**Portfolio 1 – Software Development**

**Victor Nielsen – Martin Bern**

**https://github.com/Victornrib/Portfolio1-SoftwareDevelopment**

**Introduction:**

For the first portfolio of the semester, our group will work in two different situations related to object-oriented programming and software engineering. Both of them will focus on how to implement classical OOP models and understand and apply software requirements.

The portfolio will be split in two main parts, in which each of them will target different situations:

* The first part will be based in build all the necessary environment and plan all the necessary tasks to create a small shapes model, which will contain basic forms as rectangles, triangles and circles; but also being designed with extensibility so other shapes could be added in the future without changing the basic model. For that, the first part will focus on:
* Establishing a development environment (in this case, using Git, Github, JAVA and InteliJ)
* Designing the application (using UML from building use case diagrams to static and dynamic models of the app)
* Planning the development of the application (using Trello – which is Kanban based)
* The second part will be the implementation of the model itself. It will have a common behavior, so that each shape instantiated can return:
* Its center
* Its area
* Its circumference
* Indicate whether or not a point is inside of it
* Compute the Euclidean distance from its center to the center of another shape

The design of the shapes will be made in UML and implemented in JAVA.

Resuming, the second part will be made of:

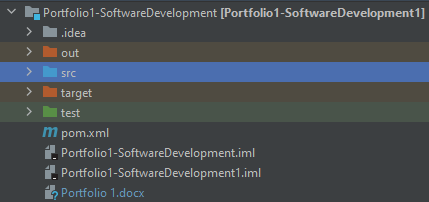
* An overall description of the shape model
* UML diagrams documenting representation and behavior of the shape model
* An implementation of the shape system
* Unity tests verifying behavior for all shapes

**First part (Shapes model):**

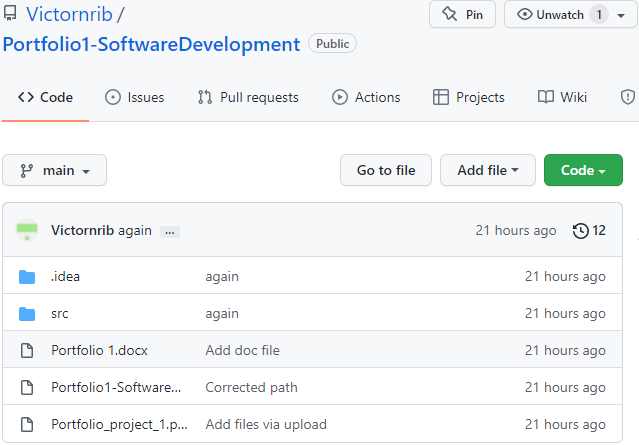
**Development environment:**

To implement a shapes model, based on the specifications given by the portfolio, we first started our development environment.

For that, we started creating a project in InteliJ using Oracle OpenJDK version 11.0.11, Maven and JUnit 5.8.2.

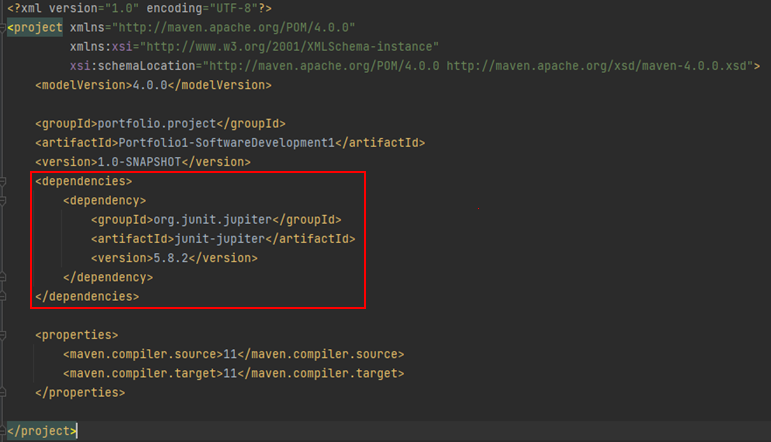


After that, we enabled Git in the project and exported it to a Github repository.



We did a remote repository because it gave us the ability to update our application from any device, so we could both make commits to the project. Also, it allowed us the search through Git History in the case of committing a version not working properly.

After we configured our pom.xml file, so that it could allow the use of the Junit 5, which would be crucial to test the class of the project.

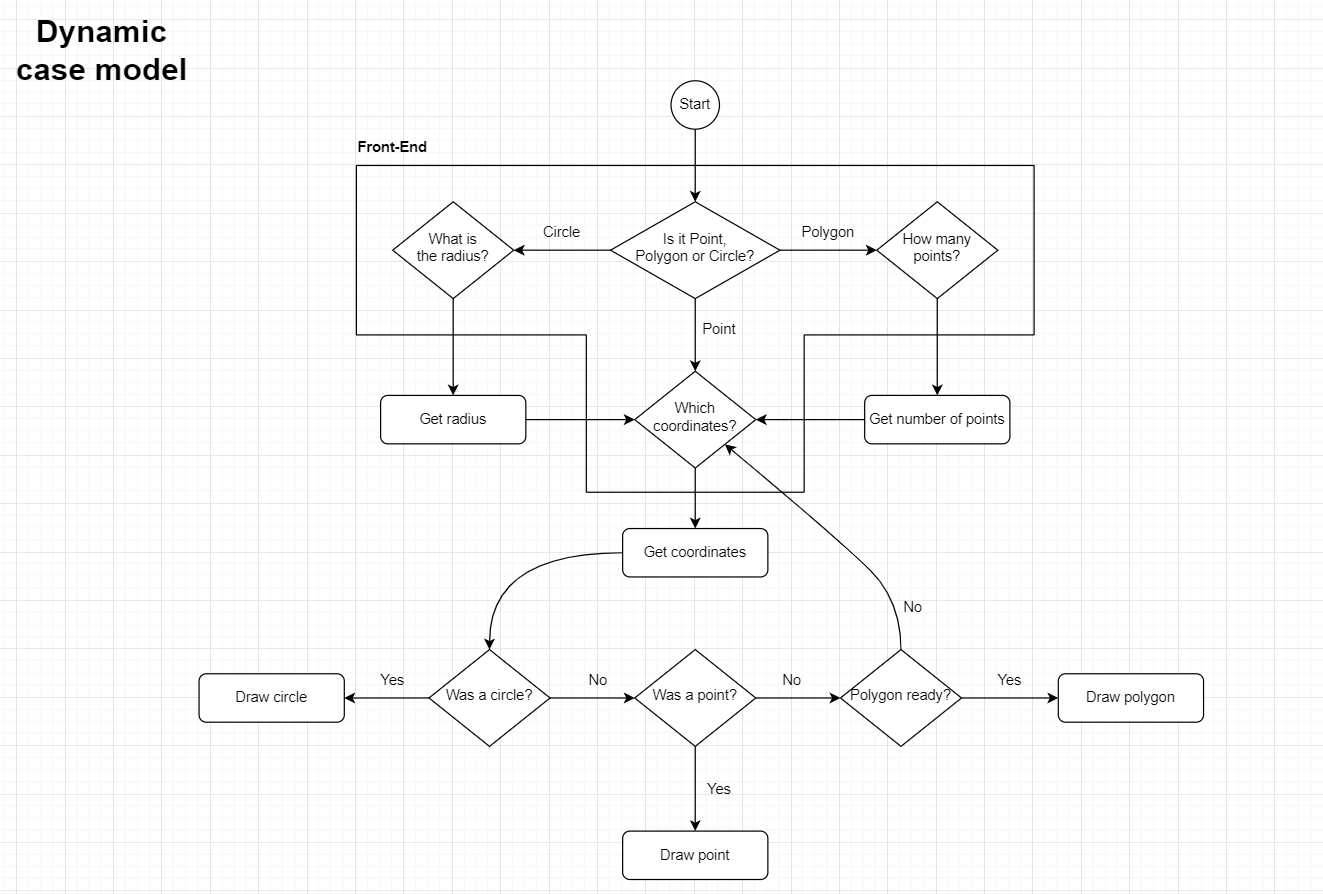


**Design of the application:**

For the app design, we started doing a dynamic case model with an Activity Diagram.

We wanted to make a model that could give us the general idea of the software and its interaction with the user.

The model would be a very useful tool to help to build the controller package, which would contain the main function for running the program.



Since we wanted to do an application with good potential to expand with new features, we only limited ourselves to three types of drawings: “Points”, “Polygons” and “Circles”.

They were chosen that way because they would have different internal representations. And also, in the case of the polygons, they would not be restricted to pre-defined forms, like squares or triangles, but to any convex polygon with 3 or more vertices.

A more in depth look into the internal representations of each one of the base classes will be showed in our Class diagram in the second part of the portfolio.