Genetic Algorithm Problem

Pink Panther Picks Diamonds

Xinyun Chen & Junjie He

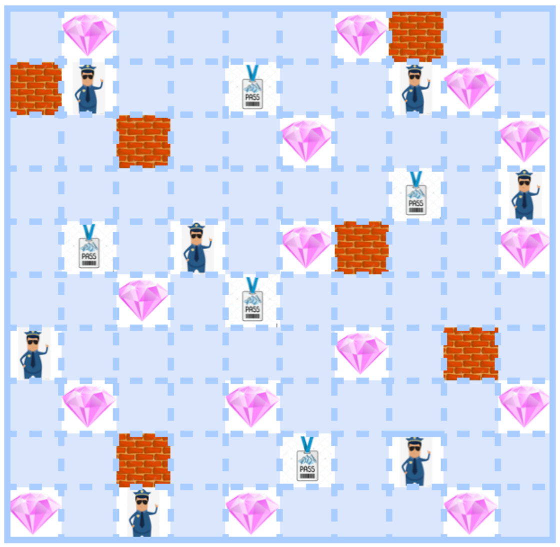
Group 509

2019 Spring

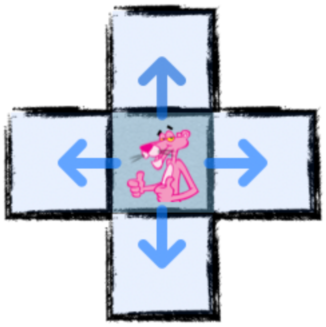
INFO6205 FINAL PROJECT REPORT

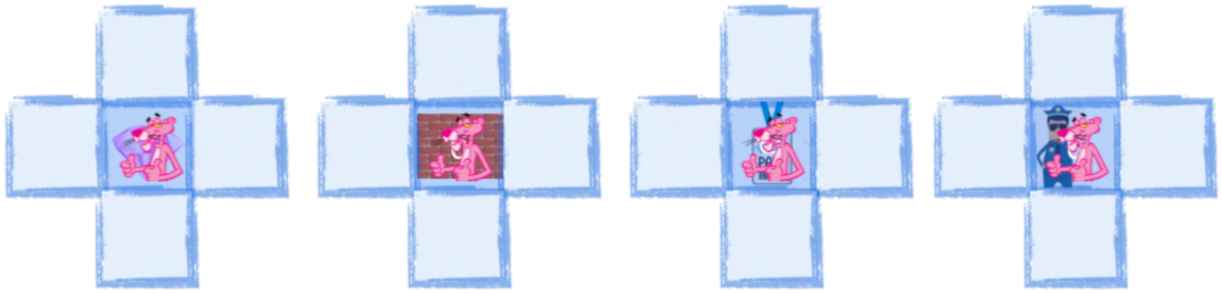
Problem Description

Pink Panther is a cartoon figure who loves collecting diamonds. There are lots of diamonds stored in a 50 * 50 maze. The pink panther's task is to collect as more diamonds as he can within 50 moves in order to get higher score. Besides, there are some cops inside the maze who's responsibility is to protect diamonds. When the pink panther encounters a cop, he will lose some scores unless he collected passes before and can use one pass each time to get through the cop without losing scores. Walls are also existed in the maze and the pink panther cannot get over the walls.

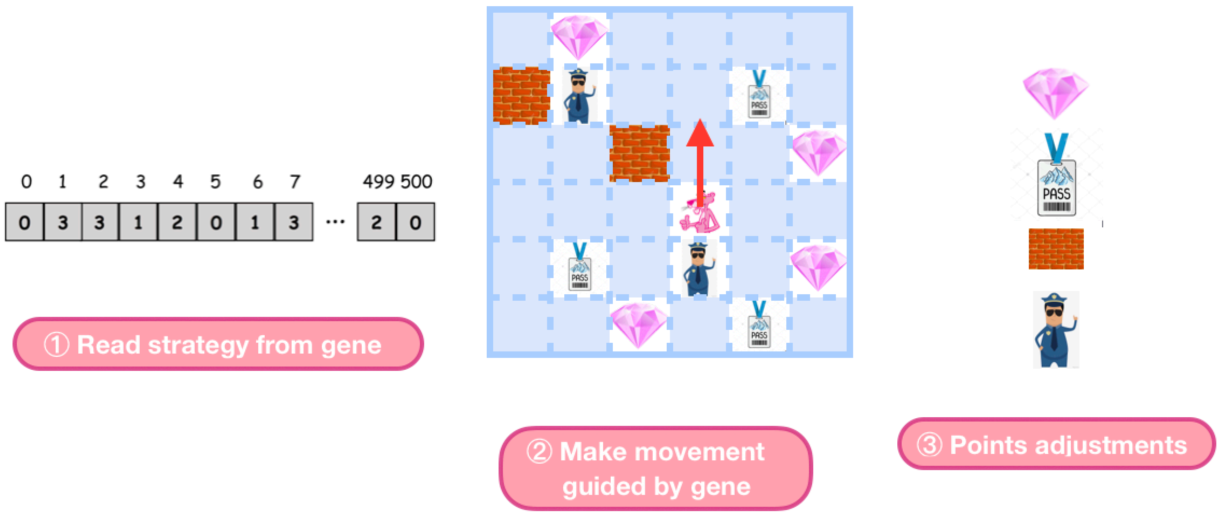


The pink panther has 4 options for next move: up, down, left, right. If the pink panther's next move leads to a diamond, he will collect the diamond. If the next move is a cop, he will lose some diamonds. If next move is out of maze or a wall, he will stay at the same place.





The collecting pattern is based on the different scenarios, which is written in each pink panther's genes. Each time before he makes a single movement, he will percept the status of his surroundings, then read the gene inside him. Pink panther will follow the gene to make actions completely.



The design of genotype (the order and content of genes) and phenotype (strategies which guide its movement) will be introduced in details later.

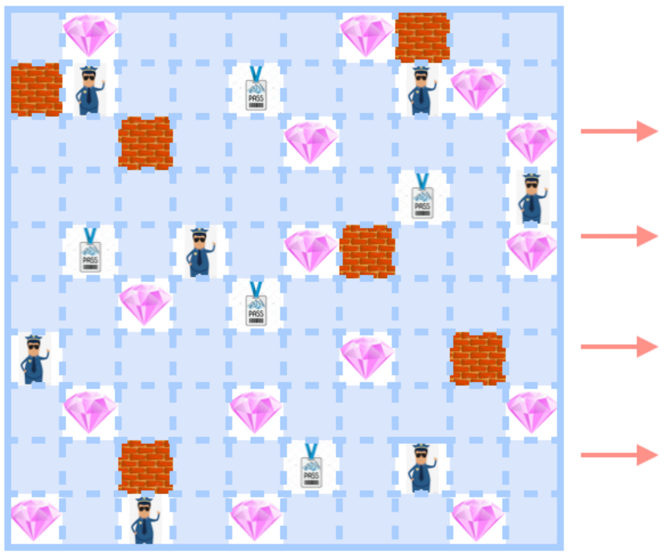
Model Design Details

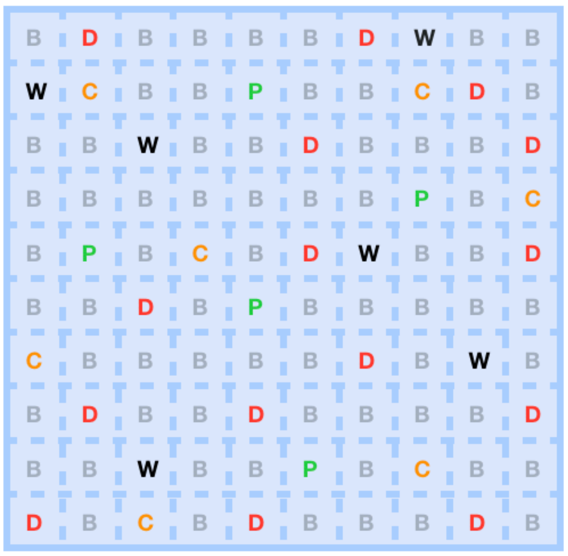
Maze Design:

There are 5 kinds of situations for a grid which are

① Diamonds ② Cops ③ Walls ④ Passes ⑤ Blank

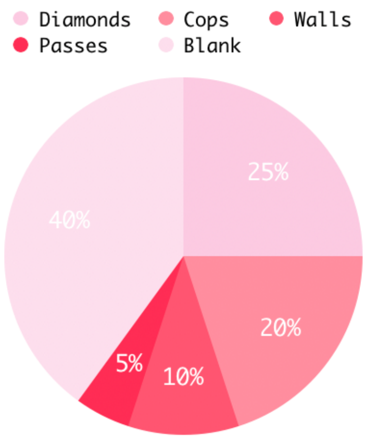
The maze is represented by a 50 * 50 matrix with 5 different characters inside (D for diamond, C for cop, W for wall, P for pass, B for blank).





B	D	B	B	B	B	D	W	B	B
W	C	B	B	P	B	B	C	D	B
B	B	W	B	B	D	B	B	B	D
B	B	B	B	B	B	B	P	B	C
B	P	B	C	B	D	W	B	B	D
B	B	D	B	P	B	B	B	B	B
C	B	B	B	B	B	D	B	W	B
B	D	B	B	D	B	B	B	B	D
B	B	W	B	B	P	B	C	B	B
D	B	C	B	D	B	B	B	D	B

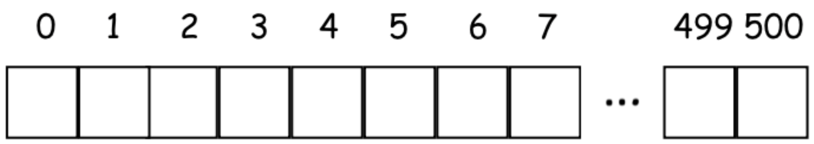
Each grid has different possibilities to have any one element described above, the possibility distribution is shown as right.



Pink Panther Gene Design:

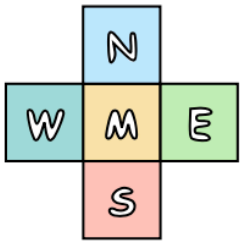
1. Gene order:

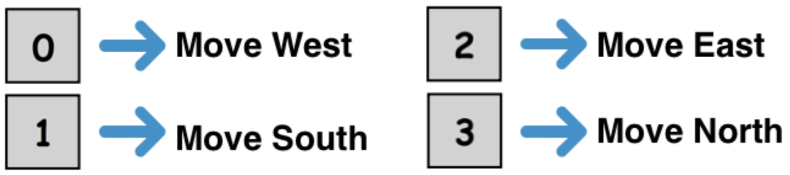
As the gene of a pink panther should be able to store the strategies for each step, so 500 gene positions should be included for each individual.



2. Genotype and Phenotype Design:

The pink panther has four directions to choose when he makes every step, so we use 0, 1, 2, 3 four numbers to represent different direction.





Fitness Function Design:

- Meet with a diamond → +5 scores
- Meet with a cop
  - if has passes → -1 pass
  - if no passes → -3 scores
- Meet with a pass → +1 pass
- Meet with a wall → -1 step, move to next step

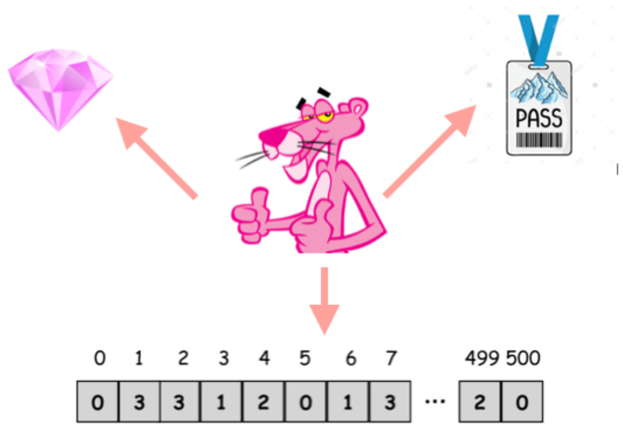
Evolution Process

Initializations:

1. Initialization of individual:

The population of each generation is set as 200 which means every generation will have 200 pink panthers (individual) .

Each pink panther will 3 properties: gene array, scores and passes.

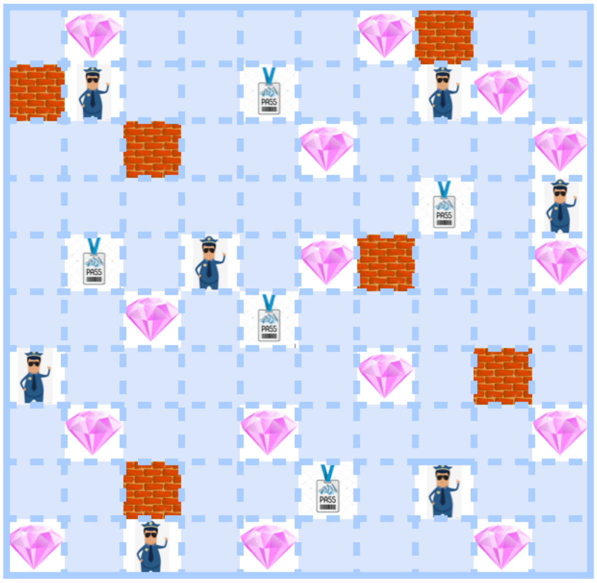


Gene array stores the strategy which decides how the pink panther moves in the maze. The passes stores the number of passes he gets in the maze. The score represents the diamonds each pink panther collects and loses during the process of traveling the maze within 500 moves. The more diamonds remain when the travel process ends, the higher score he will get.

2. Initialization of the maze and initial position:

It is a 50 * 50 2 dimensional array and all elements(diamonds, cops, passes, walls) will be added into the maze randomly. We initiate a maze once at the beginning of the application and use this maze for following process. We put a pink panther at the center of the maze each time in order to avoiding meeting the boundaries in a short time.

First Generation Initialization



Play Diamond Picking Game

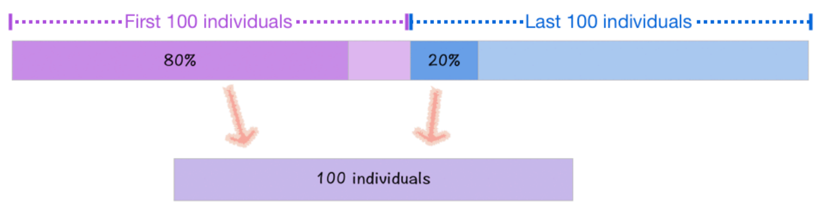
Each individual in a generation is put into the map then play the diamond picking following the strategy guided by his gene. The scores he gains are calculated by the fitness function we designed.

Fewer they met with the cops and the wall, and more diamonds and passes pick, they will definitely perform better in the game.

Rank the scores

The individual with a higher score (which is calculated by the fitness function) has a larger possibility to be selected as a parent, which means a bigger chance to propagate its genes.

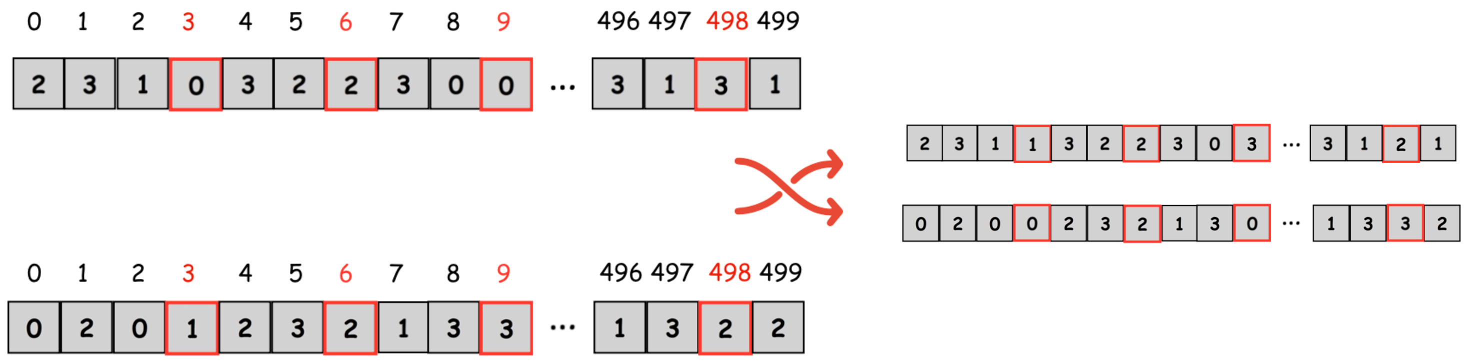
Selection Process



We choose 100 parents of the next generation in total by selecting 80% pink panthers within first 100 highest scores as superiors and 20% pink panthers will be selected randomly from remaining 100 pink panthers.

This 20% pink panthers is used to avoid local optimal solution. If all candidates are selected from high scores, it will lead to local optimal solution and prevent other candidates who maybe generate better solution after several generations selected into the group.

Reproduction Process



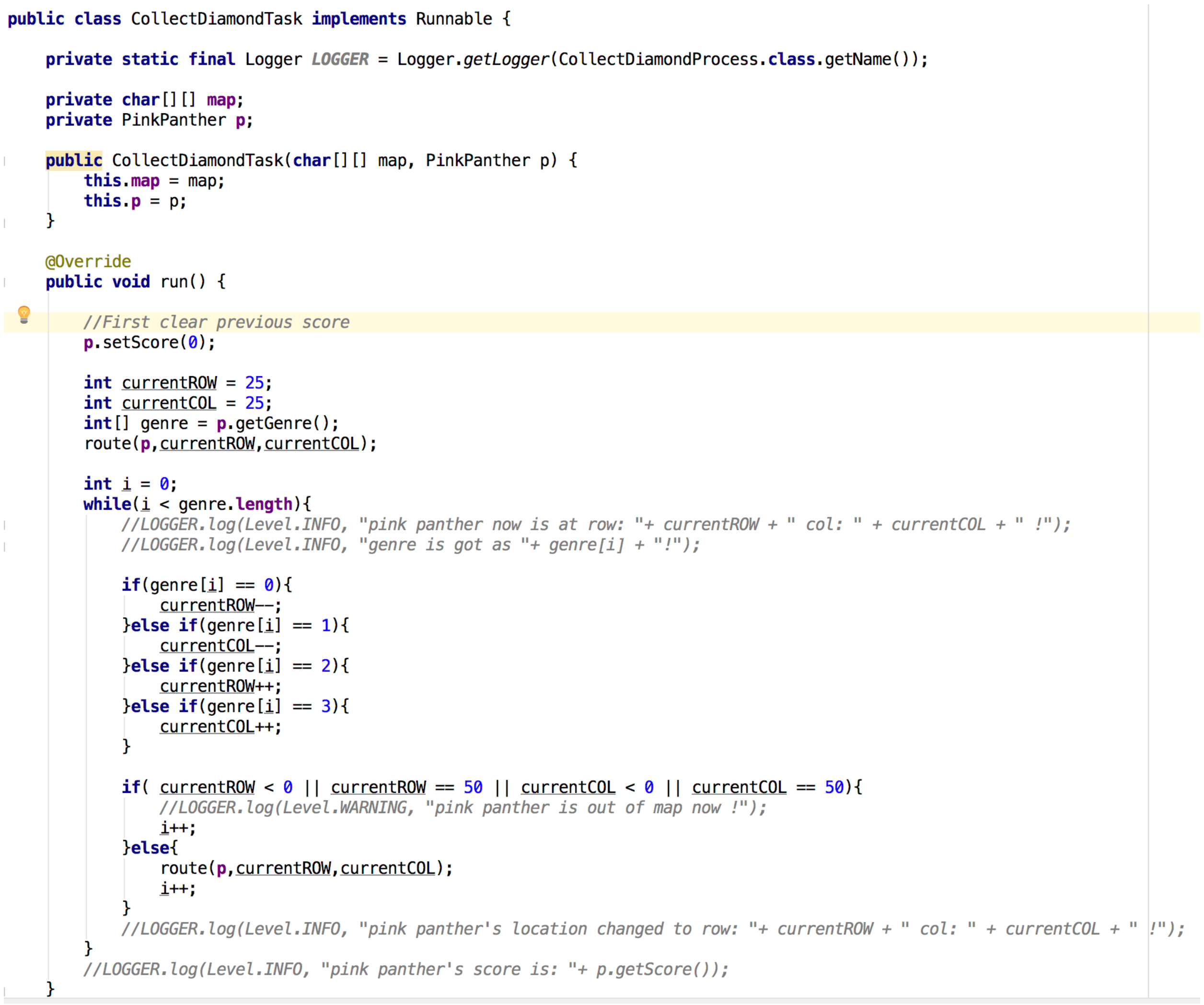
The reproduction pattern is designed as following: 100 selected parents from previous generation will generate 100 new individuals by exchange their genes on exact positions where the position of element is the multiple of 3

$$\begin{tikzcd}\text{Mutation} \arrow[d] \\ \text{New Generation}\end{tikzcd}$$

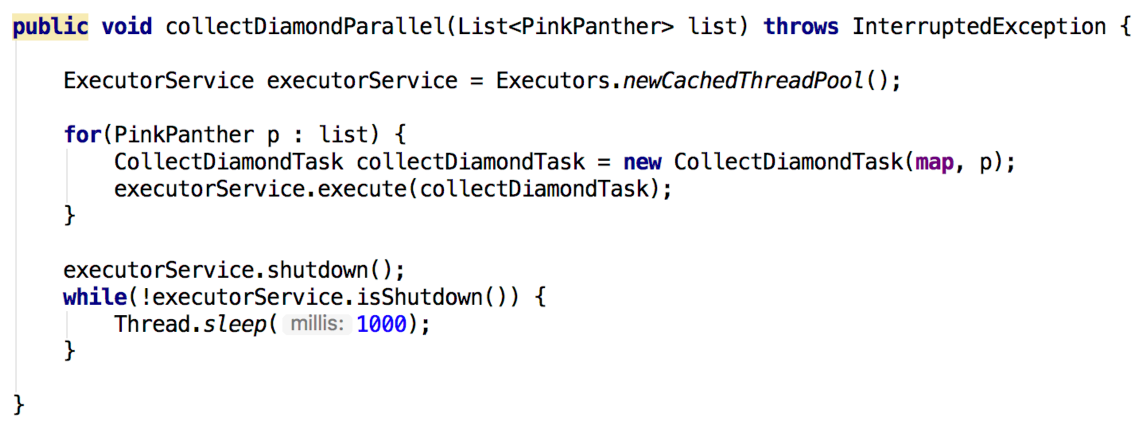
Before the new individuals are generated, mutation will be involved in. Each gene out of 500 has a 1% opportunity mutating to a new genetic code from 0-3 randomly.

Parallel Processing

The Executor Server with newCachedThreadPool is used to implement multithreaded computation. The CollectDiamondTask implements Runnable interface and it is used to compute scores for a list of individuals it receives. For each generation, 200 individuals are divided into several lists, and when the executor service shutdown, all results will collected into the result list.



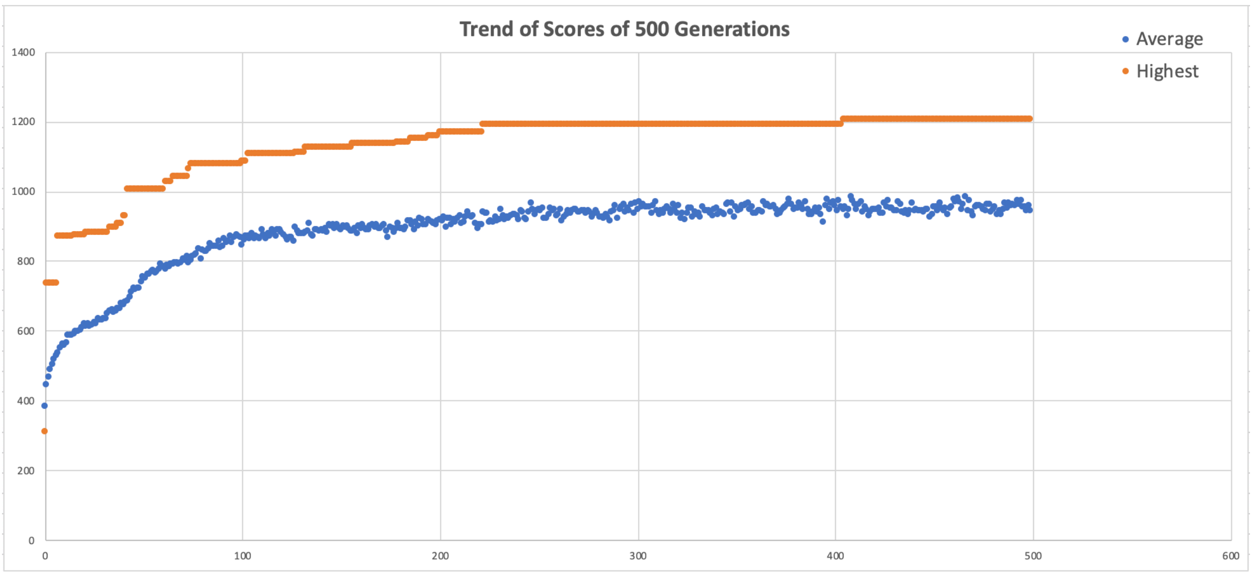
```java
public class CollectDiamondTask implements Runnable {

    private static final Logger LOGGER = Logger.getLogger(CollectDiamondProcess.class.getName());

    private char[][] map;
    private PinkPanther p;

    public CollectDiamondTask(char[][] map, PinkPanther p) {
        this.map = map;
        this.p = p;
    }

    @Override
    public void run() {

        //First clear previous score
        p.setScore(0);

        int currentROW = 25;
        int currentCOL = 25;
        int[] genre = p.getGenre();
        route(p,currentROW,currentCOL);

        int i = 0;
        while(i < genre.length){
            //LOGGER.log(Level.INFO, "pink panther now is at row: "+ currentROW + " col: " + currentCOL + " !");
            //LOGGER.log(Level.INFO, "genre is got as "+ genre[i] + "!");

            if(genre[i] == 0){
                currentROW--;
            }else if(genre[i] == 1){
                currentCOL--;
            }else if(genre[i] == 2){
                currentROW++;
            }else if(genre[i] == 3){
                currentCOL++;
            }

            if( currentROW < 0 || currentROW == 50 || currentCOL < 0 || currentCOL == 50){
                //LOGGER.log(Level.WARNING, "pink panther is out of map now !");
                i++;
            }else{
                route(p,currentROW,currentCOL);
                i++;
            }
            //LOGGER.log(Level.INFO, "pink panther's location changed to row: "+ currentROW + " col: " + currentCOL + " !");
        }
        //LOGGER.log(Level.INFO, "pink panther's score is: "+ p.getScore());
    }
```



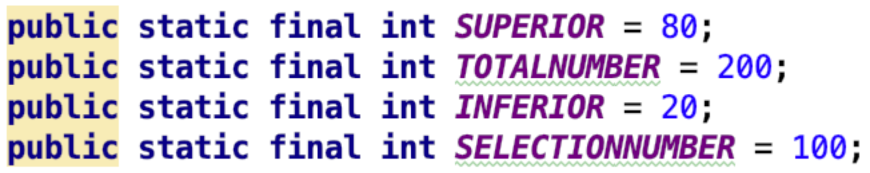
```java
public void collectDiamondParallel(List<PinkPanther> list) throws InterruptedException {

    ExecutorService executorService = Executors.newCachedThreadPool();

    for(PinkPanther p : list) {
        CollectDiamondTask collectDiamondTask = new CollectDiamondTask(map, p);
        executorService.execute(collectDiamondTask);
    }

    executorService.shutdown();
    while(!executorService.isShutdown()) {
        Thread.sleep( millis: 1000);
    }

}
```

Results & Conclusions

We can easily find that the first generation could perform really bad because their genes are generated randomly which leads to a small amount of good genes for collecting lots of diamonds and earning high scores. Hence, we select the parents of next generation by the score of the individuals, the individuals who can make relatively higher scores will be filtered to be remained and propagate their genes. The trend is in the direction of an increasing performance on the game, their genes which hold a good strategy will transfer to their child increasingly.

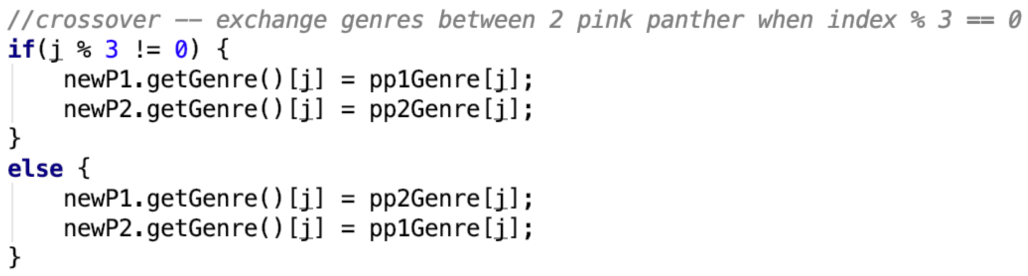


The diagram above is the result came out from one of the evolving process. Both the best and average performance in every generations continuously made progress and converge to where theocratically the best point. With the increase of the generation, the difference between two adjacent generations becomes smaller and the slope of the curve is close to 0. The curve converges around x = 200, which indicates a good strategy for selection and evolution process.



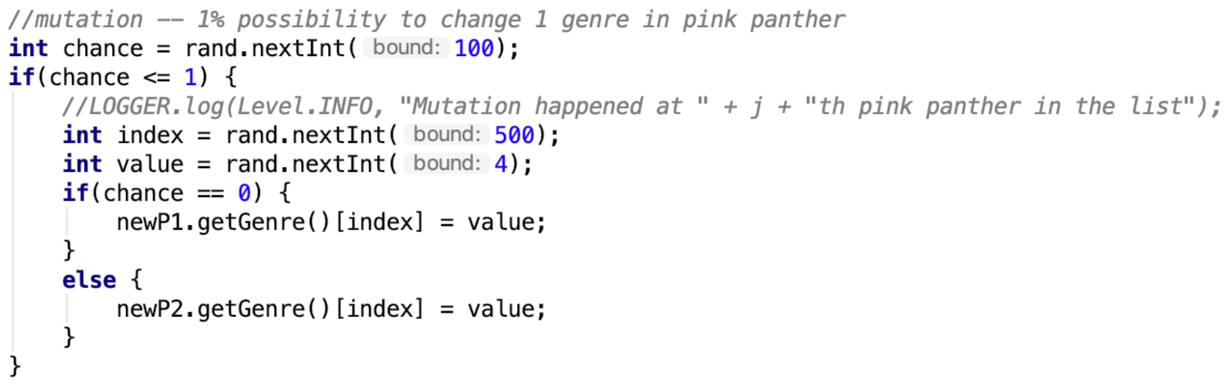
```java
public static final int SUPERIOR = 80;
public static final int TOTALNUMBER = 200;
public static final int INFERIOR = 20;
public static final int SELECTIONNUMBER = 100;
```

The screenshot above is the configuration of parameters for the evolution process. 80% of the parent individuals are selected from the first 100 individuals who got high scores and 20% are chosen from the last 100 individuals randomly. In this way, the good genes can be kept as many as possible, at the mean time, the individuals who have a low score calculated by the fitness function still have opportunities to be selected.



```java
//crossover -- exchange genres between 2 pink panther when index % 3 == 0
if(j % 3 != 0) {
    newP1.getGenre()[j] = pp1Genre[j];
    newP2.getGenre()[j] = pp2Genre[j];
}
else {
    newP1.getGenre()[j] = pp2Genre[j];
    newP2.getGenre()[j] = pp1Genre[j];
}
```

In two parents' genes array, the genre which is positioned at multiply of 3 are exchanged and as a result, two new individuals are generated. One of three amount of genes exchange can guarantee a fast speed of convergence.



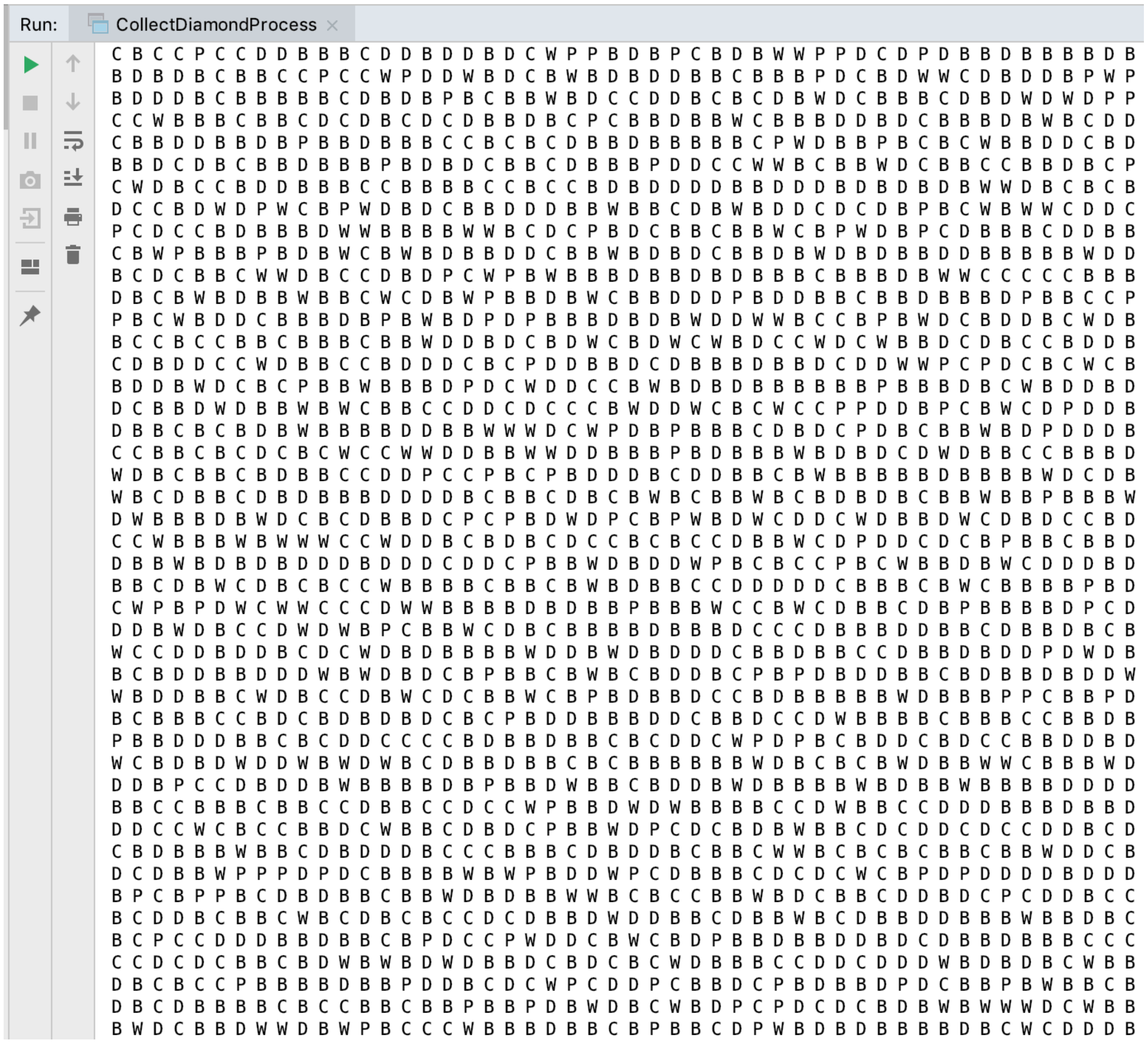
```java
//mutation -- 1% possibility to change 1 genre in pink panther
int chance = rand.nextInt( bound: 100);
if(chance <= 1) {
    //LOGGER.log(Level.INFO, "Mutation happened at " + j + "th pink panther in the list");
    int index = rand.nextInt( bound: 500);
    int value = rand.nextInt( bound: 4);
    if(chance == 0) {
        newP1.getGenre()[index] = value;
    }
    else {
        newP2.getGenre()[index] = value;
    }
}
```

Although the mutation rate for each gene is only 1%, there are 500 genes for each of 200 individuals in one generation, so there will always have a group of genes change from the old to new one.

Evidence of Running

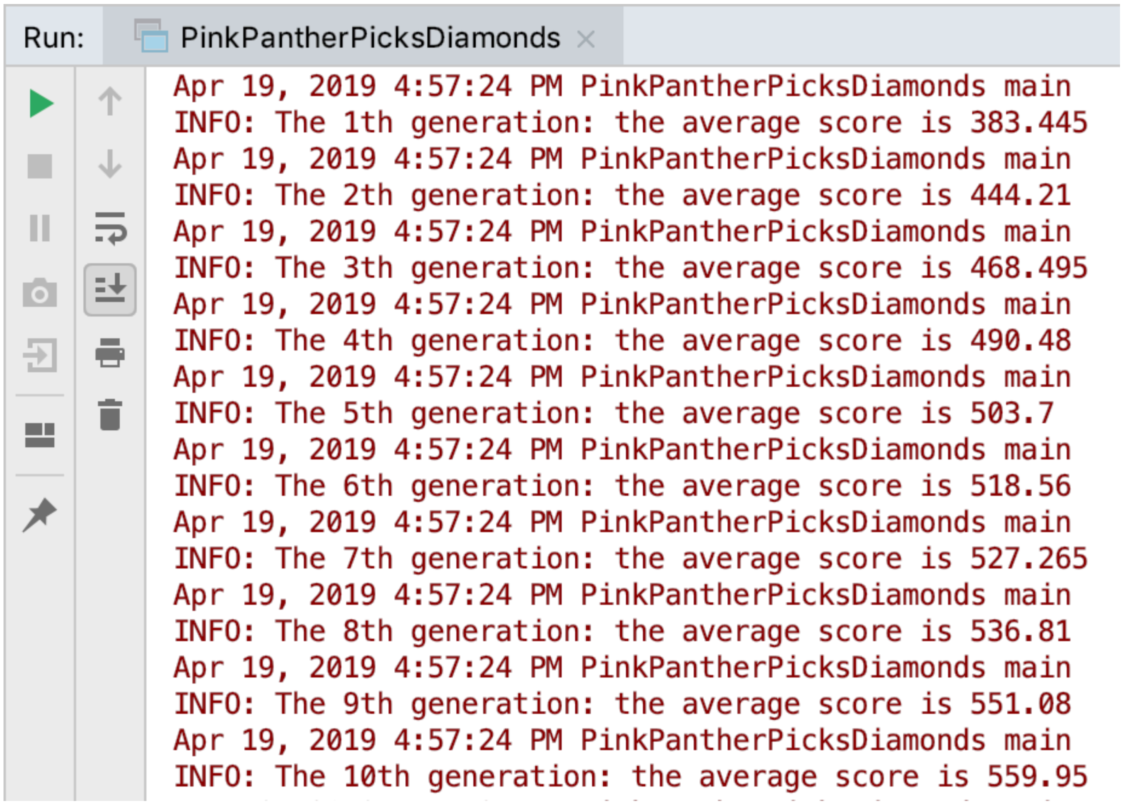
Program Outputs:

1. Map Initialization

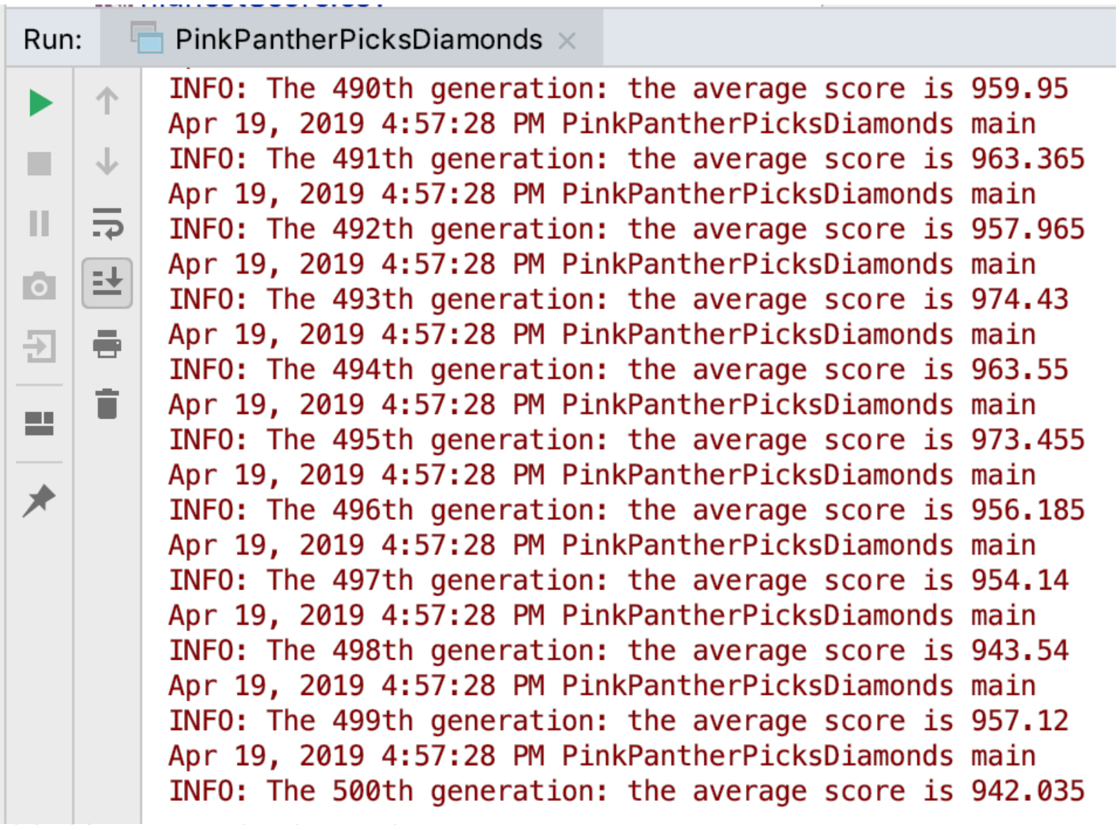


```
C B C C P C C D D B B B C D D B D D B D C W P P B D B P C B D B W W P P D C D P D B B D B B B B D B
B D B D B C B B C C P C C W P D D W B D C B W B D B D D B B C B B B P D C B D W W C D B D D B P W P
B D D D B C B B B B B C D B D B P B C B B W B D C C D D B C B C D B W D C B B B C D B D W D W D P P
C C W B B B C B B C D C D B C D C D B B D B C P C B B D B B W C B B B D D B D C B B B D B W B C D D
C B B D D B B D B P B B D B B B C C B C B C D B B D B B B B B C P W D B B P B C B C W B B D D C B D
B B D C D B C B B D B B B P B D B D C B B C D B B B P D D C C W W B C B B W D C B B C C B B D B C P
C W D B C C B D D B B B C C B B B B C C B C C B D B D D D D B B D D D B D B D B D B W W D B C B C B
D C C B D W D P W C B P W D B D C B B D D D B B W B B C D B W B D D C D C D B P B C W B W W C D D C
P C D C C B D B B B D W W B B B B W W B C D C P B D C B B C B B W C B P W D B P C D B B B C D D B B
C B W P B B B P B D B W C B W B D B B D D C B B W B D B D C B B D B W D B D B B D D B B B B B W D D
B C D C B B C W W D B C C D B D P C W P B W B B B D B B D B D B B B C B B B D B W W C C C C B B B
D B C B W B D B B W B B C W C D B W P B B D B W C B B D D D P B D D B B C B B D B B B D P B B C C P
P B C W B D D C B B B D B P B W B D P D P B B B D B D B W D D W W B C C B P B W D C B D D B C W D B
B C C B C C B B C B B B C B B W D D B D C B D W C B D W C W B D C C W D C W B B D C D B C C B D D B
C D B D D C C W D B B C C B D D D C B C P D D B B D C D B B B D B B D C D D W W P C P D C B C W C B
B D D B W D C B C P B B W B B B D P D C W D D C C B W B D B D B B B B B B P B B B D B C W B D D B D
D C B B D W D B B W B W C B B C C D D C D C C C B W D D W C B C W C C P P D D B P C B W C D P D D B
D B B C B C B D B W B B B B D D B B W W W D C W P D B P B B B C D B D C P D B C B B W B D P D D D B
C C B B C B C D C B C W C C W W D D B B W W D D B B B P B D B B B W B D B D C D W D B B C C B B B D
W D B C B B C B D B B C C D D P C C P B C P B D D D B C D D B B C B W B B B B B D B B B B W D C D B
W B C D B B C D B D B B B D D D D B C B B C D B C B W B C B B W B C B D B D B C B B W B B P B B B W
D W B B B D B W D C B C D B B D C P C P B D W D P C B P W B D W C D D C W D B B D W C D B D C C B D
C C W B B B W B W W W C C W D D B C B D B C D C C B C B C C D B B W C D P D D C D C B P B B C B B D
D B B W B D B D B D D D B D D D C D D C P B B W D B D D W P B C B C C P B C W B B D B W C D D D B D
B B C D B W C D B C B C C W B B B B C B B C B W B D B B C C D D D D D C B B B C B W C B B B B P B D
C W P B P D W C W W C C C D W W B B B B D B D B B P B B B W C C B W C D B B C D B P B B B B D P C D
D D B W D B C C D W D W B P C B B W C D B C B B B B D B B B D C C C D B B B D D B B C D B B D B C B
W C C D D B D D B C D C W D B D B B B B W D D B W D B D D D C B B D B B C C D B B D B D D P D W D B
B C B D D B B D D D W B W D B D C B P B B C B W B C B D D B C P B P D B D D B B C B D B B D B D D W
W B D D B B C W D B C C D B W C D C B B W C B P B D B B D C C B D B B B B B W D B B B P P C B B P D
B C B B B C C B D C B D B D B D C B C P B D D B B B D D C B B D C C D W B B B B C B B B C C B B D B
P B B D D D B B C B C D D C C C C B D B B D B B C B C D D C W P D P B C B D D C B D C C B B D D B D
W C B D B D W D D W B W D W B C D B B D B B C B C B B B B B B W D B C B C B W D B B W W C B B B W D
D D B P C C D B D D B W B B B B D B P B B D W B B C B D D B W D B B B B W B D B B W B B B B D D D D
B B C C B B B C B B C C D B B C C D C C W P B B D W D W B B B B C C D W B B C C D D D B B B D B B D
D D C C W C B C C B B D C W B B C D B D C P B B W D P C D C B D B W B B C D C D D C D C C D D B C D
C B D B B B W B B C D B D D D B C C C B B B C D B D D B C B B C W W B C B C B C B B C B B W D D C B
D C D B B W P P P D P D C B B B B W B W P B D D W P C D B B B C D C D C W C B P D P D D D D B D D D
B P C B P P B C D B D B B C B B W D B D B B W W B C B C C B B W B D C B B C D D B D C P C D D B C C
B C D D B C B B C W B C D B C B C C D C D B B D W D D B B C D B B W B C D B B D D B B B W B B D B C
B C P C C D D D B B D B B C B P D C C P W D D C B W C B D P B B D B B D D B D C D B B D B B B C C C
C C D C D C B B C B D W B W B D W D B B D C B D C B C W D B B B C C D D C D D D W B D B D B C W B B
D B C B C C P B B B B D B B P D D B C D C W P C D D P C B B D C P B D B B D P D C B B P B W B B C B
D B C D B B B B C B C C B B C B B P B B P D B W D B C W B D P C P D C D C B D B W B W W W D C W B B
B W D C B B D W W D B W P B C C C W B B B D B B C B P B B C D P W B D B D B B B B D B C W C D D D B
```

2. Evolution Process



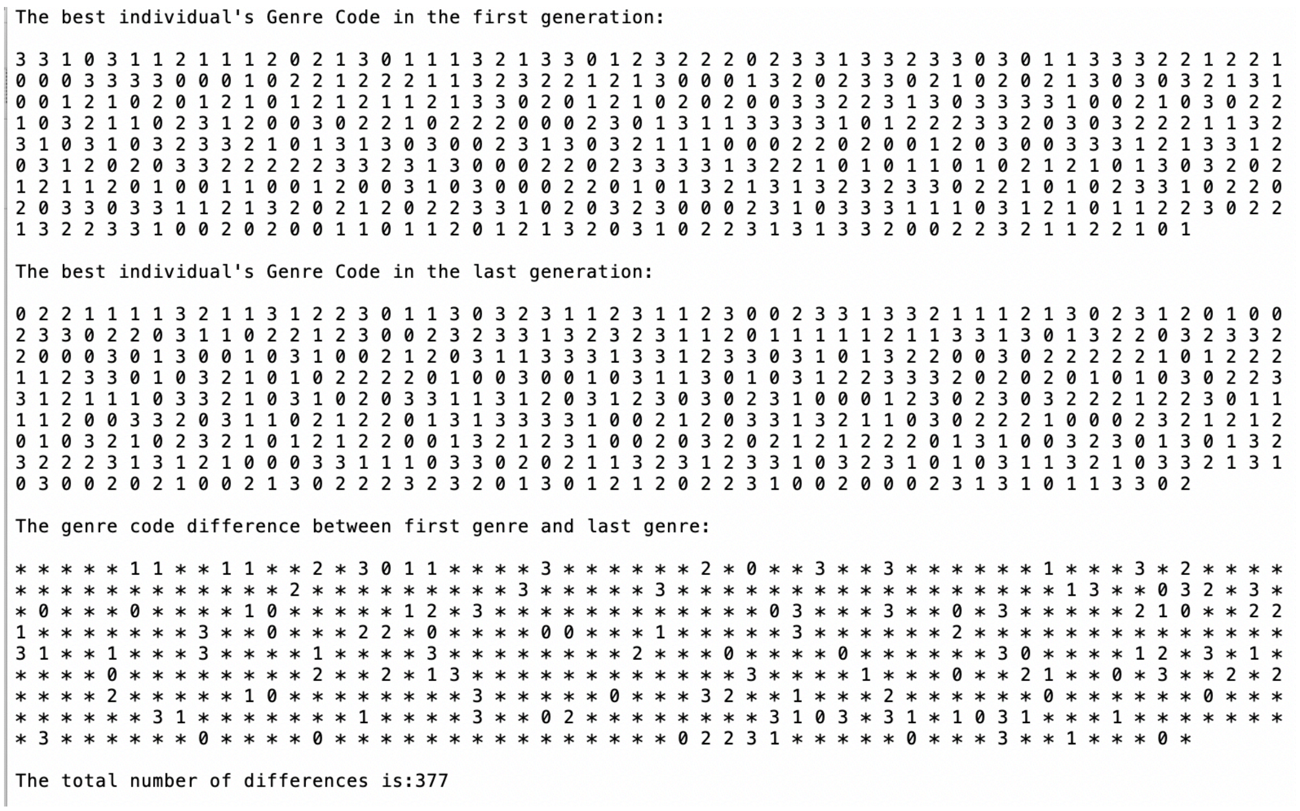
```
Apr 19, 2019 4:57:24 PM PinkPantherPicksDiamonds main
INFO: The 1th generation: the average score is 383.445
Apr 19, 2019 4:57:24 PM PinkPantherPicksDiamonds main
INFO: The 2th generation: the average score is 444.21
Apr 19, 2019 4:57:24 PM PinkPantherPicksDiamonds main
INFO: The 3th generation: the average score is 468.495
Apr 19, 2019 4:57:24 PM PinkPantherPicksDiamonds main
INFO: The 4th generation: the average score is 490.48
Apr 19, 2019 4:57:24 PM PinkPantherPicksDiamonds main
INFO: The 5th generation: the average score is 503.7
Apr 19, 2019 4:57:24 PM PinkPantherPicksDiamonds main
INFO: The 6th generation: the average score is 518.56
Apr 19, 2019 4:57:24 PM PinkPantherPicksDiamonds main
INFO: The 7th generation: the average score is 527.265
Apr 19, 2019 4:57:24 PM PinkPantherPicksDiamonds main
INFO: The 8th generation: the average score is 536.81
Apr 19, 2019 4:57:24 PM PinkPantherPicksDiamonds main
INFO: The 9th generation: the average score is 551.08
Apr 19, 2019 4:57:24 PM PinkPantherPicksDiamonds main
INFO: The 10th generation: the average score is 559.95
```



```
INFO: The 490th generation: the average score is 959.95
Apr 19, 2019 4:57:28 PM PinkPantherPicksDiamonds main
INFO: The 491th generation: the average score is 963.365
Apr 19, 2019 4:57:28 PM PinkPantherPicksDiamonds main
INFO: The 492th generation: the average score is 957.965
Apr 19, 2019 4:57:28 PM PinkPantherPicksDiamonds main
INFO: The 493th generation: the average score is 974.43
Apr 19, 2019 4:57:28 PM PinkPantherPicksDiamonds main
INFO: The 494th generation: the average score is 963.55
Apr 19, 2019 4:57:28 PM PinkPantherPicksDiamonds main
INFO: The 495th generation: the average score is 973.455
Apr 19, 2019 4:57:28 PM PinkPantherPicksDiamonds main
INFO: The 496th generation: the average score is 956.185
Apr 19, 2019 4:57:28 PM PinkPantherPicksDiamonds main
INFO: The 497th generation: the average score is 954.14
Apr 19, 2019 4:57:28 PM PinkPantherPicksDiamonds main
INFO: The 498th generation: the average score is 943.54
Apr 19, 2019 4:57:28 PM PinkPantherPicksDiamonds main
INFO: The 499th generation: the average score is 957.12
Apr 19, 2019 4:57:28 PM PinkPantherPicksDiamonds main
INFO: The 500th generation: the average score is 942.035
```

INF

3. Genes Comparison

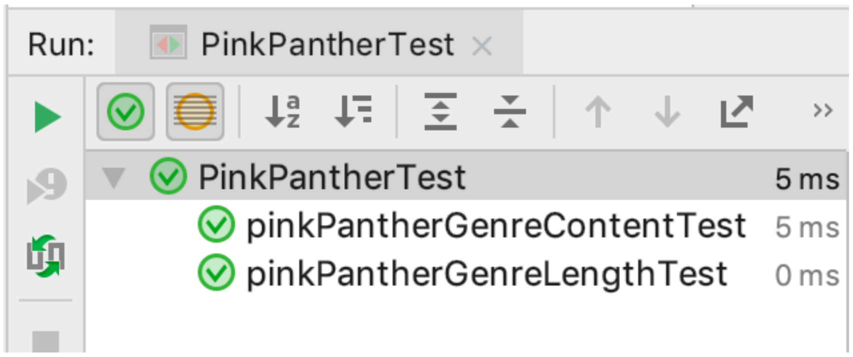


```
The best individual's Genre Code in the first generation:

3 3 1 0 3 1 1 2 1 1 1 2 0 2 1 3 0 1 1 1 3 2 1 3 3 0 1 2 3 2 2 2 0 2 3 3 1 3 3 2 3 3 0 3 0 1 1 3 3 3 2 2 1 2 2 1
0 0 0 3 3 3 3 0 0 0 1 0 2 2 1 2 2 2 1 1 3 2 3 2 2 1 2 1 3 0 0 0 1 3 2 0 2 3 3 0 2 1 0 2 0 2 1 3 0 3 0 3 2 1 3 1
0 0 1 2 1 0 2 0 1 2 1 0 1 2 1 2 1 1 2 1 3 3 0 2 0 1 2 1 0 2 0 2 0 0 3 3 2 2 3 1 3 0 3 3 3 3 1 0 0 2 1 0 3 0 2 2
1 0 3 2 1 1 0 2 3 1 2 0 0 3 0 2 2 1 0 2 2 2 0 0 0 2 3 0 1 3 1 1 3 3 3 3 1 0 1 2 2 2 3 3 2 0 3 0 3 2 2 2 1 1 3 2
3 1 0 3 1 0 3 2 3 3 2 1 0 1 3 1 3 0 3 0 0 2 3 1 3 0 3 2 1 1 1 0 0 0 2 2 0 2 0 0 1 2 0 3 0 0 3 3 3 1 2 1 3 3 1 2
0 3 1 2 0 2 0 3 3 2 2 2 2 2 3 3 2 3 1 3 0 0 0 2 2 0 2 3 3 3 3 1 3 2 2 1 0 1 0 1 1 0 1 0 2 1 2 1 0 1 3 0 3 2 0 2
1 2 1 1 2 0 1 0 0 1 1 0 0 1 2 0 0 3 1 0 3 0 0 0 2 2 0 1 0 1 3 2 1 3 1 3 2 3 2 3 3 0 2 2 1 0 1 0 2 3 3 1 0 2 2 0
2 0 3 3 0 3 3 1 1 2 1 3 2 0 2 1 2 0 2 2 3 3 1 0 2 0 3 2 3 0 0 0 2 3 1 0 3 3 3 1 1 1 0 3 1 2 1 0 1 1 2 2 3 0 2 2
1 3 2 2 3 3 1 0 0 2 0 2 0 0 1 1 0 1 1 2 0 1 2 1 3 2 0 3 1 0 2 2 3 1 3 1 3 3 2 0 0 2 2 3 2 1 1 2 2 1 0 1

The best individual's Genre Code in the last generation:

0 2 2 1 1 1 1 3 2 1 1 3 1 2 2 3 0 1 1 3 0 3 2 3 1 1 2 3 1 1 2 3 0 0 2 3 3 1 3 3 2 1 1 1 2 1 3 0 2 3 1 2 0 1 0 0
2 3 3 0 2 2 0 3 1 1 0 2 2 1 2 3 0 0 2 3 2 3 3 1 3 2 3 2 3 1 1 2 0 1 1 1 1 1 2 1 1 3 3 1 3 0 1 3 2 2 0 3 2 3 3 2
2 0 0 0 3 0 1 3 0 0 1 0 3 1 0 0 2 1 2 0 3 1 1 3 3 3 1 3 3 1 2 3 3 0 3 1 0 1 3 2 2 0 0 3 0 2 2 2 2 2 1 0 1 2 2 2
1 1 2 3 3 0 1 0 3 2 1 0 1 0 2 2 2 2 0 1 0 0 3 0 0 1 0 3 1 1 3 0 1 0 3 1 2 2 3 3 3 2 0 2 0 2 0 1 0 1 0 3 0 2 2 3
3 1 2 1 1 1 0 3 3 2 1 0 3 1 0 2 0 3 3 1 1 3 1 2 0 3 1 2 3 0 3 0 2 3 1 0 0 0 1 2 3 0 2 3 0 3 2 2 2 1 2 2 3 0 1 1
1 1 2 0 0 3 3 2 0 3 1 1 0 2 1 2 2 0 1 3 1 3 3 3 3 1 0 0 2 1 2 0 3 3 1 3 2 1 1 0 3 0 2 2 2 1 0 0 0 2 3 2 1 2 1 2
0 1 0 3 2 1 0 2 3 2 1 0 1 2 1 2 2 0 0 1 3 2 1 2 3 1 0 0 2 0 3 2 0 2 1 2 1 2 2 2 0 1 3 1 0 0 3 2 3 0 1 3 0 1 3 2
3 2 2 2 3 1 3 1 2 1 0 0 0 3 3 1 1 1 0 3 3 0 2 0 2 1 1 3 2 3 1 2 3 3 1 0 3 2 3 1 0 1 0 3 1 1 3 2 1 0 3 3 2 1 3 1
0 3 0 0 2 0 2 1 0 0 2 1 3 0 2 2 2 3 2 3 2 0 1 3 0 1 2 1 2 0 2 2 3 1 0 0 2 0 0 0 2 3 1 3 1 0 1 1 3 3 0 2

The genre code difference between first genre and last genre:

* * * * * 1 1 * * 1 1 * * 2 * 3 0 1 1 * * * * 3 * * * * * * 2 * 0 * * 3 * * 3 * * * * * * 1 * * * 3 * 2 * * * *
* * * * * * * * * * * * 2 * * * * * * * * * 3 * * * * * 3 * * * * * * * * * * * * * * * * * 1 3 * * 0 3 2 * 3 *
* 0 * * * 0 * * * * 1 0 * * * * * 1 2 * 3 * * * * * * * * * * * * * 0 3 * * * 3 * * 0 * 3 * * * * * 2 1 0 * * 2 2
1 * * * * * * * 3 * * 0 * * * 2 2 * 0 * * * * 0 0 * * * 1 * * * * * 3 * * * * * * 2 * * * * * * * * * * * * * *
3 1 * * 1 * * * 3 * * * * 1 * * * * 3 * * * * * * * * * 2 * * * 0 * * * * * 0 * * * * * * 3 0 * * * * 1 2 * 3 * 1 *
* * * * 0 * * * * * * * * * 2 * * 2 * 1 3 * * * * * * * * * * * * * * 3 * * * * * 1 * * * 0 * * 2 1 * * 0 * 3 * * 2 * 2
* * * * 2 * * * * * * 1 0 * * * * * * * * * 3 * * * * * * 0 * * * 3 2 * * 1 * * * 2 * * * * * * * 0 * * * * * * 0 * * *
* * * * * * 3 1 * * * * * * * * 1 * * * * 3 * * 0 2 * * * * * * * * * 3 1 0 3 * 3 1 * 1 0 3 1 * * * 1 * * * * * * *
* 3 * * * * * * 0 * * * * 0 * * * * * * * * * * * * * * * * 0 2 2 3 1 * * * * * 0 * * * 3 * * 1 * * * 0 *

The total number of differences is:377
```

The snapshot above shows that after 500 generations, 377 genes has changed which leads to a better performance of an individual.

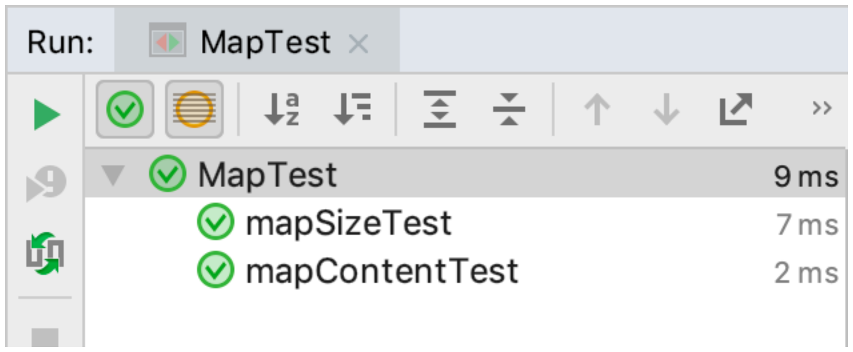
Unit Tests

Pink Panther Tests:



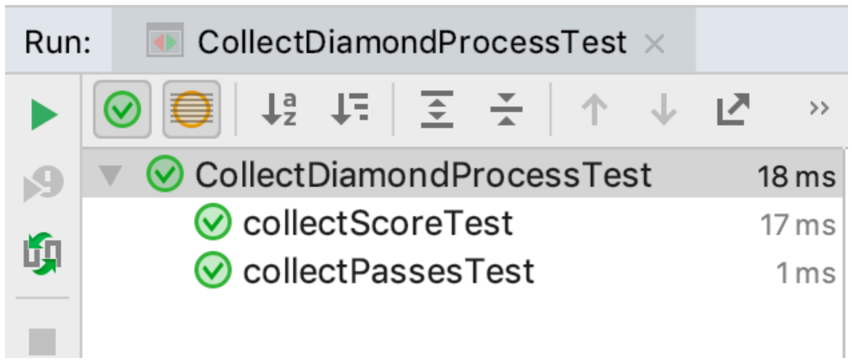
```
PinkPantherTest                    5 ms
    pinkPantherGenreContentTest    5 ms
    pinkPantherGenreLengthTest     0 ms
```

Map Tests:



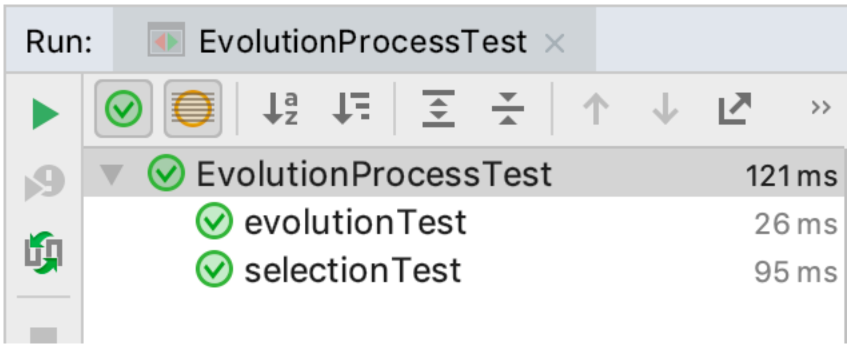
```
MapTest                 9 ms
    mapSizeTest         7 ms
    mapContentTest      2 ms
```

Collect Process Tests:



```
CollectDiamondProcessTest    18 ms
    collectScoreTest         17 ms
    collectPassesTest         1 ms
```

Evolution Tests:



```
EvolutionProcessTest   121 ms
    evolutionTest       26 ms
    selectionTest       95 ms
```