

Devoirs Fortales

Léo
Simon

Partie Théorique :

Question 1 :

Fortales est un jeu de cartes et de stratégie au tour par tour. C'est aussi un jeu qui repose beaucoup sur sa narration, qui évolue en fonction des cartes jouées et des décisions prises au fil de l'aventure.

Question 2 :

Après une heure de jeu, voici les mécaniques de jeu que j'ai rencontrées :

- Interaction avec les cartes lieux à l'aide des cartes consommables, compétences ou des cartes spéciales obtenues au fil du jeu.
- Mise en surbrillance pour indiquer les effets d'une carte si elle est utilisée sur une carte lieu ou si une carte peut être jouée sur une carte lieu.
- Cliquer sur une carte lieu pour obtenir une indication ou de la narration via un dialogue.
- Cycle des cartes lieux.
- Groupe de recherche.
- Consulter les paquets de cartes pour voir leur contenu.
- Système d'aide pour obtenir des indices et des indications.
- Affrontement :
 - Faire fuir un ennemi en utilisant des cartes consommables pour diminuer le moral.
 - Combat au tour par tour.
 - Utiliser des renforts.
 - Renforcer ses dégâts grâce aux cartes compétences.
 - Utiliser une compétence pour interagir avec les gardes (voler, étourdir, etc.).
- Perdre des PV : pas uniquement en combat.

- Se reposer.

Question 3 :

Le jeu a de multiples façons de proposer sa narration et de faire avancer l’histoire :

- Les cartes personnages en bas au milieu de l'écran parlent entre elles durant le gameplay via des bulles comme dans des BD.
- Lors d’une interaction avec une carte lieu en utilisant une autre carte : voler le marché --> gardes.
- Cliquer sur un lieu peut déclencher un dialogue qui fournit des indications et fait avancer l’histoire.
- Le système d'aide déclenche un dialogue qui (...) (phrase incomplète).
- Certaines cartes ont une interaction spéciale avec des cartes lieux qui peuvent permettre de prendre différents chemins dans l'histoire.
- Après avoir accompli une mission, l’histoire progresse en fonction de la mission accomplie.

Question 4 :

Les différents types de cartes présentes dans le jeu :

- Lieux : Elles représentent les différents lieux que l’on peut parcourir durant une mission. Elles ont des effets différents en fonction des cartes utilisées pour interagir avec elles.
- Compétences : Les compétences des personnages, elles ont des effets différents en fonction de quelle carte elles sont jouées.
- Consommables :
 - Or
 - Gloire
 - Nourriture
 - Infamie
 - Cartes spéciales comme les aides, les cartes de quête, les cartes d’outils
- Carte Repos, qui permet de récupérer des cartes compétences, parfois contre un coût.
- Cartes Créatures, qui sont les monstres qui peuvent être appelés durant un combat.
- Les cartes mission, qui servent à choisir sa mission dans le HUB des missions.

Question 5 :

Lors d'une rencontre avec des adversaires, le combat ne se lance pas immédiatement. Dans un premier temps, le joueur peut utiliser ses compétences et ses consommables pour faire fuir certains adversaires. Le groupe de monstres possède une valeur de moral et chaque monstre possède un niveau. Si le joueur fait fuir un adversaire, son niveau est déduit du moral du groupe, et si le moral tombe à 0, alors le combat est évité.

Mais si le joueur utilise certaines compétences, alors le combat est déclenché avec les monstres restants.

Le joueur peut donc éviter totalement un combat ou n'affronter qu'un certain nombre d'adversaires grâce au système de moral.

Question 6 :

Le HUB des missions est constitué de cartes représentant les possibilités pour le monde que seul le protagoniste peut voir. Certaines d'entre elles sont des "Calamités" et disposent d'un chiffre sur le dessus qui représente combien de tours le joueur dispose pour les faire.

Ignorer une calamité a un impact sur l'histoire et le déroulement du jeu, comme la perte d'un personnage. L'accomplissement des missions et des calamités a aussi un impact sur le déroulement des missions suivantes.

Question 7 :

Durant ma phase de test, le jeu m'a beaucoup plu ! Ses mécaniques sont intéressantes et originales, sa narration est prenante et m'a donné envie de continuer à jouer. Je vais probablement finir le jeu sur mon temps libre !

Partie Pratique :

J'ai décidé de mettre en place la mécanique de drag and drop de carte sur des lieux de Foretale en y ajoutant les effets de déplacement des cartes lorsqu'elles sont survolées par la souris.

Dans un premier temps, j'ai créé trois Widgets : un pour le fond, un pour les lieux et un pour les cartes en main.

Les cartes :

J'initie les cartes en leur donnant des variables d'angles, de position, de couleur et de texte qui sont toutes instance-editable et exposed-on-spawn pour pouvoir les personnaliser pour chaque carte.

Pour faire en sorte que les images se déplacent correctement lorsqu'elles sont survolées par la souris, j'ai utilisé l'événement tick pour agir sur leur Transform et Angle (j'ai aussi profité de ce moment pour garder une variable de scale pour plus tard).

Ensuite, pour m'assurer que malgré l'événement tick, les calculs ne se feraient pas tout le temps, j'ai utilisé le node Gate.

Cela fonctionne donc de la manière suivante :

Quand la souris survole une carte, je définis un float « Target_Position_Y » à la valeur que je veux pour faire monter la carte (-100 dans mon cas). Je définis aussi un deuxième float « Target_Rotation_Angle » pour redresser la carte (donc 0). Puis, j'appelle un custom event nommé « StartMove » qui ouvre la gate et attribue ces deux floats au transform en Y et à l'angle grâce à un set Render Transform. Une fois que les valeurs sont appliquées, je vérifie que les valeurs sont correctes, puis je ferme la Gate en appelant un custom event « End Move ». Pour rendre le mouvement fluide, j'utilise un node d'interpolation pour chaque paramètre que je change. Les nodes d'interpolation ont besoin du delta time (que je tire de l'événement tick) et d'une vitesse que j'ai créée : un float « Speed » (de valeur 20).

Pour ramener les cartes dans leur position : au moment où la souris quitte la carte, je réinitialise les valeurs de « Target_Rotation_Angle » et de « Target_Position_Y » aux valeurs d'origine de la carte. Puis, je réappelle l'événement Start Move pour ouvrir la Gate et appliquer de nouveau les valeurs à la carte.

Pour le drag and drop des cartes :

J'utilise 3 Overrides de fonction pour le drag and drop des cartes :

- On Mouse Button Down :

Lorsque l'on clique sur une carte, j'enregistre l'offset de ma carte, puis je détecte un Drag si une certaine touche est pressée (dans mon cas, clic gauche).

- On Drag detected :

Lorsque le drag est détecté : je crée un nouveau Widget (la carte qui va suivre la souris). En référence, je lui mets self car je veux créer un widget carte. Je lui attribue une nouvelle scale et, grâce à une variable linear color « Start Color » (qui conserve la couleur de la carte sur laquelle on a cliqué vu que je l'ai mise en instance-editable et Expose-on-Spawn pour pouvoir choisir la couleur de chaque carte individuellement), je lui attribue la bonne couleur.

Puis, je définis 2 variables : une référence de la carte sur laquelle on a cliqué et une référence de la carte que l'on est en train de drag and drop. Je crée ensuite une drag and drop operation avec

comme visuel la référence de la carte que l'on drag. Je définis ensuite l'opacité de la carte sur laquelle on a cliqué à 0 pour la rendre invisible mais pour ne pas la détruire (dans le cas où on lâcherait la carte au mauvais endroit). Enfin, je lance l'événement « Start Move » avec comme target la carte dragged pour lui attribuer les bonnes valeurs d'angle de scale etc.

- On drop :

Je cast au BP de drag and drop pour obtenir la référence de la carte sur laquelle on a cliqué et je lui remets son opacité à 1.

Dans le cas où la carte est droppée au mauvais endroit : je redonne à la carte sur laquelle on a cliqué son opacité avec un « On drag Cancelled » dans l'événement graph.

Les Locations :

Dans le pre-construct, je leur attribue une couleur aléatoire dans un tableau de linear color.

- On drop :

Je leur attribue une nouvelle couleur aléatoire dans le tableau de linear color et je m'assure que la couleur est différente de celle qu'ils ont déjà. Puis, grâce à un cast, je récupère le texte des cartes pour l'attribuer au texte de la location.

Pourquoi cette mécanique :

Je voulais comprendre les bases de n'importe quel jeu de carte, les petits mouvements de carte, les interactions très simples entre widgets et m'habituer à circuler entre les widgets et les faire marcher ensemble.