



RÉVÉLATEUR D'INGÉNIEURS
DEPUIS 1961

Cahier des charges

BlindQuest-S2



Nom du projet	BlindQuest-S2
Version	1.0
Date	17/03/15
Auteurs	CARON Clémence – c4caron@enib.fr GONI Guillaume – g4goni@enib.fr

1 DESCRIPTION GENERALE	3
2 EXPRESSION DU BESOIN	3
2.1 REGLES DU JEU	3
2.2 L'INTERFACE UTILISATEUR	3
2.2.1 IMPORTANCE DU SON	3
2.2.2 INTERACTIONS	3
2.3 CONTRAINTES TECHNIQUES	4
2.4 SCENARIO D'UTILISATION	4
S0 : SCENARIO PRINCIPAL	4
S1 : SE DEPLACER	5
S2 : COMBATTRE	5
S3 : FIN DU JEU	6
3 ANALYSE DU BESOIN	6
3.1 FONCTIONNALITES	6
3.2 CRITERES DE VALIDITE ET DE QUALITE	7
3.2.1 VALIDATION	7
3.2.2 QUALITE	7
3.2.3 IMPORTANCE DES FONCTIONNALITES	7
4 LIVRABLES	8
4.1 ECHEANCIER	8
4.2 DESCRIPTION DES LIVRABLES	8
4.2.1 CDC : CAHIER DES CHARGES	8
4.2.2 C1 : DOCUMENT DE CONCEPTION V1.0	8
4.2.3 P1 : PROTOTYPE P1	8
4.2.4 P2 : PROTOTYPE P2	8
4.2.5 VF : VERSION FINALE	8

1 Description générale

Dans le cadre des enseignements de MDD – Méthodes de développement - du semestre 2 de l'ENIB, nous allons créer un jeu vidéo, en python. Il s'inspirera de jeux de rôle et aura la particularité de guider l'utilisateur exclusivement par le son.

2 Expression du besoin

BlindQuest est un jeu de rôle pour un joueur. Il faut se déplacer à travers un monde, franchir différents environnements et vaincre le « boss » final, en se laissant guider par la musique ambiante.

2.1 Règles du jeu

Il n'y a pas d'interface graphique.

Le joueur doit se déplacer à l'aide du clavier. Il se situe dans l'environnement grâce aux sons : forêt, rivière, monstre etc. qu'il entendra.

Le but est de trouver le monstre final et de l'abattre, pour cela, il faut frapper le monstre jusqu'à ce que mort s'en suive.

Le monstre peut également triompher du joueur si ce dernier ne lui a pas infligé suffisamment de dégâts.

Sur son chemin, le joueur peut rencontrer d'autres ennemis, plus faibles, qui le ralentiront dans sa quête.

La partie est gagnée lorsque le boss final est vaincu ; elle est perdue si le joueur meurt, pour quelque raison que ce soit – environnement dangereux, monstres divers etc..

Il ne peut y avoir de match nul.

2.2 L'interface utilisateur

2.2.1 Importance du son

BlindQuest n'affiche en général rien au joueur. Il n'y a pas de carte visuelle, on ne voit ni le personnage, ni les ennemis qu'il affronte. Néanmoins, il est possible d'afficher une aide à l'écran, pour les commandes notamment.

Il est donc nécessaire, pour jouer, d'avoir des haut-parleurs ou des écouteurs en état de marche.

2.2.2 Interactions

Le joueur a la possibilité de se déplacer dans quatre directions : Nord [↑], Sud [↓], Est [→] et Ouest [←]. La touche [E] sert à attaquer, [Q] à quitter, et [H] à afficher l'aide. Le jeu peut

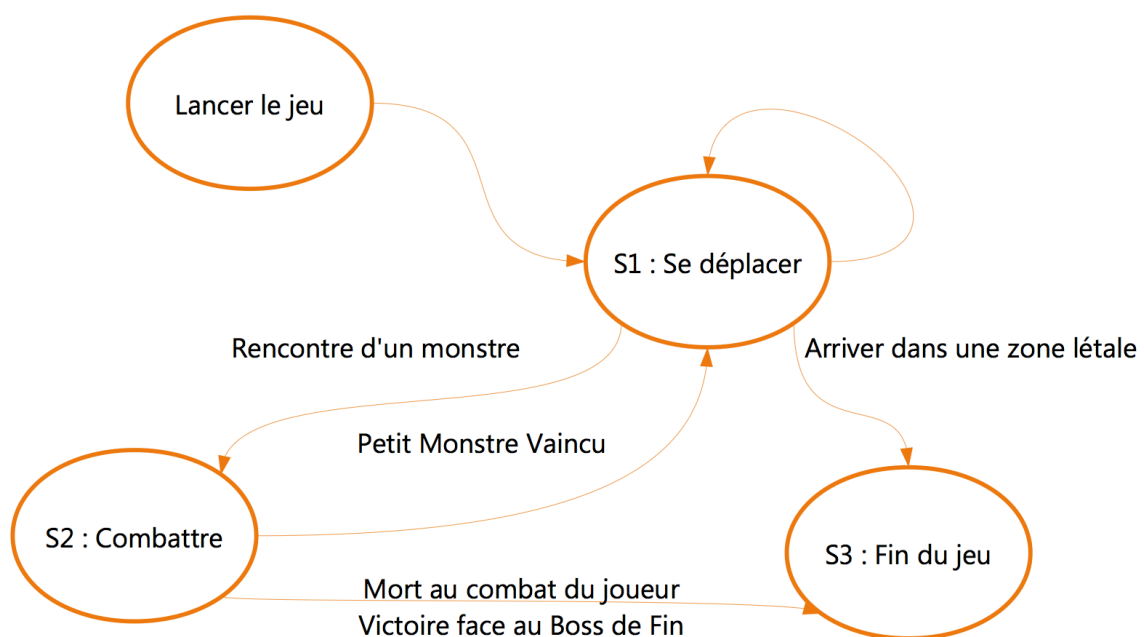
également être mis en pause [SPACE], cela coupera temporairement la musique. Les commandes pourront potentiellement être modifiées.

2.3 Contraintes techniques

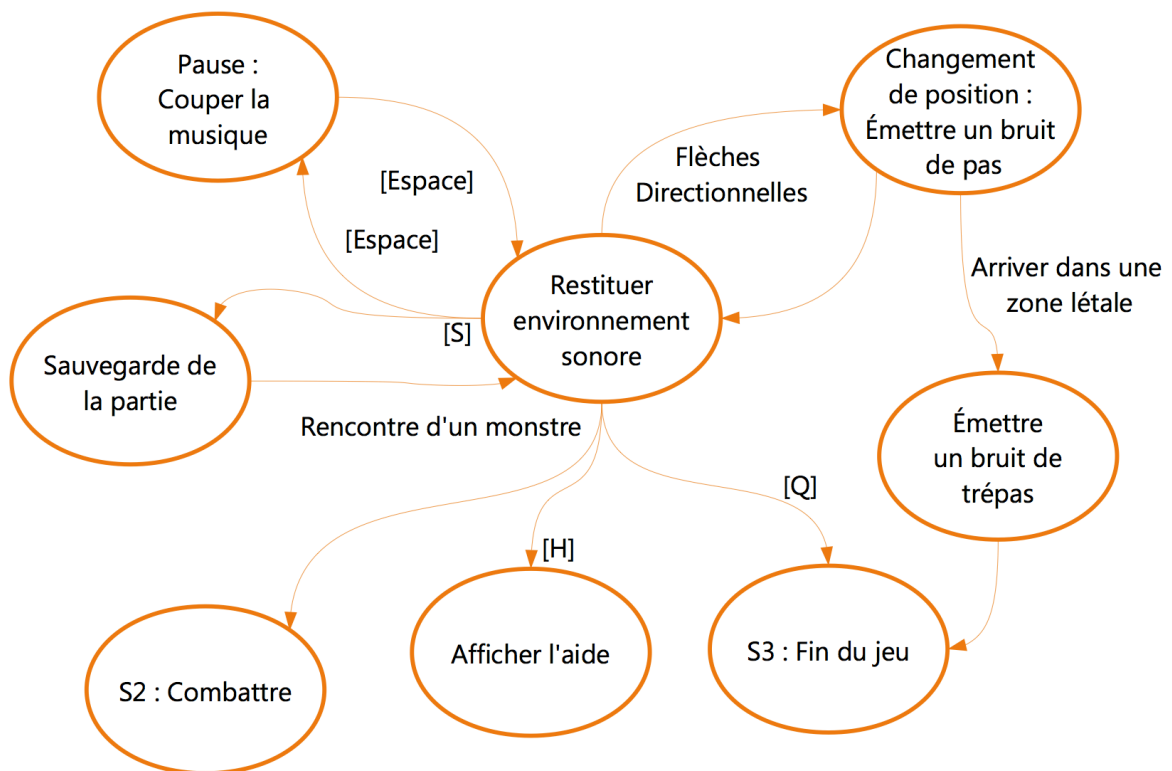
BlindQuest doit fonctionner sur les machines TP de l'ENIB, et prendre en charge les fonctionnalités audio. Le développement devra se faire en python. Nous utiliserons de la programmation objet, avec un raisonnement objet, mais un déroulement procédural pour rester en adéquation avec le cours de MDD. La barrière d'abstraction sera donc grandement employée, comme recommandé en classe.

2.4 Scénario d'utilisation

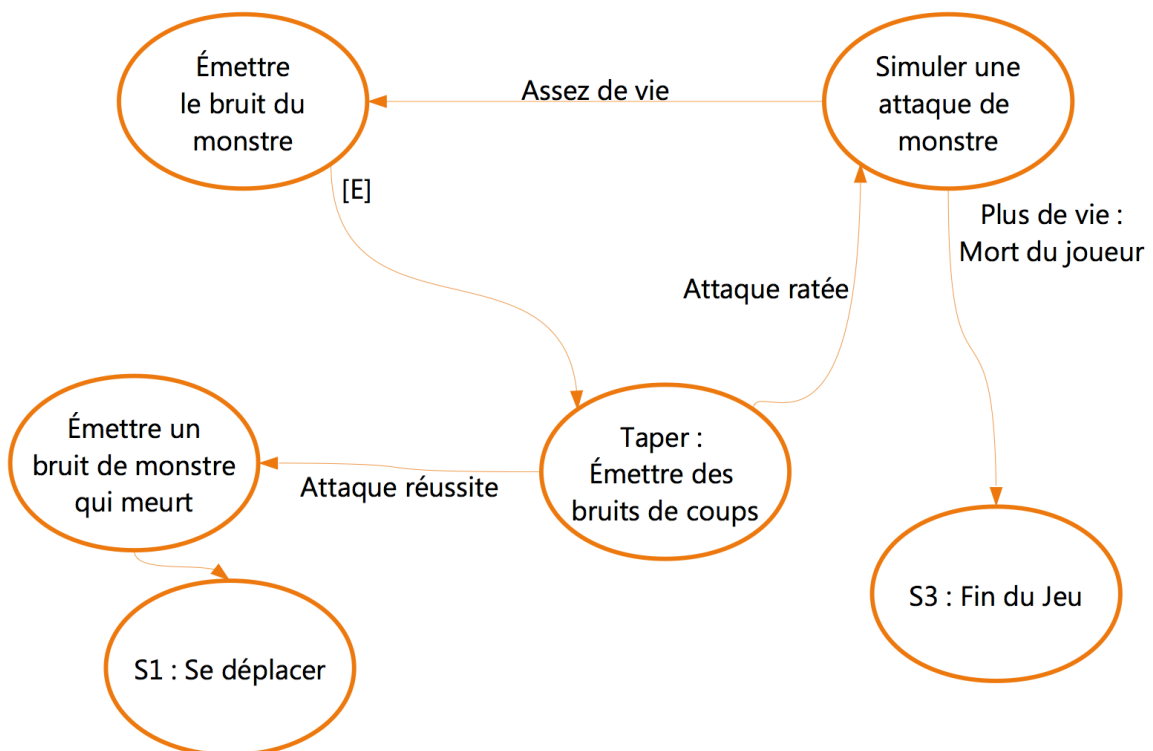
S0 : Scénario principal



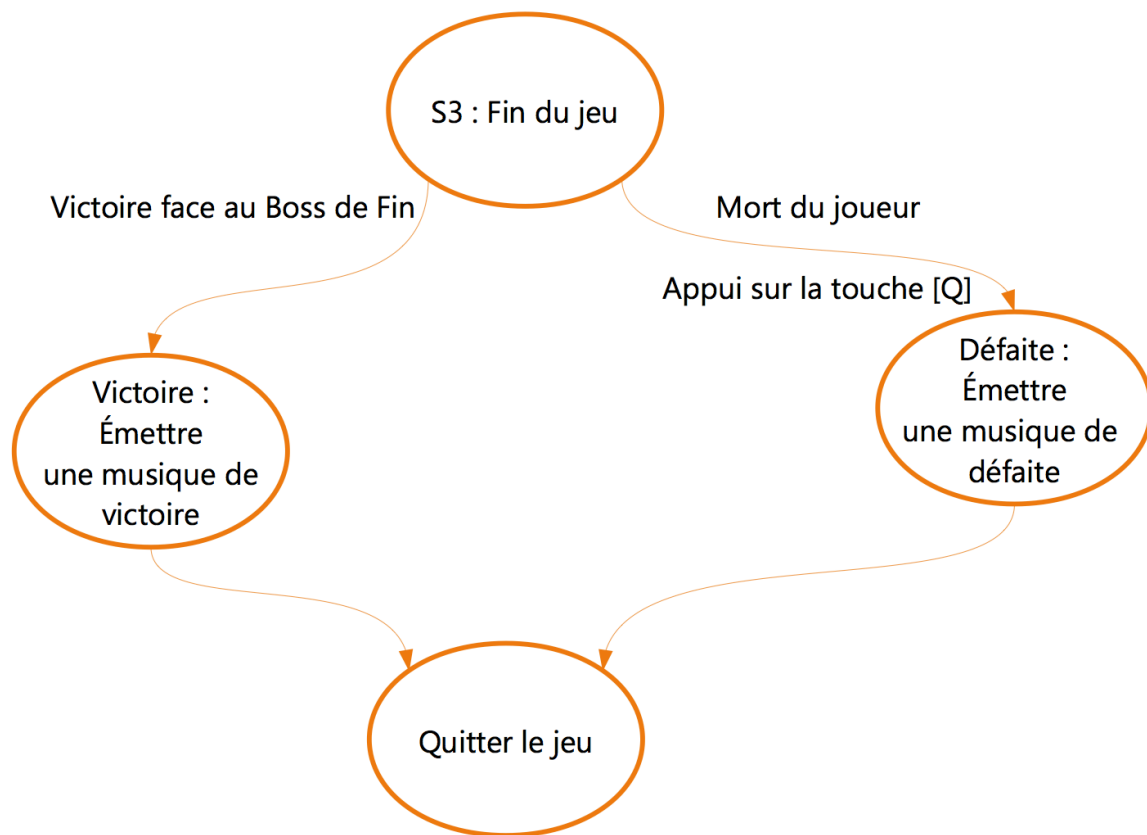
S1 : Se déplacer



S2 : Combattre



S3 : Fin du jeu



3 Analyse du besoin

3.1 Fonctionnalités

- F1 : Charger la carte
- F2 : Restituer l'environnement sonore
- F3 : Gestion des déplacements avec les touches [↑] [↓] [→] [←]
- F4 : Gestion des combats
 - F4.1 : Simulation d'attaque du joueur (succès (60%) ou échec)
 - F4.2 : Gestion de la vie du personnage
- F5 : Gestion de la mort
 - F5.1 : Zone létale
 - F5.2 : Décès lors d'un combat
- F6 : Quitter le jeu
 - F6.1 : Victoire contre le boss final
 - F6.2 : Mort du personnage
 - F6.3 : Pression de la touche [Q]
- F7 : Gestion de la sauvegarde
 - F7.1 : Sauvegarde avec la touche [S]
 - F7.2 : Charger une sauvegarde avec [C]
- F8 : Mettre en pause en appuyant sur [SPACE]
- F9 : Afficher l'aide – les commandes – avec la touche [H]

3.2 Critères de validité et de qualité

3.2.1 Validation

Le jeu sera validé ainsi :

- Le code doit s'exécuter correctement, comme explicité dans le manuel utilisateur.
- Son utilisation permettra de constater que les fonctionnalités ont bien été implémentées.

3.2.2 Qualité

La qualité du jeu sera évaluée selon différents critères :

- La jouabilité : le joueur doit être en capacité de comprendre dans quel environnement il se trouve et de savoir quand un combat commence et se termine.
- La robustesse.
- Le respect des méthodes de conception et de codage enseignées en cours.

3.2.3 Importance des fonctionnalités

0 : Indispensable

1 : Forte valeur ajoutée

2 : Facultatif

F1 : Charger la carte	0
F2 : Restituer l'environnement sonore	0
F3 : Gestion des déplacements	0
F4 : Gestion des combats	0
F4.1 : Attaque du joueur	0
F4.2 : Gestion de la vie du personnage	1
F5 : Gestion de la mort	0
F5.1 : Zone létale	1
F5.2 : Décès lors d'un combat	0
F6 : Quitter le jeu	0
F6.1 : Victoire contre le boss final	0
F6.2 : Mort du personnage	0
F6.3 : Pression de la touche [Q]	0
F7 : Gestion de la sauvegarde	2
F7.1 : Sauvegarde	2
F7.2 : Charger	2
F8 : Mettre en pause	2
F9 : Afficher l'aide	1

4 Livrables

4.1 Echéancier

24 mars 2015 : cahier des Charges
Semaine du 6 avril : conception
Semaine du 4 mai : premier prototype
Semaine du 18 mai : second prototype
8 juin : version finale

4.2 Description des livrables

4.2.1 CDC : Cahier des charges

Expression et analyse du besoin.
Fichier : cdc-BlindQuest-Caron-Goni.pdf

4.2.2 C1 : Document de conception v1.0

Fichier : conception-BlindQuest-Caron-Goni.pdf

4.2.3 P1 : Prototype P1

Ce prototype mettra en place le chargement de la carte F1, la restitution de différents environnements F2, le déplacement F3 et la possibilité de quitter le jeu F6.3.
Toutes les versions du jeu sont livrées dans une archive au format .zip ou .tgz et contiennent le manuel d'utilisation dans le fichier readme.txt

4.2.4 P2 : Prototype P2

Cette version sera complétée par des améliorations de l'environnement F5.1, notamment la création de milieux dangereux pour le joueur, les monstres et la fonction de combat F4, F6.1, F6.2, le décès du personnage F5, le mode pause F8, l'aide F9, et éventuellement la sauvegarde F7.

4.2.5 VF : Version finale

Contient toutes les versions des documents : cahier des charges et conception, prototypes, manuel d'utilisation.