

BOMBERMAN CUP

Dernière mise à jour : 19/02/2021

1. L'événement

Le but de cet événement de mettre à disposition des étudiants, un serveur du jeu pour que tout le monde puisse expérimenter la crée de bot ou d'IA sur un jeu ludique.

Il existe un client ou les développeurs peuvent jouer contre leurs bots ou visualiser la partie.

Le lien vers le client :

<https://bomberman-mibde.fr/>

Un server discord est disponible via :

<https://discord.gg/mTZbhAvSuS>

2. Le Jeu :

Le jeu Bomberman est un jeu vidéo où le joueur incarne un poseur de bombe. Le but du jeu est d'être le dernier joueur en vie dans le temps imparti.

Pour ce faire, le joueur pose des bombes sur la map, explose des blocs de brique ou tue des joueurs et récupère potentiellement des bonus.



Les cases marrons (brique) sont celles qui peuvent être explosées par le joueur à l'aide d'une bombe, il doit les exploser pour pouvoir progresser dans la map.

En explosant ces cases il se peut que le joueur découvre des bonus permettant d'améliorer ses capacités.

Les blocs gris cependant ne peuvent être explosé ni bougé.

Le tunnel que vous voyez sur l'image au centre mesure 3 cases (c'est cadeau).

Lorsqu'un joueur meurt, tous les bonus qu'il a accumulé sont redistribuer aléatoirement sur la map.

(Excusé moi, mais j'avais oublié cette composante du jeu, elle n'est pas disponible actuellement, je l'ajouterai le plus vite possible)

Lorsque la partie commence tous les joueurs partent avec une vitesse d'un block parcourue en 250ms, la capacité de ne posé qu'une bombe à la fois,

Ainsi que la puissance de la déflagration qui ne dépasse pas 1 case à partie de la bombe.

Les bonus, malus et capacité :

Le bonus « flamme » donne à notre bombe une case supplémentaire à la déflagration lors de l'explosion. (flammeUP)

Le bonus « bombe » donne au joueur la capacité de placé une bombe supplémentaire. (bombeUP)

Le bonus « vitesse » réduit le temps pour parcourir un bloc de 15ms.(speedUP)

Le bonus « kick » donne au joueurs la capacité de shooter automatiquement dans une bombe lorsque celui-ci vient à la toucher. (Fonctionnalité disponible très prochainement) « kick »

Le bonus « super bombe » donne la capacité de posé une bombe dont l'explosion est absolue, qui ne s'arrête pas au premier obstacle.

« bombeSUP »

Il existe 1 malus pour la flamme, la bombe, la vitesse.

« flammeDO », « bombeDO », « speedDO »

C'est un jeu en temps réel, le joueur n'a pas besoin d'attendre pour pouvoir rejouer et enfin il peut s'arrêter là où il le souhaite.

Les actions du joueur sont :

- Se déplacer en haut
- Se déplacer en bas
- Se déplacer à gauche
- Se déplacer à droite
- Poser une bombe
- S'arrêter

3. Structure des données et coordonnées :

La map du jeu est un tableau de taille 21x15 (21 en longueur,15 en largeur).

La position courante du joueur est représentée par les variables xBlock, yBlock.

Ces variables sont les coordonnées de la case sur laquelle le joueur se trouve dans la map.

Les variables xp et yp représentent les coordonnées du joueur à l'intérieur de cette case.

Chaque joueur possède 3 « capacités » :

1 : la puissance de ses bombes, le nombre de cases que parcourt la déflagration en partant de la bombe.

2 : la vitesse, celle-ci est en ms et représente le temps que met le joueur à parcourir 1 cases du jeux.

4. Connection au serveur :

La connexion au serveur se fait exclusivement via une connexion websocket.

Le choix du langage de programmation est libre.

Toutes les données envoyées doivent être au format JSON exclusivement.

Le lien du serveur : <wss://bomberman-mibde.fr/websocketBomber/>

Si le lien précédent ne fonctionne pas voici un lien de secours (site perso):

<wss://pi-univers.fr/websocketBomber/>

Les informations de connections sont communiqués via l'URL au format query.

Exemple : <wss://bomberman-mibde.fr/websocketBomber/?t=play&s=default&nbg=2&name=BOT>

t : play or spect (joueur ou spectateur)

s : nom de la partie

nbg : nombre de joueurs

name : nom du joueur

Lorsque que vous vous connecté au server, tous les joueurs connectés reçoivent

```
{
  "new_player" : nom,
  "id" : id
}
```

Vous devez récupérer votre id grâce à la correspondance de votre nom, cette id servira à envoyer chaque requête par la suite.

Lorsque tous les joueurs se sont connecté alors le server envoie l'évènement « start » avec un tableau 2D appelé map qui contient une

map crée par le server avec les blocks en métal incassable et les blocks de briques, comme ci-dessous en version réduite :

```
[null, null, "b","x" ]  
[null, "x", "b", "x" ]  
["b" , "b", "b", "b" ]
```

« x » représente les cases de métal incassables

« b » représente les briques

Null représente une case vide, les premières cases autour d'un joueur sont toujours vides, mais il se peut qu'il y en ait à n'importe quelle endroit de la map.

L'événement « start » :

```
{  
  "start" : 1,  
  "map" : tableau 2D de la map du jeu  
}
```

5. Action du joueur

Un joueur peut envoyer 3 instructions :

Un joueur peut envoyer la direction vers laquelle il veut aller

```
{  
  "id" : votre_id,  
  "move" : " left | right | up | down "  
}
```

Les chemins, si vous le souhaitez vous pouvez soumettre au server un chemin, qui sera composé d'une table contenant les coordonnées de chaque case que compose le chemin, avec la possibilité d'y ajouter une action comme « bombe » pour poser une bombe. Le server fera suivre au joueur le chemin, en allant jusqu'à la première case du tableau puis en passant à la case suivante.

```
{  
  "id" : votre_id,  
  "path" : [ [xBlock, yBlock, "action_facultative"],  
             [x1Block, y1Block, "action_facultative"],  
             ... [xNBlock, yNBlock, "actionFacultative"] ]  
}
```

Exemple : position de départ du joueur : [0,0]

```
{  
  "id" : mon_id,
```

```

"path" : [ [0, 2, "bombe"], [0,0] [1,0] ]
}

```

Le joueur va aller jusqu'au point 0,2 pour y poser une bombe et vas revenir se protégé de l'explosion en [1,0] en passant par la case [0,0]

Un joueur peut poser une bombe à l'endroit où il se trouve.

Pour ce faire :

```

{
  "Bombe" : "votre_id"
}

```

6. Evènement Server

Le server envoyé régulièrement au joueur les évènements :

Start : signifie que la partie vient de commencer, le server nous renvoie une map pré construit

Move : pour signalé qu'un joueur est en mouvement et comment

Alert_bombe : Lorsqu'un joueur pose une bombe

- L'évènement move intervient lorsqu'un joueur change de direction ou qu'il change de xBlock ou yBlock.

```

{
  "move" : « left | right | up | down »,
  "id" : votre_id,
  "xBlock" : xblock,
  "yBlock" : yBlock,
  "xp" : xp,
  "yp" : yp
}

```

- L'évènement « Alert à la bombe » se produit lorsqu'un des joueurs vient de poser une bombe.

```

{
  "alert_bombe" : id_du_joueur qui vient de poser une bombe,
  "x" : xBlock,
  "y" : yBlock,
  "puissance" : puissance de la bombe
}

```

- Lorsqu'un des joueurs prend un bonus, le server le notifie autre autre joueur via :

```

{
  "takenBY" : id du joueur ayant pris ce bonus,
  "bnType" : " type du bonus ",
}

```

```

"x" : xBlock du bonus,
"y" : yBlock du bonus qui a été pris,
"speed" : nouvelle vitesse du joueur dans le cas d'un speedBon
us
}

```

- Lorsqu'une bombe explose, celle-ci est notifié via l'évènement « boom » comme ce ci

```

{
  "boom" : id_du_propriétaire de la bombe,
  "x" : xBlock,
  "y" : yBlock,
  "infX" : bornINFx,
  "infY" : bornINFy,
  "supX" : bornSUPx,
  "supY" : bornSUPy,
  "puissance": puissance de la bombe,
  "bonus" : bonusArray,
  "dead" : deadPeople
}

```

Quelques explications seraient la bienvenue,
 En plus de prévenir sur la position de la bombe qui à exploser,
 Cet évènement nous donne quelque information supplémentaire.
 Les bornes c'est par rapport à jusqu'où vas la bombe à exploser et dans chaque direction.

Bonus est un array contenant tous les bonus qui ont été dévoilé lors de l'explosion de la bombe :

```

{"type" : "nom_bonus", "x": xBlock du bonus, "y": yBlock du bonus }

```

Dead est un array contenant tous les joueurs qui ont été désintégré lors de l'explosion de la bombe :

```

{
  "whoIS" : id_du_joueur,
  "x": xBlock du joueur mort,
  "y": yBlock du joueur mort
}

```

Si vous avez des questions, n'hésite surtout pas à me contacter via :

Insta : paul_cibier

Discord : Paulo-21#2202

Email : paul.cibier@gmail.com

Telephone : 0777326385

DEMO DU TRAITEMENT DES REQUETTES ET CONNECTION AU SERVER, en NodeJS

```
const WebSocket = require('ws');
const client = new WebSocket('wss://bomberman-
mibde.fr/websocketBomber/?t=play&s=default&nbj=2&name=BOT');
var joueurs = {};
/*ON cree un tableau de 2 dimensions pour représenter la map*/
var map ;
/*La map contiendra les objets*/

function Bombe (puissance, x, y, id_joueur) {
/* On crée un objet Bombe pour représenter une bombe dans la map,
Ca sera beaucoup plus facile pour la suite du développement*/
    this.type = "bombe";
    this.puissance = puissance;
    this.xBlock = x;
    this.yBlock = y;
    this.id_joueur = id_joueur;
}
function Bonus (type, x, y) {
/*Un objet bonus*/
    this.type = "bonus";
    this.typeBonus = type
    this.xBlock = x;
    this.yBlock = y;
}
function Joueur (id, move, xBlock, yBlock, xp, yp) {
    this.id = id;
    this.move = move
    this.xBlock = xBlock;
    this.yBlock = yBlock;
    this.xp = xp;
    this.yp = yp;
}

client.onmessage = e => {
    var data = JSON.parse(e.data)
    if(data.move) { // A chaque notification de mouvement, on met a jour le
joueur dans notre tableau
        joueurs[data.id]= new Joueur(data.id, data.move, data.xBlock, data.yBl
ock, data.xp, data.yp);
        console.log(joueurs[data.id]);
    }
    else if(data.alert_bombe >= 0) { // A chaque notification d'alerte à la
bombe, on pose une bombe dans notre map
        map[data.y][data.x] = new Bombe(data.puissance, data.x, data.y, data.a
lert_bombe);
        console.log(map[data.y][data.x]);
        //function whereIamSafe();
    }
}
```

```

    }
    else if(data.boom >= 0) { //Quand une bombe à explosé, on met à jour notre
map avec les données envoyés
        for(var i = data.infY; i <= data.infY; i++) {
            map[i][data.x] = null;
        }
        for(var i = data.infX; i <= data.infX; i++) {
            map[data.y][i] = null;
        }
        data.bonus.forEach((item, i) => {
            map[item.y][item.x] = new Bonus(item.type, item.x, item.y);
            console.log(map[item.y][item.x]);
        });
    }
    else if(data.new_player >= 0) { //Un nouveau joueur vient de se connecté,
on le met dans notre tableau joueur
        if(data.nom == "BOT") { //On récupère nom id de notre bot
            myId = data.new_player;
        }

        var id = data.new_player;
        var tx = (id%2)?14:0;
        if(id < 2) ty = 0;
        else if (id < 4 && id >= 2) ty = 8;
        else if(id < 6 && id >= 4) ty = 11;
        else if(id < 8 && id >= 6) ty = 19;
        players[data.new_player] = new Joueur(id, "stop", tx, ty, 0.5, 0.5);
    }
    else if(data.start) { //On récupère la map généré par le server quand la
partie commence
        map = data.map;
    }
};

```

ATTENTION CE CI N'EST PAS UN BOT FONCTIONNEL MAIS JUSTE UNE DEMO