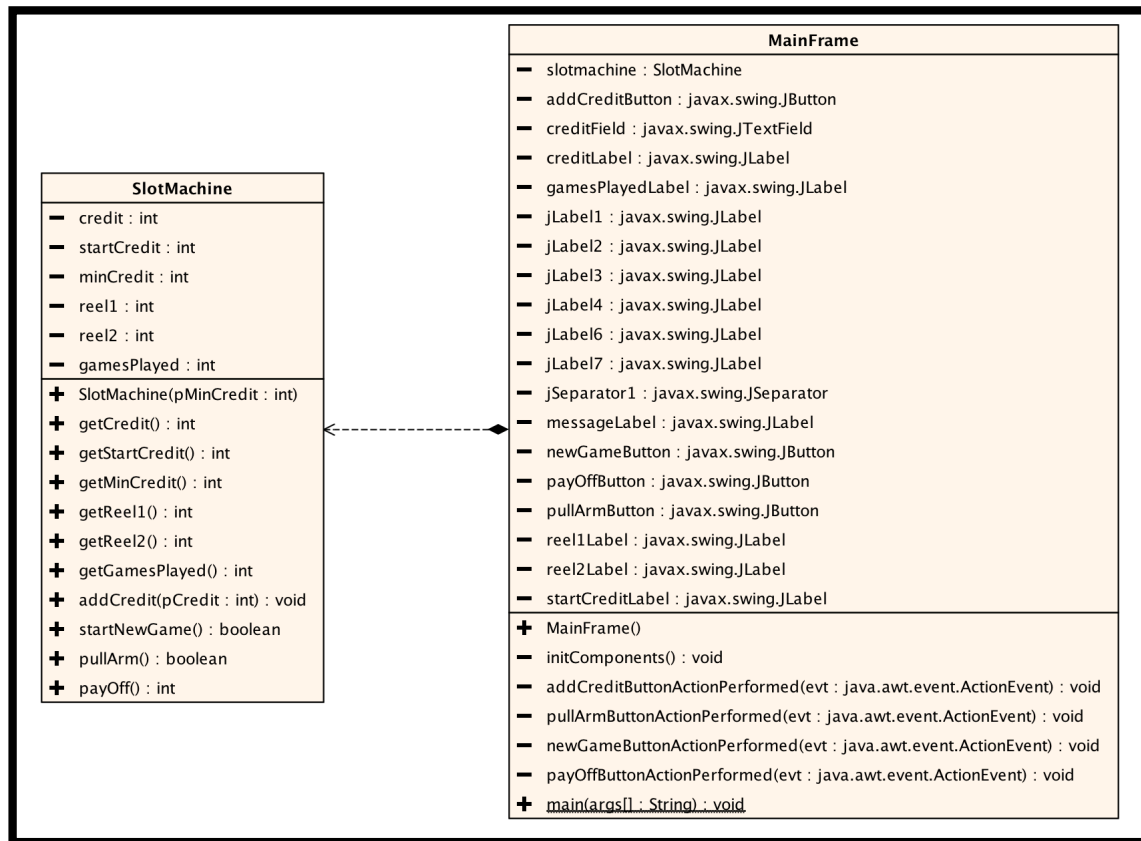


Exercice C – Machine à sous

Il s'agit de développer un programme, qui permet de simuler d'une façon simplifiée une machine à sous utilisée dans les casinos.



Partie 1 – Création du modèle

1. Créez un projet **ExerciceC** dans Unimozer.
2. Créez la classe `SlotMachine`.
3. Ajoutez les attributs suivants
 - a. `credit` représente le solde actuel du joueur
 - b. `startCredit` représente le capital de départ du joueur
 - c. `minCredit` définit le crédit minimal qu'un joueur doit posséder pour pouvoir jouer une partie. (Il s'agit du prix d'une partie.)
 - d. `reel1` et `reel2` représentent les rouleaux contenant les résultats des tirages aléatoires.
 - e. `gamesPlayed` contient le nombre de tirages aléatoires depuis le début de la partie.
4. Ajoutez des accesseurs pour tous les attributs.
5. Ajoutez un constructeur qui initialise tous les attributs à 0 sauf `minCredit` qui est initialisé à la valeur fournie en paramètre.
6. Ajoutez une méthode `addCredit` qui permet d'ajouter des crédits au solde actuel ainsi qu'au capital de départ.

7. Ajoutez une méthode `startNewGame` qui démarre une nouvelle partie à condition qu'il n'y ait plus de sous dans la machine. Elle retourne `true` au cas où une nouvelle partie a été commencée avec succès, `false` sinon. Au cas où une nouvelle partie est commencée, le capital de départ et le nombre de parties jouées sont réinitialisés à 0.
8. Ajoutez une méthode `pullArm` qui permet de lancer la génération d'une valeur aléatoire pour les champs `reel1` et `reel2`. Elle retourne `true` si le joueur possède assez d'argent pour jouer, `false` sinon. En plus la méthode actualise le crédit actuel selon les règles suivantes :
 - a. les valeurs générées pour les deux rouleaux sont 1 → le crédit du joueur est augmenté de 20€,
 - b. les valeurs générées pour les deux rouleaux sont 2 → le crédit du joueur est augmenté de 15€,
 - c. les valeurs générées pour les deux rouleaux sont 3 → le crédit du joueur est augmenté de 10€,
 - d. la valeur 1 a été générée pour l'un des deux rouleaux → le crédit du joueur est augmenté de 5€,
 - e. dans tous les cas le joueur doit payer le prix d'une partie.
9. Ajoutez la méthode `payOff` qui paye les gains au joueur et retourne la différence entre le capital final et le capital de départ du joueur.

Partie 2 – Création de l'interface graphique

1. Créez l'interface graphique suivante dans NetBeans.

2. Renommez les variables des éléments que vous devez accéder au code, tout en respectant les conventions vues au cours.
3. Etablissez un lien de communication entre la vue/le contrôleur et le modèle.

4. Définissez le comportement du bouton **Add Credit** qui prend la valeur du champ de texte adjacent pour l'ajouter au crédit actuel et au capital de départ du joueur.
5. Définissez le comportement du bouton **New Game** qui soit initialise une nouvelle partie tout en affichant le message **Insert coins to play**, soit affiche le message **Get your money first !** dans la partie inférieure de la fenêtre.

Cas 1 : solde actuel = 0 -> Nouvelle partie & Insert coins to play

The screenshot shows a game interface with a light blue background. At the top, there are three rows of text and buttons: 'Current Credit: 0€' followed by an empty text input field and an 'Add Credit' button; 'Start Credit: 0€' followed by a 'New Game' button; and 'Minimum Credit to play: 5€' followed by a 'Pay Off' button. Below these is a line 'Games Played: 0'. A horizontal separator line divides the interface. Below the line, 'Reel 1:' is followed by the number '1' and a 'Pull Arm' button. Below that, 'Reel 2:' is followed by the number '1'. At the bottom, the text 'Insert coins to play' is displayed.

Cas 2 : solde actuel ≠ 0 -> Get your money first!

The screenshot shows the same game interface as Case 1, but with different values. 'Current Credit' is now '20€' and the adjacent text input field contains the number '10'. 'Start Credit' is now '5€'. 'Minimum Credit to play' remains '5€'. 'Games Played' remains '0'. Below the separator line, 'Reel 1:' is followed by '1' and a 'Pull Arm' button. 'Reel 2:' is followed by '1'. At the bottom, the text 'Get your money first!' is displayed instead of 'Insert coins to play'.

6. Définissez le comportement du bouton **Pull Arm****Cas 1 – Le joueur a un solde suffisant pour jouer**

Si le joueur a des crédits suffisants pour jouer, un tirage aléatoire est lancé et les nouvelles valeurs des champs `reel1`, `reel2`, `credit` et `gamesPlayed` sont affichées. La partie inférieure indique le montant gagné ou perdu lors du dernier tirage.

The screenshot shows a game interface with a light blue background. At the top, there are four labels: 'Current Credit: 10€', 'Start Credit: 5€', 'Minimum Credit to play: 5€', and 'Games Played: 4'. To the right of 'Current Credit' is a text input field containing '10'. Below these labels are three buttons: 'Add Credit', 'New Game', and 'Pay Off'. A horizontal line separates the top section from the bottom section. In the bottom section, 'Reel 1:' is followed by the number '3', and 'Reel 2:' is followed by the number '3'. To the right of these numbers is a button labeled 'Pull Arm'. At the bottom of the interface, a message reads: 'You won 5€. Pull arm again or get your money.'

(**Current Credit** avant le tirage : 5€)

Cas 2 – Le joueur a un solde insuffisant pour jouer

Si le joueur a des crédits insuffisants pour jouer, un message correspondant est affiché dans la partie inférieure.

The screenshot shows the same game interface as Case 1, but with different values. At the top, the labels are: 'Current Credit: 0€', 'Start Credit: 0€', 'Minimum Credit to play: 5€', and 'Games Played: 0'. The 'Add Credit' button is now disabled. The 'Pull Arm' button is also disabled. At the bottom of the interface, a message reads: 'Not enough credit. Insert more coins.'

7. Définissez le comportement du bouton **Pay Off**

L'affichage est actualisé (nouvelles valeurs pour `credit` et `startCredit`) et soit le gain total soit la perte totale du joueur est affiché.

The screenshot shows a slot machine game interface with a light blue background and a black border. The interface is divided into two main sections by a horizontal line. The top section contains the following elements: 'Current Credit: 0€' followed by a text input field containing '10', an 'Add Credit' button, 'Start Credit: 0€', a 'New Game' button, 'Minimum Credit to play: 5€', a 'Pay Off' button, and 'Games Played: 14'. The bottom section contains 'Reel 1:' followed by the number '1', a 'Pull Arm' button, 'Reel 2:' followed by the number '1', and 'You won 15€'.

Current Credit: 0€	10	Add Credit
Start Credit: 0€		New Game
Minimum Credit to play: 5€		Pay Off
Games Played: 14		
Reel 1:	1	Pull Arm
Reel 2:	1	
You won 15€		