Ariel University, School of Computer Science, 2022
Introduction to Computer Science

Ex3 - Basic algorithmic programming: 2D arrays + Intro to OOP

Abstract:

This assignment introduces the design and implementation of algorithms on 2D arrays (aka images). The main task in this assignment is to implement a set of "image processing" algorithms over 2D arrays representing an image, a maze, or simple 2D matrix. Moreover, in this assignment you are required to implement a JUnit class - to test all your functions.

To Do

1. Download Ex3.zip, un compress it.
2. Run the Ex3_v0.1-obf.jar file (java -jar Ex3_v0.1-obf.jar). This is a complete solution to Ex3 - you should implement your solution accordingly.
3. Create a new project (named Ex3), and run the class Ex3.java, You should get a simple GUI which will allow you to test your code.
4. Implement your solution in the given **MyMap2D** and **Ex3**.java. Make sure your program creates the same GUI as shown in Figure 1.
5. Implement a detailed JUnit class (this time there is no skeleton is given to you - make sure you implement a complete testing suite in class **MyMap2DTest.java.**
6. **Add a detailed documentation (in English) to the Ex3.java** file with the related description for each function.
7. Make sure to submit ALL the needed classes + an "executable" jar fine named Ex3.jar that can be run by double clicking.

Notes:

1. Work in pairs (or alone)! - you can talk about this assignment with anyone in class - but when writing your solution DIY!. Please go over this document which covers the School's honesty policy.
2. **Make sure you write your IDs (ID1 & ID2) in the files Ex3.java and MyMap2D.java.**

3. The implementation of the function should be as efficient and elegant as possible.

4. Submission guidelines: your solution should be written as a java project the following files are required: **Ex3.java, MyMap2D.java, Ex3.jar,** make sure to submit these files! (Map2D.java, StdDraw_Ex3.java and Point2D.java are not required). You can add additional classes to your implementation.

5. Your solution should be submitted to Moodle according to the instructions - as presented to you in the TA sessions.

```java
14  public interface Map2D {
15
16      public void init(int w, int h);
17      public void init(int[][] arr);
18      public void fill(int c);
19
20      public int getWidth();
21      public int getHeight();
22      public int getPixel(int x, int y);
23      public int getPixel(Point2D p);
24      public void setPixel(int x, int y, int v);
25      public void setPixel(Point2D p, int v);
26
27      public void drawSegment(Point2D p1, Point2D p2, int v);
28      public void drawRect(Point2D p1, Point2D p2, int col);
29      public void drawCircle(Point2D p, double  rad, int col);
30
31      public int fill(Point2D p, int new_v);
32      public int fill(int x, int y, int new_v);
33      public Point2D[] shortestPath(Point2D p1, Point2D p2);
34      public int shortestPathDist(Point2D p1, Point2D p2);
35      public void nextGenGol();
36
37  }
```
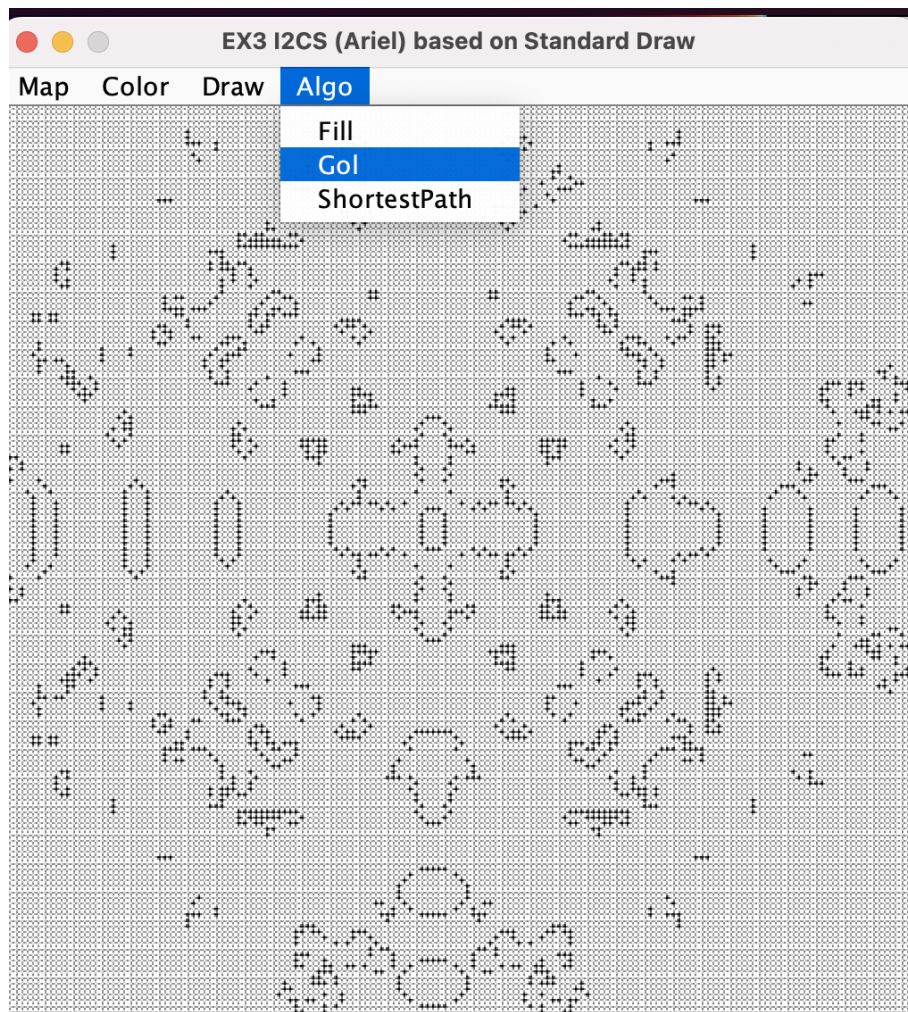
Figure 1: the Map2D interface, see

https://github.com/benmoshe/Intro2CS/blob/main/src/Exe/EX3/Map2D.java

Figure 2: GameOfLife: an example of running the attached Ex3_sol.jar file on Game Of Life implementation.
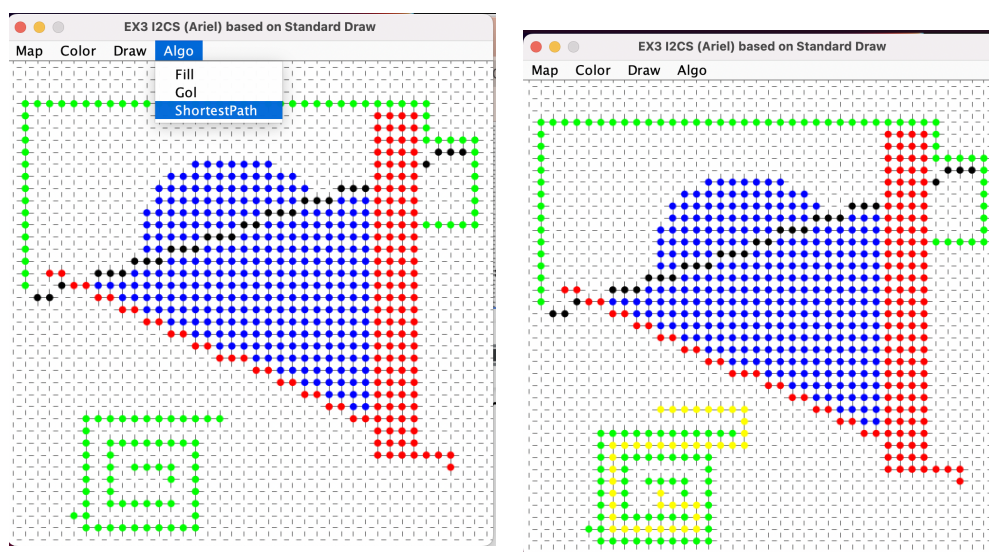

Figure3: Shortest path: Left the scenario before the yellow path. Right: the shortest path (marked in yellow).
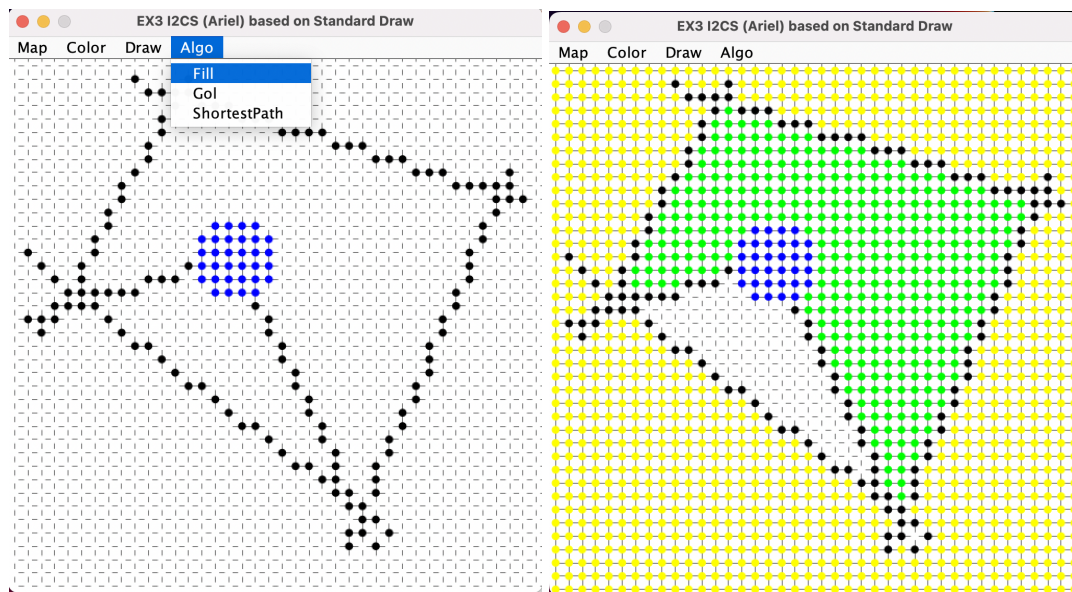
Figure 4: Fill:Left the scenario before filling in green and yellow. Right: the result of filling the map with green above the blue circle, and in yellow on the outer part.

Links:

- https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life
- https://www.youtube.com/watch?v=KiCBXu4P-2Y
- Shortest path algorithm:
  https://ariel-ac-il.zoom.us/rec/share/PcZpLqvO8F2HzeYMCl2nMV3H0XEz-qvUSpIQ3vggfBh_mQVIN8GoB8bQTVSrsZl1.9HTsafDkRDlmxXuI?startTime=1669991178000

Game of file (general links):

https://www.youtube.com/watch?v=7-97RhAZhXI

https://chakazul.github.io/lenia.html

https://www.youtube.com/watch?v=C2vgICfQawE