|  |
| --- |
| DIVTEC |
| Documentation Technique |
| Jeu du pendu |

|  |
| --- |
| Minger Sophie  02/12/2019 |

Table des matières

[Description du projet 2](#_Toc29987281)

[UML 2](#_Toc29987282)

[Convention de nommage 2](#_Toc29987283)

[Les composants 2](#_Toc29987284)

[Les variables et les méthodes 2](#_Toc29987285)

[Classe utilisé 2](#_Toc29987286)

[JavaDoc 2](#_Toc29987287)

[Conclusion 2](#_Toc29987288)

# Description du projet

Le projet de l’atelier 31 Programmation Orienté Objets, il fallait crée une application graphique à l’aide de Java ou C ++.

Pour ma part il s’agit d’un Hangman Games, un jeu du pendu en français.

# UML

L’UML concernant cet atelier est en annexe. Le fichier se nomme : MINGSOP\_UML\_31OOB

Il est disponible en format StarUML ou image.

# Convention de nommage

## Les composants

Les composants tels que les boutons sont suffixé d’un BTN\_ ou LB\_ (pour un label) suivit de sa fonction. Exemple

**Button BTN\_DemarrageJeu = new Button();**

Cette ligne de code si dessus concerne le bouton pour démarrer le jeu.

## Les variables et les méthodes

Si la variable est composée que d’un mot, celui-ci est en minuscule.

Si dans le cas, la variable est composée de deux ou plusieurs mots, chaque nouveau mot est distinguer grâce à une majuscule.

**private static final String fichierMots ="/ListMots/fr\_MotsFaciles.txt";**

Pour ce qui est des méthodes, les règles de nommages sont identiques.

# Classe utilisé

Il existe 3 classes dans mon jeu :

* TheHangmanGames
* FenetreModal
* LecteurMot

# JavaDoc

Le fichier html pour javaDoc se trouve **: /TheHangmanGames/dist/javadoc/index.html**