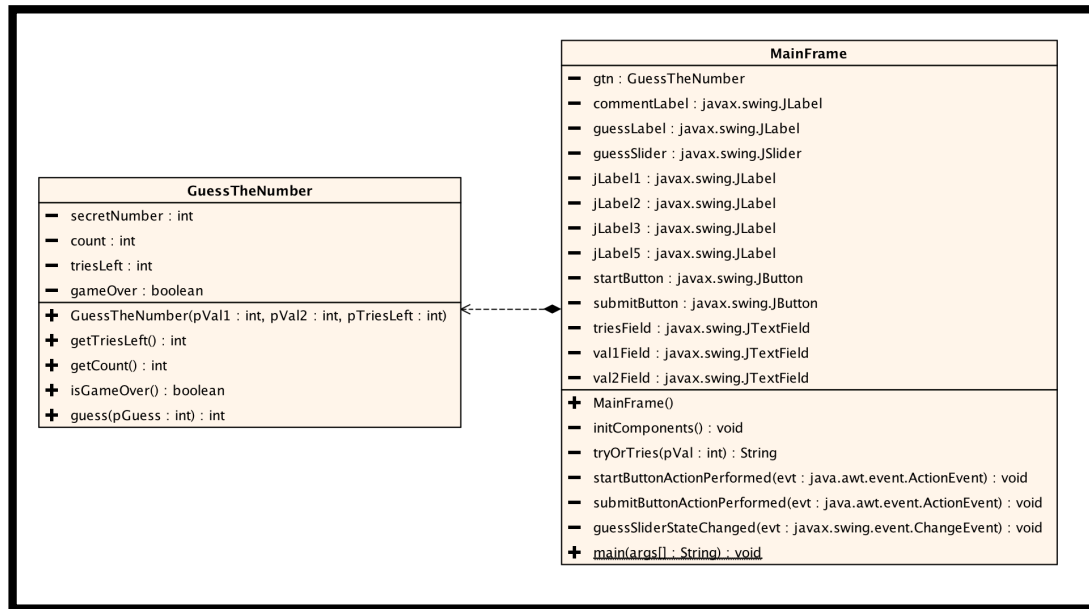


# Exercice I1 – Guess The Number

Le but de cet exercice est de créer un jeu qui consiste à deviner un nombre aléatoire généré par l'ordinateur.



## Création du modèle

1. Créez un projet **ExerciceI1** dans Unimozzer
2. Créez la classe `GuessTheNumber`
3. Ajoutez les attributs suivants :
  7. `secretNumber` à valeur entière qui contient le nombre que l'utilisateur doit deviner
  8. `count` à valeur entière qui compte le nombre d'essais de l'utilisateur
  9. `triesLeft` à valeur entière qui stocke le nombre d'essais restants
  10. `gameOver` à valeur booléenne qui indique si le jeu est terminé
4. Ajoutez des accesseurs `getCount`, `getTriesLeft` et `isGameOver` pour les attributs correspondants. Il n'y a pas d'accesseur pour le nombre secret.
5. Ajoutez une méthode `guess` qui prend un nombre entier `pGuess` en paramètre et le compare au nombre secret si le jeu n'est pas déjà terminé.
  7. Si le jeu est terminé, la méthode retourne le nombre -2.
  8. Si le jeu n'est pas encore terminé, la méthode compare `pGuess` et `secretNumber` et retourne ...
    - i. 0 si les nombres sont égaux
    - ii. -1 si `pGuess` est plus petit que le nombre secret
    - iii. 1 si `pGuess` est plus grand que le nombre secret



*N'oubliez pas d'adapter les valeurs des attributs `count`, `triesLeft` et `gameOver` au cas échéant*





## Création de l'interface graphique

Créez une classe *MainFrame.java* et construisez l'interface graphique suivante par glisser-déposer :

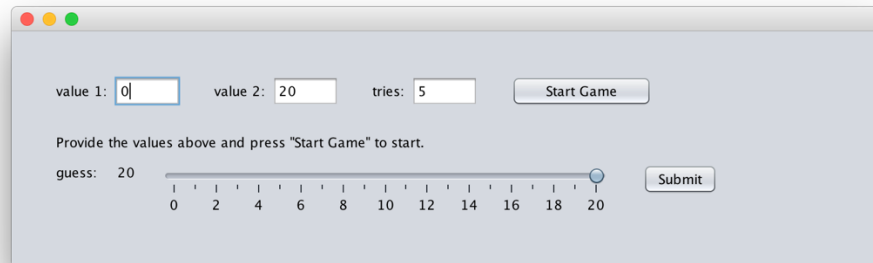


Figure 1 - Interface au démarrage du programme, les valeurs 0, 20 et 5 sont présentes au démarrage

1. Renommez toutes les variables de tous les éléments graphiques que vous devez accéder.
2. Procédez aux changements nécessaires pour que la classe *MainFrame* puisse communiquer avec la classe modèle.
3. Pourquoi n'est-il pas possible cette fois-ci de déclarer et d'initialiser la classe modèle en même temps ?
4. Comment faut-il initialiser la classe modèle cette fois-ci ?
5. Définissez le comportement du bouton **Start Game**. Quand le bouton *Start Game* est appuyé les valeurs des trois champs suivants sont reprises pour initialiser la classe modèle. Aussi l'étiquette du milieu de la fenêtre confirme l'initialisation du jeu en indiquant un message correspondant :

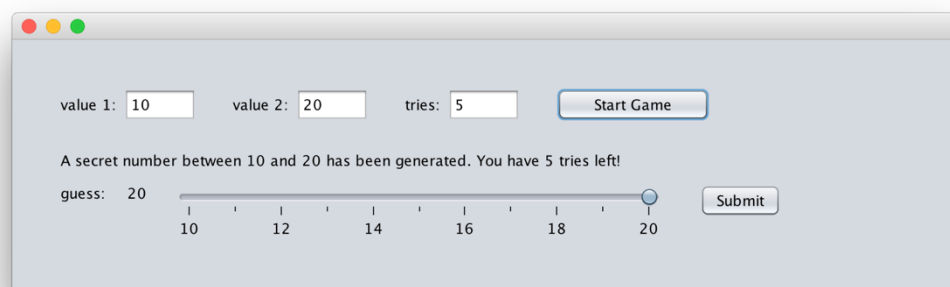


Figure 2 - Un message confirme l'initialisation du jeu avec les paramètres saisis

Les valeurs qui peuvent être saisies à travers la glissière sont adaptées selon les valeurs entrées par l'utilisateur.

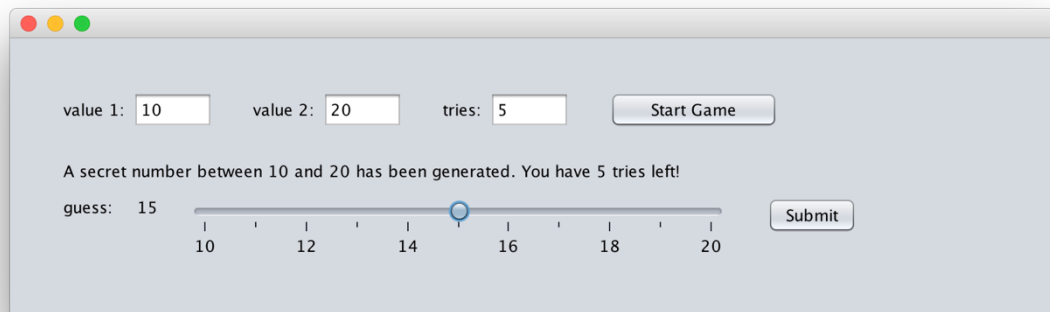
7. Avant l'initialisation du jeu, le fait d'appuyer sur le bouton **Submit** n'a aucun effet. Seulement après l'initialisation du jeu, il est possible de deviner le nombre secret à travers la glissière et le bouton **Submit**.

Après chaque tentative de l'utilisateur l'étiquette au-dessus de la glissière indique si le



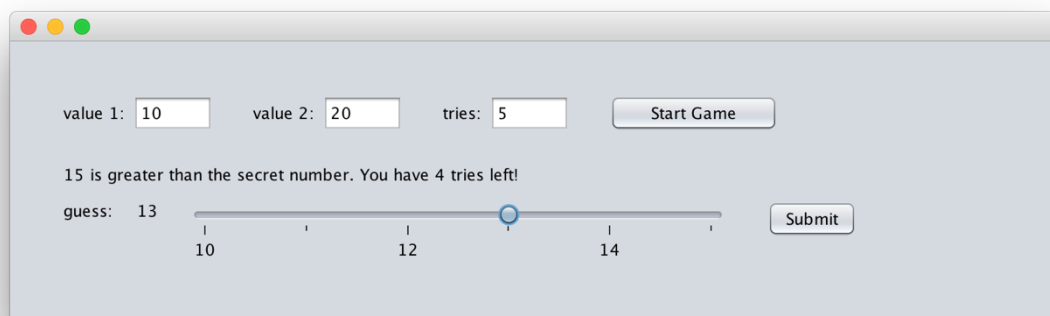
nombre saisi est égal au nombre secret ou s'il est plus petit ou plus grand que le nombre secret. A chaque fois les valeurs qu'on peut entrer à travers la glissière sont adaptées.

### Avant d'avoir appuyé sur le bouton



The screenshot shows a window with a light blue background. At the top, there are three colored window control buttons (red, yellow, green). Below them, there are three input fields: 'value 1:' with '10', 'value 2:' with '20', and 'tries:' with '5'. To the right of these is a 'Start Game' button. Below the input fields, a message reads: 'A secret number between 10 and 20 has been generated. You have 5 tries left!'. Underneath this, there is a 'guess:' label followed by the number '15'. To the right of the guess is a horizontal slider with a blue circle handle positioned at 15. The slider has tick marks and labels at 10, 12, 14, 16, 18, and 20. To the right of the slider is a 'Submit' button.

### Après avoir appuyé sur le bouton



The screenshot shows the same window as before, but with updated information. The 'value 1:' field still shows '10', 'value 2:' shows '20', and 'tries:' shows '5'. The 'Start Game' button is still present. The message now reads: '15 is greater than the secret number. You have 4 tries left!'. The 'guess:' label now shows '13'. The slider's blue circle handle is now positioned at 13, and the visible tick marks and labels on the slider are 10, 12, and 14. The 'Submit' button remains on the right.

Si le nombre saisi par l'utilisateur n'est pas le bon nombre, le nombre de tentatives restantes est affiché aussi.

Finalement, si le jeu est terminé, soit parce que le nombre secret a été deviné, soit parce que le nombre de tentatives restantes vaut zéro, le message *Game Over* est affiché aussi.

Si maintenant l'utilisateur appuie à nouveau sur le bouton *Submit*, un message lui rappelle que le jeu est terminé. Il peut cependant démarrer un nouveau jeu en appuyant sur *Start Game*. Il est possible de garder les valeurs actuelles ou de les changer avant de commencer un nouveau jeu.



**Ajoutez la méthode suivante à votre classe `MainFrame` :**

```
private String tryOrTries(int pVal){  
    if (pVal == 1){  
        return "try";  
    }else{  
        return "tries";  
    }  
}
```



**Elle vous sera utile pour créer des phrases grammaticalement correctes : ..., 2 tries left, 1 **try** left, ..., You guessed right after 4 tries, ... You guessed right after 1 **try****