ZINE-EDDINE Selma
PAN Sophie

Rapport

SPACE INVADER

À QUOI RESSEMBLE LE JEU

Le menu affiche le nom du jeu et le niveau. Il faut appuyer sur espace pour commencer le jeu

JEU

Touches clavier:

<- et -> pour déplacer le joueur

<espace> pour tirer

<p> pour pause

COMPOSITION

Les vaisseaux ennemies sont organisés en bloc. Ils descendent en se déplaçant de gauche à droite, puis de droite à gauche.
Le Joueur doit tirer sur les vaisseaux ennemies tout en évitant leurs tirs.
Des bunkers immobiles peuvent protéger le joueur, mais tout comme lui, ils n'ont que trois vies!

WIN/LOSE

Le joueur perd si les monstres atteignent la même hauteur que lui ou s'il n'a plus de vies. Si les monstres sont tous décimés alors le joueur gagne.

STRUCTURE DE PROGRAMME : CLASSE MÈRE ET FILLE

GAME

- S'occupe des différents états du jeu : Si il est en pause ou pas, si le joueur a perdu ou pas. Gère également le niveau de difficulté.
- Ajoute les éléments en début de partie

GAMEOBJECT

- Définis les positions des objets : x et y
- Définis le nombre de vie : vie
- Définis Collision()

VAISSEAU

- Construis les vaisseaux

JOUEUR

- Peut se déplacer horizontalement
- Peut activer un tir

ENEMY

- Définis différents types d'ennemies
- Peut activer un tir

BLOC ENEMIES

- Sélectionne aléatoirement des lignes d'Enemy pour construire un bloc d'ennemies.
- Gère l'augmentation de la difficulté en fonction du niveau
- S'auto-déplace dans un mouvement bien précis

BUNKER

- Gère les bunkers

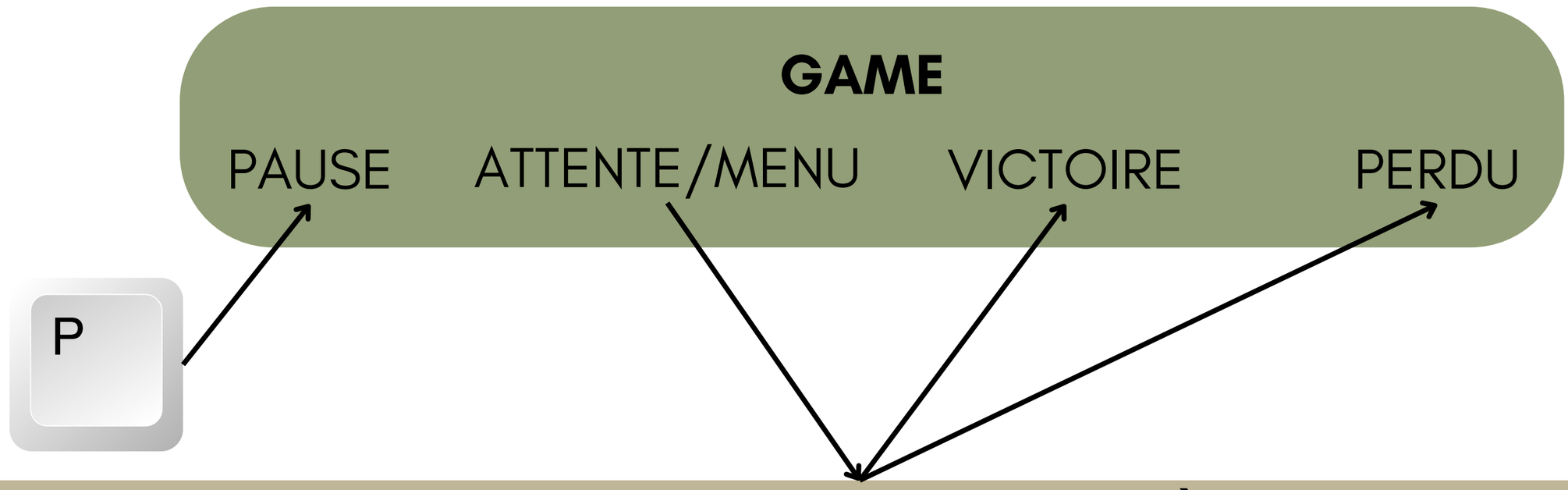
MISSILE

- Se déplace de bas en haut ou de haut en bas en fonction de son type.
- Se supprime s'il entre en collision ou sort de l'écran

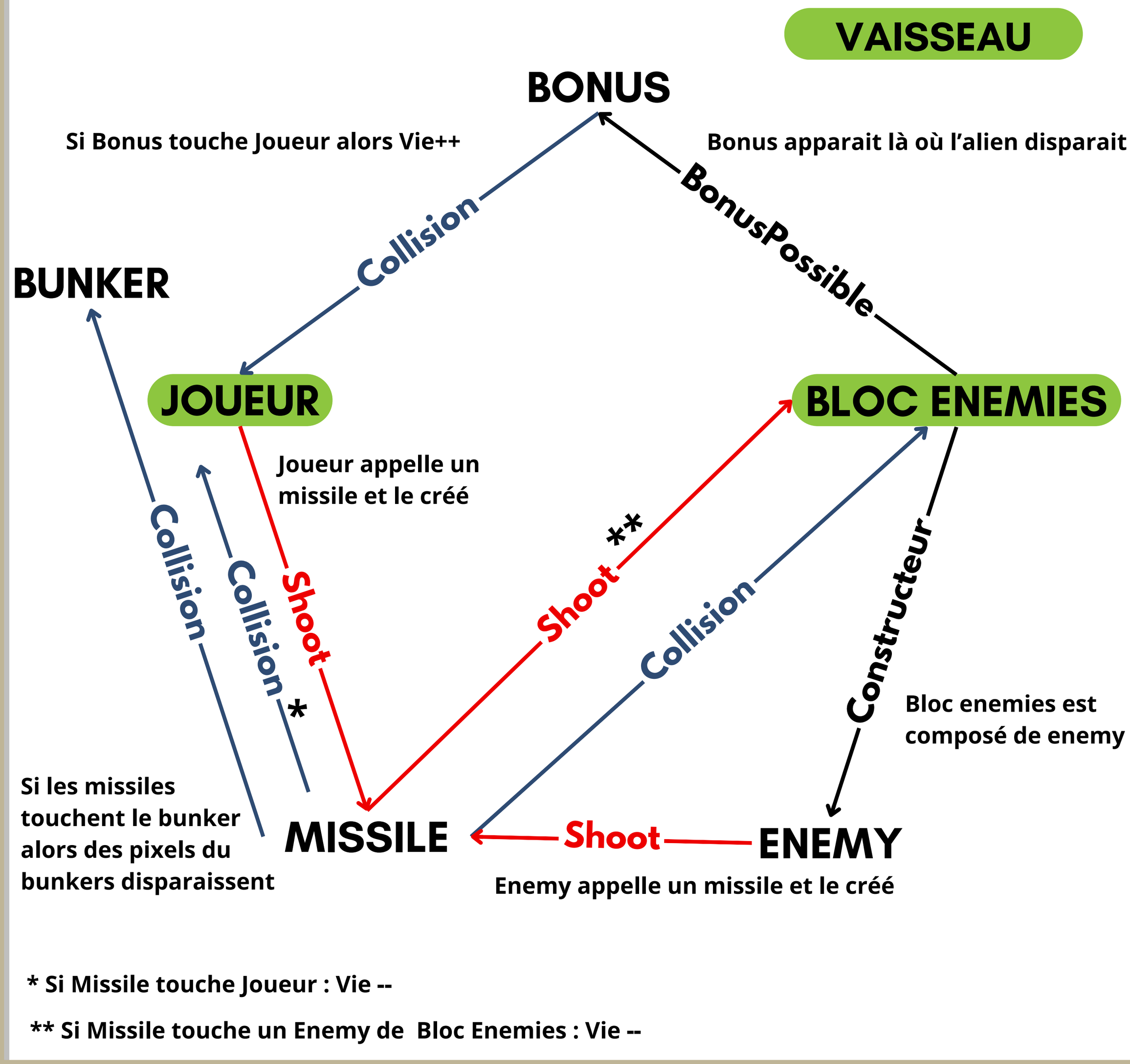
BONUS

- Peut apparaître lors de la mort d'un ennemie, se déplace alors de bas en haut.
- Permet au joueur d'acquérir une vie supplémentaire

FONCTIONNEMENT DU PROGRAMME : LES APPELS DE FONCTIONS ENTRE CLASSE



GAMEOBJECT : CLASSE MÈRE



* Si Missile touche Joueur : Vie --

** Si Missile touche un Enemy de Bloc Enemies : Vie --

NOS PROBLÈMES RENCONTRÉS ET NOS SOLUTIONS APPORTÉES

LE JOUEUR GROSSIT QUAND IL SE DEPLACE

Problème : Quand le joueur se déplace de droite à gauche, il grossit.
AddNewGameObject() était dans le update, ainsi un joueur était ajouté à chaque update.
Pour résoudre le problème, nous avons ajouté le joueur dans l'état Attente, quand la touche espace est pressé, pour qu'il ne soit ajouté qu'une fois au jeu.

```
if (keyPressed.Contains(Keys.Space))
{

    AddNewGameObject(newVague);
    AddNewGameObject(player);
    AddNewGameObject(bunkermilieu);
    AddNewGameObject(bunkergauche);
    AddNewGameObject(bunkerdroite);



    //gamestart = true;
    state = Gamestate.Play;

    // release key space (no autofire)
    ReleaseKey(Keys.Space);


}
```

LES ENNEMIES QUI TOMBENT DIRECTEMENT

Problème : Le Bloc Ennemie arrive au bord et tombe directement tout en bas au lieu de descendre et de changer de direction car le condition pour descendre était toujours valide.

- Ajouter un int direction qui permet de déterminer le sens et de ne pas rester coincé dans la condition qui faisait descendre

```
if (enmy.X <= 0)
{
    direction = 1;
    descend();
}
if (enmy.X >= gameInstance.gameSize.Width-50)
{
    direction = -1;
    descend();
}
```

LES COLLISIONS BUNKER

Problème: Lors d'une collision entre un missile et le bunker, le bunker est supprimé directement alors qu'il est censé avoir 3 vies.
Le problème vient du fait que même si le missile était supprimé du game, on ne l'avait pas supprimé de la liste des missiles et il faisait donc des dégâts à chaque update même si à l'écran, il avait bien disparu.
Pour corriger le problème, nous avons donc supprimé de la liste des missiles les missiles "dead".

```
foreach (GameObject gO in gameObjects)
{
    foreach (Missile m in missiles)
    {
        if (gO.Collision(m, image, this))
        {
            if (newVague.BonusPossible() != null)
            {
                float bx = newVague.BonusPossible()[1];
                float by = newVague.BonusPossible()[2];
                if (boost == null || !boost.IsAlive()) boost = CreeBonus(bx, by, newVague);
                if (boost != null && boost.IsAlive()) AddNewGameObject(boost);
            }

            break;
        }


    }
    missiles.RemoveAll(m => !m.Alive);
}
```

NOS PROBLÈMES RENCONTRÉS ET NOS SOLUTIONS APPORTÉES

LES ENNEMIES QUI TIRENT DEUX MISSILES EN MEME TEMPS

Problème : La condition pour activer le tir donne le temps au jeu de tirer deux fois.
Pour résoudre le problème, nous avons ajouté un booléen qui devient “true” après un certain délai passé, le temps que la condition ne soit plus rempli.

```
peutTirer = false;
tir = m;
// Planifiez la réinitialisation de l'état du tir après un certain délai (par exemple, 1 seconde)
Task.Delay(1000).ContinueWith(t => peutTirer = true);
```

LES COLLISIONS MISSILES ENNEMIES

Problème: Les ennemies se tire dessus car les collisions testent pour chaque objet et pour chaque missile.
Pour résoudre cela, nous avons crée un type dans missile. Les tirs des ennemies sont de types 1 alors que les tirs du joueur sont de type 0. Si le tir est de type 1, la fonction Collision() de BlocEnemies retourne qu’il n’y a pas de collision.

Mise en transparence des pixels lors des collisions

Problème : Seulement un pixel devenait transparent lors de la collision.
En effet, seul le pixel qui subissait la collision devenait transparent. Pour régler son problème, nous avons fait en sorte que tous les pixels sur toute la ligne sur une certaine épaisseur deviennent transparent.

```
for (int epaisseurX = -1; epaisseurX < 2; epaisseurX++)
{
    for (int epaisseurY = -15; epaisseurY < 15; epaisseurY++)
    {

        if (positionY - epaisseurY >= 0 && positionX + epaisseurX < Image.Width
            && positionX + epaisseurX >= 0 && positionY - epaisseurY < Image.Height)
        {
            Image.SetPixel(positionX + epaisseurX, positionY - epaisseurY, Color.Transparent);
        }
    }
}
```

Addons

-Ajout de bonus qui permet au joueur de récupérer une vie
-Ajout de niveaux. La difficulté augmente à chaque nouveau niveau (les ennemies sont plus rapide et leur probabilité de tirer est plus élevée)
-Un son est déclenché quand le joueur tir
-Ajout de couleurs au jeu