Cahier des charges :

# Description du projet

Nous sommes partis sur la création d’un jeu de type Bomberman, se basant sur les technologies suivantes : HTML/CSS/JS/Twig/PHP/SQL.

Bomberman est un **jeu multi-joueurs** consistant à se déplacer sur **un plateau de jeu à damier, fermé**. L’objectif pour chaque **joueur** est détruire leur(s) adversaire(s) grâce à la pose de **bombes**. Il peut utiliser ses bombes afin de se frayer un chemin sur le plateau de jeu en détruisant les obstacles sur son passage (des **« blocs » destructibles**). Les graphismes seront axé « oldschool » avec un rendu 8-bit.

Nous utiliserons les technologies et langages abordés au cours de notre formation :

* **HTML/CSS** : le visuel de notre site se basera sur Bootstrap pour l’habillage CSS, ainsi que l’utilisation d’un sprite comportant les modèles du jeu Bomberman officiel.
* **JS/JQuery** : notre site comportera de nombreuses animations (déplacements des joueurs, pose et explosion des bombes, création/destruction des éléments dans nos cases…).
* **PHP/SQL** : le « back-office » sera construit avec PHP et une base de données sous SQL. L’ensemble de notre projet utilisera le framework **Symfony3.**

## Fonctionnalités principales

Les différents éléments nécessaires pour rendre notre jeu possible sont les suivants :

* **Un plateau :** il sera notre base de jeu, il fera office de « fond d’écran » de jeu et de base pour tous nos cases et nos joueurs.
* **Une partie :** elle sera la référence en base de données, elle n’aura pas d’impact sur le déroulement d’une partie.
* **Des cases :** elles nous permettront de gérer avec facilité les déplacements et les différentes interactions joueurs/objets du jeu.
* **Des joueurs** et **des utilisateurs** (ou user) : nous avons décidé de séparer l’utilisateur (qui se connecte et s’identifie sur le site) du joueur (qui est l’incarnation de l’utilisateur dans une partie) pour alléger chaque classe et permettre une gestion plus précise de ces 2 éléments. L’user sera stocké de façon pérenne en base de données, et à terme comportera des statistiques relatives à ses différentes parties, tandis que le joueur recevra les instructions de déplacements et gérera les actions dans la partie.
* **Des objets ou items :** ils correspondent à ce que l’on peut rencontrer sur les cases de notre partie, comme par exemple les bombes posés par les joueurs, les murs destructibles ou indestructibles et les bonus (voir éléments plus bas).

## Fonctionnalités avancées

Quelques améliorations seront à rajouter suivant l’avancement de notre projet :

* L’ajout de plusieurs bonus différents (plus de bombes pour le joueur, bombes plus puissantes, vitesse de déplacement accrue…).
* La création d’autres modes de jeu. Tel qu’un système de points attribués lors de la destruction de blocs, la récolte de bonus ou l’élimination d’autres joueurs.
* Plusieurs styles de jeu et visuels.
* Avoir la possibilité d’être spectateur des parties en cours

## Limites du jeu

Notre première question est la jouabilité et la fluidité que l’on pourra avoir dans notre jeu. Il y a beaucoup d’interactions entre les différents joueurs et avec les éléments de notre partie (déplacements, pose de bombes, destruction de blocs de manières simultanées par différents joueurs) et nous sommes inquiets du rendu que pourrait avoir toutes ces actions.

# Diagramme de classe

Voir GitHub « DiagrammeClasse.pdf »

# Storyboard

Voir GitHub « projetMok.pdf »