Projet de programmation: Slice & Dice

Cahier des charges

Le but de ce projet est de développer un jeu de style Slice & Dice en Java, en nous aidant des bibliothèques awt et Swing.

Charges minimales:

- Le projet doit pouvoir se lancer depuis un terminal, à l'aide d'une commande spécifique. Le lancement depuis un-IDE est proscrit.

- **Le menu d'accueil** doit contenir des boutons permettant la navigation entre les différentes fenêtres du jeu. De plus, une fois la partie terminée, le joueur doit avoir le choix entre « Recommencer », « Revenir à l'écran d'accueil », « Quitter le jeu », cette dernière option devant mettre fin à l'exécution du programme. Au démarrage, la page d'accueil doit contenir le logo de notre jeu, et trois boutons **« FACILE », « MOYEN » et « DIFFICILE »**, chacun de ces boutons doit démarrer une partie correspondant au niveau indiqué par ce-dernier. Toujours sur la page d'accueil, un autre bouton (plus petit et sur le côté) doit être présent. Ce dernier doit permettre au joueur, une fois que ce dernier a cliqué dessus, de se renseigner sur les différents personnages : quels dés leur correspondent, en quoi ils peuvent être améliorés etc.

- **Chaque personnage (héros/ennemi) doit posséder un dé unique.** Entendre par là que le **pool d'attaque de chaque personnage doit être différent** (le pool d'attaque et non les attaques, un healer et un voleur peuvent tous les deux posséder une attaque faisant 2 de dégâts, mais leur pool d'attaque est différent puisque le healer possède des capacités de guérison, le voleur lui peut posséder des attaques faisant 3 de dégâts). Pour ce faire, on créera un constructeur d'objet « De » qui prendra en paramètres 6 objets de type « Capacites ». Les dés qui contiennent moins de 6 capacités auront le même constructeur que ceux qui en possède 6, simplement, les capacités seront égales à « null » dans leur constructeur.

- Le dé d'un héros peut contenir **jusqu'à 6 capacités**. Les équipements seront affichés, pendant la partie, aux côtés du dé MAIS ils ne font pas partie du dé. **Les équipements** devront être stockés d'une autre manière (par exemple avec un attribut de type tableau d'équipements de taille 2 dans la classe Heros).

- **Le tour du lancer de dé doit être séparé du tour des attaques.** Autrement dit, le joueur doit avoir droit à un premier tour, dit de préparation durant lequel il peut lancer les dés autant de fois qu'il le veut (dans la limite imposée), et durant lequel il peut attribuer le dé correspondant au héros, mais également retirer le dé attribué à un héros durant le même lancer de dé. Le deuxième tour doit quant à lui permettre l'utilisation des capacités acquises durant le tour précédent, et doit permettre au joueur de choisir sur quelle entité utiliser la capacité. Il faudra bien sûr penser à vérifier qu'une attaque ne soit pas utilisée contre un allié, et qu'une guérison ne soit pas utilisée sur un ennemi. **Il faut donc penser à toutes les données que le jeu doit retenir pour prévoir les éventuels retours en arrière que le joueur peut vouloir effectuer. Ces données peuvent être les dés qui ont été choisis, les héros qui ont été soignées, les ennemis qui ont été attaqués, les attaques générales qui ont été utilisées, etc.**

- Implémenter une **transition entre les niveaux.** Une piste serait d'utiliser un gif qui vienne cacher l'écran avec un ou des éléments et qui voit ces mêmes éléments repartir et dévoiler le nouveau niveau. Exemple : Clash of Clans, lorsque l'on cherche un village à attaquer, le jeu utilise des nuages pour cacher l'écran et charger un nouveau village.

- **Les ennemis sont toujours les premiers à lancer leurs dés, mais attaquent toujours en dernier.** Cela permet au joueur d'anticiper. Ainsi, il est impliqué que le joueur doit pouvoir voir quelles capacités ont été acquises par les ennemis, et également quels héros les ennemis vont attaquer. De même, les points de vie du héros ciblé qui resteraient après que l'ennemi ait attaqué doivent être affichés de sorte à ce que le joueur puisse anticiper. Le jeu de base affiche cette information à l'aide d'un changement de couleur (jaune) des coeurs représentant les points de vie du héros.

- À chaque fin de niveau, le joueur doit pouvoir **améliorer un de ses héros** (le joueur ne peut pas choisir directement quel héros améliorer, la ou les propositions seront aléatoires). De même, le joueur doit pouvoir obtenir de l'équipement/bonus à chaque fin de niveau qu'il peut automatiquement équiper à l'un de ses héros. Cependant, le joueur doit également pouvoir refuser d'équiper l'objet/bonus obtenu, auquel cas cet objet/bonus doit pouvoir être stocké dans un inventaire pour pouvoir être réutilisé plus tard. Cela implique donc que, contrairement à l'amélioration d'un héros, l'équipement des héros peut se faire à tout moment durant la partie.

- Les équipements ne doivent pas tous être des bonus simples d'attaque ou de défense, certains doivent avoir des particularités spéciales, comme permettre l'attribution de trois attaques de niveau 3 sur le dé d'un personnage par exemple. Cela implique donc de devoir modifier le dé d'un personnage, et de plus, prévoir le retour du dé à la normale si l'équipement est retiré.

- Tout comme le jeu de base, le joueur possède une attaque permanente nommée BURST, qu'il peut utiliser s'il possède assez de mana. Deux autres emplacements doivent être présents, ces-derniers correspondants aux attaques permanentes qui dépendent des personnages faisant parti de l'équipe. Exemple : si le joueur décide prendre un guérisseur et un mage de glace dans son équipe, alors en plus de BURST, il aura accès aux capacités permanentes « HEAL » et « FROZE » (ce sont des exemple, à nous d'être créatifs). Il faut donc que le programme vérifie à chaque début de partie, ainsi qu'à chaque fois qu'un héros est éliminé et enfin à chaque amélioration si le joueur n'a plus accès à une attaque générale qui lui était conférée par un héros qui n'est plus sur le terrain ou au contraire, si l'arrivée d'un nouvel héros (après amélioration ou si ressuscité) lui confère une nouvelle attaque générale.

- Cliquer sur un dé qui a été lancé doit permettre l'attribution automatique au personnage correspondant. Cliquer sur le personnage doit afficher le pool de capacités (sous la forme d'un dé déplié ?) disponible pour ce personnage. Cliquer sur le dé une fois qu'il a été attribué doit afficher la description de la capacité obtenue. Cliquer sur un des ennemis doit afficher quels héros vont se faire attaquer par ce-dernier (comme le jeu de base, on peut soit l'indiquer par des lignes, ou alors simplement afficher les sprites des héros concernés sous l'étiquette de l'ennemi). Cliquer sur le dé de l'ennemi doit indiquer une description de l'attaque.

- Naturellement, un bouton « Relancer » est nécessaire pour permettre au joueur de relancer les dés non choisis.

- Si tous les dés ont été choisis, alors le bouton « Relancer » devient le bouton « Undo » (ou « Annuler ») qui doit permettre au joueur de revenir en arrière jusqu'à la première action effectuée (si il y a au moins 1 reroll restant, alors le retour en arrière doit permettre au joueur de pouvoir revenir jusqu'au moment où il pouvait relancer les dés).

- **Les ennemis sont générés aléatoirement.**

- Le jeu doit posséder **plusieurs niveaux de difficultés**. Les paramètres sur lesquels s'appuyer pour déterminer qu'une partie est plus difficile qu'une autre peuvent être choisis librement. Faire attention à l'équilibrage.

Charges optionnelles/supplémentaires :

- Un affichage des stats à la fin de la partie.

- **Une boutique** qui permette au joueur d'acheter des équipements/bonus pendant la partie, et ce, à tout moment. Implique la création d'une monnaie qui peut s'incrémenter en tuant les ennemis par exemple.

- **Création de son propre héros** : Permettre au joueur de créer son propre héros. Cela implique de nombreuses **vérifications à faire** :
- Tout d'abord, le joueur peut créer sa « propre classe » de héros. En réalité, le code devra contenir une classe qui étend la classe Heros mais dont le nom ne fait allusion à aucune classe supplémentaire. On peut par exemple appeler cette classe de héros la classe Custom. Ainsi, une instance de Custom est un héros crée par le joueur. Ce sera au joueur de donner les noms correspondants à chacun des niveaux d'améliorations de Custom.

- En ce qui concerne les capacités de Custom : le joueur aura accès à toutes les capacités qu'il aura débloquées. Cependant, il ne pourra pas attribuer des capacités librement. En effet, il faudra mettre en place un système de points de compétences qui varie en fonction du niveau du héros qui est en train d'être personnalisé. Le système fonctionnera ainsi : pour la forme de Custom de base, c'est-à-dire lorsqu'il s'agit de Custom au niveau 1, le joueur disposera de 8 points de compétence , au niveau 2, le joueur disposera de 13 points de compétence, et enfin au niveau 3, il disposera de 18 points de compétence. Ainsi, à chaque fois que le joueur attribue une capacité à son héros, le programme doit vérifier quel est le niveau de la capacité attribuée et décrémenter le nombre de points de compétences par ce niveau. Exemple : je possède 8 points de compétences, j'attribue à mon héros la capacité Bouclier de niveau 2, je n'ai plus que 6 points de compétences.

- Apparence de Custom : pas encore de solution précise trouvée. Quelques pistes cependant :
    1. Avoir une banque d'assets sur lesquels le joueur peut cliquer pour construire petit) petit son héros. Par exemple, 4 sections : Couvre-Chef/Cheveux ; Yeux ; Bouche ; Visage/CouleurPeau. Chacune de ces sections contient un bon nombre d'assets déjà faits et donc la position sur un template est déjà défini. Ainsi, le joueur doit simplement cliquer pour voir apparaître sur un template l'asset à sa positon définie. (Exemple : l'éditeur de Mii sur la Wii mais sans aller jusqu'à permettre le mouvements des assets)
    2. Trouver un moyen de permettre au joueur de pouvoir dessiner son propre héros lui-même soit directement dans le jeu, soit via un site (comme Piskel). Si deuxième option retenue, alors il faut trouver un moyen de sauvegarder l'image ainsi créée pour pouvoir l'attribuer au héros nouvellement crée ainsi qu'à ses améliorations.
    3. Avoir une banque cette fois non pas d'assets, mais simplement de sprites de héros divers déjà faits. Le joueur devra alors simplement choisir quels sprites il veut pour représenter son héros ainsi que ses améliorations.
- Type de Custom : Slice&Dice est un jeu dans lequel chaque héros possède une couleur qui correspond à son type. Ainsi, le joueur devra également pouvoir choisir le type de son héros Custom. Il sera bien sûr limité aux couleurs de base présentes dans le jeu. De ce fait, le joueur ne pourra utiliser son héros Custom uniquement en remplaçant le héros de base qui possède la même couleur que Custom. Exemple : si Guérisseur est de couleur rouge, et que mon héros Custom est de couleur rouge, alors je ne peux utiliser mon héros Custom seulement en remplaçant le Guérisseur par mon héros Custom. Cela a son importance notamment pour les héros de la même couleur que le Guérisseur et le Mage : ces deux héros, lorsqu'ils sont présents sur le terrain , permettent l'utilisation de sorts qui leurs sont propres. Or, si on permet la présence de deux héros de même couleur (par exemple le Mage et mon Custom), cela veut dire que le joueur aura possiblement accès à 4 sorts au total. Cela ne devrait pas être possible, du moins pas dans la version que l'on programme en ce moment. (À voir si l'on permet au joueur de joueur avec plus de 5 héros à la fois, auquel cas il faudrait prévoir plus de slots pour les sorts).

- Sorts pour Custom : ainsi, pour que Custom puisse posséder des sorts, il faut qu'il soit de la même couleur que le prêtre ou que le Mage, du moins, dans la version que l'on programme pour le moment. Pour le moment, si l'on veut garder une logique dans le jeu, les sorts seront attribués automatiquement en fonction de la couleur et du niveau de Custom. Donc, si Custom est de niveau 1, et qu'il possède la même couleur que le Guérisseur, alors son sort au niveau 1 sera le même que le Guérisseur au niveau 1 (et ainsi de suite (pareil si il est de la même couleur que le Mage)).

- L'interface de création : Mis à part l'attribution de noms, tout le processus de création doit pouvoir se faire, dans l'idéal, entièrement via l'interface graphique.

- Implémenter une vidéo Tutoriel « Comment jouer » qui serait accessible à tout moment à partir du menu.

- Ajout d'effets sonores ainsi que d'une ou plusieurs musiques de fond.

- Implémenter une cinématique au début du jeu qui raconte brièvement le lore du jeu.

- Créer une boutique en dehors des parties, qui permette au joueur d'acheter des capacités qu'il pourrait attribuer à son héros Custom. Cela implique donc la création de potentielles nouvelles capacités. Cette boutique de vrai se présenter sous la forme d'un onglet dans lequel les images des capacités sont affichées avec leurs prix juste en dessous. En passant la souris sur chaque image, la description qui correspond à cette capacité doit s'afficher. Une fois que le joueur aahète une capacité, cette dernière doit rester dans la botutique mais ne doit plus être achetable. Donc il faudra par exemple griser l'image, ou alors mettre le message « Déjà acheté » à la place du prix, en bloquant l'achat de cette capacité.

- Les évènements aléatoires (pas encore sûr) : des évènements qui ont faible probabilité d'arriver au début de chaque niveau (environ 1 chance sur 24). Ces évènements peuvent être l'ajout de mana gratuitement au début du premier tour, des dégâts infligés aux ennemis ou aux héros, ou les deux au début du premier tour, ou alors l'enlèvement de tous le mana stocké par le joueur jusque là etc etc.

- QTE : les qte sont des courtes séquences d'actions que le joueur doit réaliser pendant la partie pour atteindre un objectif. Dans notre jeu, ces qte peuvent servir à enlever l' » étourdissement » d'un héros, ou alors permettre au joueur de lancer des sorts, etc. Pour ce qui est de la forme de ces qte, un exemple serait : cliquer le plus de fois possibles en deux secondes, si le nombre de clics dépasse 8, alors le qte est réussi, sinon non. Un autre exemple serait de cliquer sur des endroits précis dans un temps imparti, si le joueur clique sur le mauvais endroit ou qu'il n'a pas eu toutes les cibles dans le temps imparti, alors le qte n'est pas réussi. D'autres idées de qte peuvent être ajoutées par la suite. Enfin, ces qte, lorsqu'il y a utilisation de la souris, devront se réaliser dans une fenêtre interne délimitée (en gros un rectangle dans lequel les clics sont comptabilisés, cela rendra le tout plus agréable et compréhensible pour le joueur et pour nous).

- Ajout d'une barre de « santé mentale » : cette jauge devrait influer sur le nombre de dégâts infligés et subis par les héros. Certaines capacités pourraient la booster et d'autres capacités provenant des ennemis pourraient la diminuer.

- Ajout de nouvelles capacités :

- **BouclierSoin :** capacité qui donne du shield à un héros tout en le soignant. Le nombre de shield ainsi que le nombre de points de vie soignés est le même et dépend du niveau de la capacité. Coûte 20g dans la boutique.

- **Poison :** capacité qui inflige le statut poison. Le niveau de la capacité correspond au nombre de tours durant lequel l'entité concernée va subir le poison. Le poison quant à lui, inflige toujours deux points de vie par tour. Coûte 15g dans la boutique.

- **HacheSoin :** inflige des dégâts tout en se soignant soi-même. Le nombre de dégâts dépend du niveau de la capacité, tandis que les points de vie restaurés sont toujours égaux à 5. Coûte 25g dans la boutique.
- **SoinsAccrus :** soigne tous les points de vie d'un héros choisi. Elle n'a donc qu'un seul niveau : le niveau 1. Coûte 50g dans la boutique.
- **MortSubite :** élimine un ennemi (excepté ceux de type « boss ») en un coup.
- Donc cette capacité ne possède qu'un seul niveau. Coûte 70g dans la boutique.

Liste des sorts à implémenter :

- **Burst :** Toujours disponible ; Consomme 2 mana ; Inflige 3 dégâts
- **Flare :** Disponible si le Mage est en vie ; Consomme 4 mana ; Inflige 5 dégâts
- **Renew :** Disponible si le Guérisseur est en vie ; Consomme 2 mana ; soigne 4 coeurs
- **Balance :** Disponible si le Prêtre est en vie : Consomme 4 mana ; Soigne 2 points de vie aux alliés et inflige 2 dégâts à tous les ennemis
- **Drop :** Disponible si l'Arcaniste est en vie ; Consomme 3 mana ; Inflige 4 dégâts à l'ennemi qui possède le plus de points de vie
- **Scald :** Disponible si le Sorcier est en vie ; Consomme 3 mana ; Inflige 2 dégâts à tous les ennemis qui ont déjà subi des dégâts
- **Infuse :** Disponible si l'Apothicaire est en vie ; Consomme 3 mana ; Soigne 6 points de vie (jusqu'aux points de vie maximum si ces derniers sont atteints) au héros avec le moins de points de vie
- **Liquor** : Disponible si Miracleur est en vie ; Soigne 10 points de vie à tous les alliés et donne 1 mana ; Consomme 4 mana
- **Revive :** Disponible si Starseer est en vie ; Ranime l'allié mort le plus en haut de la liste et régénère tous ses points de vie ; Consomme 4 mana
- **Bind :** Disponible si Démoniste est en vie ; inflige 8 dégâts à tous les ennemis mais fait perdre 5 points de vie au héros qui possède le plus de vie ; Consomme 3 mana - Blaze : Disponible si Shaman est en vie ; Inflige 13 dégâts ; Consomme 6 mana

ANNEXE:

Maquette Custom