Functional Specification

Snake reinforcement learning agent with interactive user front-end

Nicolas Oyeleye - 19359231
Ionut Eusebiu Andrici - 19702361

10/12/2021

Table of contents

1. Introduction

1.1 Overview

The project that will be developed is a snake game desktop application. The game consists of a player controlling a snake which roams around the board trying to pick up the food that appears. Every time the snake eats the food, its length increases and the difficulty of the game raises as the snake gets longer and longer. It will implement a single player version of the snake game which will provide a player the ability to interactively choose the rules of the game with consequent options of playing the game directly or run an AI agent which will play the game. The player will also be able to visualize the agent playing the game.

The main focus of this project is to develop an AI agent using Q-learning, a value based reinforcement learning algorithm, train it to maximise the score and analyse the performance of the agent in learning the game and adapting when the rules of the environment change.

The primary programming language that will be used for the implementation of the snake game and the agent is Python and the graphical interface of the game will be implemented using the Pygame library.

1.3 Glossary

AI	*Artificial Intelligence is intelligence demonstrated by machines, as opposed to natural intelligence displayed my humans*
Reinforcement Learning	*Machine learning sub-field, in which a computer's actions are taken based on the current state of the environment and the previous results of actions.*
Agent	*An entity that perceives/explore the environment and act upon it*
Q Learning	*Value-based reinforcement learning algorithm to learn the value of an action in a particular state*
Q Value	*A value given to an action in a certain state.*
GUI	*Graphical User Interface*
Pygame	*Is an external graphical and sound Python library which allows the implementation of python applications.*

2. General Description

2.1 Product / System Functions

The game will focus on a player controlling a snake as it wanders through the board. The player will be able to choose to play the game directly or to run the AI agent which will play the game. The rules can be chosen by the player before starting a game. The snake starts from the centre of the board and the goal is to eat food, which appears randomly on the board, and increase the score. As the snake eats, it increases in length and the game will end only when the snake dies. The functions that will be available at each step of the game are:

- **Main menu**
  - Play
  - Run agent
  - choose game rules

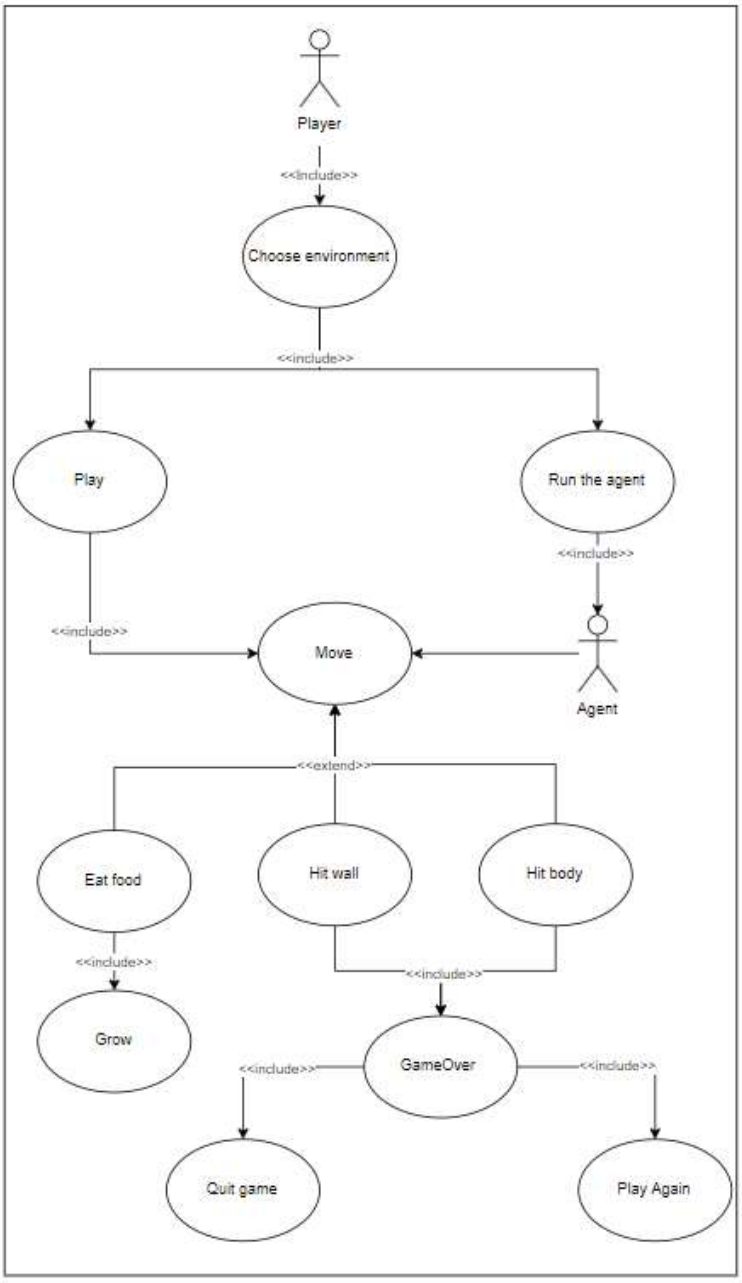
- **Game screen**
  - move
  - restart
  - spawn food
  - return to main menu

2.2 User Characteristics and Objectives

The users that might use the application are those who are interested in AI and want to test if they can perform better than an AI agent. Generally our application is done only for the analysis of the agent performances in different environment rules but we do not exclude other possibilities.

2.3 Operational Scenarios

2.3.0 Use Case Diagram



2.3.1. Start the game for the first time

Current System State

In this system state, the player has launched the executable file and is met with a menu page. The Highest score is 0 as the player has never played the game or has no previous game scores. The options on the menu are to play a game or run the agent.

Scenario
The player chooses between playing the game or watching the AI agent play. After the player has chosen an option, they are met with a second page where they can choose the rules of the environment the snake is to move in.

Next Scenario
When this is done the player is loaded into a game and is ready to start. If they had chosen to run the agent the agent is loaded into the game and is ready to play.

2.3.2. Agent in the middle of a game

Current System State
The current system state consists of the agent in the environment on the move.

Scenario
The agent decides what direction it wants to move in based on the position of the food items, the direction the snake is going and the position of the danger elements. The snake will grow after it eats a food item. The agent will use the Q score to move the snake and it will update the Q table with a new value. If the snake doesn't eat anything it continues to move. If the snake eats a food item it will grow in length. If the snake's head comes in contact with its body or a wall, the snake will die and the game is over.

Next scenario
This process is repeated numerous times until the snake's head comes in contact with its body or a wall, in which case the game is over.

2.3.3. The Game is over

Current System State
In the current system state the game is over.

Scenario
The player is given the option to play again and quit.

Next Scenario
If the player chooses to play again, they are loaded back into a game. If the score from the previous game is higher than the current highest score value, the highest score will be

updated to the score from the last game. If the player chooses to save the game the highest score evaluation will be done again and it will be saved so next time you load the game the highest score is the highest score from this game. If the player decides to quit the game he/she will be brought back to the main menu.

2.4 Constraints

Time constraint: as the project has a limited amount of time to be completed we have to allocate enough time for the allocation for the implementation and the testing of the application before implementing other non critical functions.

Agent efficiency: the agent must be able to perform the calculations for the next action to execute in a short amount of time which will not compromise the gameplay.

Technological requirements: in order to efficiently train an agent the processing power must be adequate.

Knowledge: Our knowledge can be a limit when researching new software or studying the algorithm.

System constraints: The application will be only available to be run on a Windows machine as pyinstaller produces an executable that only works on the same machine.

3. Functional Requirements

3.1 Choosing the rules of the game

- **Description:** The human player must be able to choose the rules of the game before starting a game. The rules that the player will be able to select form are deciding if the snake will die from hitting the edge of the board or not, if the board will contain walls or not and if the snake will increase its length in the head or in the tail after eating the food. A normal game is when the snake dies when hitting the edge of the board, there are no walls and when the snake eats the food, the length is increased in the tail.
- **Criticality:** This has high importance since the main aim of the project is to analyse the performance of the agent in changing the rules of the environment.
- **Technical issues:** Two issues that must be addressed is that the player must be able to choose the rules of the game from a graphical player interface and that only a set of rules can be applied to the game.
- **Dependencies with other requirements:** This is the functionality on which playing the game is based, since the rules chosen will influence the strategy of how the human agent and AI agent will play.

3.2 Human agent plays the game

- **Description :** We believe that a player has to have the possibility to manually play the game and try and get a higher score than the AI agent.
- **Criticality:** Not critical, but relevant to the entertainment of the player.
- **Technical issues:** The player must be able to visualize the board and the movements of the snake through a graphical interface and control it using keyboard commands.
- **Dependencies with other requirements:** The gameplay and the actions possible to the snake are influenced by the game rules chosen.

3.3 AI agent plays the game

- **Description:** This is the main functionality of the system. The AI agent we create has to have the ability to gradually learn how to play the game and improve its performances over time. The agent must be able to get what the current board state is and determine what is the best next move to execute.
- **Criticality:** This is the most crucial part of the system, since a player can test if it is possible to get a higher score than the agent and the performance of an agent playing a game can be analyzed.
- **Technical issues:** Processing power is the one of the main concerns when training an agent, so hardware with adequate processing power must be used. Our limited knowledge about reinforcement learning can be considered an issue which will be addressed by conducting detailed research on the topic and consulting our supervisor.
- **Dependencies with other requirements:** The way the AI agent will play the game will be highly influenced by the rules chosen before starting the game.

3.4 GUI interaction

- **Description:** The GUI will define the way the player will interact with the application and it must display the correct information. Elements of the game which will be displayed are the main menu, choice of the game rules, the board, the food, the snake moving through the board, the score and other information.
- **Criticality:** The graphical interface interaction is important, since it can be the source of frustration if the system does not display the correct information or implement and perform the choices made by the player.
- **Technical issues:** The size of the clickable elements must be appropriate as well as the contrast between colour and text. Clickable elements, when selected, must run the appropriate functions.

- **Dependencies with other requirements:** In order for the system to work correctly the clickable elements must mup to the appropriate functions and the game must be able to recognize the keyboard commands that the player inputs.

4. System Architecture

This section describes a high-level overview of the anticipated system architecture showing the distribution functions across (potential) system modules. Architectural components that are reused or 3rd party should be highlighted.

4.1. Agent

The agent is the entity that will explore the game environment and act upon it.

4.2. Player

The game will be a singleplayer game where the player will play against the AI player. The player can also watch the agent play after they have trained.

4.3. Game Scripts

The game scripts are the files that will run the game. Both the backend and frontend will be written in Python.

4.4. Game Executable File

You can either play the game in singleplayer mode or watch the agent play the game. The game will be run by an executable file using Pyinstaller. This will be the application file that when run will launch the game.

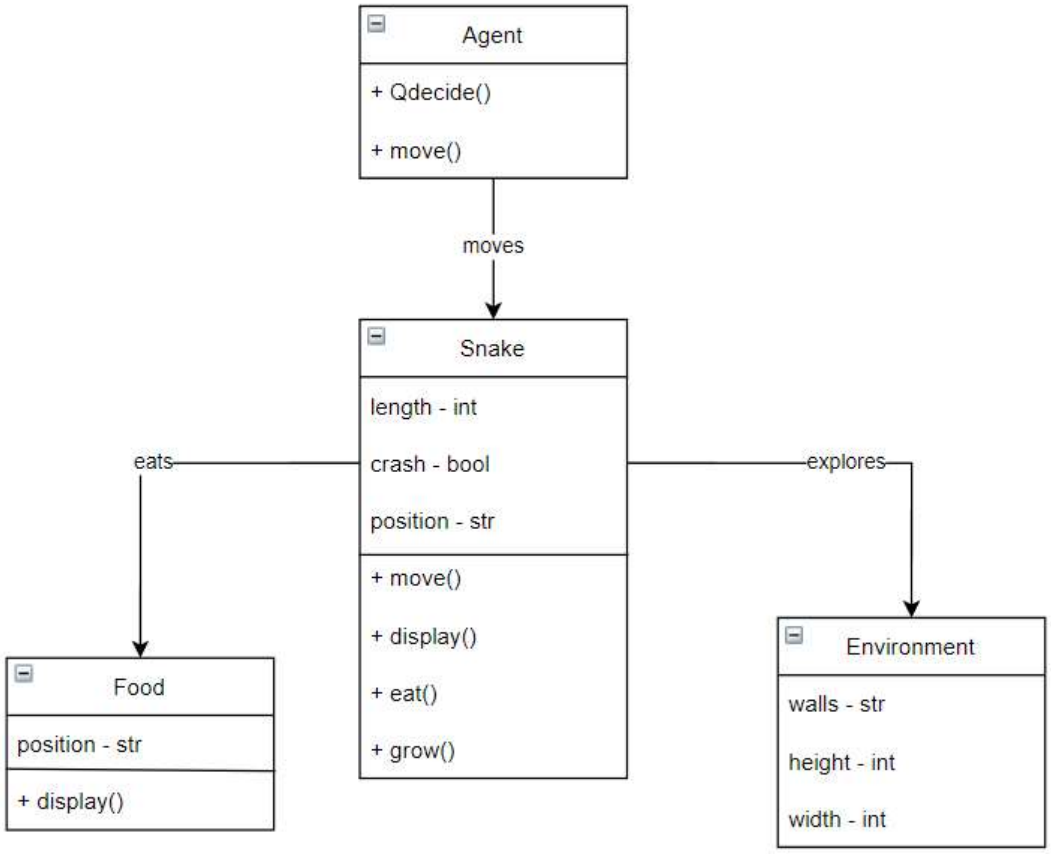
4.6. A.I.

As previously mentioned the player can choose to play the game or what the agent train. The A.I is the program the player can choose to play against.

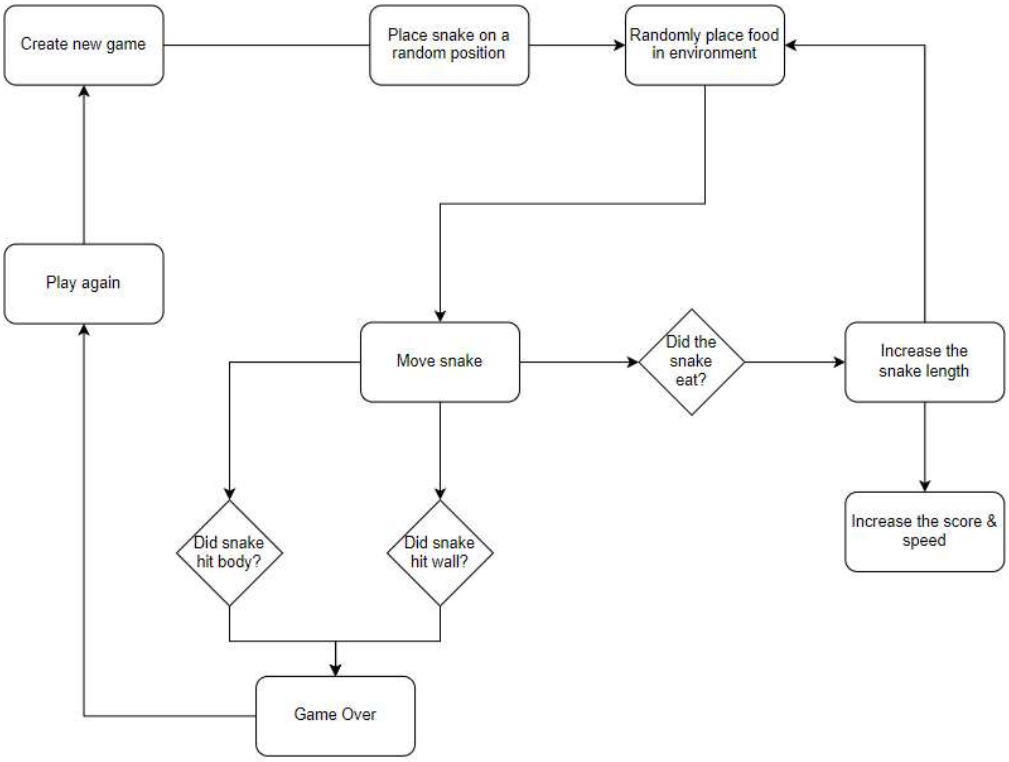
5. High-Level Design

This section should set out the high-level design of the system. It should include one or more system models showing the relationship between system components and the systems and its environment. These might be object-models, DFD, etc.

5.1 Class Diagram

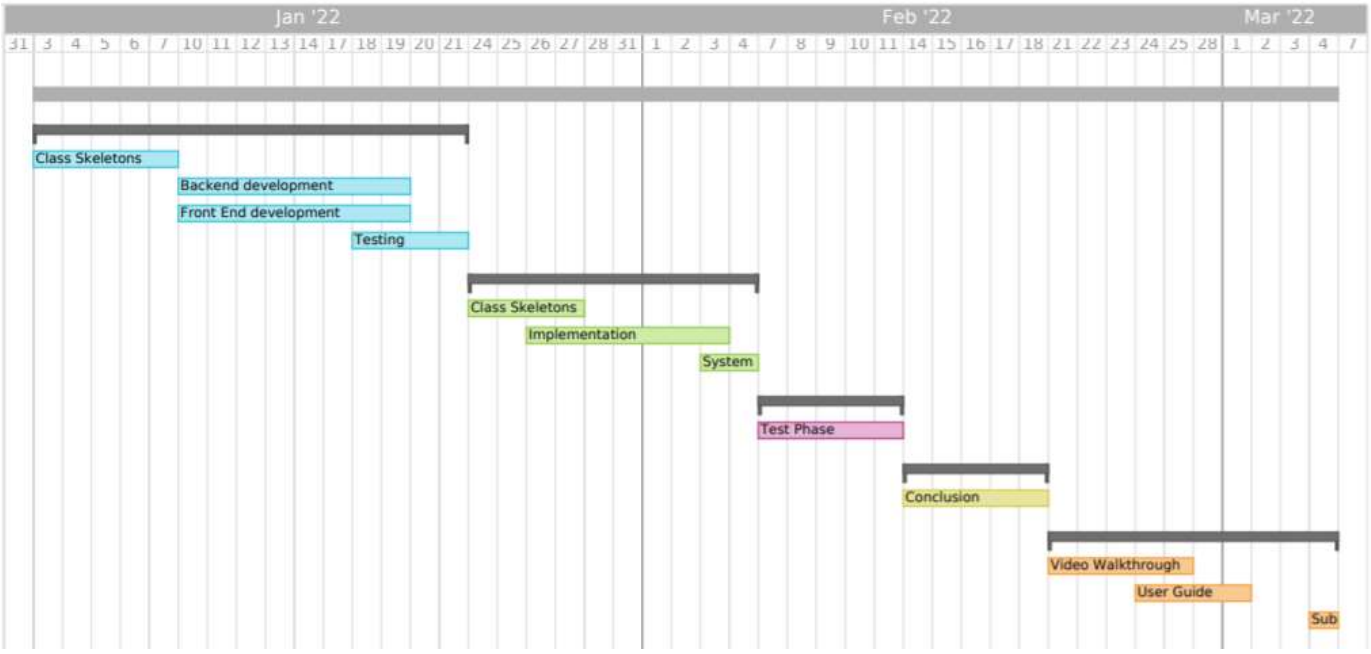


5.2 Data Flow Diagram


Create new game
Place snake on a random position
Randomly place food in environment
Play again
Move snake
Did the snake eat?
Increase the snake length
Increase the score & speed
Did snake hit body?
Did snake hit wall?
Game Over

6. Preliminary Schedule

CA326	start	end
Game Development	**03/01/22**	**21/01/22**
Class Skeletons	03/01	07/01
Backend development	10/01	19/01
Front End development	10/01	19/01
Testing	18/01	21/01
Agent Creation	**24/01/22**	**04/02/22**
Class Skeletons	24/01	27/01
Implementation	26/01	03/02
System Integration	03/02	04/02
Testing & Training	**07/02/22**	**11/02/22**
Test Phase	07/02	11/02
Analysis	**14/02/22**	**18/02/22**
Conclusion	14/02	18/02
Project Submission	**21/02/22**	**04/03/22**
Video Walkthrough	21/02	25/02
User Guide	24/02	01/03
Submission	04/03	04/03



7. Appendices

Specifies other useful information for understanding the requirements.

1. En.wikipedia.org. 2021. *Snake (video game genre) - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Snake_(video_game_genre)> [Accessed 1 December 2021].

2. Medium. 2021. *Simple Reinforcement Learning: Q-learning*. [online] Available at: <https://towardsdatascience.com/simple-reinforcement-learning-q-learning-fcddc4b6fe56> [Accessed 1 December 2021].

3. Pathmind. 2021. A Beginner's Guide to Deep Reinforcement Learning [online] Available at: <https://wiki.pathmind.com/deep-reinforcement-learning> [Accessed 3 December 2021].