Equipe
Walid LARABI
Yon KOOIJMAN
Natan BEKELE
Gabriel DJEBBAR

Synthèse du projet de l'équipe T (DataDrinkers)

Fonctionnalités réalisés

Moteur du jeu

- L'ensemble des concepts (Caractères, bambous, irrigation, aménagements, météo..)
- Un ensemble d'objectifs : parcelle, panda, jardinier (couleurs et orientations)
- Une météo qui donne le nombre de tours, ainsi que un coup/meteo
- Un arbitre: donne la liste des coups possibles et valide les objectifs
- Partie graphique (non demandée) affichant parcelles, caractères, et irrigation
- Gestion du flux d'informations: logs console, statistiques.

Bots pseudo-intelligents

Tous les bot jouent selon la météo.
- Random bot: joue aléatoirement tout le jeu (parcelles, irrigations, aménagements..)
- Smart bot 1 : joue intelligemment les objectifs panda.
- Smart bot 2 : joue intelligemment les objectifs Parcelle.

Fonctionnalités non-réalisés

Bots pseudo-intelligents

- Smart bot 1 : Ne joue pas les objectifs Parcelle , Jardinier et les aménagements
- Smart bot 2 : Ne joue pas les aménagements et les objectifs jardinier.

Niveau de confiance sur la réalisation

Evolution en fonction de temps et statistiques

Release 3 : Création du `randomBot` qui pose les parcelles, et irrigations au hasard

Release 4 : L'introduction du `smartBot1` qui joue les objectifs panda intelligemment, le nouveau bot, gagne à environ 80%

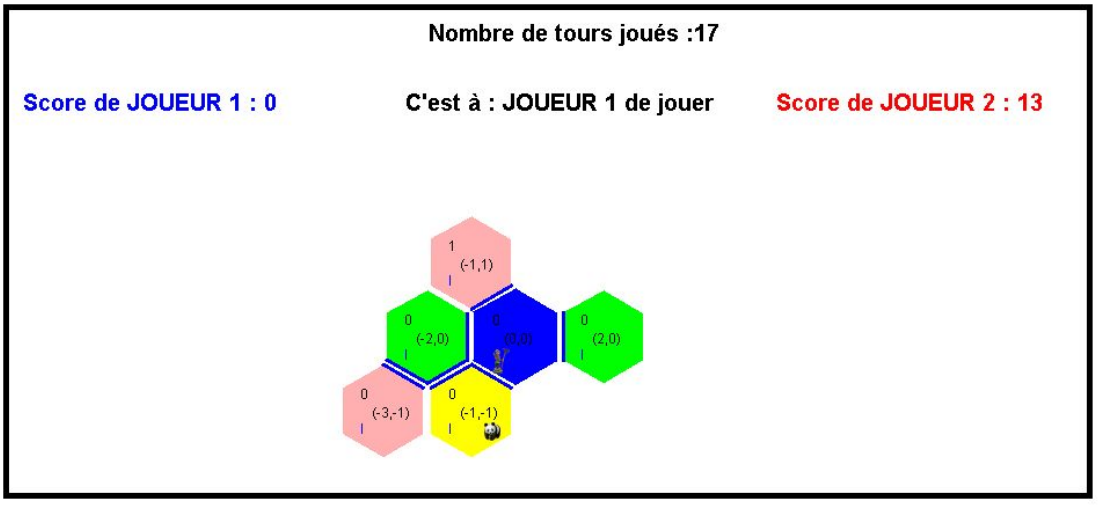
Release 6 :
Introduction du `smartbot2` qui jouera les parcelles et irrigations avec intelligence
Optimisation de l'intelligence sur la manière de jouer les objectifs panda sur le `smartbot1,` qui gagne 100% des parties contre le `randomBot et 95% le smartbot2`

Release 7 :

- Le `smartbot2` contre lui même fait 45% de victoires et 10% d'égalités , le premier qui joue n'est pas avantagé, et gagne à 90% contre le `randomBot` parce qu'il est plus difficile de gagner avec les objectifs parcelle que panda.

Partie graphique et loggers pour détecter les anomalies

Un niveau de logs est défini dans le jeu pour suivre les actions, ainsi pouvoir évaluer si les actions sont pertinents, les bon joueurs joue et qu'aucun problème apparaît.
L'équipe a développé une partie graphique du jeu, pour pouvoir visionner le déroulement du jeu. Mais également servir à faire une démonstration au client, et ainsi augmenter sa confiance.



Une capture d'écran du projet Takenoko, de l'équipe DataDrinkers. Dec 2017

Un suivi par couvrage avec des tests unitaires et qualité du code

Tests unitaires de la partie de Bambo
Coté Moteur du jeu
Fonction qui test si le pousse de bambou automatique lors de l'irrigation

Possibilities
Fonction qui test les parcelles dans le plateau ou on peut poser un bambo

Validation:
Couvrir l'ensemble du processus de validation concernant les objectifs panda (qui mange des bambo) par la vérification la disponibilité des bambo pour que l'objectif se valide, que le score augmente, et que les bambo se retirent de la fiche.

Tests unitaires de la partie de Amenagements

Coté moteur du jeu

Vérifier qu'on respecte les règles du jeu en proposant le bon nombre de parcelles aménagés ainsi que lors de prise d'aménagement et poser un aménagement, le processus est complet et valide

Coté possibilités

Fonction qui regarde l'ensemble des parcelles, pour proposer ceux ou on peut poser un aménagement

Tests unitaires de la partie d'Irrigation

Coté moteur du jeu

Vérifier que le processus de poser irrigation se passe bien, et quand une irrigation est posée, que les deux parcelles qui entourent le segment, s'irrigue.

Coté possibilités

Pouvoir vérifier qu'on pose les irrigations entre deux parcelle et seulement dans les endroits autorisés

Tests unitaires de la partie mouvement de panda et jardinier

Coté moteur du jeu

Fonctions qui test si le panda se déplace et mange le bon nombre de bambo sur les parcelles, ça test également si le bambou a été retiré de la parcelle et ajouté sur la fiche du joueur concerné

Coté possibilités

Fonction qui vérifie qu'on n'a pas le droit de déplacer le panda sur l'aménagement nopanda, qu'il se déplace sur une ligne continue de parcelles
Une autre fonction qui vérifie qu'on peut déplacer le jardinier seulement sur une ligne continue de parcelles

Parties du code/tests sont de bonne ou de mauvaise qualité

Bonne qualité

Les tests concernant le moteur du jeu sont de bonne qualité, et teste tout les fonctionnalités et leur cas limite

Mauvaise qualité

Les tests concernant les stratégies de bots ne sont pas assez complets.

Notre feedback sur le projet

Organisation

+ Une attribution de rôles et de tâche égale
+ Une bonne entente du groupe, entraide et communication
- Des périodes d'irrégularités et manque d'implication

Découpage

+ Un découpage réaliste, et faisable
+ Livraison continue (agile), et intégration des nouvelles fonctionnalités par release
- Les bots n'ont pas avancés au même niveau que le système du jeu

Code et tests

+ Code maintenable / classes logiques.
+ Le moteur de jeu est bien couvert de tests, pour respecter les règles du jeu
+ Code et tests commenté et documenté
- Architecture aurait pu être mieux
- Certaines méthodes sont longues (donc une possibilité d'incompréhension)
- Manque de tests sur les bots pseudo-intelligents

Livraisons et démos

+ Certains livraisons (notamment la 6, et 7 ont prit du retard)
- Dans la démo, l'oublie des annexes sur le projet pour une clarté (doc/graphiques..)

Retrospective

Quels sont les pratiques que vous conserverez

- La bonne communication, l'entente et création d'atmosphère amicale
- L'intégration d'un code testé, documenté pour éviter les bugs et anomalies

Quels sont les pratiques que vous améliorerez

- Contrôle régulier de la qualité avant les release, tags et respect des deadlines
- Améliorer l'architecture, et le concept du single responsability et maintenabilité
- Utilisation des méthodes agiles pour faciliter la gestion du projet et avancement
- Mise en place de réunions régulièrement, pour faire du brainstorming, et planifier

Quels sont les pratiques que vous supprimez

- Création des méthodes complexes et longues, et négligence de qualité de code
- Le non-respect des deadlines pour livrer du code interne (mais aussi externe)