**Automated Maze Generation for Ms. Pac-Man Using Genetic Algorithms**

Aykut Burak Safak, Erkan Bostanci and Ali Emre Soylucicek

# Abstract

Ms. Pac-Man has been a very popular arcade game since its release in 1982. The original game is based on a single maze structure which can make game play lose its attractiveness eventually. This paper aims to generate sets of various mazes through an evolutionary approach. A genetic algorithm was designed to create optimal mazes by specifying a fitness function to create different mazes which can allow the game to be finished by the player. Evolutionary approach was chosen due to its unique yet acceptable results. Results show that different maze structures are possible to obtain in addition to the classical design.

**Index Terms**—Genetic algorithm, evolutionary maze generation, Ms. Pac-Man, artificial intelligence.

# INTRODUCTION

Automated generation of game levels is an interesting and active research field as well as finding solutions to mazes using various techniques . Such mazes allows generation of different unpredictable game environments for arcade games like Ms. Pac-Man. Heuristic approaches such as genetic algorithms allow finding optimal maze structures through a number of iterations in which best candidates are chosen. Genetic algorithms, proposed by Alan Turing in 1950 by the idea of a learning machine which would parallel the principles of evolution , are heuristic search algorithms that mimics the natural process of evolution. These algorithms are mostly used to generate solutions to optimization and search problems. Although Turing was the one who proposed the idea of evolutionary machines, Nils Aall Baricelli simulated the idea on a computer in 1954 . Alex Fraser, Australian Geneticist, published a series of papers on simulation of artificial selection of organisms using genetic algorithms. Hans-Joachim Bremermann published a series of papers in the 60s that adopted a population solution to optimization problems, undergoing recombination, mutation and selection. Although Baricelli designed a genetic algorithm that plays a simple game, artificial evolution became a widely recognized optimization method as a result of the work of Ingo Rechenberg and Hans-Paul Schwefel in the 60s Rechenberg’s group was able to solve complex engineering problems using genetic algorithms. Related to Ms. Pac-Man game, there are already different studies that try to improve the intelligence of the enemies and the main character such as evolving location evaluator or using tree search methods for safe locations. More examples of adding the game with more sophisticated intelligence can be found in including creation of map models or generation of an agent using Ant Colony Optimization. This paper explores the use of genetic algorithms for automated generation of mazes for this game. Original maze of the Pac-Man can be seen at the Maze.1. The main aim here is to generate mazes that allow the game to be completed. These different mazes can be played as sequential levels for the original game which consisted of a single level. The rest of the paper is structured as follows: Section II describes the approach for modelling this problem in order to find a solution using genetic algorithms. Section III presents the graphics library used in the work. Section IV results of automated maze generation followed by Section V where the paper is concluded.

# APPROACH

Genetic algorithms mimic the process of natural selection, reproduction of fit members of a population and evolution. The following sections describe the design stages for the maze generation.

1. Steps of Genetic Algorithms

The algorithm used here can be described in a few simple steps:

1) Producing new members by crossover

2) Calculating the fitness of every member in the population

3) Mutating some random members

Describing the genes correctly and efficiently is one of the most important steps of the genetic algorithms since the structure is the piece of information that will evaluate to a solution to the problem.

1. Genes

Genes are the symbolization of an individual's solution to the problem, so describing them correctly is important for finding an optimal solution.

Genes that are used in the project have the main information about the blocks constructing the maze. This main information is the position of a block in a two dimensional space, whether the block is horizontal or vertical and it’s size. Every maze in the population has 48 blocks in it and every block has four parameters, which constructs the gene for every member as a two dimensional array which has 48x4 members.

Storing genes in an array makes it easier to crossover the genes and to make necessary calculations. These mazes are then can be easily rendered by the graphics system that is used in the game.

1. Fitness of Individuals

Calculating the fitness is the most important step and a very crucial to the algorithm. This step describes what the algorithm is looking for and what will be solution to the problem, so it is really important to describe and implement this function correctly.

The fitness function used in the project was using the information about mazes; whether the maze is playable or not, whether the blocks have spread to the maze homogeneously or haven’t, total count of blocks (considering the area of them), ratio of horizontal and vertical blocks.

First algorithm to determine if the maze is playable was to auto-play the game until it is finished, but this approach took too much time and was not reliable so the algorithm was reconstructed and redesigned to a much more effective and swift one. The algorithm used for this operation is as follows:

1) Select one of the dots from the maze and push it to a stack and flag it as checked

2) Pop a dot from the stack and check whether it have any neighbours (top, left, bottom, right)

3) Push any unchecked neighbours of the dot to the stack

4) Iterate second and third steps until there is no members in the stack

5) If the iteration of second and third steps are equal to the count of the dots than it means the game is playable, else is not playable.

1. Cross-over

Crossover, in the genetic algorithms, is used for creating new candidate solutions from the existing ones. It is a simulation of natural crossover as the genetic algorithms are the simulation of nature.

Crossover method used here is three point crossover method. Each selected candidate gene divided into four equal pieces to generate two new candidates as depicted in Fig. 2.

1. Selection

Supervised randomizing is used to select parent candidates to crossover. Supervised randomness is accomplished by selecting randomly while giving fit candidates more chance than the others in the reproduction process.The main idea behind this is that expectation that fit parents are more likely to produce fitter siblings for the following generations.

Roulette selection algorithm is used to apply this idea to the paper. Working principle of this algorithm is to select a random point from a wheel which was created considering the fitness of the candidates (Fig. 3). Here the likelihood of selection for reproduction of a fitter parent is higher than a parent with a lower fitness value.