

Rapport de Projet : Dice Forge

Sujet : Implémentation d'un Bot Dans le jeu Dice Forge.



Encadrée par : Philippe Collet et Sébastien Mosser

Réalisée par : Mekouar Youssef
Florian Ainadou
Bastien Jard
Yannis Falco

Année 2018-2019

I-Synthèse du projet :

Les fonctionnalités réalisées :

Moteur de jeu :

Ensemble des concepts :

- Création du temple et de toutes les faces que contient les bassins du temple.
- Création du Plateau des Iles qui contient toutes les îles du jeu.
- Un arbitre : Qui nous permet déterminer le Bot vainqueur de la partie.
- Gestion d'actions :
 - Effectuer une action dans le temple
 - Effectuer une action dans le plateau des îles.
- Création des cartes ainsi que leurs effets (Effet Immédiat , Effet Permanent) :

1. La Méduse.	7. Les Herbes Folles.
2. La Pince.	8. Les Sabot D'argent
3. Le coffre du forgeron.	9. Les Satyres.
4. Le Minotaure .	10. L'Hydre.
5. L'énigme.	11. L'ancien.
6. Le passeur .	12. Le Marteau du forgeron
	13. Les Ailes de la gardienne.

Bots-Intelligent :

- Création d'un **Bot** super-intelligent **Émeline** .
- Création d'un **Bot** qui effectue **des actions** de manière aléatoire.

Les fonctionnalités qui n'ont pas été réalisées :

- Création du **jardin dans le temple**.
 - Création des faces spéciales dans le jardin du temple.
 - Création des cartes qui ont un rapport avec le jardin du temple :
 1. Le casque d'invisibilité.
 2. Le miroir Abyssal.
- Création des cartes promotionnelles.
- Le fait qu'un joueur ai le droit de rejouer un nouveau tour s'il se fait exclure d'une île.

Niveau de confiance sur la réalisation :

Évolution de Notre Bot :

- Notre bot aléatoire Josseline effectue des choix aléatoire :
 - Le choix d'actions principal choisir de partir au temple ou aux îles.
 - Le choix d'acheter des faces dans le temple.
 - Le choix d'accéder à quelle île .

- Le choix d'acheter une cartes parmi autres.
- Création du bot Intelligent Émeline qui gagne jusqu'à 60% du temps contre le bot aléatoire et 98% contre lui-même.
- Correction d'Émeline qui gagne jusqu'à 50% du temps contre elle-même.
- Émeline notre Bot intelligent gagne jusqu'à 90% du temps contre le bot aléatoire grâce à une stratégie.

II-Quelles sont les parties du code qui sont bonnes ou mauvaise ?

Commençons par la partit obscur de notre code et laissant le meilleur pour la fin.



Mauvaise qualité : (Voir Annexe Page 6)

- Nous avons effectué une mauvaise première soutenance du à notre irrégularité de notre travail et surtout dû à une mauvaise organisation durant la soutenance.
- Dans la classe Joueur pour implémenté l'effet de la carte « Le marteau du forgeron » qui a un impact sur le lancer de dé n'es pas bien implémenté (concept objet).
- Les tests pour les bot ne sont pas optimales ce qui est un mauvais points pour le sureté et la maintenabilité de notre projet.



Bonne qualité : (Voir Annexe page 6)

- **Les tests concernant les éléments du jeu :**
 - Tout les tests sont bien détaillée pour chaque partit du jeu.
 - Les tests du bassins.
 - Les tests des faces.
 - Les tests des lles.
 - Les tests des effets des cartes et des cartes.
- **La partie objet :**
 - Tout les classes sont particulièrement tous bien implémenté au niveau de la conception objet à part pour quelques exceptions prêt .

Notre évaluations sur ce projet :

Notre Organisation :	
• Une bonne entente dans le groupe.	😊
• Une bonne communication.	😐
• Un travail bien partagé (Bonne division des taches).	😊
• Une bonne régularité de travail et la bonne implication du groupe.	😐
Notre Code et Nos Tests:	
• Une régularité continue des tests en fonction de l'implémentation.	😐
• Le code est maintenable	😊
• Architecture des classes/Concept objet	😊
• Test sur le système de Jeu	😊
• Test sur les Bot du Jeu	😐
• Une bonne documentation complète.	😊
Notre slicing et Utilisation du GitHub :	
• Un bon découpage des taches du projet.	😊
• Une bonne utilisation du support de travail GITHUB .	😐
• Respects des Livraisons	😐

○ Quels sont les pratiques que vous conserverez :

- Un code est bien testé et bien documenté nous permet de justifier ainsi conclure le résultats attendu.
- Une bonne entente dans un groupe .

○ Quels sont les pratiques que vous améliorez :

- Une régularité des Tests pour évoluer dans un code fiable.
- Un contrôle régulier des taches sur GitHub.
- Une meilleure organisation dans les rendus de Livraison.
- Une amélioration de l'architecture et de la structure des objets.

Quels sont les pratiques que vous supprimez :

- Le non-respects des Dead Lines des rendu de projet.
- Un mauvaise organisation de temps.



Bonne conception
et de bon tests

Annexe :



Conception
moyenne et tests
insuffisants

