

Sujet de Projet Tutoré EISAIC

Projet tutoré par Alexandre Leroux (alex@sherpa.one)

Document v1. 27/9/2020

Vous réaliserez une application web de gaming en ligne.

Concept du jeu Eisaic

L'objectif d'un joueur est de faire traverser son personnage dans un labyrinthe, le plus rapidement possible, en évitant les dangers et en récoltant un maximum de bonus.

Une partie réunit jusqu'à 4 joueurs en simultanée. Le vainqueur est le joueur qui réussit à traverser le labyrinthe en premier.

Chaque joueur ne voit que son propre personnage à l'écran mais peut suivre l'évolution du score de ses adversaires en temps réel.

La partie se termine :

- lorsqu'un joueur a franchi le labyrinthe,
- si les joueurs ont tous perdu (contact avec un danger),
- si les joueurs se déconnectent,
- automatiquement après 5mn (partie chronométrée),

Lorsqu'une partie est terminée, une nouvelle partie avec un niveau de jeu plus complexe.

Spécifications fonctionnelles

L'application proposera les fonctionnalités suivantes :

- Création de compte joueur,
- Personnalisation du personnage incarné (nom, apparence...),
- Classement des joueurs (par nombre de victoires et bonus),
- Organisation de parties regroupant jusqu'à de 1 à 4 joueurs en simultanée,
- Permettre à un joueur d'initier une partie en configurant le niveau de jeu et d'attendre que des joueurs le rejoignent,
- Génération de niveau de jeu (vue du dessus ou isométrique, cf. Pacman, Zelda).
- A chaque partie : création dynamique d'un labyrinthe et positionnement des bonus et dangers de façon aléatoire mais logique. Déplacement du personnage à l'aide des flèches du clavier,
- Animations au format Gif,

- Augmentation croissante de la complexité des niveaux de partie en partie,
- Interface back office permettant à un administrateur de valider les inscriptions, ajouter de nouveaux bonus, dangers et murs de labyrinthe (sous forme d'images),

Spécifications techniques

- HTML5,
- CSS3,
- JavaScript,
- Node.js,
- MongoDB,
- Socket.io,
- Docker,
- API Rest,

Spécifications d'architecture

Vous mettrez au point une architecture découplée mettant en relation une Single Page Application avec une API liée à une base de données.

Vous utiliserez Docker (compose) pour la mise en place des différents services.