

2021

Producer&Consumer Simulation Program

Simulate Your Production Line Easily

ENGINEERING ASSIGNMENT

OOP COURSE

5

1

Youssef Hany Shamsia
18015025

es-YoussefHany2023@alexu.edu.eg

2

Omar Ashraf Kotb
18011111

es-Omar.Kotb2023@alexu.edu.eg

3

Moaz Nabil Abdallah
18011824

es-moaz.nabil2018@alexu.edu.eg

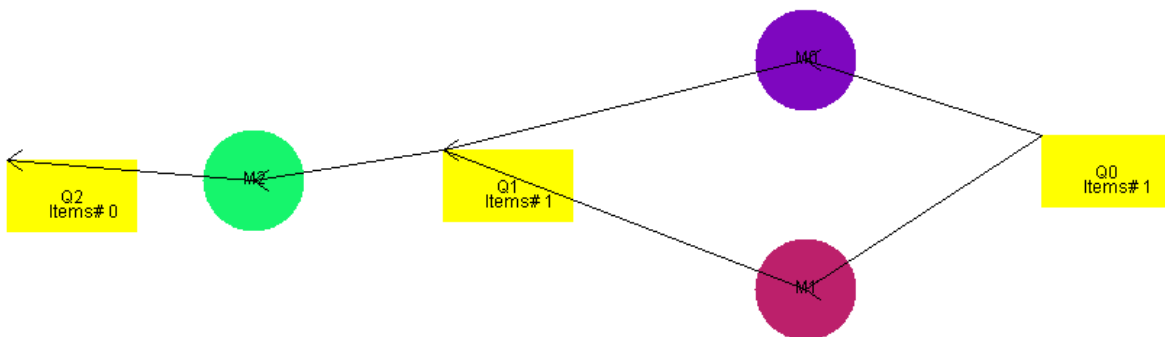
4

Ahmed Tarek Mohamed
18010147

es-ahmedtarek23@alexu.edu.eg

Contents

- Used Design Patterns
- How To Run Simulation Program Application
- How To Use User Interface
- UML Class Diagram



Used Design Patterns:

(1) Producer&Consumer

(2) Observer

(3) Snapshot

Let's talk about every single design pattern and how we apply it in our simulation program application.

1) Producer and Consumer Design Pattern:

The machine class implements Runnable allowing each machine to be run as a separate thread. Machines consume items/products from their input queues when they have available input and after making sure that no other machines are currently attempting to de-queue an item/product from that queue. When there are multiple input queues the machine chooses a random queue that is not empty. After the machine receives the item/product, it changes its color to that of the item and the thread sleeps according to the random runtime of the machine. The thread sleeping simulates how much time the machine takes to process the item. After processing the item, the machine produces the item to one of its output queues (chosen randomly if there are multiple). The machine's color also returns to the default machine color and the machine marks itself as ready to receive further input. Each machine runs on a separate thread to implement concurrency. The simulation ends once all of the items that entered the input queue of the system reach the output queue of the system (being processed by machines along the way).

2) Observer Design Pattern:

First implementing abstract **class observer** which have update method and machine attribute and **concrete class input ready** which allows constructor to initialize machine object by the run time machine object we want and implements update method which notify the queues that the machine is ready.

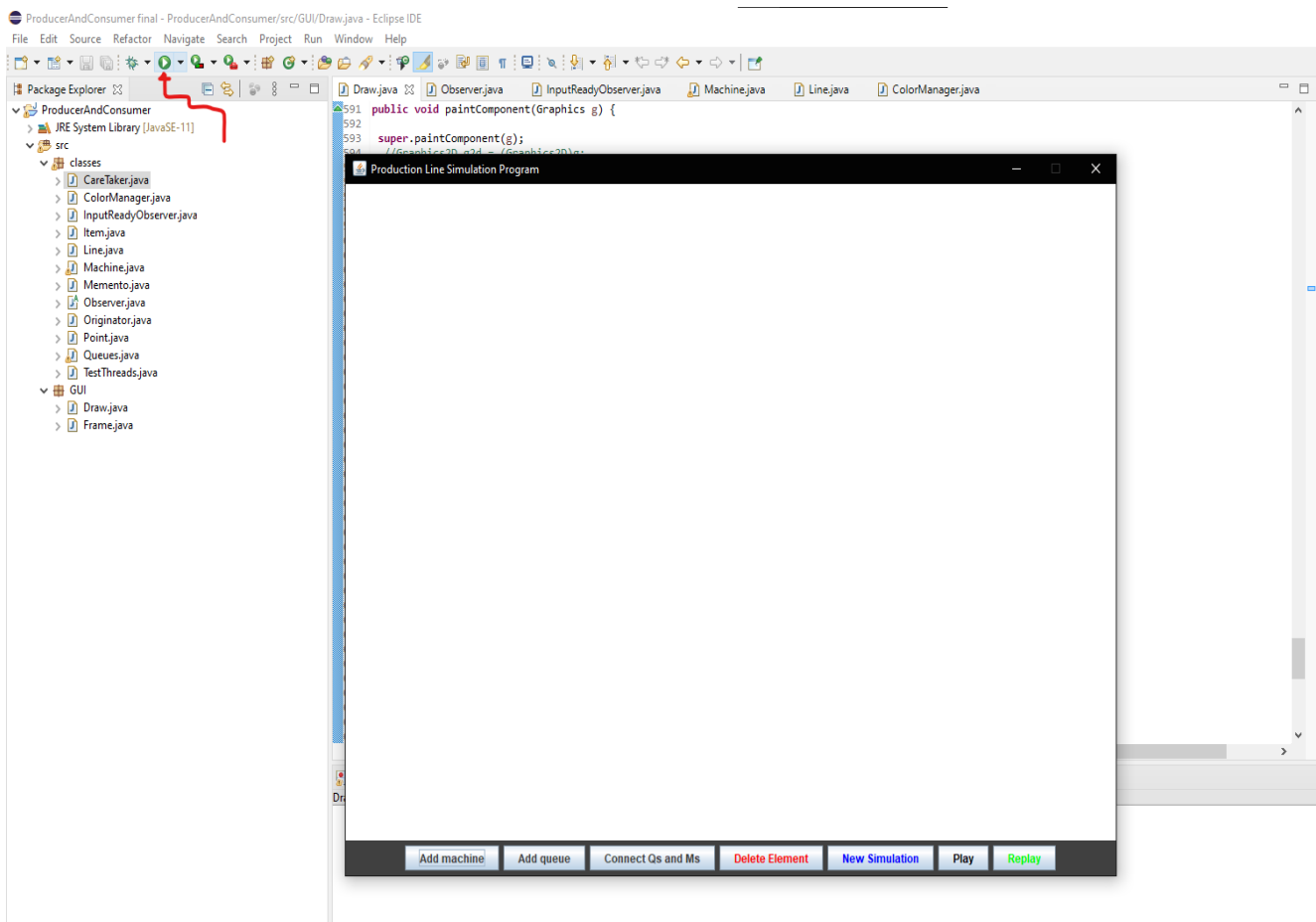
3) Snapshot Design Pattern:

We decided to use it to implement the feature of replaying the simulation as it is considered design pattern which is specialist at restoring a state of an object to a previous state. We first used the **memento class** to contain the state of the object which is represented in our project at the color of items and time of machines to be restored. And the **originator class** to create and store the states in Memento objects. And finally the **caretaker** to restore the states (**items colors and machine time**) from memento class.

How To Run Simulation Program Application:

Assuming: user uses eclipse compiler program.

- Importing the project folder.
- Running the program.



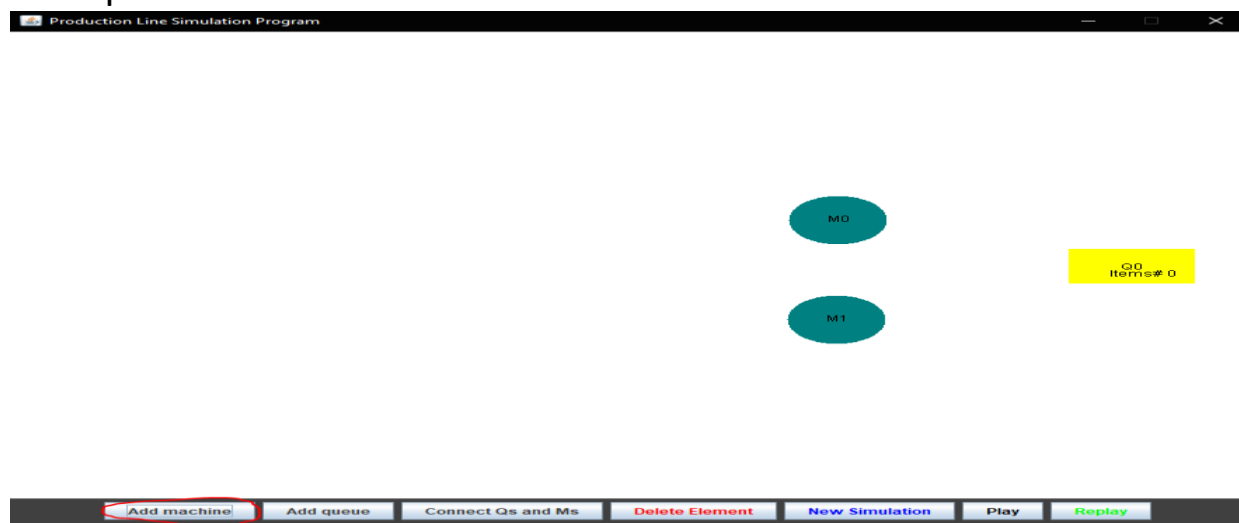
How To Use User Interface:

Assumption:

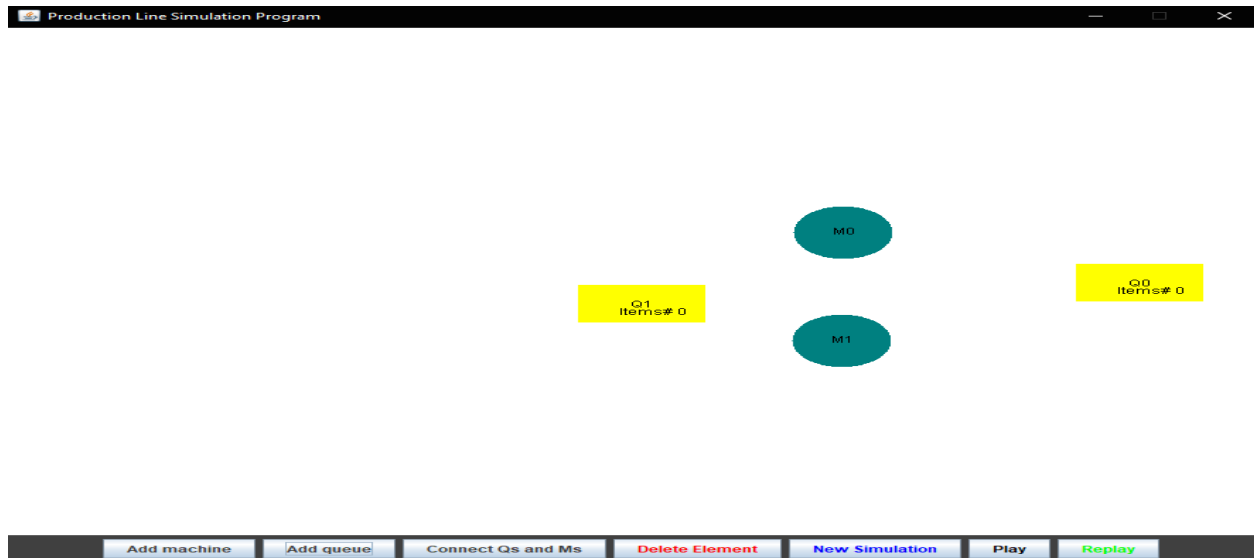
- **First Queue** added is the **input** queue.
- **Last Queue** added is the **output** queue.
- Input items are random number between **5** and **10**
 - First, you must add the input queue (Q_0) by clicking on **add queue** button as shown in the next figure.



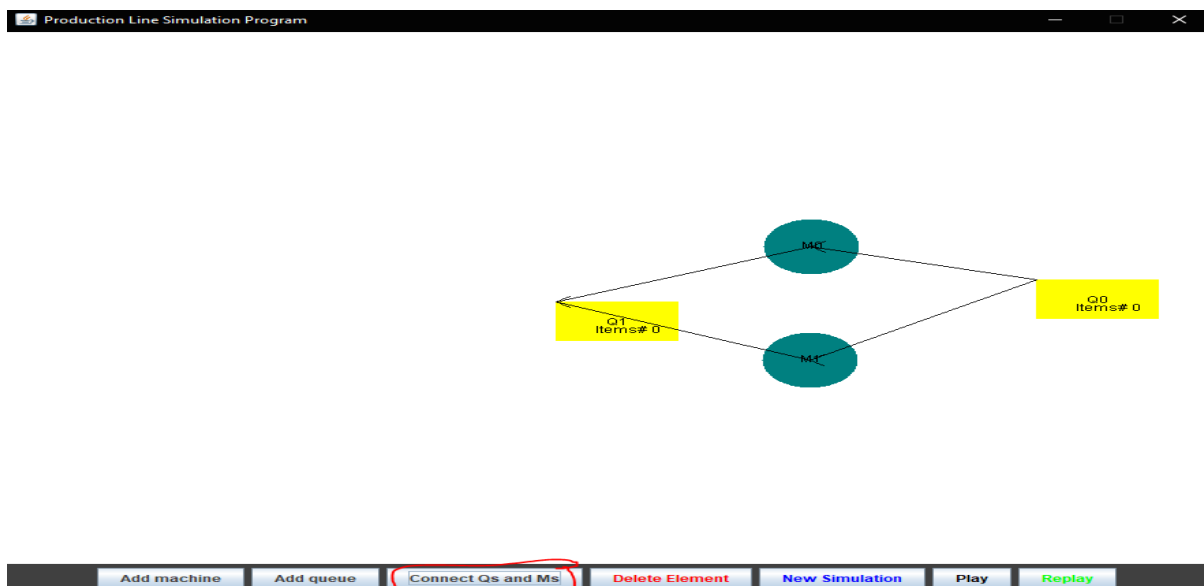
- Second, add machines by clicking on **add machine** button as shown in the next figure. You can also add more intermediate queues.



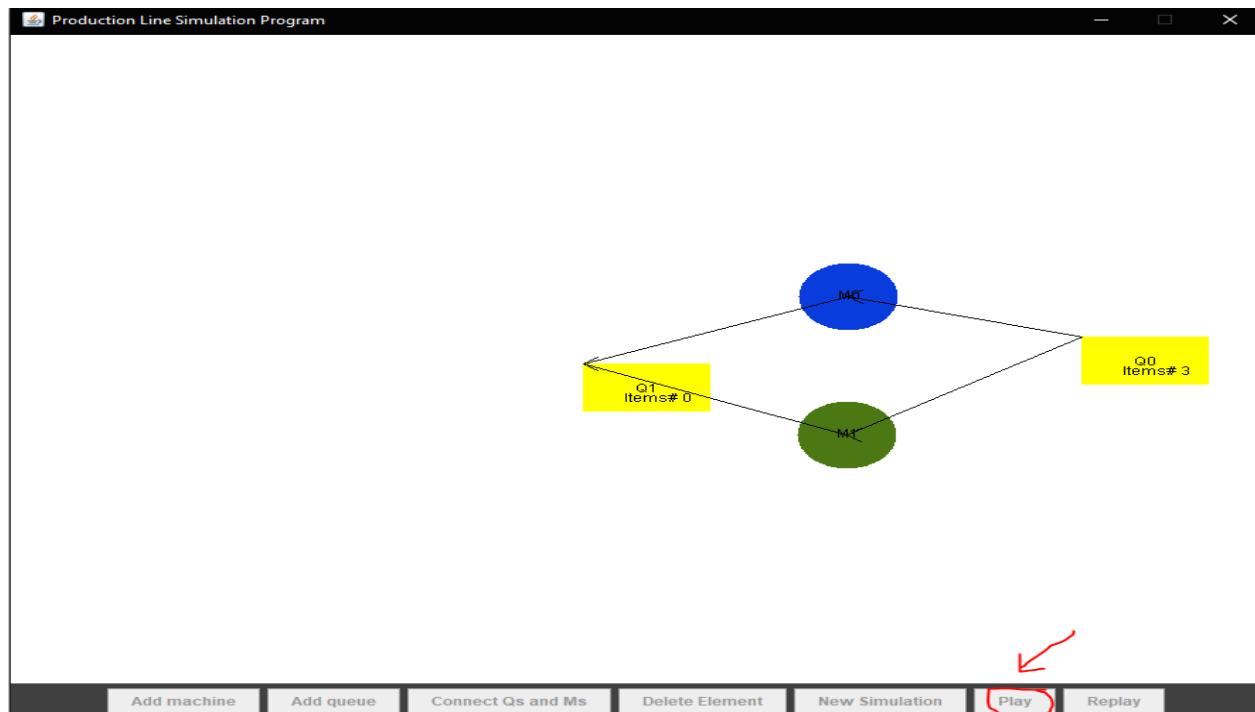
- Third, you must finish by adding the last queue as output by clicking on **add queue** button.



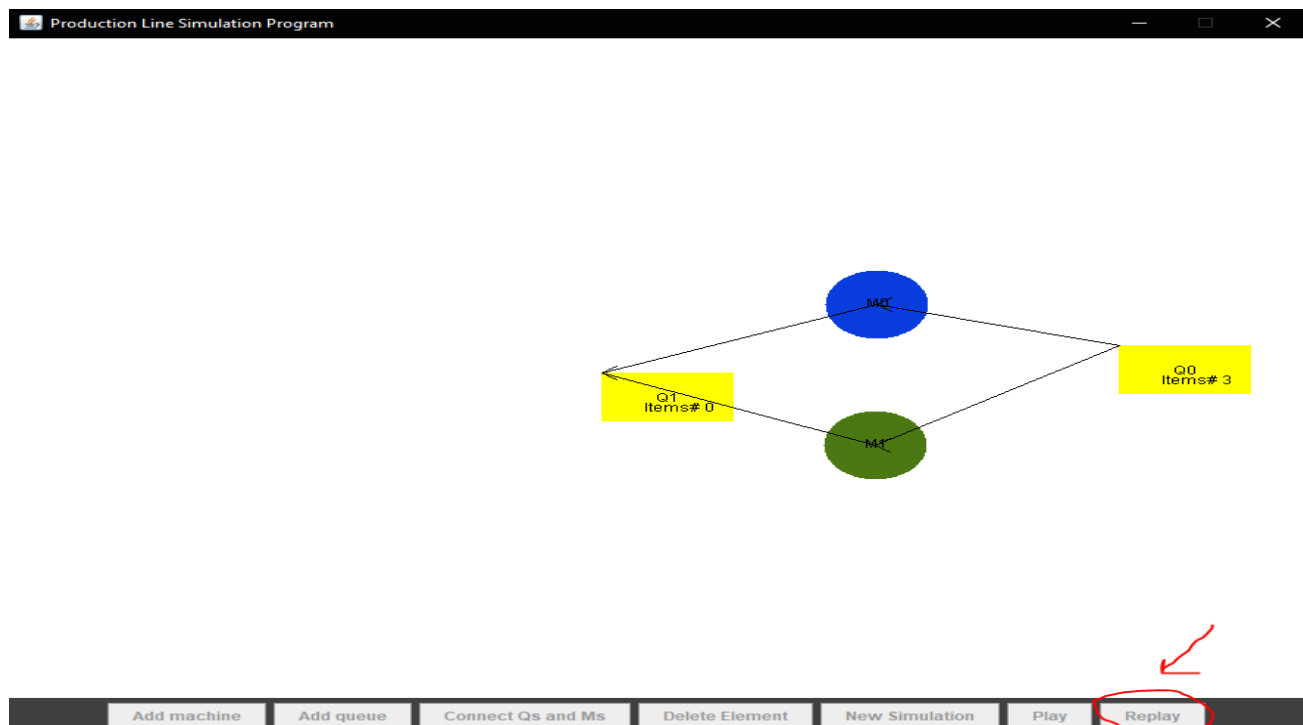
- After that, click **first** on the queue **then** click on the machine that you want to connect, **finally** click on **connect Qs and Ms button** to connect them together such that the queue provides input to the machine. Clicking on the machine then the queue before clicking on **connect Qs and Ms button** will direct the flow of items from the machine to the queue (queue receives output of machine).



- To start simulation, click on **play button**.



- If you want to replay the simulation click on **Replay** button.



Some other features:

- You can delete any element (machine or queue) created by mistake by clicking on **Delete element** button. You can also delete the line connecting a queue and machine to delete the connection between them.
- You also can clear the simulation to make a new one by clicking on **New Simulation** button.

Video Link:

<https://youtu.be/9D5WGn9ztD8>

UML Diagram: (Available as pdf in the folder)

