



AIX-MARSEILLE UNIVERSITÉ

M3105 - CONCEPTION ET PROGRAMMATION
OBJET AVANCÉES

Jeu de Pente

Auteurs :
Lucien Aubert
Auguste Taillade

Enseignant :
Sébastien Thon

Table des matières

1	Introduction	2
1.1	Présentation du projet	2
2	Analyse	2
2.1	Classes utilisées	2
2.2	Relations entre les classes	2
2.3	Fonctionnement global	2
3	Réalisation	2
3.1	Choix techniques	2
3.2	Algorithmes "complexes" utilisés	3
4	Utilisation	3
4.1	mode d'emploi	3
4.2	configuration requise	3
5	Conclusion	3
5.1	Bilan	3
5.2	Optimisations possibles	3
5.3	Extensions possibles	4

1 Introduction

1.1 Présentation du projet

2 Analyse

2.1 Classes utilisées

Faire un diagramme et changer le titre en "Diagramme de classe"

2.2 Relations entre les classes

Cf diagramme au dessus.

2.3 Fonctionnement global

Si on a la foi, faire un diagramme de séquence. Sinon, expliquer comment ça fonctionne.

3 Réalisation

3.1 Choix techniques

Nous avons choisi d'utiliser Swing ([Ajouter ref.](#)), bibliothèque graphique présente dans les JFC ([Ref. https://en.wikipedia.org/wiki/Java_Foundation_Classes](https://en.wikipedia.org/wiki/Java_Foundation_Classes)) car elle est plus performante que AWT ([Idem, ref.](#)). De plus, elle est équipée d'une interface `ChangeListener` ([Ref. https://docs.oracle.com/javase/8/docs/api/index.html?javax/swing/event/ChangeListener.html](https://docs.oracle.com/javase/8/docs/api/index.html?javax/swing/event/ChangeListener.html)) qui permet de gérer le survol d'un élément graphique en toute simplicité, comme n'importe quel autre événement utilisateur.

Nous avons fait hériter nos classes de celles de Swing afin de les personnaliser. Ainsi nous avons pu modifier le comportement et l'apparence des widgets

utilisés, comme les pions, par exemple, qui sont dessinés procéduralement et non stockés sous forme d'image.

Utilisation de swing pour les widgets

↔ Création de classes enfants afin de les personnaliser

Surcharge de la méthode `Component::paintComponent()`

3.2 Algorithmes "complexes" utilisés

Explication de la méthode `Board::checkMove()`.

4 Utilisation

4.1 mode d'emploi

Tu lances, tu joues.

4.2 configuration requise

Lol

5 Conclusion

5.1 Bilan

5.2 Optimisations possibles

On peut trouver une implémentation plus propre pour le survol des pions.

5.3 Extensions possibles

Améliorer la création et l'accès aux/de parties en ligne (un pool de parties en cours/en attente d'adversaire, un mode observateur, un classement en ligne)

Références