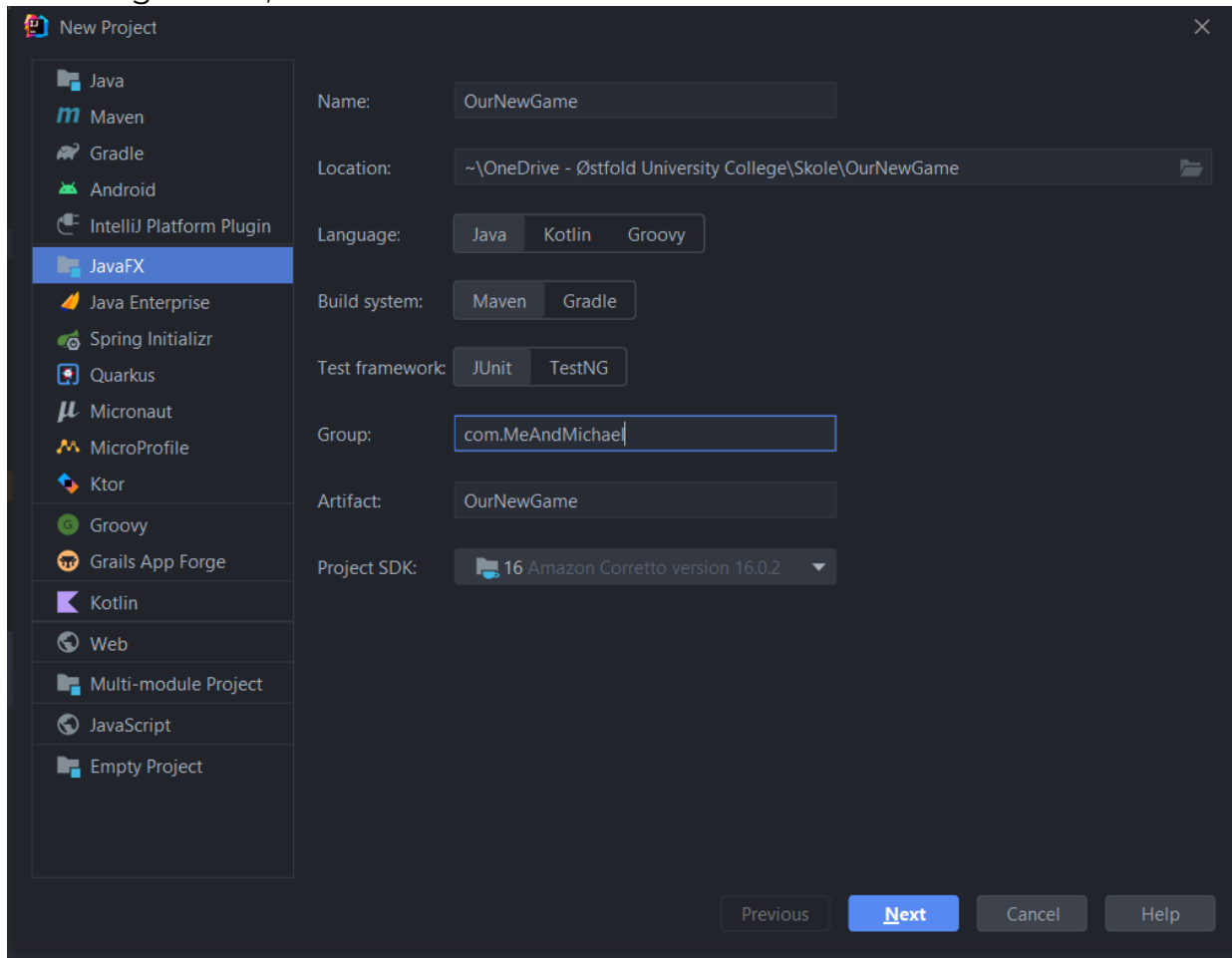


# Getting Started with TRPGF

Follow this getting started guide made by JavaFX, and install Version 17.0.3

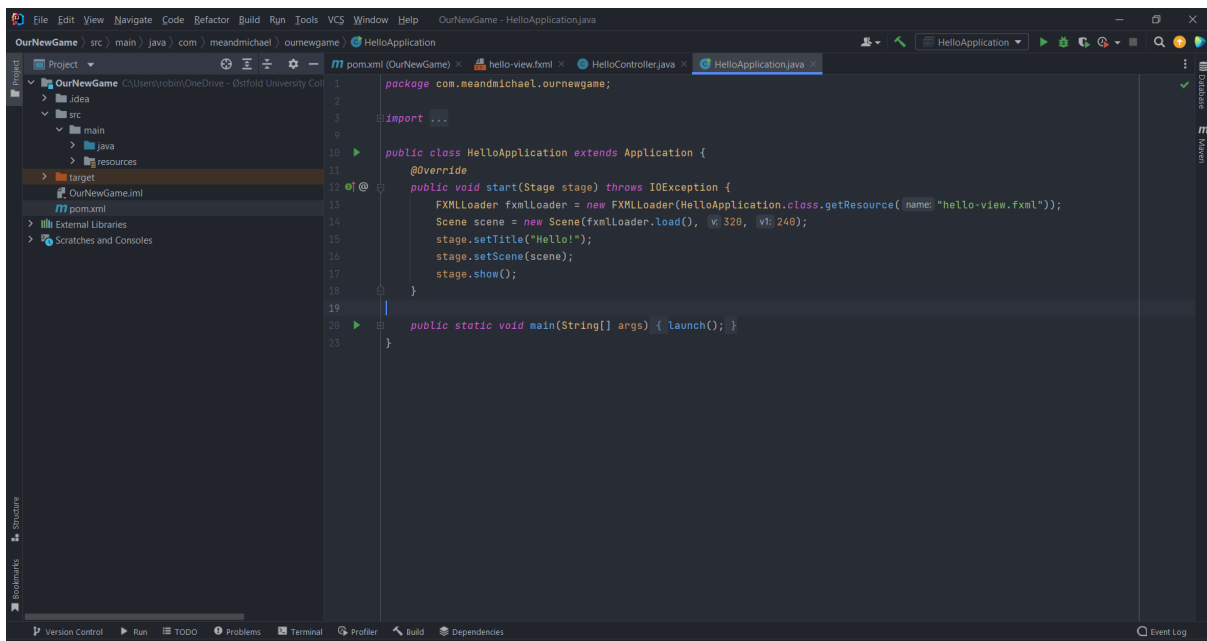
<https://openjfx.io/openjfx-docs/#introduction>

Then setup a new JavaFX project using your chosen IDE, for this one we are using IntelliJ,



It should look like this.

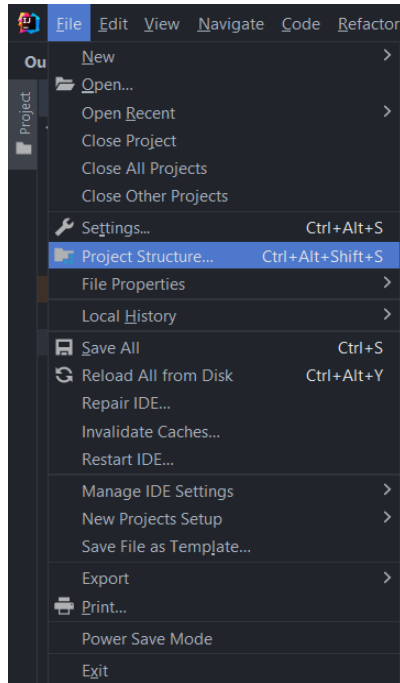
We do not need to add any extra dependencies from the next tab.



Your project should look like this.

Then run the project as is to see if you have any issues with JavaFX, this is the HelloJavaFX application made by JavaFX.

Next you're going to want to add TRPGF.jar to your project.



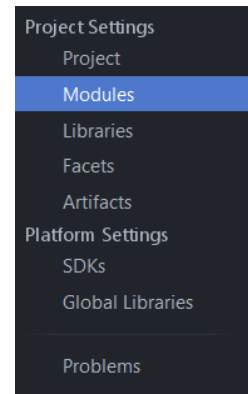
To do this first open up Project

Structure like so, or use

Ctrl+Alt+Shift+S to open it.

Then proceed to the modules tab.

From here we are going to be adding a new dependency, which is the TRPGF.jar

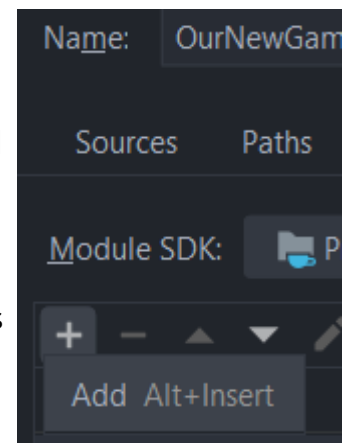


From here we are going to want to click add, or use Alt+Insert, and we want to select JARS or Directories.

Then locate wherever you saved TRPGF.jar to, and add it as a dependency, after this hit Apply and close the Project Structure.

Now we are going to want to add some dependencies for TRPGF, so we want to head over to pom.xml since we are using Maven for this project (we currently only support Maven, since it has been used during development).

Then you're going to want to add this snippet of code to your dependencies;



```
<dependency>  
  <groupId>org.json</groupId>  
  <artifactId>json</artifactId>  
  <version>20220320</version>  
</dependency>
```

```
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-databind</artifactId>
  <version>2.13.2.2</version>
</dependency>
```

Make sure to load the Maven changes using Ctrl+Shift+O, or by using the Maven reload icon in the top right.

Now we have to head over to Module-info.java, we will make some changes.

Your Module-info.java should look like this, do not be alarmed if IntelliJ does not autofill org.TRPGF

```
module com.meandmichael.ournewgame {
    requires javafx.controls;
    requires org.TRPGF;

    opens com.meandmichael.ournewgame to org.TRPGF;
    exports com.meandmichael.ournewgame;
}
```

Now we can delete HelloApplication.java, HelloController.java and hello-view.fxml, and make our own Application.java.

```
1  package com.meandmichael.ournewgame;
2
3  import javafx.application.Application;
4  import javafx.stage.Stage;
5  import org.TRPGF.Screen;
6
7  public class Timerun extends Application {
8      @Override
9      public void start(Stage primaryStage) throws Exception {
10         Screen screen = new Screen(primaryStage);
11
12         screen.startingScreen( gameTitle: "Ninjas in space",
13                               startingText: "There are ninjas in space," +
14                                             " with deadly laser kunais");
15     }
16
17     public static void main(String[] args) {
18         launch();
19     }
20 }
```

And now you are ready to start developing your own Text Based RPG, using our Framework, please refer to the javadoc if you need more help with the functions.

## Example code for Screen:

```
Screen screen = Screen.getInstance(stage);

screen.startingScreen("Kråkevisa", "The man went into the woods\n" +
    "Hey fara into the woods\n" +
    "When a crow sat in the meadow and crawled\n" +
    "Hey fara. Faltu riltu raltura");

screen.characterScreen("Make your character!");

String path = "Absolute Path to an Image, or a link";
screen.addTableScreenMap("Map", path);
screen.tableScreen();

screen.addEndingScreen(1, "The end", true, true);
screen.addEndingScreen(2, "The end 2", true, false);
//FinishEndingScreens have to be after adding them.
screen.finishEndingScreens();
}
```

## Example code for Dialog:

```
Dialog dialog = new Dialog();

// TESTING FOR SOME OPTIONS
dialog.createDialogBox(1, "Default Content");
dialog.addOption(1, 1, "I GO TO 2", 2);
dialog.addOption(2, 1, "I GO TO 5", 5);

dialog.createDialogBox(2, "I AM BOX 2");
dialog.addOptionPrevious(1, 2, "Chose 1, in 1", 1, 1, 3, 4);
dialog.addOptionPreviousRequirement(2, 2, "CHOSE 1, IN 1, + STAT 15 STR", 1,
1, "Str", 15, 3, 4);
dialog.addOptionPreviousReward(3, 2, "CHOSE 1, IN 1, +2 STR", 1, 1, "Str", 2, 3,
4);
```

```

dialog.createDialogBox(3, "GOOD CHOICE");
dialog.addOptionWithReward(1, 3, "CON +10", "Con", 10, 6);
dialog.createDialogBox(4, "BAD CHOICE");
dialog.addOptionEnding(1, 4, "You Died.", 2);

dialog.createDialogBox(5, "MORE OPTIONS!");
dialog.addOptionPreviousRequirementReward(1, 5, 1, 1, "CHOSE 1, IN 1, +
STAT 15 STR +2 DEX", "Str", 15, "Dex", 2, 3, 4);
dialog.addOptionWithRequirement(2, 5, "STAT 15 STR", "Str", 15, 3, 4);
dialog.addOptionWithRequirementReward(3, 5, "STAT 15 STR, +2 DEX", "Str",
15, "Dex", 2, 3, 4);

dialog.createDialogBox(6, "GOOD AGAIN!");
dialog.addOption(1, 6, "Go back?", 1);
//Ending on success means if you meet the requirements then the game
will end, likewise you can set it to false and end the game if you do not meet the
requirements
dialog.addOptionPreviousRequirementEnding(2, 6, "Chose 1 in 1, >+ 15 Str,
ENDING 1", 1, 1, "Str", 15, 1, 1, true, 1);

// You have to have finishStory last because it compiles all the functions into
one datafile for the table screen to display.
dialog.finishStory();

```

## Example Code for Character Creator:

```

CharacterCreator characterCreator = CharacterCreator.getInstance();

//characterCreator.setName("B");
characterCreator.addNameOption();
//You can only use either function, they will override each other if you try to
use both

characterCreator.addStat("Dex",0,20);
characterCreator.setStat("Dex", 10);
// A stat must first be created and then the generation method can be set.
The name of the stat is case-sensitive
characterCreator.addStat("Wisdom",0,20);

```

```
characterCreator.setStat("Wisdom", 10);
```

```
characterCreator.addStat("Str", 0, 20);  
characterCreator.setStat("Str", 15);
```

```
characterCreator.addStat("Con", 0, 20);  
characterCreator.setStat("Con", 20);
```

```
characterCreator.finishCharacter();
```

*// Remember to use this function, it saves the info above so that it can be used by the rest of the framework*

## Reminder:

You have to have `launch();` at the very end of the code.