

Rapport de soutenance- Projet S2

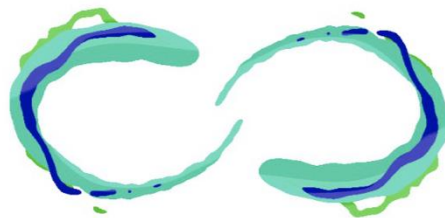
Epita 2022

Sacha Reggiani

Clément Fossati

Aboubakar Charf

Eytan Benhamou



CEAS

Sommaires

1 Récapitulatif – page 3

1.1 Principe du jeu 3

1.2 L'avancement 3

2 Problème du multijoueur – page 4

3 Ajout de fonctionnalités utiles – page 9

3.1 Mort 9

3.2 Chaines d'explosions. 10

3.3 Bombes solides. 11

3.4 Réduction de la puissance des bombes 12

4 Ajout des items – page 13

5 Avancement du site – page 14

6 Objectif – page 16

6.1 Objectif 2ème soutenance 16

6.1 Objectif 3ème soutenance 17

Récapitulatif :

Principe du jeu :

Tout d'abord un peu de contexte, notre jeu est un bomber-man c'est à dire un jeu vu du dessus avec des blocs destructibles, l'objectif de chaque joueur est d'éliminer ses opposants sur la carte via des bombes qui explosent au bout d'un certain temps. Pour ce faire le joueur va devoir naviguer dans un niveau carré ou il sera bloqué par des blocs, mais grâce à ses bombes il pourra les détruire et progresser. De plus les blocs ont un pourcentage de chance de laisser tomber des power-up, qui permettent d'augmenter les statistiques du joueur par exemple pouvoir poser plus de bombes, avoir une plus puissance pour les bombes etc. ...

L'avancement :

La dernière fois nous avons commencé le projet après s'être réparti les tâches.

Nous avons donc nos 1ers graphismes avec une map et les textures des différent blocs (destructible et indestructible).

Le système multijoueur était opérationnel, il était possible pour plusieurs personnes sur différents ordinateurs de rejoindre la même partie, mais avait de nombreux bugs notamment avec les bombes qui n'étaient pas vues par tous les joueurs connectés à la partie.

Concernant le menu, la seule chose qui marchait était le bouton play qui donnait ensuite la possibilité d'héberger ou de rejoindre une partie.

A propos du jeu en lui-même, il était impossible pour les joueurs de mourir, et les bombes détruisaient tous les blocs dans un rayon de 3 cubes en croix, de plus il était possible pour les joueurs de passer à travers les bombes, il était par contre possible de se déplacer sans problème, et les blocs indestructible résistaient bien aux bombes contrairement aux blocs destructibles qu'ils étaient possibles de détruire grâce aux bombes.

Nous avons donc une base pour commencer notre projet, mais le jeu n'était clairement pas jouable.

Problème du multijoueur :

En reprenant le projet depuis là où il en était je me suis heurté à un gros problème. Le multijoueur, le problème avec le multijoueur depuis la dernière fois c'est que les bombes, les explosions et les destructions de blocs n'étaient pas reliés, c'est à dire qu'un joueur pouvait faire ce qu'il voulait, ça n'avait pas d'importance pour les autres joueurs.

Le multijoueur marchait car on pouvait très bien voir tout le monde se déplacer sur la carte mais aucune possibilité d'interagir. Donc ce fut le problème majeur, le tout premier qu'il fallait résoudre avant de pouvoir faire quelques autres implémentations, Mais le défi était de taille, c'était probablement l'étape la plus dure car le groupe n'ayant aucune expertise en multijoueur il fallait commencer à zéro.

Heureusement il y avait quelques tutoriels bien ficelés et détaillés qui pouvaient aider à la résolution de ce problème. Malheureusement il y avait très peu d'information concernant des maps modulaires qui peuvent être modifié au cours du temps.

C'est donc ainsi que commença la construction d'un multijoueur fonctionnel. Tout d'abord je me suis penché sur la documentation officielle de Photon (qui est notre logiciel qui permet d'héberger des serveurs, sur lesquelles on peut connecter tous les joueurs), et j'y ai rapidement la bonne méthode, il fallait juste ajouter PhotonNetwork.(ma fonction) afin d'envoyer sur le serveur les bombes.

```
if (Input.GetKeyDown("space"))  
{  
    PhotonNetwork.Instantiate("Bomb",  
}
```

Là on pourrait se dire que s'était quand même plutôt simple compte tenu des paragraphes si dessus, sauf que les problèmes arrivent maintenant. Une fois cet ajout fait j'ai lancé le jeu et remarqué qu'il y avait pas mal d'erreur, je me suis renseigné, j'ai trouvé et j'ai corrigé.

Le problème était que chaque joueur devez faire disparaître sa version, pour les bombes ou les explosions, il fallait donc ajouter un composant sur les bombes et les explosions afin de savoir qui a quoi.

```
PhotonView view;  
  
void Start ()  
{  
    view = GetComponent<PhotonView>();  
}
```

```
if (view.IsMine)  
{  
    PhotonNetwork.Destroy(gameObject);  
}
```

Une fois ceci fait la majorité des erreurs avaient disparues, mais le problème de taille qui s'est imposé juste après a été et est encore le plus difficile de tout le projet. Le multijoueur.

Ce problème si difficile à résoudre vient du fait qu'au cours de la partie, tout semble se dérouler comme prévu lorsque qu'une animation de bombe s'effectue deux fois, en réappuyant sur la touche pour poser une bombe je me rends compte que là il n'y a plus qu'une seule animation je réessaie plusieurs fois et à chaque fois j'obtiens un résultat différent sans aucune explication logique. J'essaie en marchant, en ne bougeant plus en posant deux bombes côte-à-côte, mais rien n'y fait il n'y a rien qui puisse expliquer ce phénomène bizarre. Alors c'est naturellement que je lance la session sur différents ordinateurs afin de comprendre ce qui se passe, et là je constate que lorsqu'il n'y a qu'une explosion les destructions occasionnées par la bombe ne sont pas retransmises pour tous les joueurs. Et lorsque qu'il y en a plusieurs les destructions sont bien réparties pour tous les joueurs, et un autre fait bizarre c'est que plus il y a de joueurs connectés en même temps plus il y a d'explosions lorsqu'elles sont multiples (elles sont d'ailleurs proportionnelles au nombre de joueurs).

A ce moment ça ne fait aucun doute que le problème vient de mon script, mais pour toute la partie multijoueur il n'y a probablement aucune erreur car le code utilisé revient très souvent dans de nombreux tutoriels ainsi que sur la documentation officielle de Photon qui est notre serveur

```
void Start()
{
    PhotonNetwork.ConnectUsingSettings();
}

public override void OnConnectedToMaster()
{
    PhotonNetwork.JoinLobby();
}

public override void OnJoinedLobby()
{
    SceneManager.LoadScene("Lobby");
}
```

```
public InputField createInput;
public InputField joinInput;

public void CreateRoom()
{
    PhotonNetwork.CreateRoom(createInput.text);
}

public void JoinRoom()
{
    PhotonNetwork.JoinRoom(joinInput.text);
}

public override void OnJoinedRoom()
{
    PhotonNetwork.LoadLevel("PlayTest");
}
```

Après avoir relu tout mon code pour pouvoir identifier d'où venait le problème, je pense enfin l'avoir trouvé, c'est uniquement dans l'appel des explosions qui sont en charges de les répartir autour de la bombe et de gérer le cas où il y aurait des blocs à détruire. Une fois ceci trouvé, il y a eu deux autres problèmes, je ne sais pas si le problème vient directement de mes fonctions qui gèrent l'explosion des bombes ou si ça vient de l'appel aux fonctions qui gèrent les explosions des bombes.

```
void Update ()
{
    countdown -= Time.deltaTime;

    if (countdown <= 0f || explode)
    {
        FindObjectOfType<MapDestroyer>().Explode(transform.position, 2);
        if (view.IsMine)
        {
            PhotonNetwork.Destroy(gameObject);
        }
    }
}
```

Voilà après de nombreuses tentatives pour corriger cette fonction je n'ai pas réussi à trouver ce qui n'allait pas avec

Ajout de fonctionnalités utiles :

Mort :

Dans cette section nous aborderons plein d'ajout fait en vrac qui ne sont pas très long, nous les traiterons ici car ils ne méritent pas de sections à part mais constituent quand même des éléments importants dans la construction et amélioration du jeu.

Premièrement la mort. Pour l'instant le jeu marche à peu près comme convenu, sauf qu'il manque un élément clé dans les mécaniques basique du jeu, la mort.

Pour l'implémenter il aura simplement suffi de regarder si un joueur et sur le sprite d'une explosion et si oui on fait disparaître le joueur tout simplement

```
PhotonView view;

void Start ()
{
    view = GetComponent<PhotonView>();
}

private void OnCollisionEnter2D(Collision2D other)
{
    if(other.gameObject.tag == "Explosion")
    {
        if(view.IsMine)
        {
            PhotonNetwork.Destroy(gameObject);
        }
    }
}
```

Pour effectuer cela on a créé un tout nouveau script qui ne sera dédié qu'à ça. Comme on peut le voir il y a un composant Photon view qui permet de garder en mémoire qui est qui et aussi surtout de pouvoir détruire l'objet sur chaque version du jeu pour chaque joueur. Sinon le reste parle de lui-même, on a juste rajouté une collision à la bombe ce qui permet de lancer la fonction `OnCollisionEnter2D` qui vérifie s'il y a une collision entre l'objet et autre chose, en l'occurrence si c'est une explosion on détruit le joueur représenté la par `gameObject`.

Chaines d'explosions :

Une mécanique emblématique des jeux bomber-man c'est évidemment les chaînes d'explosions, qui permettent des réactions inattendues et souvent spectaculaires. Encore c'est assez simple à implémenter

```
if (countdown <= 0f || explode)

if(other.gameObject.tag == "Explosion")
{
    explode = true;
}
else
{
    explode = false;
}
```

On le voit ici on a juste à rajouter un booléen qui va dire si ça doit exploser ou non en plus du compteur classique. Pour modifier ce booléen on ajoute un composant déclencheur aux explosions ce qui va permettre de savoir si la bombe et sur une explosion et ainsi déclencher prématurément l'explosion de la bombe

Bombes solides :

Encore une mécanique phare des jeux de cette licence, les bombes qui agissent comme des murs, et oui sinon comment pourrions-nous bloquer nos adversaires pour les faire exploser. Pour cette mécanique c'était un petit peu plus dur car juste ajouter une composante solide aux bombes et une fausse bonne idée, pourquoi ? Pour la simple et bonne raison que si la bombe était instantanément solide et bien le joueur se ferait expulser en dehors de la bombe. Donc pour il faut que la bombe n'ait pas de corps solide au moment de la pose mais dès que le joueur ne touche plus la bombe elle devient solide et agit comme un mur.

```
private void OnTriggerEnter2D(Collider2D other)
{
    if (other.gameObject.tag == "Player" && IsIn)
    {
        collide.enabled = false;
        IsIn = false;
    }
    else
    {
        collide.enabled = true;
    }
}
```

Pour faire ça c'est à peu près pareil que pour les chaînes d'explosions à savoir on check les collisions, même si là c'est un déclencheur qui n'est pas tangible au moment de la pose mais qui le devient dès que plus aucun joueur n'est dans la bombe.

Réduction de la puissance des bombes :

Pour l'instant les bombes étaient un petit peu trop puissant, c'est à dire que s'il y avait un obstacle à détruire elle le détruisait mais continuer quand même après ce qui est une mauvaise chose, car on pourrait se croire en sécurité derrière des blocs mais la bombe explose au travers et nous tue. Pour remédier à cela :

```
bool ExplodeCell (Vector3Int cell)
{
    Tile tile = tilemap.GetTile<Tile>(cell);

    if (tile == wallTile)
    {
        return false;
    }

    if (tile == destructibleTile)
    {
        tilemap.SetTile(cell, null);
        Vector3 pos1 = tilemap.GetCellCenterWorld(cell);
        PhotonNetwork.Instantiate("Explosion", pos1, Quaternion.identity);
        return false;
    }

    Vector3 pos = tilemap.GetCellCenterWorld(cell);
    PhotonNetwork.Instantiate("Explosion", pos, Quaternion.identity);

    return true;
}
```

Pas besoin de tout comprendre ici juste à savoir que si c'est la tile est une destructibletile alors on return false ce qui bloqué la propagation des explosions.D

Ajout des items :

Pour cette soutenance, nous avons aussi implémenté nos premiers items.

Dans notre jeu il y a deux grandes catégories d'Items :

- _Les malus (qui sont temporaires et qui ont tous la même apparence)
- _Les bonus (qui sont soit permanent et concerne donc des améliorations de stats (comme la vitesse) ou sont temporaires)

Pour l'instant il existe 5 items différents fonctionnels, il y a 4 malus et 1 bonus permanent :

_Malus inversion : qui inverse les commandes du joueur pendant 5 secondes (gauche devient droite, haut devient bas etc....)

_Malus ralentissement : qui réduit de beaucoup la vitesse du joueur qui le ramasse pendant 5 secondes.

_Malus vitesse max : qui augmente énormément la vitesse du joueur qui le ramasse pendant 3 secondes.

_Malus No bomb : qui désactive le fait de pouvoir des bombes pendant 5 secondes, donc pendant 5 secondes le joueur ne pourra plus poser aucune bombe.

_Bonus permanent de vitesse : qui augmente un peu la vitesse du joueur qui le ramasse de manière définitive.

Tous ces malus ont la même apparence (une pastille rouge) et le bonus est pour l'instant sous la forme d'une pastille jaune. Mais d'ici la prochaine soutenance ces pastilles seront changées par une meilleure icône, cependant les tous les malus devront toujours avoir la même apparence pour qu'il soit impossible de savoir sur quel malus on va tomber quand on en ramasse un.

Un des buts qui n'est pas encore pas encore rempli va être de faire en sorte que quand on casse un mur, un item tombe à la place du mur, ceci pour tous les joueurs présents dans la partie qui devront voir le même item car l'item qui remplacera le bloc est aléatoire. Il y aura une chance sur deux que l'item soit un bonus ou un malus puis l'item sera choisi aléatoirement de manière équiprobable parmi tous ceux de sa classe (les probabilités des items risquent d'être légèrement modifiées pour un souci d'équilibrage).

Avancement du site :

A la dernière soutenance, on avait juste un site avec 4 pages qui était en ligne mais presque intégralement vide à l'adresse :

<https://yeckes.github.io/Ceas.website/Index.html>

Depuis, le site a été rempli avec les informations concernant notre jeu et notre avancement.

Il y a maintenant sur la page d'accueil une description des membres du groupe où il est notamment spécifié leurs rôles selon notre répartition des tâches.

Sur la page “présentation du jeu” qui ne possédait avant qu’une brève description du fonctionnement prévu pour notre jeu. Il y a maintenant une description complète de notre avancement selon le travail qui a été fait aux dates des anciennes soutenances, ainsi qu’une description de ce qui est prévu d’avoir accompli pour la prochaine et dernière soutenance.

En bas de cette page se trouve aussi des liens qui permettent d’accéder à nos rapports de soutenance et cahier des charges sous format pdf.

Un dossier a donc été ajouté au site contenant tous ces pdf.

Sur la page “Logiciels utilisés” où se trouvait avant seulement une image avec quelques logos de logiciels que nous avons utilisé, ce trouve désormais une liste avec les logos de tous les logiciels que nous avons utilisé jusqu’ici ainsi qu’une description de ces logiciels et de leurs utilités. De plus un lien vers le site principal de ses logiciels est attaché à chacun des logos, donc si vous cliquer sur le logo d’un de ces logiciels vous allez arriver sur le site de celui-ci.

Pour finir la page “Installer” est la seule page qui n’as pas subi de grosse mise à jour du fait que nous n’ayons pas implémenter d’installateur pour l’instant.

Notre site est donc maintenant rempli d’information mais doit donc être régulièrement mis à jour, cependant le site a donc bien avancé et a par les mises à jour concernant les données qu’il contient qui doivent être faites au fur et à mesure le seul autre objectif serait d’implémenter un design un peu plus agréable à regarder avec notamment plus d’images.

Objectif :

Objectif 2ème soutenance (28 avril 2022) :

Pour cette seconde soutenance l'objectif principale est d'avoir quelque chose de jouable sur lequel on peut lancer une partie, ce ne sera évidemment pas le rendu final, il manquera sûrement de nombreuses fonctionnalités. Les fonctionnalités attendues sont :

- Avoir une ou deux cartes avec le style graphique souhaité qui sont terminées.
- Le cœur du jeu est fini, avec toutes les différentes idées proposer au début du projet, quelque power-up seront ajoutés dans les parties.
- Le multijoueur doit être stable et fonctionnel.
- Le menu/option doit être avancé notamment sur la partie configuration des touches, gestion du volume et modification de la résolution.
- Une musique devrez aussi être présente dans le menu ou les parties.
- L'intelligence artificielle devrez être commencé bien que très limité dans ses choix et actions.
- Avoir un site présentable avec plusieurs pages et une mise en page correcte

Objectif 3ème soutenance (semaine du 6 juin 2022) :

Si tout c'est bien passer à cette soutenance, le jeu devrait être fini. Il doit comporter un menu/option fonctionnel avec un bouton jouer qui vous connecte directement au serveur de Photon pour jouer en multijoueur comme en solo (pour jouer en solo il faudra quand même être connecté à internet). Le multijoueur sera fini, on pourra se connecter pour jouer à 20 joueurs maximum (limite imposée pas photon) et choisir d'inclure ou non une ou plusieurs intelligences artificielles. Le gameplay sera fini avec tous les power-up et l'IA doit avoir une difficulté faible. La musique et les bruitages doivent aussi être fini.

Comme on peut le voir juste au-dessus il y a les objectifs de cette soutenance, voyons voir jusqu'où nous avons réussi à aller.

Pour le premier point c'est globalement une réussite le style graphique est bon, et il y a une carte de disponible pour pouvoir jouer.

Pour le second c'est encore une fois accompli, toutes les mécaniques du jeu sont présente, notamment l'interaction entre plusieurs bombes, le fait que les bombes deviennent solides après les avoir posées ou encore des choses très simple comme le fait de mourir au contact d'une explosion. Donc pour le cœur du jeu c'est réussi.

Pour le menu/option c'est réussi bien qu'il manque des options spécifiques comme modifier la résolution du jeu, ou encore attribuer des touches.

La musique aussi est présente avec en plus de ça divers bruitages comme le son d'explosion des bombes, ou encore les bruit quand on clique sur un bouton dans les menus.

Malheureusement l'intelligence artificielle est le plus gros défaut de cette seconde soutenance, nous n'avons pas réussi à créer un début d'IA, par manque de temps et surtout car il y avait des problèmes beaucoup plus importants à régler avant. Mais de toute façon le cœur du jeu ne réside clairement pas dans son IA ce qui n'est donc pas très grave dans notre cas.

Et pour finir le site, il n'a pas été modifier visuellement mais a surtout été bien fourni en termes de contenu, il y a beaucoup plus d'informations concernant le jeu, l'équipe ou même le précédent cahier des charges ou planning.