pacman has a collision with an object or a ghost. If Pacman gets touched by a ghost, its life attribute goes down, and when it reaches 0, the Game status change to 0 (GameOver). The function CheckWin() will check the number of dots present on the map, and see if Pacman has won or not.