**2805 ICT**Systems and Software Design  
**Assessment | Milestone 1  
Student Name:** Mitchell Roles **Student ID:** s5132278  
**Student Email:** [mitchell.roles@griffithuni.edu.au](mailto:mitchell.roles@griffithuni.edu.au)

# Table of Contents

**Contents**

[1. Table of Contents 2](#_Toc46472818)

[2. Project Requirements 3](#_Toc46472819)

[2.1 Functional Requirements 3](#_Toc46472820)

[2.2 Use Cases 4](#_Toc46472821)

[2.2.1 Actor Description 4](#_Toc46472822)

[2.2.2 Documented Template 4](#_Toc46472823)

[2.2.3 Activity Diagram 4](#_Toc46472824)

[2.3 Use Case Diagram 4](#_Toc46472825)

[2.4 Non-Functional Requirements 4](#_Toc46472826)

[2.5 Constraints 4](#_Toc46472827)

[3. Project Risk 5](#_Toc46472828)

[4. Project Feasibility 6](#_Toc46472829)

[5. Prototype Demonstration 7](#_Toc46472830)

[6. Conceptual Design 8](#_Toc46472831)

[6.1. Automatic Document generation 8](#_Toc46472832)

# Project Requirements

## Functional Requirements

Functional Requirements are, as the name implies, the needed functionality of a system or what that system needs to achieve with legal/expected input from a user. These functional requirements will detail what precisely is the desired behaviour of the product. For this specific project the goal is to create a version of the game Pac-Man so the functional requirements will reflect the needs to meet this end.

|  |  |  |
| --- | --- | --- |
| **ID:** FR-1 | **Name:** Broad GUI Requirements | |
| Functional Requirement | | Rationale |
| The system will need to offer a user an interface to which they can interact with the game in an intuitive and simple way, allowing them to select their desired settings and attempt to beat the game. | | Having a functional and intuitive GUI will allow for an enjoyable user experience, one which they can access what they want quickly and efficiently. |
| **ID:** FR-2 | **Name:** Functional Tutorial | |
| Functional Requirement | | Rationale |
| The system will require a dedicated section of itself to educate the user on how the core gameplay aspect functions as well as the win and lose conditions. | | The tutorial will allow a user to smoothly be introduced to the gameplay aspect of the system, avoiding any possible frustrations without some amount of guidance |
| **ID:** FR-3 | **Name:** Game Score | |
| Functional Requirement | | Rationale |
| The system should display some information, ascertaining to how well a user is doing in the game. “Eating” pellets, power-pellets or fruit all should increase the score. | | Having some form of score will allow a user to, at a glance, determine how well they are currently doing within a play state. |
| **ID:** FR-4 | **Name:** Win Condition | |
| Functional Requirement | | Rationale |
| The system needs to recognise when a user has completed a level. This will be done when the final pellet has been “eaten” | | Having a win condition will give a user a visible goal to move towards. This will also allow the game to finish and not continue forever. |
| **ID:** FR-5 | **Name:** Lose Condition | |
| Functional Requirement | | Rationale |
| The system will need to recognise when a user has lost the level. This will be when a ghost has touched “Pac-Man” and they are not under the effect of a power-pellet. | | This will give the user something to avoid and add some challenge to the experience, to prevent it from becoming boring. |
| **ID:** FR-6 | **Name:** Power Pellet | |
| Functional Requirement | | Rationale |
| The game aspect of the system will need to recognise when the user has “eaten” a power-pellet and have them enter a power up state, where the ghost will scatter and can be eaten themselves. | | This will give the player more choice as to how they wish to complete the level, going for the power up early or waiting until there are little pellets left. |

This might need to change.

## Use Cases

### Actor Description

|  |  |  |
| --- | --- | --- |
| **Actor Name:** User | **Other Names:** Player | |
| Actor Description | | Actor Assumptions |
| The actor for this system will be a user, or player. This user will be the main driver of the program, as most the systems functions will remain unused unless they specifically use it. The user will be able to interact with the system through a combination of the mouse and keyboard. With the mouse primarily being used for the menu and the keyboard being used to navigate the maze. | | The user will have the expectation that the system will work faultlessly with no visible errors during gameplay. |

### Documented Template

|  |  |
| --- | --- |
| **Use Case:** Playing the Game | |
| **Function Requirement Satisfied:** FR-1 | **Actor:** User |
| Risk | |
|  | |
| Pre-Conditions | Post Conditions |
|  |  |
| Basic Flow | |
|  | |
| Corresponding Diagram | Alternate Flow |
|  |  |

### Activity Diagram

## Use Case Diagram

## Non-Functional Requirements

## Constraints

# Project Risk

# Project Feasibility

# Prototype Demonstration

# Conceptual Design

## Automatic Document generation

**Conceptual Design**