Dossier de Validation IHM

Projet S6 - 2023

Table des matières

Introduction

Nous avons, à la suite de plusieurs retours et idées au sein groupe, construit l'interface des interactions humains-machines pour permettre aux utilisateurs de notre programme de pouvoir facilement prendre en main et effectuer une ou plusieurs partie de ce jeu de plateau que nous avons adapté sur ordinateur à l'aide du langage de programmation java et de la bibliothèque graphique Swing.

Affichage du plateau de jeu

Affichage du plateau

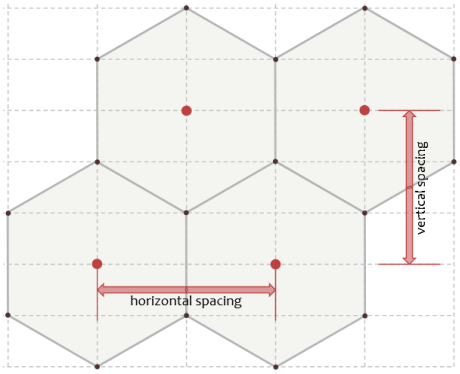
Affichage des tuiles hexagonales du plateau de jeu

Nous avons décidé d'afficher les tuiles hexagonales du plateau de la même manière que le jeu d'origine, c'est-à-dire avec des hexagones pointés verticalement. Le jeu est composé de 8 lignes de tuiles composé chacune à tour de rôle de 7 et 8 hexagones, toutes les lignes impaires sont composées de 7 hexagones et les lignes pairs de 8 hexagones.

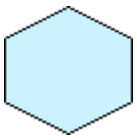
Pour afficher tous ces hexagones nous avons utilisé des images hexagonales de tailles carré (64x64 pixels). Comme chaque hexagone doit être affiché de manière à être positionné côte à côte. Comme la pointe des hexagones est orientée à la verticale. Il faut faire un certains calcul pour obtenir l'écart de l'espace horizontal et vertical à ajouter entre chaque hexagone créer un quadrillage hexagonal, et aussi ajouter un léger décalage horizontal au début d'une 1 rangée sur 2 pour aligner le quadrillage, ce décalage vaut la moitié de la largeur d'un hexagone.

- Le calcul de l'espacement horizontal entre les hexagones est la largeur d'un hexagone L.
- Le calcul de l'espacement vertical est la hauteur d'un hexagone H multiplié par ¾ . H*¾.

Schéma de l'espacement pour les hexagones :

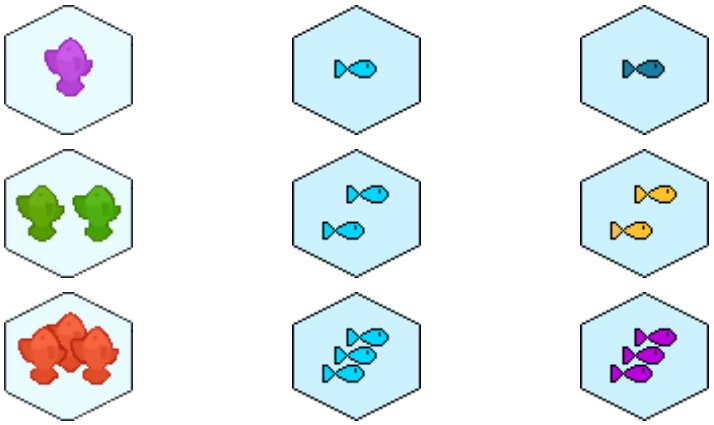


Tuile hexagonale de notre plateau sans information dessus :

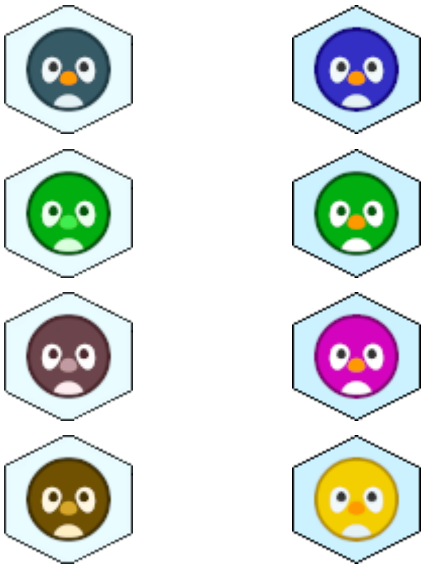


Après de multiples retours nous avons décidé à plusieurs reprises de modifier les images correspondantes à l'allure des pingouins et des poissons. Les pingouins ont la même forme et 4 couleurs différentes. Les poissons ont la même forme et la même orientation, ils se situent dans des paternes de 1, 2 ou 3 poissons les uns au-dessus des autres. Les poissons ont une couleur différente pour chaque nombre de poissons présent sur la tuile, pour pouvoir repérer plus rapidement la valeur de chaque tuile. Nous avions comme idée d'ajouter différents accessoires de différentes couleurs sur chaque pingouins et avoir les pingouins de la même couleur, mais nous nous sommes retenus d'implémenter cette idée pour rester dans les contraintes de temps du projet et ne pas nous retarder avec des fonctionnalités moins importantes. Ces décisions sur l'allure des pingouins et poissons sont le résultat de multiples tests et retours de différents utilisateurs.

Poissons début : Poissons Milieu : Poissons Fin :



Pingouins début : Pingouins Fin:



Affichage du type d'une tuile en fonction de l'état du jeu

Dans le jeu il existe au minimum 7 types de tuiles hexagonales :

- Tuile a 1 poisson (1)

- Tuile a 2 poisson (2)
- Tuile a 3 poisson (3)
- Les Pingouins du joueur numéro 1 (4)
- Les Pingouins du joueur numéro 2 (5)
- Les Pingouins du joueur numéro 3 (6)
- Les Pingouins du joueur numéro 4 (7)

Il existe implicitement le type des tuiles vides qui peuvent prendre le nombre 0.
Pour afficher une tuile d'un certain type, on charge différentes images hexagonales et on affiche une image en fonction du nombre lu sur la variable `plateau[][]`. Le reste de types d'images s'affichent sur le plateau selon ces 8 valeurs ainsi qu'en fonction de l'état du jeu et de l'historique.

Mise des hexagones à la bonne échelle

Pour mettre les images des tuiles à la bonne échelle, selon la taille de la fenêtre, on met à jour les dessins du plateau quand on redimensionne la fenêtre. Pour cela la taille de l'image est dépendante de la largeur et de la hauteur de la fenêtre par une multiplication. On récupère la largeur de la fenêtre avec `getWidth()` et sa hauteur avec `getHeight()`.

Gestion des interactions avec le plateau

Gestion des clics sur le plateau pour correspondre aux coordonnées d'un hexagone sur le plateau.

Pour qu'un utilisateur puisse jouer et faire avancer le jeu, son clic sur le plateau doit être transformé en coordonnées qui correspondent à une tuile du plateau dans le moteur de jeu.
Le plateau de jeu est stocké dans une matrice, il nous faut donc 2 coordonnées qui correspondent aux indices à transférer au contrôleur lors d'un clic valide sur le plateau.

Depuis un `mouseEvent e` on récupère une valeur X et une valeur Y du point du clic, on effectue ensuite les calculs suivants pour déterminer la tuile correspondantes : On divise X par la largeur et Y par la hauteur d'un hexagone. A cause des pointes verticales des hexagones on multiplie Y pour couvrir ces zones. A toutes les lignes pairs, il y a un décalage d'une moitié de la largeur d'un hexagone, on enlève ce décalage pour X lorsque la nouvelle valeur de Y est impair (On compte à partir de 0 la première donc à la première ligne pair Y == 1). Après ces transformations X et Y sont envoyés au contrôleur en tant que L et C respectivement le numéro de ligne et le numéro de colonne. Pour éviter les clics de coordonnées invalides, on ne n'envoie rien si X ou Y est supérieur à 7, à cause des 8 lignes et colonnes du plateau, d'indices 0 à 7. Pour respecter les contraintes de temps nous n'avons pas centré le plateau par rapport aux autres composants Swing, et la fenêtre car la gestion des cliques aurait dû être recommencée après le centrage du plateau.

Affichage des pingouins d'un joueur durant l'état de déplacement

Pour exprimer visuellement que les pingouins sont sélectionnables par le joueur du tour actuel, nous avons de façon similaire aux cases de poisson accessible, d'encadré en violet

les pingouins du joueur. De plus une fois un pingouin cliquez, pour montrer lequel est sélectionné, on ajoute 2 flèches rouges à horizontal aux bords de l'hexagone.
Pingouins sélectionnable :

Début : Milieu : Fin :

Pingouin sélectionné :

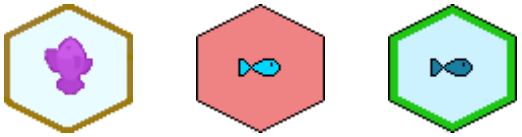
Début : Milieu : Fin :

Affichage des coups possibles lors des déplacements

Nous avons décidé d'afficher les coups possibles d'un pingouin après avoir cliquer sur lui.
A l'aide de l'appel de la méthode `hexAccessible(int x, int y)` avec les coordonnées d'où se trouve le pingouin et nous renvoie les coordonnées dans une `ArrayList[]` de toutes les cases accessibles par ce pingouin. Nous avons testé plusieurs versions pour afficher les cases accessibles, d'abord nous avons décidé d'un encadrement marron, puis de colorier le fond d'une tuile en rouge et enfin de faire un encadrement autour des poissons. Après les tests nous avons décidé de faire un encadrement vert qui suit les bords de la tuile.

Case de poisson sélectionnable (ici à 1 poisson)

Début : Milieu : Fin :



Affichage des coups possible durant l'état de placement du jeu

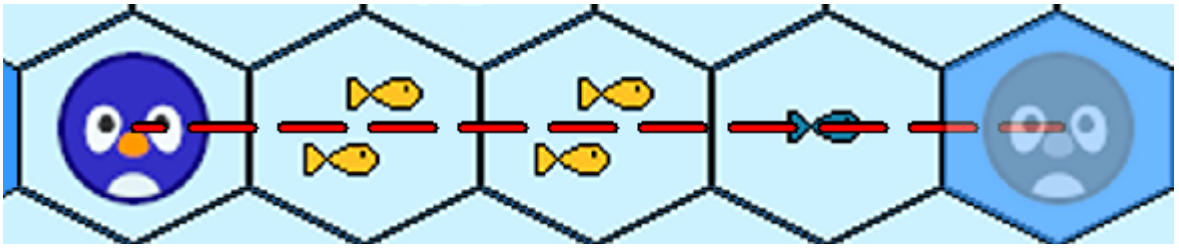
Nous avons décidé d'afficher les cases où l'on peut poser les pingouins comme les cases de déplacements possible, dans un encadré hexagonal vert. Pour trouver les cases où l'on peut poser un pingouin, on cherche donc les cases à 1 poisson pour les encadrer avec la méthode `getCases(1)`;

Fonctionnalités supplémentaires

Affichage de l'historique du dernier coup après un déplacement

Pour signaler un déplacement nous avons choisi de dessiner un pingouin "fantôme" à l'emplacement de la tuile qui vient d'être enlevée, ainsi qu'un trait en pointillé rouge allant du centre de la tuile quitté au centre de la tuile d'arrivée du pingouin. Pour dessiner ce trait, on calcule 2 points à l'aide des coordonnées `Coup.srcL` `Coup.srcC` pour le premier point et `Coup.dstL`, `Coup.dstC` pour le second en récupérant ces valeurs depuis l'historique du jeu.

Coup de l'historique :



Pingouins historique :

Nous avions pensé à faire des pingouins fantômes de chaque couleur pour les différents joueurs, mais nous nous sommes avisés de ne pas surcharger en couleur le plateau. Voici les versions en couleur :

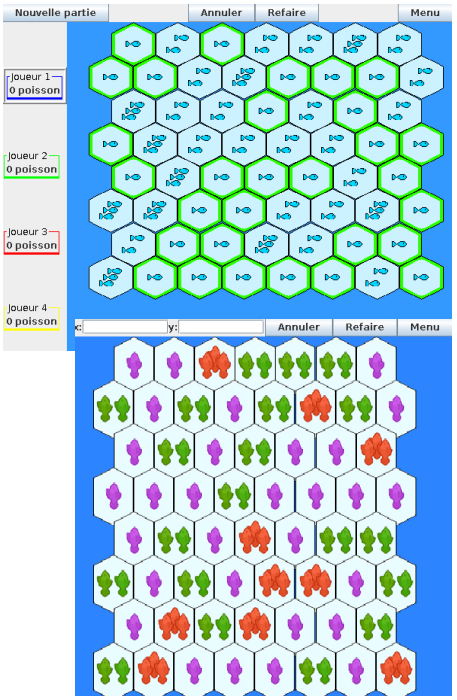
Affichage des pingouins éliminés

Les pingouins éliminés de la partie, ou plus simplement qui ne peuvent plus bouger, étaient à la base effacés du plateau, mais nous avons décidé après plusieurs retours de les afficher sous la forme de tuiles de pingouins K.O. Pour les afficher, on met à jour le `plateau[][]` de jeu avec une 8ème valeur que l'on assigne à une case au lieu de le retirer du jeu.

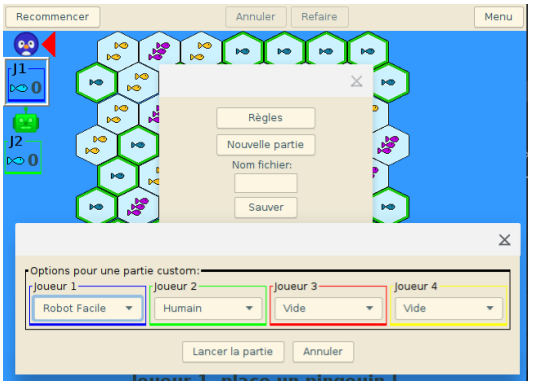
Pingouin K.O. :

Nous avions pensé à faire un pingouin K.O. de couleur différentes, pour chaque joueur, mais pour éviter de surcharger les couleurs nous avons décidé de garder le pingouins en nuance de gris. Même si cela rend les pingouins éliminés et les pingouins de l'historique similaires.

Interface Graphique



Après avoir parlé du plateau, intéressons-nous désormais à l'interface autour du jeu. Les limitations de Swing sont apparues au cours de ce projet, notamment au niveau style et taille. Java Swing est principalement fait pour être fonctionnel et ne permet pas de grande customisation.
Nous avons essayé quelques librairie qui changeait légèrement le thème, celles-ci n'étaient pas compatible avec toutes nos machines.
Pour la taille des éléments de l'interface, nous avons essayé différent type de layout, notamment le GridBagLayout permettant de customiser la taille des éléments avec une plus grand précision, cependant, le plateau de s'affichait uniquement dans le FlowLayout par défaut, nous ne sommes pas parvenu à résoudre celà.

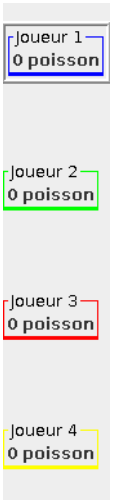


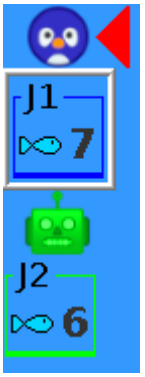
L'information disponible au joueur

Premièrement, parlons des informations primordiales à la partie. Nous nous sommes mis d'accord sur 3 informations essentielles pour le joueur: le score, l'indicateur de joueur courant, et un autre indicateur pour montrer que l'IA réfléchit.

Les score

Le panneau de score des joueurs se tient sur la gauche de la fenêtre. Au début, il affichait les 4 joueurs possibles même si la partie en avait moins, ce qui a été réglé et il affiche désormais uniquement le nombre de joueurs nécessaire.
La couleur n'étant pas sélectionnable, elles ont toujours le même ordre et, par exemple, il ne peut pas y avoir de rouge s'il n'y a que 2 joueurs.
Dans les retours reçus, il revenait que les couleurs de score et de pingouins n'étaient pas toujours identiques, et pas simple pour les daltoniens. La couleur des pingouins a changé plus tard pour mieux représenter les scores, mais nous avons quand même ajouté l'icône du pingouin (ou robot pour les IA) au-dessus des scores.



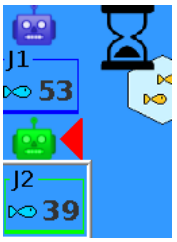


Le joueur courant

Au début, l'affichage du joueur courant se faisait avec un encadrement sur le tableau des scores. Les retours étaient tous d'accord que ce n'étais pas très visible ou suffisant. En plus du contour des pingouins sur le plateau, nous avons aussi mêlé l'ajout du pingouin/robot à une flèche indiquant le joueur courant.

L'IA qui réfléchit

En plus de l'afficheur du joueur courant, nous avons ajouté un petit sablier afin de montrer que le jeu ne freeze pas pendant que l'IA travaille. Nous avons aussi ajouté du texte au bas de l'écran pour montrer la même chose, ce que nous avons ensuite adapté pour afficher aussi lorsque que c'est à un joueur de jouer.



Les boutons de l'interface

Ensuite, parlons des boutons accessibles depuis le jeu. Nous avons essayé de garder un très peu nombre de boutons, donc ce qui n'était pas essentiel d'accès rapide a été mis dans un menu.

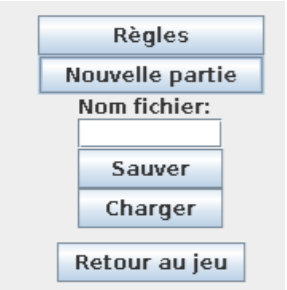
Annuler/Refaire

Ces boutons, centrés au milieu de la barre haute, permettent au joueurs d'utiliser l'historique. Ils sont présents depuis le pré-projet mais ont été améliorés suite à des retours: Ils sont grisés s'ils ne peuvent pas être utilisés (pas d'historique précédent ou suivant).

Recommencer

Le bouton "Recommencer" était auparavant le bouton "Nouvelle Partie", l'option de partie

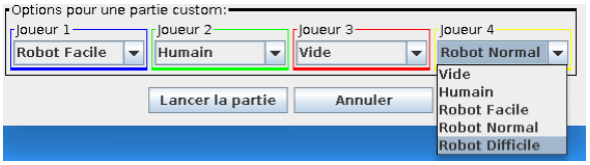
rapide et de partie custom ne composent qu'un seul panneau.
Nous avons premièrement eu l'idée d'un bouton "Partie Rapide" qui lançait juste une partie avec 1 humain et 1 IA facile, qui a ensuite évolué pour relancer une partie avec les mêmes paramètres mais un plateau différent.
Ce panneau de partie custom a été modifié et déplacé dans le menu.



Le menu

Le menu contient tous les boutons ne nécessitant pas d'accès immédiat. On y trouve notamment les règles, la capacité de lancer une partie personnalisée et de sauvegarder ou charger une partie.

Partie Personnalisée



Maintenant que l'option partie rapide a son propre bouton, le menu de partie personnalisée n'a désormais que l'option de lancer une partie avec les paramètres indiqués.
Afin de choisir les types de joueurs, un menu déroulant est disponible pour chaque emplacement où l'on peut choisir entre vide (pas de joueur), un humain (contrôlé à la souris) ou une IA avec sa difficulté.
Il faut noter que si les joueurs sont décalés vers la gauche s'il y a un joueur vide. Par exemple, si on crée la partie comme dans l'image avec un joueur 4 mais sans joueur 3, alors après la création, le joueur 4 deviendra le joueur 3.
Ce menu n'a pas beaucoup évolué hormis la séparation de partie rapide car il n'y a eu aucun problème dans les retours.

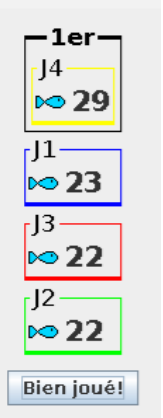
Règles

Le panneau des règles est assez simple et ne contient pas d'images. Dans nos playtest, les joueurs avaient tendance à intuiter les règles assez vite, soit directement, soit après avoir entendu le principe du jeu, donc nous n'avons pas vraiment eu de retour de ce côté là.

Sauver/Charger

L'option de sauvegarder et de charger une partie est aussi disponible depuis le pré-projet. L'option d'écrire un nom de fichier est présente mais non nécessaire (un nom par défaut est présent). Nous avons considéré utiliser un sélectionneur de fichier (comme dans les logiciel quand on utilise "enregistrer sous"), cependant les sondages effectués auprès des playtesters nous ont appris que les joueurs ne se voyait pas avoir besoin de sauvegarder et préfèreraient soit finir la partie directement, ou en recommencer une plus tard. Nous ne nous sommes donc pas penchés sur cette fonctionnalité vu qu'elle fonctionnait déjà.

Fin de partie



Lors de la fin de partie, toute l'interface reste utile. Pour relancer la même partie avec un plateau différent on a recommencer, et on peut annuler le dernier coup gagnant comme n'importe quel coup, cependant cliquer sur le plateau tant que le jeu est en position de fin de jeu ne fera que rouvrir la fenêtre de fin de partie.
Cette fenêtre affiche le score des joueurs dans l'ordre de 1er au 4e, les joueurs ayant le même score étant départagés par le nombre de tuiles qu'ils ont mangé (non affiché sur l'interface).

Conclusion

Nous sommes satisfait de l'aspect final du plateau, même si nous aurions aimé faire plus . Mais pour respecter les contraintes de temps nous avons décidé de nous focaliser sur les aspects que nous considérons plus importants d'abord. Nous pensons avoir atteint une bonne balance avec la lisibilité et les couleurs sans surcharger le plateau.

Au niveau de l'interface hors plateau de jeu, nous avons trouvé que Swing était très limitant si on veut créer une interface intéressante. On sent le côté daté de Swing et l'aspect purement fonctionnel, mais celà ne veut pas dire que l'on a pas essayé de changer celà. Certaines librairies afin de changer légèrement le thème de l'interface sont disponibles en ligne, cependant, celle-ci ne marche pas avec toutes nos machines, ni les machines de l'université. Nous nous sommes donc concentrés sur les fonctionnalités et avons fait de notre mieux pour rendre intéressant l'interface (surtout au niveau des scores).
Malgré celà, nous avons pu implémenter toutes les fonctionnalités désirées et aucun aspect de l'interface n'as pas été compris dans nos derniers playtests.