2805ICT PACMAN REQUIREMENTS ANALYSIS

Sebastian perry

S5132483

Contents

At a glance

This document will contain the various functional, and non-functional requirements (in addition to project constraints) that have been generated in relation to the Pacman project.

Functional Requirements

Below is a list generated of the functional requirements for the project.

Identifier	Priority	Functional requirement
F-R-1	4	The game should offer a graphical interface through which the player can see Pacman, the ghosts, walls, and any other objects which get added to the game.
F-R-2	4	Pacman and the ghosts should be able to move around the maze, moving at a constant speed in a constant direction until decided otherwise by the player or the A.I.
F-R-3	4	Pacman and the ghosts should collide with walls and consider them impassable
F-R-4	4	Pacman should be able to change his direction whenever, and the ghosts should decide their direction based on heuristic algorithms calculated every time they reach a crossroad.
F-R-5	4	If Pacman collides with a ghost, Pacman should lose a life and the game should restart
F-R-6	4	Pacman should be able to collide with small pellets, which would have the effect of removing the pellet and raising Pacman's score
F-R-7	3	Pacman should receive a penalty to his score depending on the amount of time it took to complete the level
F-R-8	3	The game should also have the ability to run on three different display / map settings, a square cell configuration, a hexagonal grid configuration, and an arbitrary grid configuration
F-R-9	2	At least one ghost should move towards Pacman through the use of Dijkstra's algorithm
F-R-10	2	The game should be able to randomly generate valid mazes
F-R-11	2	Pacman should be able to collide with large pellets, which would have the effect of removing the pellet and temporarily changing the Pacman ghost rule so that it is instead the ghosts which temporarily die, the will also try to get away from Pacman
F-R-12	1	If all pellets have been consumed, Pacman should gain a life and the game should restart
F-R-13	1	Players should be able to easily manipulate various game options (for example: map type, Pacman speed, ghost speed, map size)

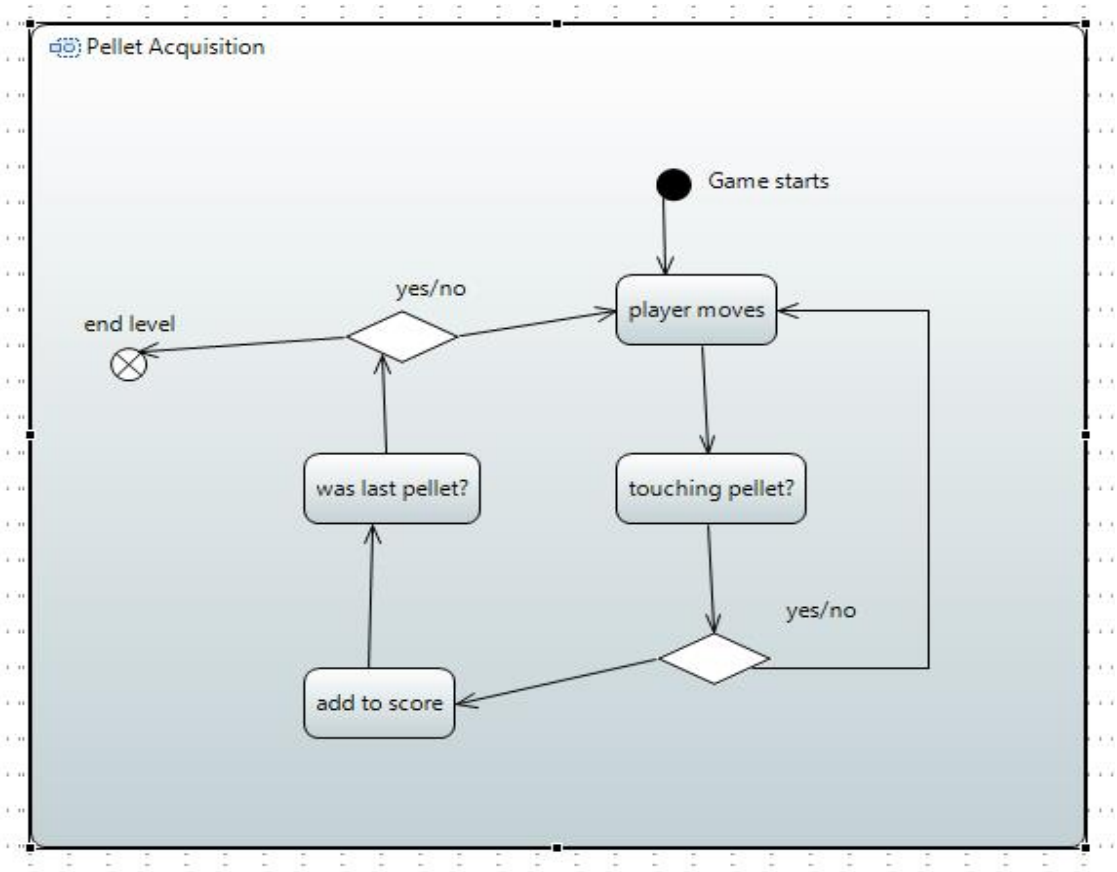
Use cases

Actor list

Actor: User	
Description: this is the person who plays the game that is being created. It is assumed that they will be able to complete this task through their access to a keyboard and mouse.	
Alias: Player	Inherit

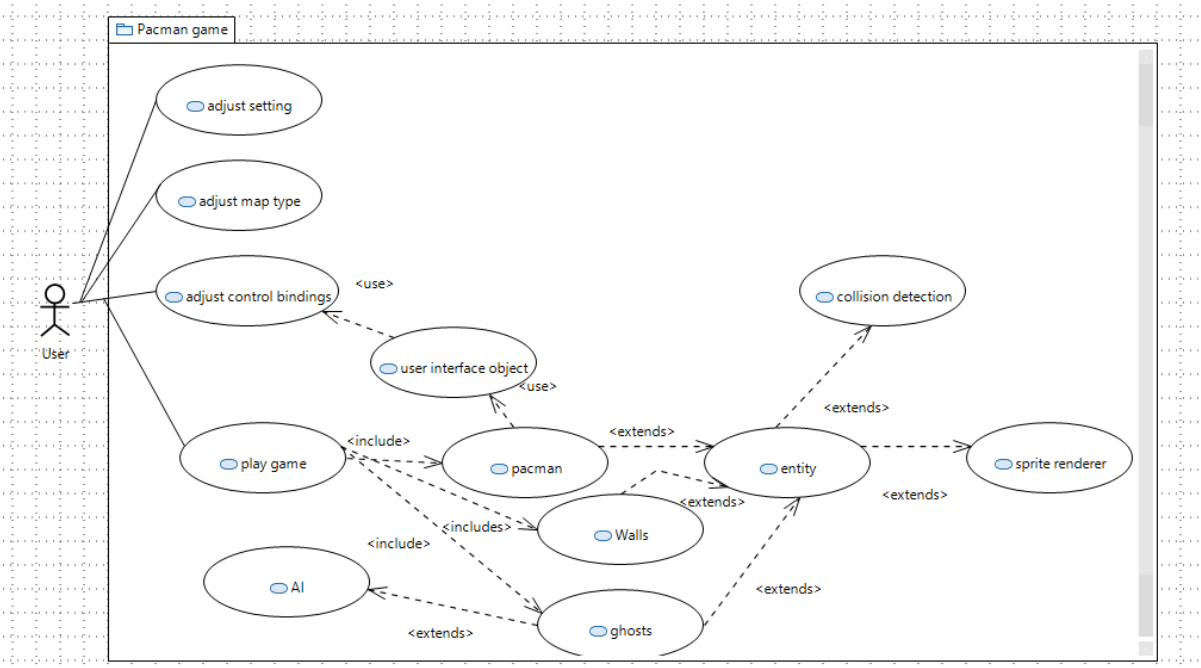
Actor: small pellets	
Description: this is a small circular object which can be found and collected in around the map in order acquire points.	
Alias: Player	Inherit

Use case: Pacman picking up pellets
Requirements satisfied: F-R-6
Brief Description: the player picks up pellets with the ultimate goal of colleting all of them and progressing to the next level
Actor: User, small pellets
Priority: 4
Risk: this requirement describes a core element of the game, without it a user has very little reason to play the game at all
Pre-conditions: the player has successfully started the game after having reviewed the options and control bindings
Post-conditions: the player will have had a fun time
Basic flow: 1. Initially there are many small pellets throughout the map 2. Iteratively the player will collet them by moving Pacman over them Upon pickup, score will be added to a counter variable, and if the pellet picked up was the last pellet, the level end
Activity Diagram: refer to the activity diagram below
Alternative flow: the user exits the game before completion



Use case diagram for typical playthrough

Below, is a use case diagram of a typical playthrough



Non-Functional Requirements

Below is a list generated of the functional requirements for the project.

Identifier	Priority	Non-Functional requirement
NF-R-1	3	Universality: the game should be able to run on both 32 and 64 bit versions of windows
NF-R-2	3	Reliability: the game should work without crashing, producing errors, or violating defined behaviours
NF-R-3	2	Ascetics: the finished game should look and feel as close to the original retro Pacman as possible within time limits and requirements (obviously this will not be possible with some of the aspects, but it would be preferable for the overall ascetics to match throughout)
NF-R-4	1	Performance: the finished game should be able to run at a stable frame rate
NF-R-5	1	Useability: players should be able to change their control bindings and other game options via a file placed in the same directory as the executable

Constraints

Below is a list generated of the functional requirements for the project.

Identifier	Constraint
C-1	Tool Constraint: It has been chosen that the game is going to be getting compiled using a 32bit architecture version of MinGW in addition to 32 bit packages of SDL, and SDL image. Because of this choice, an effort must be made to ensure memory usage stays under 4 gigabytes
C-2	Tool Constraint: The packages require that certain .dlls are present in the same directory as the executable when running the production version
C-3	Time Constraint: The whole Pacman game must be completed by the time of the third assignment milestone
C-4	Platform Constraint: The game should be able to run on 2 vastly different operating systems
c-5	Version Control Constraint: During the creation of the game, a version control software should be used extensively