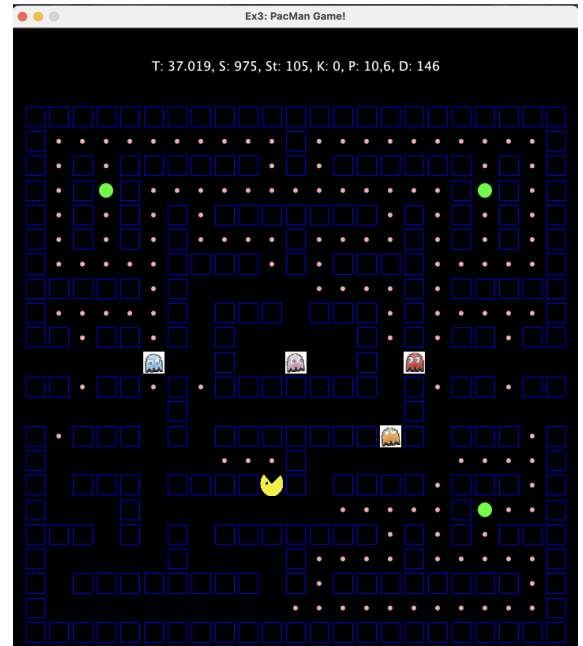


Ariel University, School of Computer Science, 2026
Introduction to Computer Science

Ex3 - Object-Oriented Programming & Gaming
Aka: Pac-Man is “Going Places”

Abstract: This assignment focuses on OOP, code reuse, and putting it all together (aka integration). We will utilize a modified version of the Map2D algorithms (implemented in [Ex2](#)) in order to design and implement a complete version of the [Pac-Man](#) game. [Ex3](#) has three main stages: (i) Design the Pac-Man algorithm. (ii) Implement the Pac-Man module (the client side). (iii) Implement the game module (the server side). Make sure to work according to the stages' order.



To Do

1. Create a GitHub repository (private) named I2CS_Ex3 - from now on, work on this repository - commit frequently. Download [Ex3.zip](#) (with all the required files) Add the Ex3 files to the repository and play with the game (in manual mode) - get a general idea of the project structure. At the end of [this video](#), one can find a technical explanation regarding how to set up the project from the given files.
2. Go over this [short video](#), and design a detailed algorithm for the Pac-Man (the yellow “thing”). The goal of the Pac-Man is to “eat” all the “pink” dots while avoiding the ghosts. At each step, the Pac-Man can move to one of the four directions (UP, DOWN, LEFT, RIGHT). Upon eating a green dot, the ghosts become “eatable” for a few seconds (the ghosts’ icons get smaller). The algorithm should be written in the “Readme”, adding figures and a video clip is recommended.
3. Implement your solution to the client package (the Pac-Man side), in particular the Ex3Algo, including the needed JUnit tester.

4. Test your solution on level 4, ensuring your implementation wins level 4, and report your results on this [form](#). If needed, do update your algorithm's docs (in the readme).
5. Implement an improved game server by implementing your version of the game classes, (design and implement your version of the PacManGame interface, implement the required GUI classes - using StdDraw) - Note: here you have artistic freedom to implement a nice project, split between the client and the server-side. You can go "crazy"; adding levels, boards, music, file support, etc, yet the minimal requirement is to implement a well-organized project with a "game" server-side as given to you in the jar file.
6. You are asked to add a downloadable **version** of the project, as a Jar file (and all the source code + resource files), see this [short explanation](#) regarding how to create a "runnable" jar in IJ. Make sure you have the following jar files: Ex3_2.jar: the solution to part 2 - client. Ex3_3.jar: the full solution to part 3 - client + (your) server.
7. Add detailed documentation (in English) server-side. Add the needed JUnit tests.
8. Create a short (120 seconds max) video - explaining your implementation of the server-side and demonstrating the actual game of the Pac-Man on your implementation.
9. Make sure your Github repository has a release (at least one) with the following files:
 - a. Ex3_docs.zip
 - b. Ex3_2.jar (or Ex3_2.zip, in case several files are needed)
 - c. Ex3_3.jar (or Ex3_3.zip, in case several files are needed)
 - d. Ex3_all_src.zip

See this [short explanation](#) regarding how to add releases to your Github repository.

10. Submission: fill the [following form](#) with your Github link (to Ex3 repository). Make sure to also submit a text file with a link to your Github (of Ex3) to Moodle.

Notes:

1. Work 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. Ex3 contains several classes and files: you are welcome to add additional classes and interfaces. Adding packages may improve your project.
3. The implementation of the methods should be as efficient and elegant as possible. The main algorithms & interfaces should be well-documented.