TEACHER: GLAUBER TORRES

BY

JULIEN PONTBRIAND

Student # : 202135020

Course Number : 420-CT2-AS

Course Title: Object-Oriented Programming Concepts

Session: Winter 2022

FINAL PROJECT REPORT

MY DROID FACTORY

Table Of Contents

[1 - PROJECT DESCRIPTION 2](#_Toc101266672)

[2 - PROJECT DOCUMENTATION 2](#_Toc101266673)

[2.1 - Provide a high-level explanation of the layers that integrate your solution. Describe the responsibility of each layer. 2](#_Toc101266674)

[2.2 Describe each created class. 3](#_Toc101266675)

[2.3 Provide a class diagram (The one provided is for the Droids) 3](#_Toc101266676)

[2.4 Provide a user-friendly documentation that explains how to use the entire system. Include screenshots. 4](#_Toc101266677)

[3 – CONCLUSION 9](#_Toc101266678)

[4 – Bibliography 10](#_Toc101266679)

# 1 - PROJECT DESCRIPTION

To personalize and improve upon the already exciting Bike Factory project, I decided to customize this project’s theme and turn it into a Droid Factory. Hence, you will find in MyDroidFactory project all the required technical specifications, altered with different names and characteristics. For instance, instead of Enumerations about Tire Types or Suspension Types, you will find AiChip, Armament, Equipment, Plating and so on.

A whole lot of my inspiration for this project came from Star Wars episode II, especially from the droid factory scene.

# 2 - PROJECT DOCUMENTATION

## 2.1 - Provide a high-level explanation of the layers that integrate your solution. Describe the responsibility of each layer.

This project makes use of 3 main layers of code.

The 1st one is the Business layer. This layer encapsulates the instantiable classes of the project along with its Enumeration lists, which contain names and associated Id numbers.

The second layer is the Data Layer which contains 2 classes, one dedicated for the Users of the Application and another one for the Droids contained within the application. Both make use of encoding and decoding processes to store or collect data from a .txt document. The 1st class is making use of a Xml serialization process while the 2nd class uses a Binary Formatter.

The third and last layer of code for this project is the UI layer. Within this layer, I created 3 Forms which contain the different User Interfaces usable for MyDroidFactory. All of these Forms have been designed with UI controls (buttons, forms, and so on) as well as with underlying code to make the user experience seamless.

## 2.2 Describe each created class.

From the Business Layer, there are 4 main classes, with 3 of which may be instantiated. As exposed in 2.3, the Droid class is the abstract class from with the 3 other droid classes inherit. The BattleDroid and the AstroMechDroid both inherit directly from the abstract Droid class while the AssassinDroid is a derived class from BattleDroid. The BattleDroid adds 4 armament properties as well as Plating, the AssassinDroid adds an AiCip to the properties of the BattleDroid and the AstroMechDroid adds Plating as well as 2 additionnal equipment slots to the Droid properties. At first there were many more types of droids which I wanted to include to the application such as protocolar, interrogation, mechanical and spying droids. This explains why the Plating property was not added to the Droid class, since protocolar droids, for instance, would not have been given the Plating property as they are not meant for combat situations.

## 2.3 Provide a class diagram (The one provided is for the Droids)

Graphical user interface, text

Description automatically generated

## 2.4 Provide a user-friendly documentation that explains how to use the entire system. Include screenshots.

When you launch the application, this is what should be displayed:

Graphical user interface, application

Description automatically generated

There are 3 main buttons, 2 input fields and a Loading Screen. The exit will asks you if you wish to exit, the Account Creation lets you create an account to be able to login and the login lets you have access to the next Form of the Application. You may use ALT+C, E or L for a quicker access to the buttons. The way it has been coded, the loading screen will stay until the main video beneath it will have finished loading.

Once is has finished loading, a Star Wars video should run, given that the machine running the application is connected to the internet. I really wanted to include a way to load a video from the internet, but I did not wanted to have ads running. Since I used WebView2 to load an instance of Youtube from Invidious, the ads were taken tare of, but even while using the URL parameters to request an automatic play of the videos, modern browsers such as the one used by WebView2 make it impossible to launch a video automatically unless it is muted.

Graphical user interface, application, website

Description automatically generated

https://invidious.snopyta.org/embed/ZaxMHBxck5I?loop=1&autoplay=1&iv\_load\_policy=3&t=59s&quality=hd720&thin\_mode=true&t=59s&controls=0

From this link (which you can find in the source of the webview control), you can see that I request a link with parameters that define many specifications to the video that I request, such as specific video quality, an autoplay function (which does nothing since I do not request a muted video, but is anyhow essential to make this process work), the looping parameter and also a way to prevent users from controlling the video (play, pause, etc).

As a matter of fact, the solution that I found was to include code to go and automatically force a click to the video in the stead of the user. To make sure the client stays unaware of this process, I replaced the mouse cursor to the original position right after moving it onto the webview2 control to click it make start the video. This code has been placed in the UIIntelligence class as well as in the the Login Class from the UI layer.

Text

Description automatically generated

Text

Description automatically generated

Once the login step has been successful, the user will be taken to the second Form : the MainMenu.

Graphical user interface, PowerPoint

Description automatically generated

A somewhat similar process to the Login Form happens for the loading screen of the MainMenu, except this time the sound comes from a file included in the Project, which allows for a different music than the one from the loaded video.

There are many buttons on this from. Manufacture New Droid and Update Selected Droid both bring you to a 3rd From to Add a Droid to the listview or to Edit the ones that are currently being displayed on the listview.

The ID Search button allows Users to search for a specific Droid, which will be displayed in the listview if it matches with the ID of a droid in the list.

The Display Type Combo box lets users display a filtered List by Droid type (All Droids, Battle Droids, Assassin Droids or AstroMech Droids).

The Secure All Droids Button lets you Save your current list of Droids for further usage.

The Sell Selected Droid lets you remove a specific droid from the list.

The Exit Button will let you exit the application.

Graphical user interface

Description automatically generated

While browsing this form, you may receive Mail at some point! (Timer-Based MessageBox)

Graphical user interface, text, application

Description automatically generated

While both Manufacture New Droid and Update Selected Droid bring you to the 3rd Form, they bring you to it with a few differences both code-wise and design-wise. Also, to update, you need to have selected a droid before. The option to select many droids has been disabled.

Graphical user interface

Description automatically generated

In the image above, we ca see that the Update Droid button is Yellowed-Out. This is because is has been disabled since it is been accessed from the Manufacture New Droid Button. The opposite is also true when accessed from the Update droid Menu (this will be displayed on an image below). As you start creating a Droid, the Equipment, armament, plating and AI chip combo boxes are first darkened. This is because only the Droid Abstract Class Properties are displayed while no Droid Type has been specified. Once specified, the required combo boxes will be enabled. Necessary validations are always required. If you are updating a droid, all droids properties will be loaded into the form for a convenient experience.

Graphical user interface

Description automatically generated

In either the Droid Creation or Update case, returning to the Main Menu will automatically update the List according to the Display Type selected (Default being All Droids).

In the case of updating a droid, the user will be brought back automatically to the main menu after updating its selected droid from the list view.

The music played in the background will always loop since it is programmed to launch again a couple of seconds after it has finished.

# 3 – CONCLUSION

From this project, I learned about many new concepts, technologies and functions I knew nothing about. For instance, making use of Webview2 to embed web content in my applications through Microsoft Edge was something I had no idea about. I believe its usage made for a fun and interactive addition to my project.

I also had the opportunity to learn about controlling the Cursor of a user using the application, which also helped me greatly to figure out a way to make the videos start automatically.

Making user of timers to help me implement some small features such as the Mail Messages or the automatic music loops for my .wav file was also a lot of fun.

Another important feature to mention was the use of a dictionary and of referenced objects passed between several UI Forms, noticeably through my most famous BoolCarrier class, contained in the MyDroidFactoryUI class, or via the listOfDroids being transferred by reference.

Also, I found interesting ways to make my code more concise, for instance by removing the block associated to the namespaces, removing an indent level, or also by returning directly my regex validations.

That being said, I believe I did not make full use of polymorphism in my code and this is something I believe I will have to refactor in my code in a future version release.

For me, having the liberty to make this project more customizable was a great way to bring more passion to it and helped me greatly to want to browse the internet in a quest to improve my application. I had to experiment a lot on my own and I strongly believe it paid off in the end!

# 4 – Bibliography

Microsoft Windows Forms Documentation:

<https://docs.microsoft.com/en-us/dotnet/desktop/winforms/overview/?view=netdesktop-6.0>

Microsoft Web View 2 Documentation:

<https://docs.microsoft.com/en-us/microsoft-edge/webview2/>

Invidious GitHub Page for URL parameter documentation

<https://github.com/iv-org/documentation/blob/master/docs/url-parameters.md>

Google – Youtube’s URL parameter documentation

<https://developers.google.com/youtube/player_parameters>

StackOverflow for many questions regarding this project

<https://stackoverflow.com/>