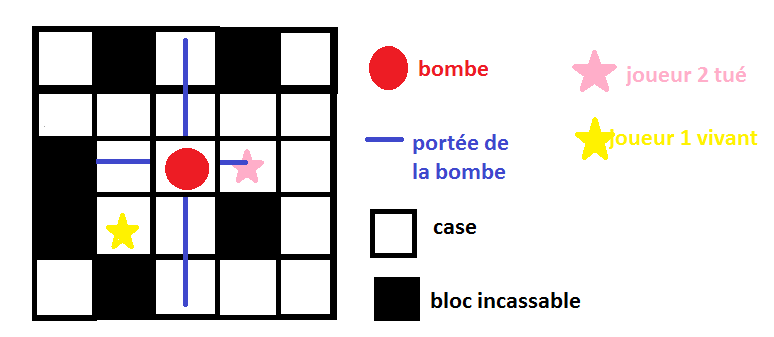
Bomberman

Principe du jeu :

Les différents joueurs dirigent chacun un personnage. Leur but est de tuer le personnage des autres joueurs. Pour cela, ils peuvent se déplacer (haut, bas, gauche, droite) sur les cases qui le permettent (c’est-à-dire pas sur les blocs incassables) et ils disposent de bombes qu’ils peuvent poser sur la case située à côté d’eux (haut, bas, droite, gauche). Ces bombes explosent et tuent les personnages mais ne détruisent pas les blocs incassables. Ces blocs incassables stoppent la portée de l’explosion des bombes comme pour les joueurs touchés par la bombe. Ces bombes possèdent une portée de 2 cases en haut, en bas, à gauche et à droite. Quand il ne reste plus qu’un joueur dans la partie, il a gagné.



## Actions possibles du joueur :

* Se déplacer : haut, bas, gauche, droite dans les cases qui le permettent (c’est-à-dire pas dans les blocs incassables)
* Poser une bombe qui explosera … secondes après le placement de la bombe.
* Quitter la partie : bouton « echap »

Le joueur 1 possède 6 boutons : les 4 flèches pour se déplacer, le bouton « …» pour poser une bombe, le bouton « echap » pour quitter la partie

Le joueur 2 possède 6 boutons : « z »,  « q », « s », « d » pour se déplacer, le bouton « … » pour poser une bombe, le bouton « echap » pour quitter la partie

## Principe de la map:

Pour créer la map, définir un tableau map. Ce tableau a une propriété 2D : nombres de cases en longueur & nombres de cases en largeur. On choisit le nombre de cases en fonction du nombre de joueurs  -> exemple : pour 2 joueurs, prendre pour nombre de cases en largeur 10 et pour nombre de cases en longueur 8.

Dans ce tableau créer 2 types de cases : les cases vides et les cases pleines que l’on appellera respectivement : les «  cases » et les « blocs incassables ». Le nombre de blocs incassables est défini en fonction du nombre de joueurs -> exemple : pour 2 joueurs, prendre pour nombre de blocs incassables 20.

Grâce à la fonction « random », les blocs incassables seront placés aléatoirement sur la map.

Exemple de map

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Joueur 1 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | Joueur 2 |

## Programmation :

* Définir une variable map
* Faire un tableau avec la propriété 2D
* Définir un objet joueur : pour avoir 2 joueurs, instancier cet objet 2 fois, etc…
* Définir un objet bombe : l’instancier autant de fois que le joueur utilisera l’action « poser une bombe » - l’objet détruit l’instance joueur quand celui est à portée.

Idées :

* associer les touches d’action du clavier à une manette (PS3/Xbox)
* blocs cassables ? items dans blocs cassables ? améliorations (vitesse, portée) grâce à ces items ?
* faire un timeur : au bout de 5 minutes, la partie s’arrête, il n’y a pas de vainqueur.