A Component-based 2D Asteroids Game

The purpose

# DesignLab

# JavaLab

*Resume of GameLab:* in the GameLab exercise the original monolithic code base were divided into modules such as Core, Common, Player, etc. and when module A depends on module B, module B will be a dependency imported in the module A’s pom.xml. The instantiation of a module B is done by calling the constructor directly in module A with “new ClassName();”.

In JavaLab this instantiation/assembly is automated with Java’s native service loader, to use this service loader the SPILocator class is provided. The SPILocator allows with the use of these three methods *(figure 1)* to get a collection with the implementations of a specific interface, instead of accessing an instance by referring to the value at which the instance from the “new ClassName();” call’s return value is stored like done in the GameLab exercise.

Et billede, der indeholder tekst

Automatisk genereret beskrivelse

Figure Game.java within the Core module.

These collections returned by the methods *(figure 1)* can then be loop thou with the use of a for loop *(figure 2)* and the individual implementations of these interfaces can then be accessed.

Et billede, der indeholder tekst

Automatisk genereret beskrivelse

Figure Game.java within the Core module.

To publish an implementation of an interface for the SPILocator to include one file per implemented interface is placed within the following path “src > main > resources > META-INF > services” in the module that provides the implementation, this file is then named the path to the java class file *(figure 3)* which implements the interface and the context of the file is the same as the name.

Et billede, der indeholder tekst

Automatisk genereret beskrivelse

Figure The structure of the Player module.

# NetBeansLab1

# NetBeansLab2