

# RAPPORT DIABALLIK



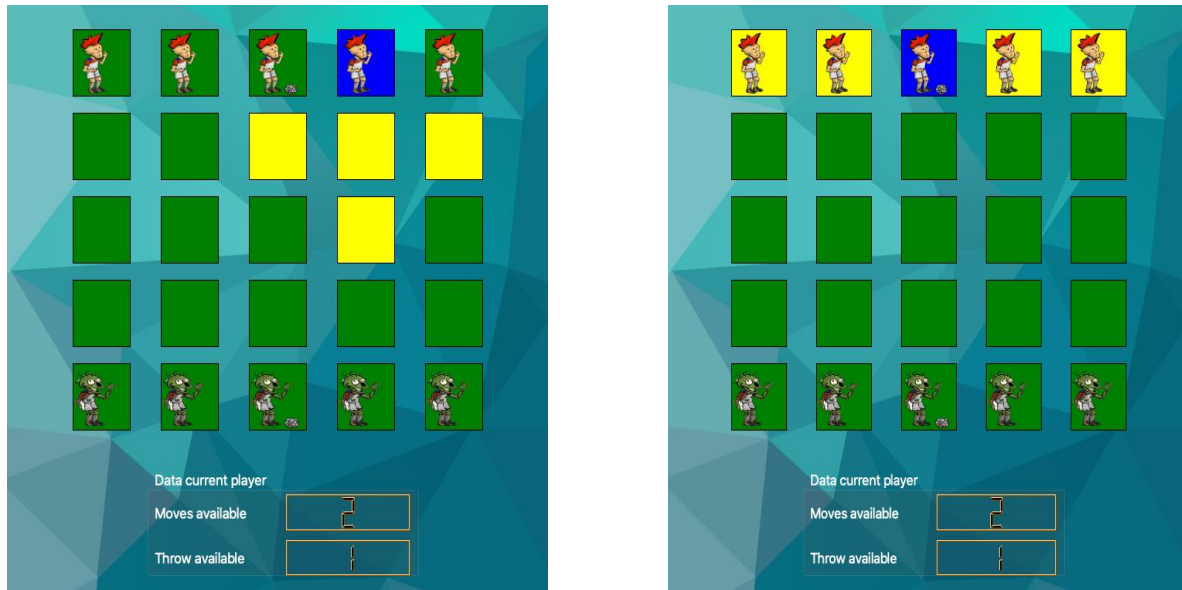
DE SOUSA PEREIRA, ANDRE FILIPE - 51999 | TOUDA LACHIRI, ANAS - 43256

## Table des matières

<b><i>Rajouts de fonctionnalités par rapport à l'énoncé: .....</i></b>	<b><i>3</i></b>
Gestion des actions: .....	3
Retour d'informations : .....	3
<b><i>Écarts/bogues par rapport à l'énoncé :.....</i></b>	<b><i>4</i></b>
<b><i>Problèmes rencontrés : .....</i></b>	<b><i>4</i></b>
<b><i>Estimation du temps passé sur cette partie de l'énoncé : .....</i></b>	<b><i>4</i></b>

## Rajouts de fonctionnalités par rapport à l'énoncé:

### Gestion des actions :

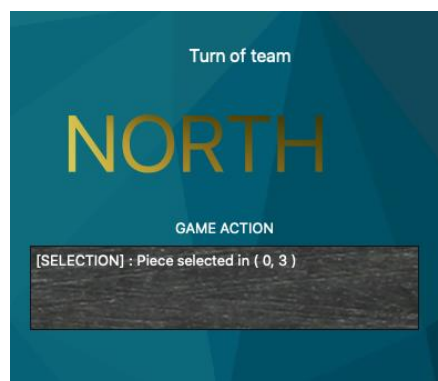


Les déplacements & lancés possibles sont mis en surbrillance sur le plateau de jeu. Grâce à l'ajouts des méthodes suivante dans le modèle, qui vont donc permettre de savoir si une position dans le plateau est jouable ou pas selon le type d'action:

```
bool Diaballik::canPass(const Position & pos) const
```

```
bool Diaballik::canMove(const Position & pos) const
```

### Retour d'informations :



Le retour d'informations sur les actions des utilisateurs se fait dans la partie droit de l'écran. De ce fait les utilisateurs sont notifié de leur sélection de pièce, mauvaise sélection...

### Écarts/bogues par rapport à l'énoncé :

Aucun écart par rapport à l'énoncé, ni de présence de bogues à notre connaissance. La vue utilise directement le modèle de la version console. Aucun nouveau bug a été rencontré dans la version graphique.

### Problèmes rencontrés :

Décrit les solutions trouvées et peut être non idéales pour les problèmes rencontrés :

- **Utilisation de Classes CSS**

Afin de modifier les couleurs de chaque SquareUI il était envisagé d'utiliser des classes CSS pour modifier la couleur de fond. La méthode utilisée pour assigner une classe CSS à chaque SquareUI était :

```
this->setProperty('class', 'SquareUI');
```

En effet cette méthode était valide, mais lors d'un deuxième appel pour attribuer une autre class CSS par exemple **SquareUISelected**, ce deuxième appel n'était pas tenu en compte :

```
this->setStyleSheet("");
```

```
this->setProperty('class', 'SquareUISelected');
```

Donc les méthodes `setInteractable()`, `setSelected()`, `setDefaultBackground()` dans SquareUI font directement appel à **this->setStyleSheet('...')** ;

- **Retourner au menu principal après la partie**

Lorsque l'utilisateur décidait de revenir au menu principal après avoir commencée une première partie, des traces de la première partie étaient affichées durant la deuxième partie. Plusieurs après plusieurs tentatives, la solution trouvée était de redémarrer de core de l'application.

- **Sélectionner un autre joueur après avoir sélectionné le joueur qui contient la balle**

En utilisant un seul clic (par exemple clic gauche de la souris), il était impossible pour le programme de déduire si l'utilisateur veut passer la balle ou si l'utilisateur veut sélectionner un nouveau joueur, après avoir sélectionné le joueur qui contient la balle. Donc il est possible d'utiliser le clic droit uniquement pour sélectionner.

### Estimation du temps passé sur cette partie de l'énoncé :

Le temps passé sur la partie graphique du projet Diaballik est d'environ 3 semaines, avec une moyenne d'environ 15h par personne.