Intelligent Computing Project

Sadula Vishwa (20074028)
Sweta Singh (20075091)
Yamini Jha (20074035)
Ivy Treesa Siby (20074036)

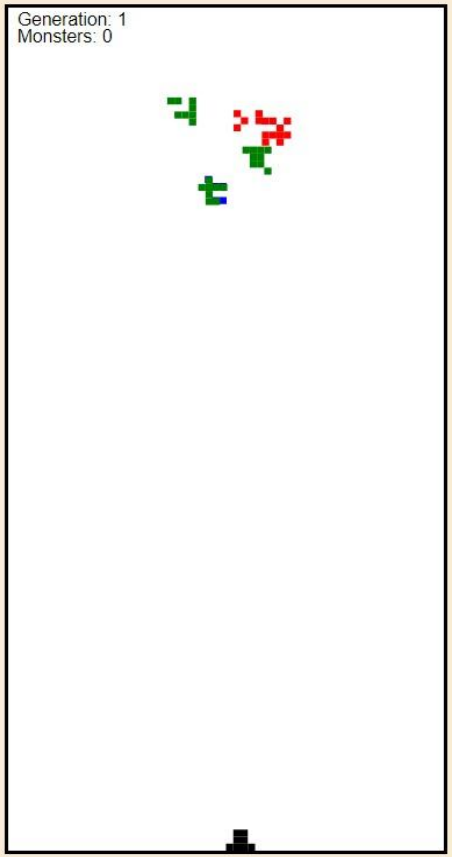
Monster Shooting Game

By using Genetic Algorithms

Overview of game

- This is a shooting game in which the player has to shoot the monsters before it surpasses the player.
- There are 6 monsters in each generation. If 5 of the monsters passes through, the game will be over.
- Player gets to shoot one bullet at the time. Player does not get another shot until the previous bullet disperse into the space or hits a target.
- Each monster has a shape that influences how they move.
- The monsters which have highest fitness values will be replicated in the next generations.
- After each 7 generations of monsters, the genetic algorithm generates a complete new population of monsters keeping only the very best of the last 7 generations.

Screenshot Of the game



Flow of Genetic Algorithm

- Encoding technique
- Population generation
- Fitness evaluation
- Selection of parents
- Crossover of parents
- Mutation of child
- Elitism & evolution

Encoding Technique (Player)

0	0	0	0
0	1	1	0
0	1	1	0
1	1	1	1

- 4x4 matrix is created to represent the shape of the player.
- If the value of the grid cell is 1 then it is colored otherwise not.
- To determine the shape of the player we convert 2D grid to 1D array.

Encoding Technique (Monster)

- We apply similar encoding techniques for monsters as that of the player.
- Initially, the shape of monster is created by generating random numbers for each grid cell.

Population Generation & Fitness evaluation

- A random population of 6 monsters are generated at the beginning of the game.
- The fitness score of the monster is how far it travelled in the vertical direction of the screen before it gets hit.

Selection of the parents

- We try to find best parents according to the fitness value of the monsters.
- For this process first we create a random probability.
- Then we compare this random probability with the fitness of a monster. If the fitness value is less than the probability then we subtract the fitness value from the probability value.
- We iterate the process till we get a monster with fitness value greater than the probability value.
- Through this process we get two parents for the crossover.

Crossover of the parents

The crossover could happen in 6 different ways:

- First horizontal half of the first parent with the second horizontal half of the second parent.
- First horizontal half of the second parent with the second horizontal half of the first parent.
- First vertical half of the first parent with the second vertical half of the second parent.
- First vertical half of the second parent with the second vertical half of the first parent.

Crossover of the parents (continuation)

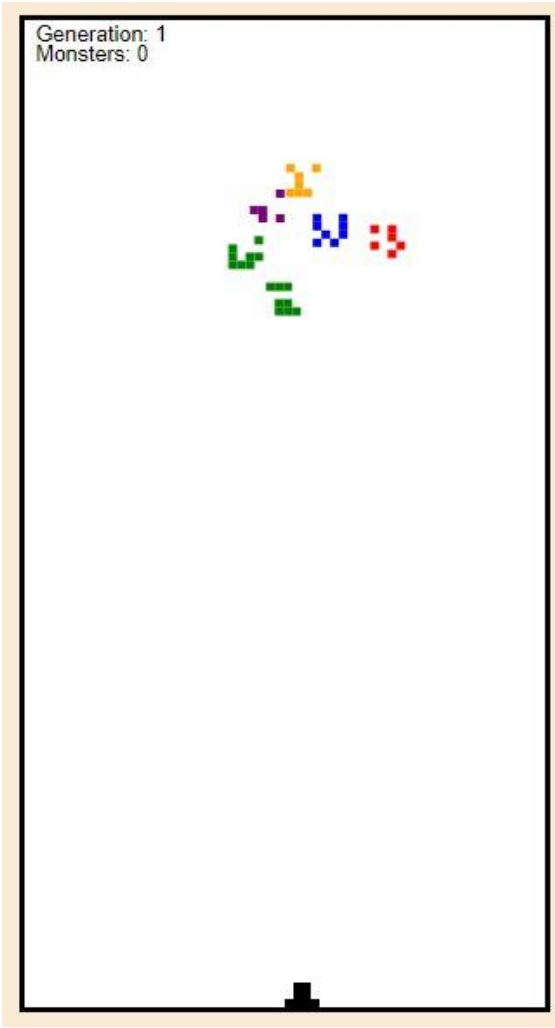
- Odd genes from the first parent, even genes from the second parent.
- Even genes from the first parent, odd genes from the second parent.

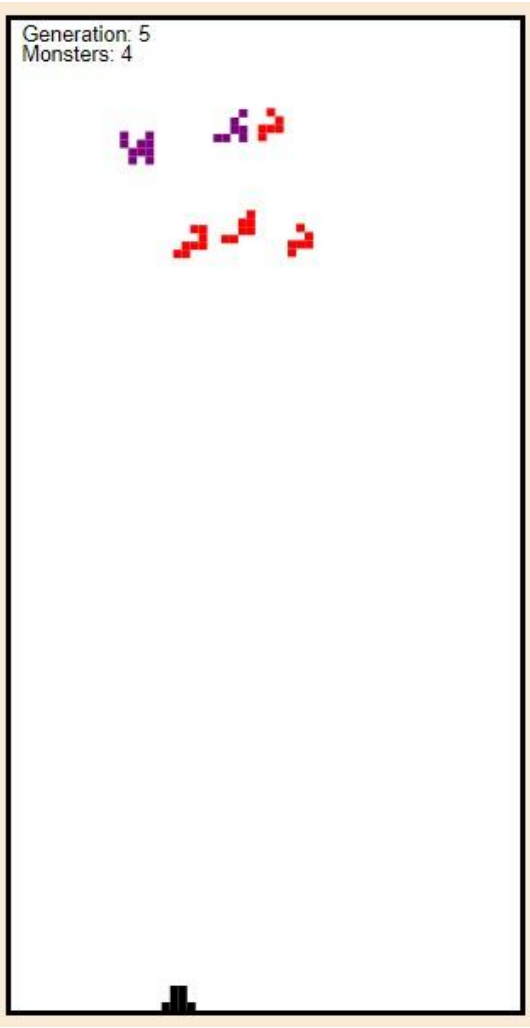
Mutation of child

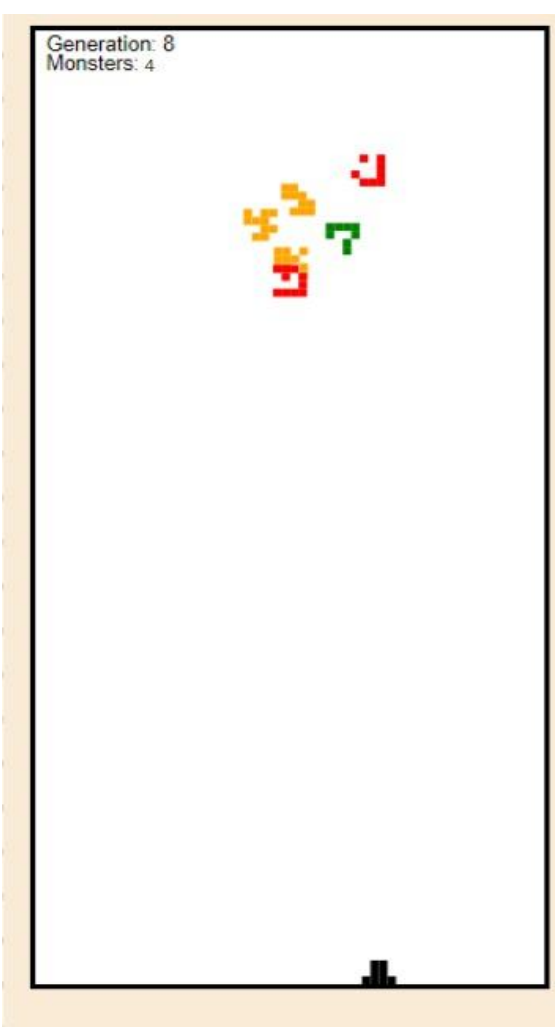
- There is a 10% chance of mutation after each crossover.
- For this process a random position is created.
- We are flipping the bit of the monster’s shape at this random position.
- Due to mutation their body shape will change ultimately their movement will also change.

Elitism & Evolution

- In each generation we create a new population and keep one monster of highest fitness value from the previous generation.
- To keep the game interesting, after 7 generations a new population of monsters will be created from the beginning and only the best of the 7 past generations will be kept.


Generation: 1
Monsters: 0


Generation: 5
Monsters: 4


Generation: 8
Monsters: 4

Thankyou