

# Paint Project

OOP Fall 2016

Aia Lotfy - Fady Nabil - Mohamed Raafat

## **1. Application Overview**

Introducing “The Wool Ball” paint application, the second assignment during the Object Oriented Programming course. The application supports basic functionalities which should be available in any paint program such as:

- Drawing basic geometric shapes.
  - Circle
  - Ellipse
  - Square
  - Rectangle
  - Line
  - Triangle
- Moving shapes
- Resizing shapes
- Saving and loading canvas in two extensions ( .JSON & .XML).
- Undoing and redoing applied actions.
- Importing images and applying edits.
- Adding plug-ins to the application to enrich its functionalities.

## **2. Application Design**

As a beginner Object Oriented Programmers we followed the OOD in our programs as much as a beginner can. The program was divided into separated packages each collects bundle of classes related to each other.

### **Main Package (Assembler Classes)**

The main package contains the main class and 2 helper classes used to facilitate move and resize function in the application.

### **Creator Classes**

To create a shape we must have a shapes’ package which contains the create x classes -where x stands for the built-in shape, the create classes implement one interface and are appealing the “Singleton Design Pattern”.

### **Move and Resize**

Since editing a shape is a need for any designer, a special class was implemented to perform the desired edits from moving or resizing any shape or image on the main canvas, this class is one of the most complicated classes in the project which may break some OOD principles unfortunately.

### **Saving and loading**

Concerning saving an the a canvas, the program allows the user to choose from two main file formats / extensions ( .JSON & .XML). - as it was mentioned before -, saving classes, briefly the main idea for saving in the application is to save every single object in an external file which can be easily parsed recreated and added to the drawing canvas.

### **Dynamic Class Loading**

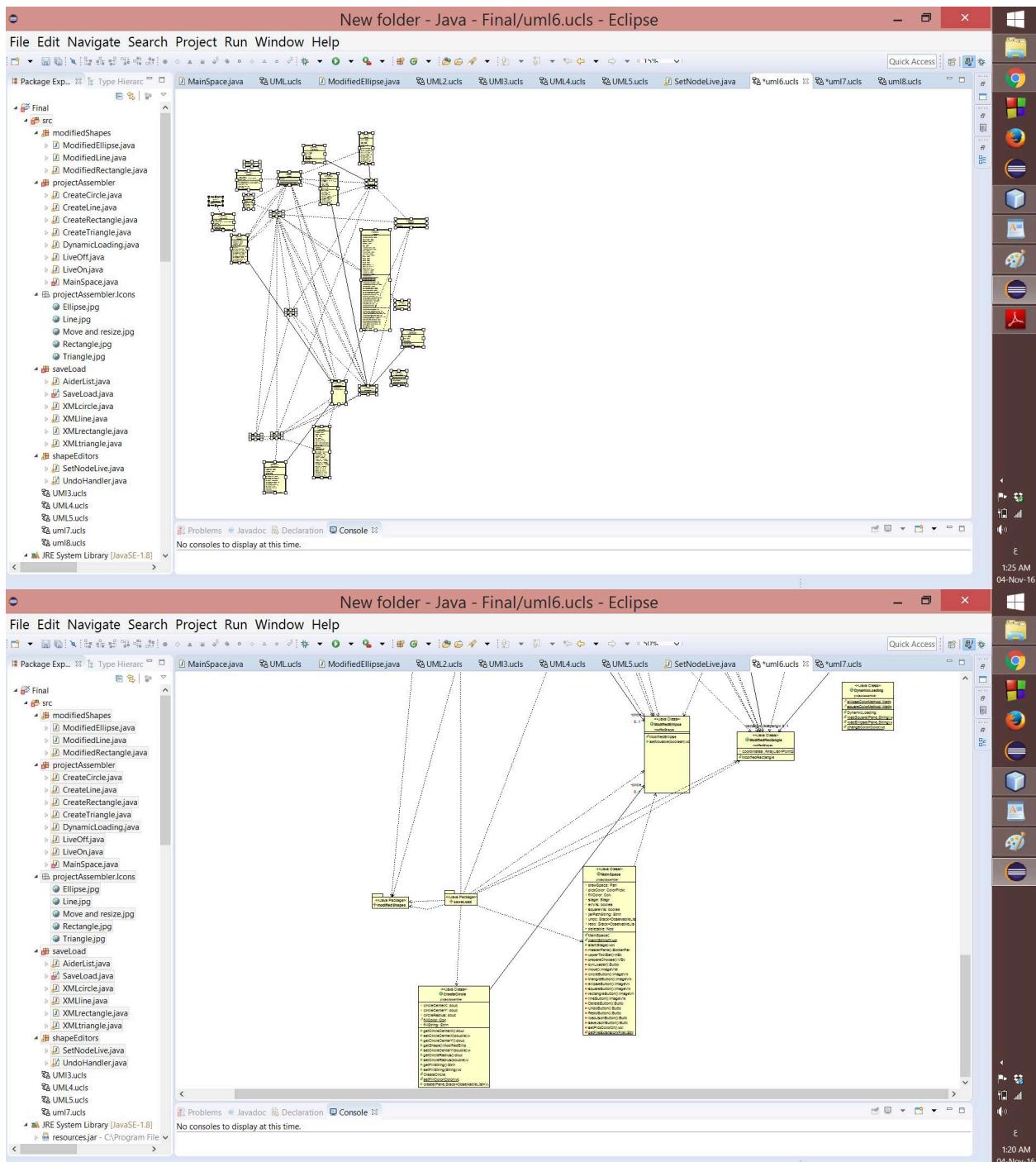
Extendability of an application was a main requirement in the application design, so, the term (Dynamic Class Loading) appears in our project through adding a plugin

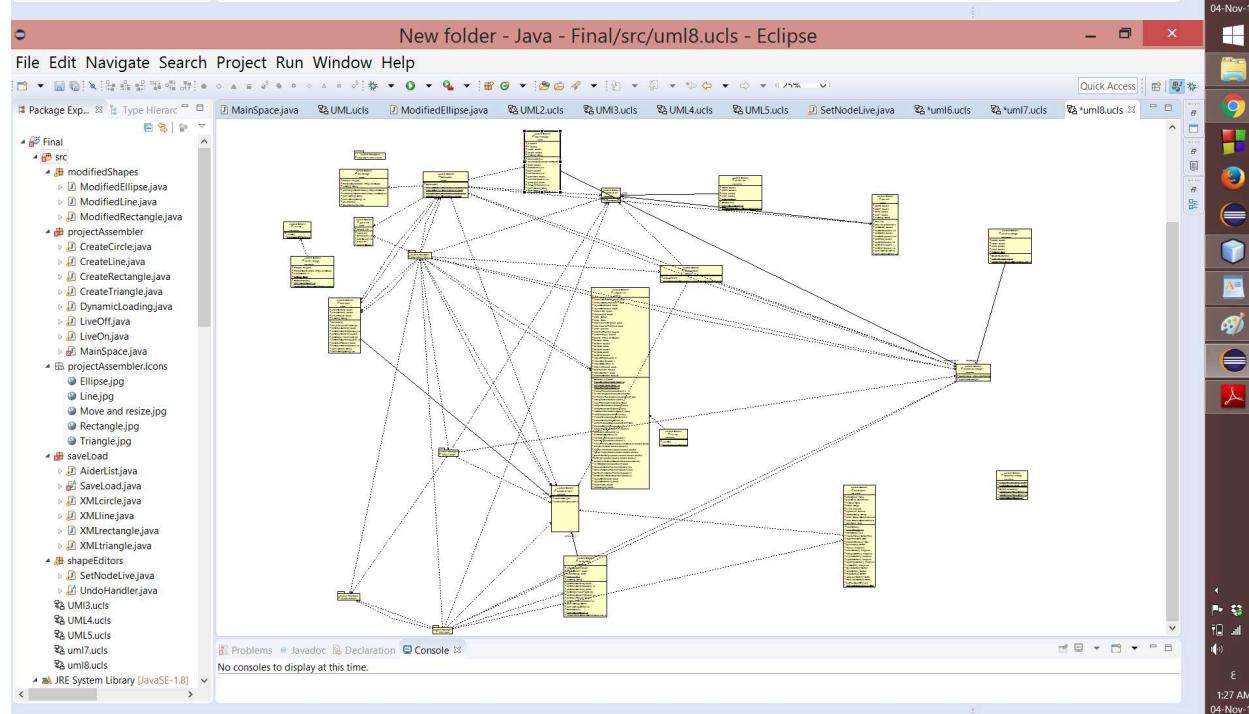
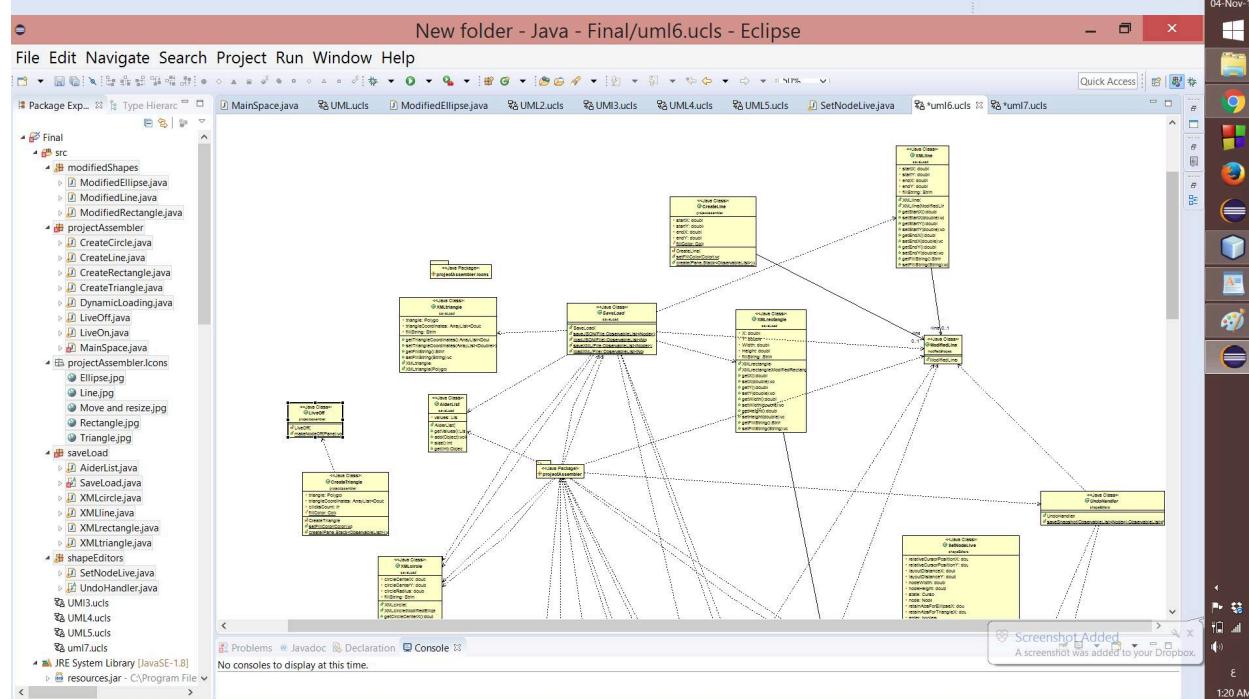
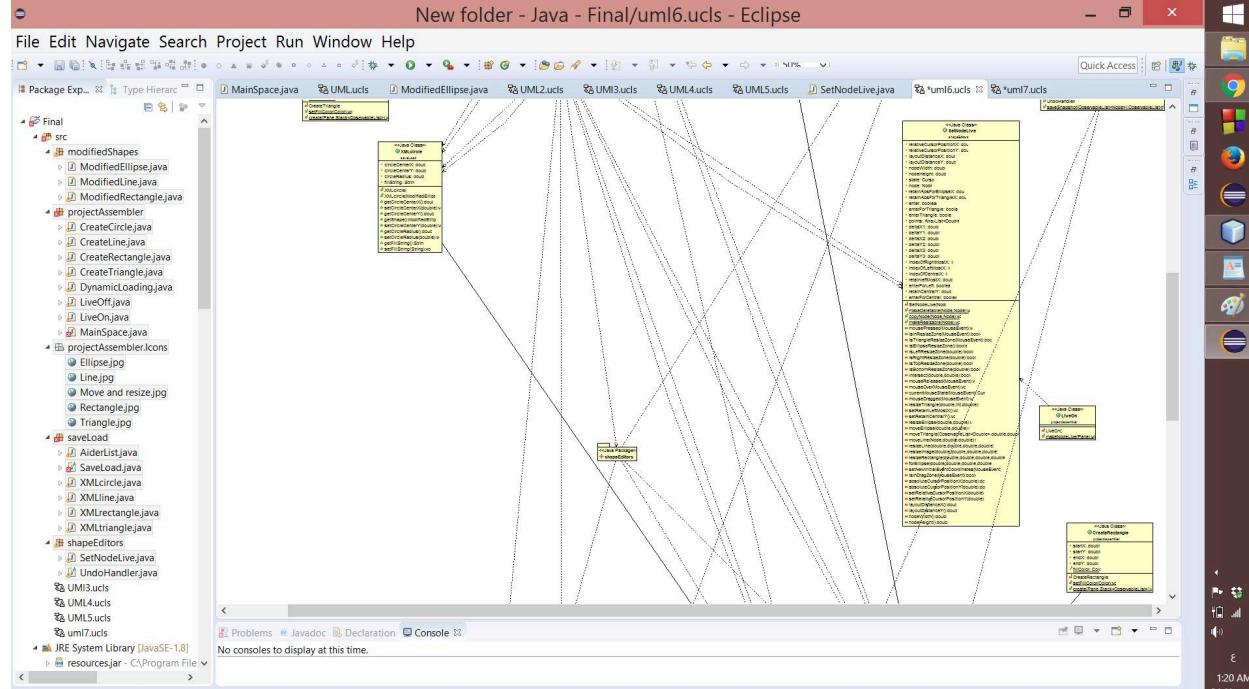
button to allow the user to add shapes and functionalities *compatible with the program* through a file-chooser panel.

### Java Network Launch Protocol

Networks and remote access form a base need in our lives nowadays, so we added a remote access capability in our project which allows users to access our server and enjoy a good experience on using our app through JNLP.

## 3. Unified Modeling Language Diagrams





## 4. Screenshots

