1. **INTRODUCTION**

An exhaustive analysis of the different areas related to the project will be seen through this chapter. In addition to that, the different objectives of the project will be explained. The scope and limitations will be introduced as well. Finally, a methodology and the system to validate the objectives will be shown.

* 1. **CONTEXT**

In this section, we are going to describe the different areas of interest related to this project. Besides, the actors that are going to be affected by this project will be presented.

* + 1. **Areas of interest**

There are three main areas of interest in this project. The first one is the video game Pac-Man. The second one is Machine Learning (ML) applied to video games. The last one is the technical aspects related to the development of the project.

* **Pac-Man:** This project has the goal of developing a ML for the Ghosts of the video game Pac-Man. Because of that, a platform based on this video game is required.

The game consists in eating all dots found around a stage, while there are four Ghosts with different patterns that want to catch the player. The key is to run away from the Ghosts or else eat them. The Ghosts can only be eaten if they are frightened and this will happen if Pac-Man is energised. Thus, Pac-Man has to take some energisers distributed all over the stage. Not only should the player aim at eating all dots but the high score has to be beaten as well. All eaten elements will add points into the score. In the following list there are the rules of the score:

* Dot 10 points. .
* Energiser 50 points.
* Ghosts 200 points.
* **Machine learning applied to video games:** It is a branch of artificial intelligence (AI) in which a computer generates rules based on data that has been fed into it. Video games are applications that usually need the generation of different rules depending on some environments. Even then, there are some reasons for declining to use ML in video games.

When designing a ML, there are some complexities to take into account. Looking upon a learning model too adjusted for the difficulties of the video game, the player will not have a great challenge. But considering another one too prepared for its dangers, the player will not have any opportunity to win. In addition to that, the ML algorithms must be created with caution because when the problem is quite complex a high process time might be needed.

Nevertheless, many companies have chosen to develop ML on video games and there have been very good results such as Black & White which is considered one of the revolutionaries in those video games with ML.

* **Technical aspects:** We need to take into account some technical aspects due to some requirements of the project.

First of all, a game engine is needed for developing the platform of the video game with all possible elements and its interactions. Furthermore, a tool to navigate on the stage is required too. For that reason, we will use different cameras offered by the game engine. Game engines will be more detailed in the section 1.2 State of the art.

Besides, we need one way to obtain or design 3D models from the original 2D models of Pac-Man, since the platform has to be in 3D. A modelling and rendering software will be required for this project accordingly. This tool offers the possibility of creating different polygonal shapes as well as making deformations and joints with other polygonal forms. We can also get some deformations by moving some of its vertexes or edges. Once the deformations are completed we will get a final form: the 3D model searched.

1. **SYSTEM ANALYSIS**
   1. **SYSTEM OBJECTIVES**

The main objective of this project is to develop a machine learning model for the Ghosts of the video game Pac-Man. They will improve their movements in accordance with Pac-Man’s actions. If Pac-Man takes all dots and eats the frightened Ghosts rapidly, the Ghosts will begin to move better. Nevertheless, if Pac-Man has greater difficulties taking all dots, the Ghosts will be more kind and they will move worse.

The second objective is to develop a machine learning model for the Pac-Man character. When Pac-Man is playing, it will not know a lot about the system. Pac-Man should learn which things are positive (dots, energisers, fruits and the frightened Ghosts) and which ones are negative (the non-frightened Ghosts). Then, when Pac-Man is playing, it begins with some experience but not totally complete yet. Because of that, it will recognise some positive and negative factors but it might not recognise them all. Finally, when Pac-Man is playing again, it begins with the maximum experience. Thus, it will know what to do in any situation from the first moment of the game.

When Pac-Man is playing beginner, it starts without experience and it will not know a lot about the system. Pac-Man should learn which things are positive (dots, energisers, fruits and the frightened Ghosts) and which ones are negative (the non frightened Ghosts). Then, when Pac-Man is playing medium, it begins with some experience but not totally complete yet. Because of that, it will recognise some positive and negative factors but it might not recognise them all. Finally, when Pac-Man is playing expert, it begins with the maximum experience. Thus, it will know what to do in any situation from the first moment of the game.

When Pac-Man is playing beginner, it starts without experience and it will not know a lot about the system. Pac-Man should learn which things are positive (dots, energisers, fruits and the frightened Ghosts) and which ones are negative (the non frightened Ghosts). Then, when Pac-Man is playing medium, it begins with some experience but not totally complete yet. Because of that, it will recognise some positive and negative factors but it might not recognise them all. Finally, when Pac-Man is playing expert, it begins with the maximum experience. Thus, it will know what to do in any situation from the first moment of the game.

**2.1.1 Scope and obstacles**

In order to solve all the objectives, developing a video game platform based on the classic pac-man is needed.

We have to consider keeping some features from the classic game:

* Number of characters (The four Ghosts and Pac-Man)
* Different Items (dots, energisers and fruits)
* Restrictions in the moment of characters(The Ghosts cannot change to the opposite direction).
* Difference in speed between Pac-Man and the Ghosts
* Reduction of the Ghosts speed and increase in Pac-man speed. When Pac-Man eats an energiser.
* Delayed time to the Ghosts departure.
* Different targets for each Ghosts.

On the other hand, the forecast features to discard are:

* Reducing speed when Pac-Man eats a dot.
* Reducing speed when the Ghosts are I tunnel.
* Shorting way Pac-Man turns left or right.
* Original system collision between Pac-Man and the Ghosts.

The platform needs to allow two modes a game mode where Pac-Man is controlled by the user and the Ghosts by CPU, and a simulation mode where Pac-Man and Ghosts are controlled by CPU and the user controls the camera of the scene. Game models will be modelled in 3D owing to the fact that one of the previous objectives was to navigate among the characters views.

The algorithms to keep in mind for developing the machine learning are the based on the reinforcement learning. We are going to evaluate different reinforcement learning models, but we will choose one of them for the final application. Other algorithms implementation will be needed to evaluate some characters attributes.

**2.2 METHODOLOGY**

We will explain the methodology that we are going to apply to formulate and develop the project. We will define the tools that we are going to use and what function they will do. Finally the project has to be validated the work is well done. Thus we will expose how we are going to validate them.

This Project visualizes the results of the machine learning techniques implemented.

**Module 1: Search**

We implement depth-first, breadth-first, uniform cost, and A\* search algorithms. These algorithms are used to solve navigation and traveling salesman problems in the Pac-Man world.

**Module 2: Multi-Agent Search**

Classic Pac-Man is modelled as both an adversarial and a stochastic search problem. We implement multi-agent minimax and expectimax algorithms, as well as designing evaluation functions.

**Module 3: Reinforcement Learning**

We implement model-based and model-free reinforcement learning algorithms, applied to the AIMA textbook's grid world, Pac-Man, and a simulated crawling robot.

**Module 4: Ghostsbusters**

Probabilistic inference in a hidden Markov model tracks the movement of hidden Ghosts in the Pac-Man world. We implement exact inference using the forward algorithm and approximate inference via particle filters.

**Module 5: Classification**

We implement standard machine learning classification algorithms using Naive Bayes, Perceptron, and MIRA models to classify digits. We extend this by implementing a behavioural cloning Pac-Man agent.