

**Software Engineering**  
**Modeling and Implementation Project**

---

**“Simulator of a Cassette Deck”**

**- Implementation -**

---

By Constant Théo and Essafsyfy Abdelkrim

Academic year 2018-2019

# Contents

<b>1</b>	<b>Implementation choices</b>	<b>1</b>
1.1	<i>Scene Builder</i> . . . . .	1
1.2	<i>Maven</i> . . . . .	1
1.3	<i>IDEs</i> . . . . .	1
1.4	<i>MVC</i> . . . . .	1
<b>2</b>	<b>Modeling differences</b>	<b>2</b>
<b>3</b>	<b>Design Patterns</b>	<b>3</b>
<b>4</b>	<b>Known Issues</b>	<b>3</b>
<b>5</b>	<b>Miscellaneous</b>	<b>3</b>
5.1	<i>Git</i> . . . . .	3
5.2	Video link . . . . .	4

# 1 Implementation choices

## 1.1 *Scene Builder*



For not only the modeling but also the implementation phase of the project, we have used *Scene Builder* to design the *JavaFx GUI*.

*FXML* files were generated and introduced to our project.

## 1.2 *Maven*



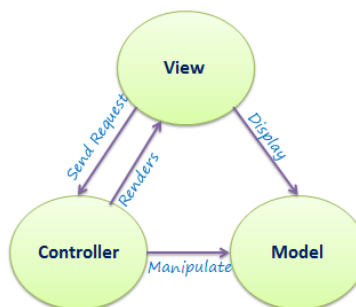
We have chosen *maven* to easily manage jar files and organize the project.

## 1.3 *IDEs*



To be sure the application is *IDE*-independent, we have used each a different java *IDE*, *Netbeans* and *Eclipse*. It is note worthy that we have encountered some behavioral differences caused by these *IDEs*, noticeably, the path each one uses to acces the *fxml* files.

## 1.4 *MVC*

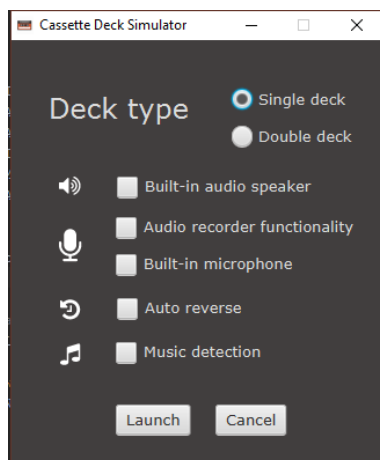


We have tried to use as best we could the Model, View, Controller architecture, the project uses 2 of the 3 packages, "model" and "controller", for the "view" we have encountered -as said above- differences between the *IDEs* which forced us to place the *fxml* files in the *src/main/resources* folder.

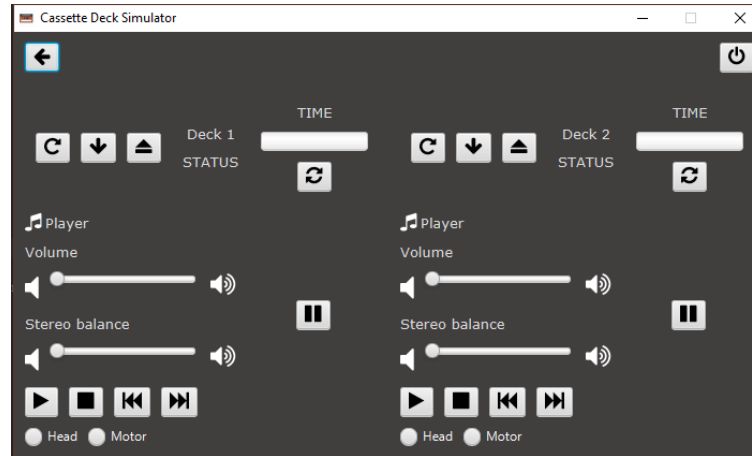
## 2 Modeling differences

There have been some changes to the model since its conception in the **Modeling Report**, notably:

- The use of *GridPaneLayout* and *H/VBoxes* to contain the different nodes.
- In the Launcher, the function "Built-in microphone" is no longer disabled if the "Audio recorder functionality" isn't checked, also, a "Music detection" *checkbox* has been added.



- In case of a double deck, the functionality is split to two decks instead of "Player" and "Recorder" in the previous model.
- Added missing *Previous/Next Song*, *Flip Cassette* and *Source* -if the deck has a built-in audio Speaker- buttons.



- Replaced buttons' text with icons and a mouse-over tooltip to display their name.
- Removed "Remove cassette" button, due to its redundancy with the Eject button. In this simulation, ejecting the cassette holder and removing the cassette within are both done by clicking the "Eject" button.
- "Pause" button is now centered between the "Player" and "Recorder".

### 3 Design Patterns

For this application, we implemented the "state machine" design pattern to handle the different states of the Deck class. Indeed, the deck (s) of the device will not have the same behavior at the start of a button depending on the state in which they are (s). A deck can be turned off, on hold, playing, fast-forwarding, fast-rewinding, or recording. We have chosen not to make a pause state, because, in our opinion, it simply is a possible action when the deck is in the reading or recording state. The pause button is a button that can be held down at the same time as the others, and can be engaged at any time. This is not the case for other buttons that disengage themselves when pressing another button (other than pause), and that can not be engaged at any time. The state machine consists of a State interface that inherits each possible state. The class that can take these states (Deck in this case) has a State attribute and the behavior of all Deck methods that are likely to vary by state is delegated to State. This State object will change each time the state changes, thus another class will take care of Deck's behavior.

## 4 Known Issues

- JUnit tests have not been completed.
- 

## 5 Miscellaneous

### 5.1 *Git*

Throughout the project, we have used version control *git* and *github* to organize and avoid conflicts while working on the project. After receiving feedback from the modeling report, we have decided to make the repository private. We invite you to send a contribution request to either *remagkes@gmail.com* or *to.constant@gmail.com*

### 5.2 Video link

As asked in the requisites, a video demonstrating the basic features of the simulation can be accessed via this link: