

Cahier des charges fonctionnel

Projet Génie Logiciel

Acteurs

Nom	Qualité / rôle
Raphaël Montet	Chef de Projet
Loïc Menguy	Testeur
Malvina Gontard	Architecte / Designer
Ludivine Nouveau	Testeur / Marketing
Thibault de Villèle	Architecte / Designer
Emmanuel Ferreira	Client

Validations

Nom	Date	Validation O/N	Commentaires

Historique des modifications

Version n°	Date	Etat	Description de la modification

1.Cadrage du projet

The Game

Conception d'un jeu vidéo paramétrable par le client.

Résumé

Dans ce jeu, le joueur contrôle un héros dont la quête principale est celle de débusquer, combattre et vaincre tous les monstres dissimulés dans un environnement prédéfini en 2D.

Attendus

1. Personnage

Sélection du personnage

Pas de changement de personnage durant une partie

Caractéristique d'un personnage paramétrable par le client

*(en **gras** sont paramétrables par le joueur):*

- *nom de classe*
- ***nom du joueur***
- *nombre de points de vie max*
- *nombre de points de vie actuels*
- *nombre de points de mana max*
- *nombre de points de mana actuels*
- ***compétences choisies***
- *compétences*
- *fiche descriptive*

Caractéristiques des ennemis paramétrables par le client:

- *nom de l'ennemi*
- *nombre d'ennemis sur la carte*
- *emplacement des ennemis sur la carte*
- *Nombre de points de vie max*
- *Nombre de points de vie actuels*
- *compétences possibles*
- *compétences choisies*

2. Interface

Vue du dessus (formalisée par des cases à l'écran) en 2 dimensions.

3. Déplacements

Contrôles : haut, bas, droite, gauche.

Certaines cases interdites (décor)

4. Systèmes de jeu

4.1. Combats

Découverte monstres : personnage dans la même case qu'un monstre

Combat au tour par tour

Déroulement tour :

- *Le joueur choisit une compétence à utiliser en fonction de son nombre de mana*
- *L'ennemi utilise une de ses compétences possibles*
- *Utiliser un ordre dans l'application des effets des compétences*
- *Condition d'arrêt combat : Combat terminé quand points de vie d'un des combattants à zéro.*

4.2. Progression

Déplacement du joueur

Combat contre des ennemis

5. Conditions d'arrêt du jeu

Le jeu s'arrête quand le joueur n'a plus de points de vie.

Le jeu s'arrête quand le joueur a vaincu tous les monstres.

Le jeu s'arrête quand le joueur le décide.

Analyse fonctionnelle

Modélisation du besoin

1. PC / NPC

1.1. Le(s) héros

Sélection du personnage

Pas de changement de personnage durant une partie

*Class personnage (en **gras** sont paramétrables par le joueur):*

- *nom de classe*
- ***nom du joueur***
- *nombre de points de vie max*
- *nombre de points de vie actuels*
- *nombre de points de mana max*
- *nombre de points de mana actuels*
- ***compétences possibles / choisies***
- *compétences*
- *fiche descriptive?*

Base de données personnage (fichier texte) :

- *nom*
- *points de vie*
- *compétences*
 - *points de dégâts*
 - *description*

1.2. Les monstres

Monstres/persos config:

Affichage des monstres/persos existants.

Création par le client d'un nouveau monstre (vérification du nom du monstre/perso).

Si le monstre/perso existe, on le remplace (suppression lignes dans le fichier texte, réécriture).

Si le monstre/perso n'existe pas, on le rajoute dans le fichier.

Monstres:

Nombre max de dégâts pour un monstre < pv max du joueur.

Base de données monstre (fichier texte):

nom

compétences

points de dégâts

description

(vitesse)

2. Interface

Vue du dessus (formalisée par des cases à l'écran) en 2 dimensions.

[En vue d'une éventuelle et spoiler interface avec une UI graphique, penser à encapsuler TOUTES les sorties dans une classe spécifique].

{ sous réserve d'un GUI }

{

lors d'un combat:

- on continue d'afficher la carte en haut de l'écran (taille : $\frac{3}{4}$ de l'écran) et en bas le combat entre le monstre et le personnage*
- personnage à gauche de l'écran et monstre à droite*

}

3. Déplacements

*Contrôle haut bas droite gauche.
Certaines zones interdites (décor).*

Logique du mouvement:

- *Le personnage est sur une case*
- *Si la case est vide :*
 - *Choix de bouger au nord SI CASE ACCESSIBLE (z)*
 - *Choix de bouger au sud SI CASE ACCESSIBLE (s)*
 - *Choix de bouger à l'est SI CASE ACCESSIBLE (d)*
 - *Choix de bouger à l'ouest SI CASE ACCESSIBLE (q)*
 - *Si la case est vide :*
 - *on recommence*
 - *sinon on fait un combat*
- *Si la case est inaccessible ne pas bouger le personnage et re-demander au joueur où il veut aller (ou simplement le faire rejouer? l'obstacle est visible).*

Base de donnée carte (fichier texte, matrice):

- *tableau à double entrée*
- *une case:*
 - *eau (case impossible d'accès par le joueur)*
 - *arbre (case impossible d'accès par le joueur)*
 - *terre (case possible d'accès par le joueur)*
 - *(terre régénératrice (case possible d'accès par le joueur))*

4. Systèmes de jeu

4.1. Combats

Découverte monstres : perso' dans même case qu'un monstre

Affichage d'une interface de combat en dessous de la carte.

Combat au tour par tour

Déroulement tour :

- *Joueur choisit compétences à utiliser en fonction de si il l'a et si assez de mana*
- *Même chose pour l'ordi.*
- *Ordre dans application des effets des compétences*
- *Condition d'arrêt combat : Combat terminé quand point de vie d'un à zéro.*

Logique d'un tour :

- *Le joueur choisit son attaque*
- *Le monstre choisit une attaque (hasard ?)*
- *Fonction d'application des effets du tour :*
 - *(Test de réussite du coup: toucher / rater selon jet random) POUR JOUEUR*
 - *(Test de réussite du coup: toucher / rater selon jet random) POUR MONSTRE*
 - *Si les deux loupent, tour fini.*
 - *Si un des deux loupe, on n'applique que l'attaque de celui qui a réussi son jet de toucher. Tour fini.*
 - *Sinon :*
 - *Selon les paramètres de la partie, le premier coup sera déterminé par soit une stat de vitesse, soit celle qui fait le plus de dégâts, soit le joueur (ou monstre) joue toujours en premier.*
 - *Quand l'ordre d'application est choisi, on applique le premier coup :*
 - *Application des dégâts/soins selon technique utilisée.*
 - *Si le joueur/monstre meurt, le tour s'arrête.*
 - *Sinon, on applique le second coup.*
 - *Application des dégâts/soins selon technique utilisée.*

Logique d'un combat :

- *Faire des tours tant que personne ne meurt*
- *Il y a 3 conditions d'arrêt :*
 - *le joueur meurt (Game Over)*
 - *le monstre meurt (le jeu continue)*
 - *le monstre meurt, mais c'était le dernier (jeu fini)*

[Combats: suggestion de passer par une pile, ce qui permet d'avoir autant de joueurs et monstres qu'on veut, on applique les effets et attaques dans la pile => meilleure évolutivité du code].

4.2. Progression

(Système de niveaux (régén vie/mana) tous les x monstres tués).

Affichage d'informations à certains moments clés de la partie ("vous avez tué environ la moitié des monstres"; "il reste 1 monstre")

Affichage d'indications sur l'emplacement du dernier monstre (x cases du joueur). Message sur l'écran indiquant l'éloignement (fluidifie le end game).

5. Conditions d'arrêt du jeu

- *Joueur n'a plus de points de vie.*
- *Joueur a vaincu tous les monstres.*
- *Volonté du joueur.*

Clarifications

Questions	Réponses
Fonctionnalités personnages?	Choix dans une liste (par classes par ex.) avec des compétences fixes. <i>Type: objet Bob, classe Perso</i>
Création carte ?	Peut être soit configurable par l'utilisateur, soit générée automatiquement au début du jeu (configurable semble plus simple, mais cartes impossibles à prévoir/empêcher).
Représentation mode de combat ? Interface de combat ? (style pokémon ?)	Interface de combat apparaît en dessous de la carte
Comment déterminer l'ordre d'application?	A choisir. Plus simple: hasard, ou le joueur tape tout le temps en premier
Système de niveaux (plusieurs cartes)?	Plusieurs cartes (choix d'une au début du programme). Une seule par partie.
Compétences: pool à partir duquel on choisit pour les assigner à un perso/monstre? Ou création pour chaque personnage?	Comme on veut.
Monstre case départ? Ou dans zones enclavées?	NON (cf. Strawpoll), Il ne peut pas y avoir d'ennemi ou de zones de départ dans des zones enclavées
Pool mana minimum? Le client choisit? Lié aux compétences assignées au perso?	
Doit-on considérer que les ennemis peuvent se déplacer, déplacements au tour par tour?	Non. Trop de complexité, et empiète sur la jouabilité.

Caractérisation de chaque fonction

Intitulé	Détails	Paramétrable par	Importance
Personnages	Nombre de personnages différents	Client	
	Nom de classe	Client	
	Nom du joueur	joueur	
	Nombre de points de vie max	Client / Dev	
	Nombre de points de mana	Client / Dev	
	Compétences	Client	
	Compétences sélectionnés	joueur	
	Description	Client	
Monstres	Nombre de type d'ennemis	Client	
	Nom de l'ennemi	Client	
	Nombre de points de vie max	Client / Dev	
	Compétences possibles	Client	
Environnement	Taille de la carte	Client / Dev	
	Nombre d'ennemis sur la carte	Client	
	Emplacement des ennemis	Client / Dev	
	Emplacement du décor	Client	