


 IUT ANNECY UNIVERSITÉ SAVOIE MONT BLANC	NOM DU GROUPE : INFO-12 RABEHI Milhane, TSANEV Pavel, BIGOT Gassien	Saé S1_01_02 Développement en C# Librairie Monogame	
	Escape Game		

Table des matières

1	PRESENTATION	2
1.1	DESCRIPTION GENERALE	2
1.2	REGLES DU JEU	2
1.3	CINEMATIQUE DES ECRANS	3
2	CONCEPTION – DIAGRAMME DE CLASSE	5
2.1	PRESENTATION GENERALE	5
2.2	PRESENTATION DETAILLEE GAME1	6
2.3	PRESENTATION ...	8
3	PARTIE ALGORITHMIE	11
4	CONCEPTION GRAPHIQUE	11
5	CAHIER DE RECETTES	12
5.1	TESTS DE VALIDATION	12
5.2	TESTS DE PERFORMANCE	13

	<p>NOM DU GROUPE : INFO-12 RABEHI Milhane, TSANEV Pavel, BIGOT Gassien</p>	<p>Saé S1_01_02 Développement en C# Librairie Monogame</p>	
	<p>Escape Game</p>		

1 Présentation

1.1 Description générale

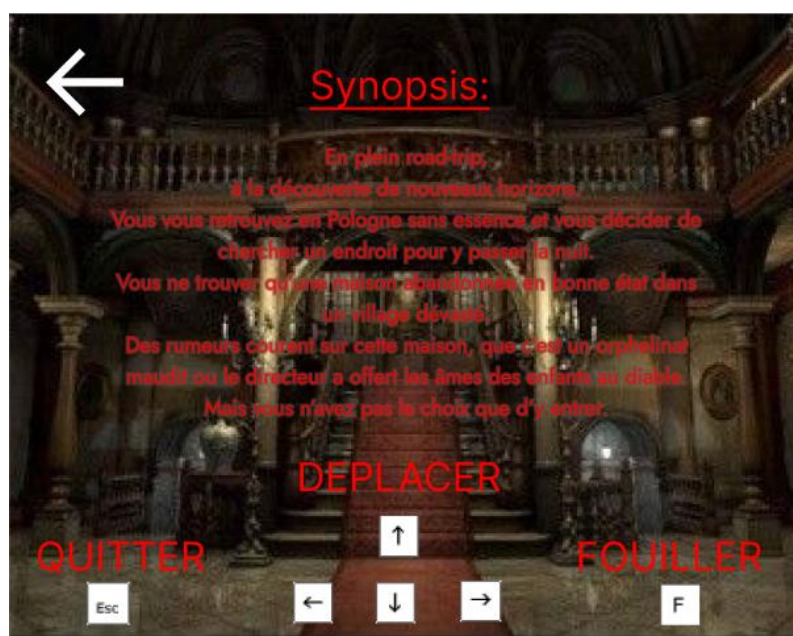
Le jeu se base sur un simple jeu d'Escape Game, le but du joueur est de sortir du manoir hanté en un temps imparti, vivant en répondant juste aux énigmes du directeur.

1.2 Règles du jeu

Les règles de jeu consistent à s'enfuir des différentes pièces à temps en sans se tromper sur une des questions du directeur.

Synopsis :

En plein road-trip,
A la découverte de nouveaux horizons,
Vous vous retrouvez en Pologne sans essence et vous décidez de chercher un endroit pour y passer la nuit.
Vous ne trouvez qu'une maison abandonnée en bonne état dans un village dévasté.
Des rumeurs courent sur cette maison, que c'est un orphelinat maudit ou le directeur a offert les âmes des enfants au diable.
Mais vous n'avez pas le choix que d'y entrer.



Les flèches d'ordinateur : Haut pour se diriger vers le nord
Bas pour se diriger vers le sud
Gauche pour se diriger vers l'ouest
Droite pour se diriger vers l'est

F : pour fouiller

Esc : Quitter



Réponse des énigmes : V pour dire que c'est vrai
M pour dire que c'est faux.

O : Options

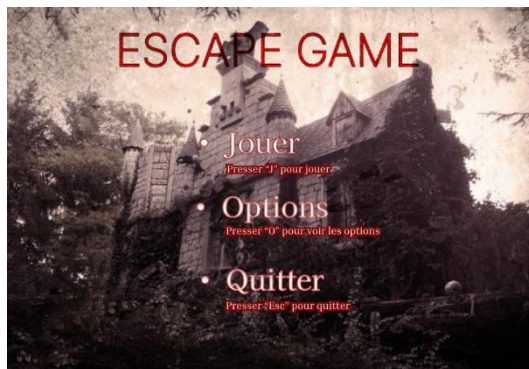
Flèche « retour » : accéder au menu en étant sur l'écran option.

Espace : Pause

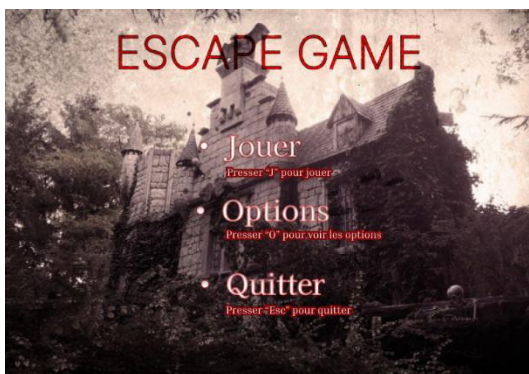
Entrer : Reprendre

	NOM DU GROUPE : INFO-12 RABEHI Milhane, TSANEV Pavel, BIGOT Gassien	Saé S1_01_02 Développement en C# Librairie Monogame	
	Escape Game		

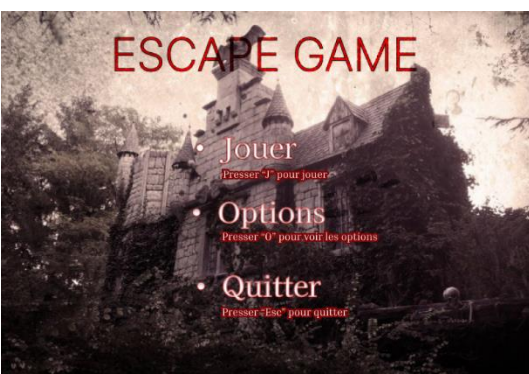
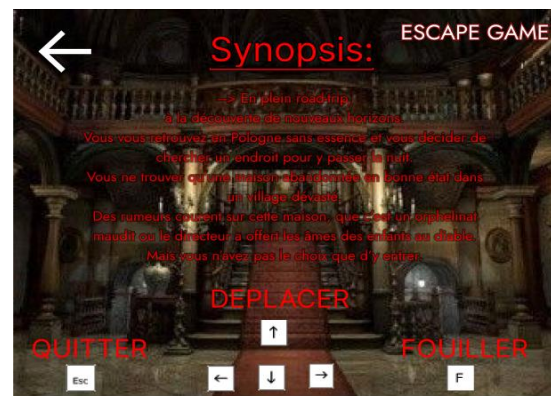
1.3 Cinématique des écrans



Affichage de la MapExt
Quand appuyer sur J





Affichage du menu
Si appuyer sur O



Quitte le jeu
Si appuyer sur Q

```
Quit
AL lib: (EE) alc_cleanup: 1 device not closed

Sortie de C:\Users\tsanevp\source\repos\SAE-1.01-DEV\Test\Test\bin\Debug\net6.0\Test.exe (processus 20704). Code : 0.
Pour fermer automatiquement la console quand le débogage s'arrête, activez Outils->Options->Débogage->Fermer automatique
ment la console à l'arrêt du débogage.
Appuyez sur une touche pour fermer cette fenêtre. . .
```

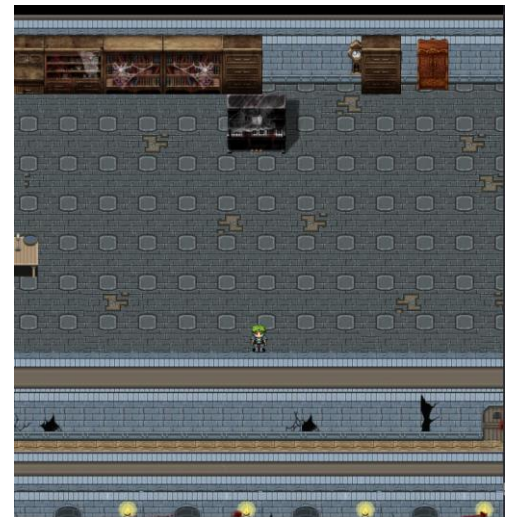

	<p>NOM DU GROUPE : INFO-12 RABEHI Milhane, TSANEV Pavel, BIGOT Gassien</p>	<p>Saé S1_01_02 Développement en C# Librairie Monogame</p>	
	<p>Escape Game</p>		





Affichage de la 2^e Map
quand on passe par la
porte



Affichage de la 3^e Map
Quand on passe par la
porte qui est au fond



	NOM DU GROUPE : INFO-12 RABEHI Milhane, TSANEV Pavel, BIGOT Gassien	Saé S1_01_02 Développement en C# Librairie Monogame	
	Escape Game		

2 Conception – Diagramme de classe



2.1 Présentation générale



Game1 : C'est le « chef d'orchestre » : Elle contrôle l'état de la map, la possibilité de revenir de la map extérieur (MapExt) au menu de base (MenuIntro), mais dès qu'on rentre dans une map intérieur cette fonctionnalité est bloquée, elle gère aussi les chargements/changements de map entre elle.

Perso : C'est une classe Sprite où l'on génère les déplacements ainsi que les collisions, le personnage en lui-même pour le faire déplacer avec les flèches. Cette classe est créée afin de pouvoir contrôler le personnage principal avec tous les déplacements, collisions etc. C'est une classe qui permet la factorisation du code.

MenuIntro : C'est une classe où l'on charge le menu, notre écran principal. Elle est primordiale pour pouvoir lancer le jeu, voir les règles et les touches puis quitter.

	<p>NOM DU GROUPE : INFO-12 RABEHI Milhane, TSANEV Pavel, BIGOT Gassien</p>	<p>Saé S1_01_02 Développement en C# Librairie Monogame</p>	
	<p>Escape Game</p>		

MenuOptions : C'est la classe où l'on charge l'écran avec les options et les règles à l'intérieur. C'est une classe qui améliore le code en lançant un fond avec les règles et les différentes touches pour pouvoir jouer.

MapExt : C'est la classe pour la génération du personnage sur notre 1^{ère} map qui mène à la Map2. Ceci est notre Map mère. C'est d'ici que commence notre histoire.

Map2 : C'est la classe pour la génération du personnage sur notre 2^{ème} map qui mène à la Map3. Une map faite pour améliorer le code afin de pouvoir commencer les « énigmes ».

Map3 : C'est la classe pour la génération du personnage sur notre 3^{ème} map qui mène à la fin du jeu. La map 3 est aussi une amélioration du code. Ceci est la Map finale où si on finit, cela nous lance l'écran victoire.

Perdu : est le chargement de texture (fond) qui définit la défaite du jeu afin de rejouer directement, revenir sur le menu d'accueil, ou bien quitter. Une classe faite pour améliorer en plus.

Victoire : C'est la classe qui permet le chargement d'une texture(fond) si vous réussissez à gagner le jeu. La classe a été conçue si le personnage arrive à sortir du manoir hanté.

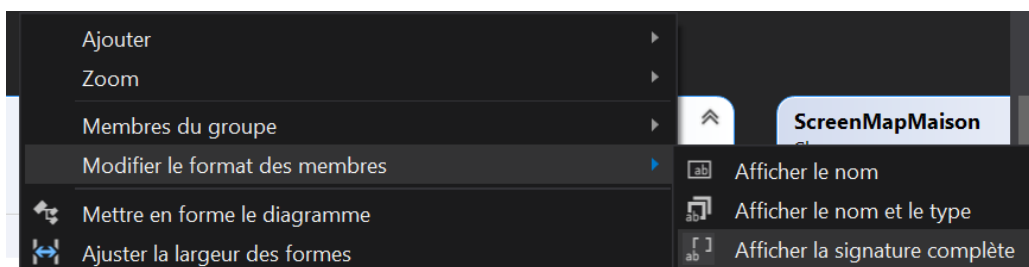
Camera : C'est la classe qui permet le chargement de la caméra afin de se déplacer sur la map. Une classe en option qui permet juste l'amélioration du jeu. Elle se lance à partir de la MapExt.

Monstre : C'est la classe d'un personnage qui se dirige à partir de la map extérieur à la map intérieur. Une classe qui est faite simplement pour rajouter des frayeurs et un peu de suspense.

2.2 Présentation détaillée Game1



Puis faites une impression écran détaillée de chaque classe avec leur signatures accompagnée d'explications un peu plus détaillées.

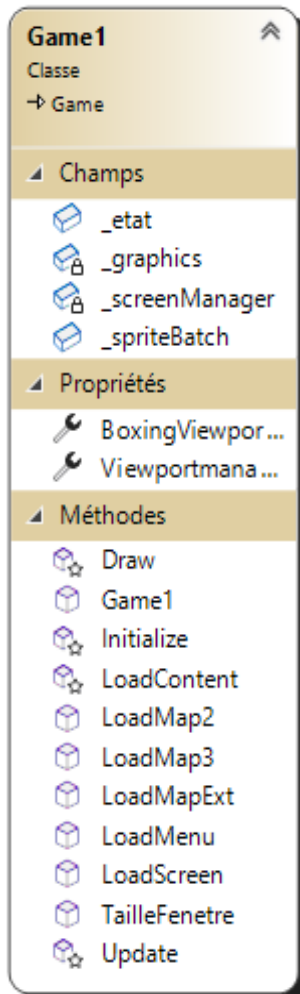
Aide : clic droit sur le fond de fenêtre vide puis Menu ci-dessous :





Game1 : instancie les 3 états, les graphics, le screenManager et les différents SpriteBatch. Game1 charge toutes les Maps et Menu. Elle contient en tout 4 champs, 2 propriétés et 11 méthodes.

- `_screenManager` : c'est un objet de classe ScreenManager qui sert à gérer les différents chargement des GameScreen

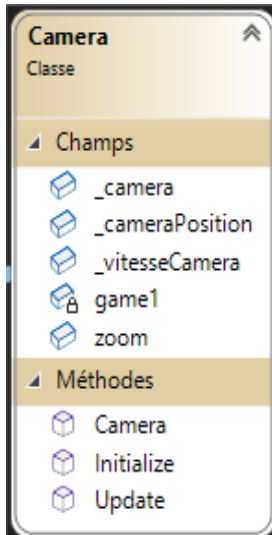
 <div>IUT ANNECY UNIVERSITÉ SAVOIE MONT BLANC</div>	NOM DU GROUPE : INFO-12 RABEHI Milhane, TSANEV Pavel, BIGOT Gassien	Saé S1_01_02 Développement en C# Librairie Monogame	
	Escape Game		



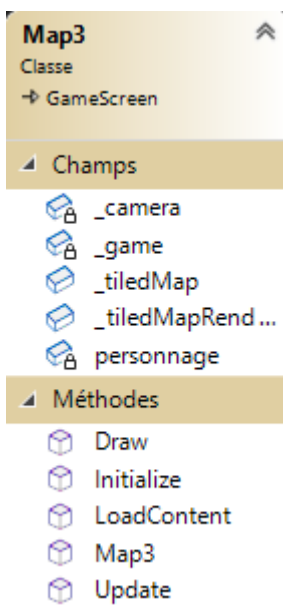
- `_graphics` : c'est un objet de classe `GraphicsDeviceManager`, il sert à gérer les périphériques graphiques. Il permet de définir les caractéristiques de l'affichage
- `_etat` : c'est un objet de classe `Etats`, il sert à décrire l'interface utilisateur ou les actions que le joueur peut effectuer à un moment donné.
- `_spriteBatch` : c'est un objet de classe `SpriteBatch`, il permet de dessiner des sprites(images) à l'écran.

	NOM DU GROUPE : INFO-12 RABEHI Milhane, TSANEV Pavel, BIGOT Gassien	Saé S1_01_02 Développement en C# Librairie Monogame	
	Escape Game		



2.3 Présentation ...

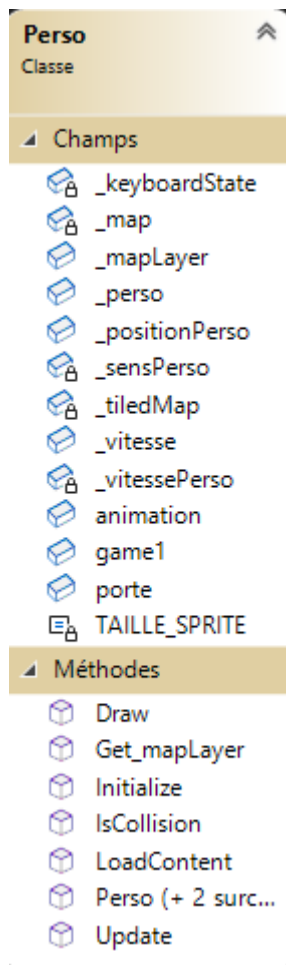


- `_camera`: c'est un objet de classe `OrthographicCamera`, il sert à gérer les paramètres graphiques. Il permet de définir les caractéristiques de l'affichage de celle-ci sur le Personnage.
- `_cameraPosition` : c'est un objet de classe `Vector2F`, celui-ci définit le déplacement qui restera au-dessus du personnage.
- `_vitesseCamera` : c'est un `Int`, il définit la vitesse que la camera doit avoir (celle-ci doit avoir la même vitesse que celui du personnage).
- `game1` : c'est un objet de la classe `Game1`, celui-ci sert à faire la liaison entre la classe `Camera` et la classe `Game1`.
- `zoom` : c'est un `float` qui définit si la caméra est plus ou moins proche du personnage.





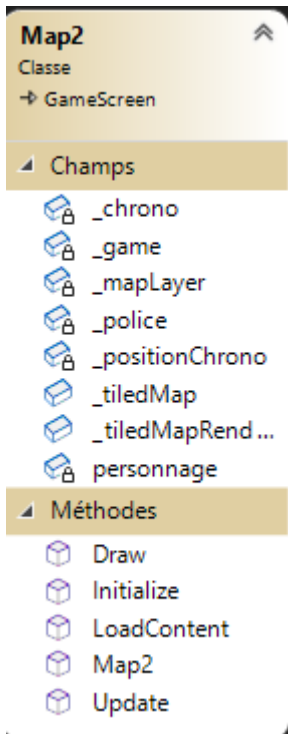
- `_camera`: c'est un objet de classe `Camera`, celui-ci sert à faire la liaison entre la classe `Camera` et la classe `Map3`.
- `_game` : c'est un objet de la classe `Game1`, celui-ci sert à faire la liaison entre la classe `Map3` et la classe `Game1`.
- `_tiledMap` : de la classe `TiledMap`, permet d'initialiser le tiled en tant que map avec des collisions prédéfinis.
- `_tiledMapRenderer` : c'est un objet de la classe `TiledMapRenderer` , Permet de régler la map en y ajoutant les collision a la bonne map a l'aide d'un petit réglage.
- `personnage` : c'est un objet de la classe `Perso` qui permet de faire le lien entre la classe `Perso` et la classe `Map3`.

	<p>NOM DU GROUPE : INFO-12 RABEHI Milhane, TSANEV Pavel, BIGOT Gassien</p>	<p>Saé S1_01_02 Développement en C# Librairie Monogame</p>	
	<p>Escape Game</p>		

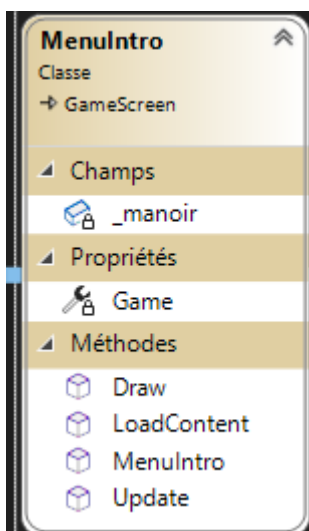


- `_keyboardState`: c'est un objet de classe `KeyboardState`, celui-ci sert à faire la liaison entre les touches du clavier et le jeu.
- `_map`: c'est un objet de la classe `MapExt`, celui-ci sert à faire la liaison entre la classe `MapExt` et la classe `Perso`.
- `_mapLayer`: de la classe `TiledMapTileLayer`, permet d'initialiser le tiled en tant que map avec des collisions.
- `_perso`: c'est un objet de la classe `AnimatedSprite` qui permet de définir le personnage comme étant une collision avec laquelle l'utilisateur joue.
- `_positionPerso`: c'est un objet de la classe `Vector2f` qui permet d'obtenir une direction du personnage afin que celui-ci puisse bouger.
- `_sensPerso`: c'est un objet de classe `Vector2f` qui permet d'avoir un sens pour que le personnage se tourne. Celui-ci est complémentaire à `_positionPerso`.
- `_tiledMap`: de la classe `TiledMap`, permet d'initialiser le tiled en tant que map avec les collisions prédéfinis.
- `_vitesse`: est un float qui définit la Vitesse dont va le jeu (permet d'accélérer tout sauf le temp qui défile).
- `_vitessePerso`: est un float qui définit seulement la vitesse du personnage.
- `_animation`: est un string permettant d'initier les animations (droite gauche haut bas) du personnage.
- `game1`: c'est un objet de la classe `Game1`, celui-ci sert à faire la liaison entre la classe `perso` et la classe `Game1`.
- `porte`: est un booléen permettant de savoir si la hitbox du personnage touche la hitbox d'une porte.



 IUT ANNECY UNIVERSITÉ SAVOIE MONT BLANC	NOM DU GROUPE : INFO-12 RABEHI Milhane, TSANEV Pavel, BIGOT Gassien	Saé S1_01_02 Développement en C# Librairie Monogame	
	Escape Game		



- `_chrono` : est un float qui permet d'avoir un chrono qui s'affiche afin de connaître le temps restant.
- `_game` : c'est un objet de la classe `Game1`, celui-ci sert à faire la liaison entre la classe `Map2` et la classe `Game1`.
- `_mapLayer` : de la classe `TiledMapTileLayer`, permet d'initialiser le tiled en tant que map avec ses collisions.
- `_police` : de la classe `SpriteFont` permet de changer le style de la police du chrono.
- `_positionChrono` : c'est un objet de la classe `objet` permet de positionner le timer (chrono) a une position precise sur l'écran.
- `_tiledMap` : de la classe `TiledMap`, permet d'initialiser le tiled en tant que map avec les collisions prédéfinis.
- `_tiledMapRenderer` : c'est un objet de la classe `TiledMapRenderer`, Permet de régler la map en y ajoutant les collision a la bonne map a l'aide d'un petit réglage.
- `Personnage` : est un objet de la classe `Perso` qui sert a faire la liaison entre `Perso` et `Map2`.



- `_manoir` : c'est un objet de la classe `Texture2D`, sert a créer une texture dans la taille choisie

	NOM DU GROUPE : INFO-12 RABEHI Milhane, TSANEV Pavel, BIGOT Gassien	Saé S1_01_02 Développement en C# Librairie Monogame	
	Escape Game		

3 Partie Algorithmie

```

if (game1._etat == 0)
{
    this._mapLayer = MapExt._tildeMap.GetLayer<TiledMapTileLayer>("obstacles");
    if (_mapLayer.GetTile(x, y).GlobalIdentifier == 2153 || _mapLayer.GetTile(x, y).GlobalIdentifier == 2152)
    {
        game1._etat += 1;
        game1.LoadMap2();
        Perso._positionPerso = new Vector2(400, 959);
        //Passer de mapEXT a map2 !
        this._mapLayer = Map2._tildeMap.GetLayer<TiledMapTileLayer>("obstaclesmap2");
    }
}
if (game1._etat == 1)
{
    this._mapLayer = Map2._tildeMap.GetLayer<TiledMapTileLayer>("obstaclesmap2");
    if (_mapLayer.GetTile(x, y).GlobalIdentifier == 1942)
    {
        Perso._positionPerso = new Vector2(1765, 121);
        game1._etat += 1;
        game1.LoadMap3();
        //Passer de map2 a map3 !



        this._mapLayer = Map3._tildeMap.GetLayer<TiledMapTileLayer>("obstaclesMap3");
    }
}
//Retour en arriere-

```

Cet extrait nous permet de changer de map par rapport aux collisions des tuiles qui sont prédéfini grâce à Tiled. Ainsi que la position du personnage lors du changement de scene.

4 Conception graphique



Les maps sont des créations sorties de notre imaginations, celle-ci ont été créés de toutes pièces. Les tuiles des maps proviennent de pinterest. Ainsi que de site comme <https://itch.io/>. On a pris des sons libres de droits sur le site <https://pixabay.com/music/>. Les maps sont faite par nous juste on a pris différents tileset pour pouvoir les construire.

	NOM DU GROUPE : INFO-12 RABEHI Milhane, TSANEV Pavel, BIGOT Gassien	Saé S1_01_02 Développement en C# Librairie Monogame	
	Escape Game		

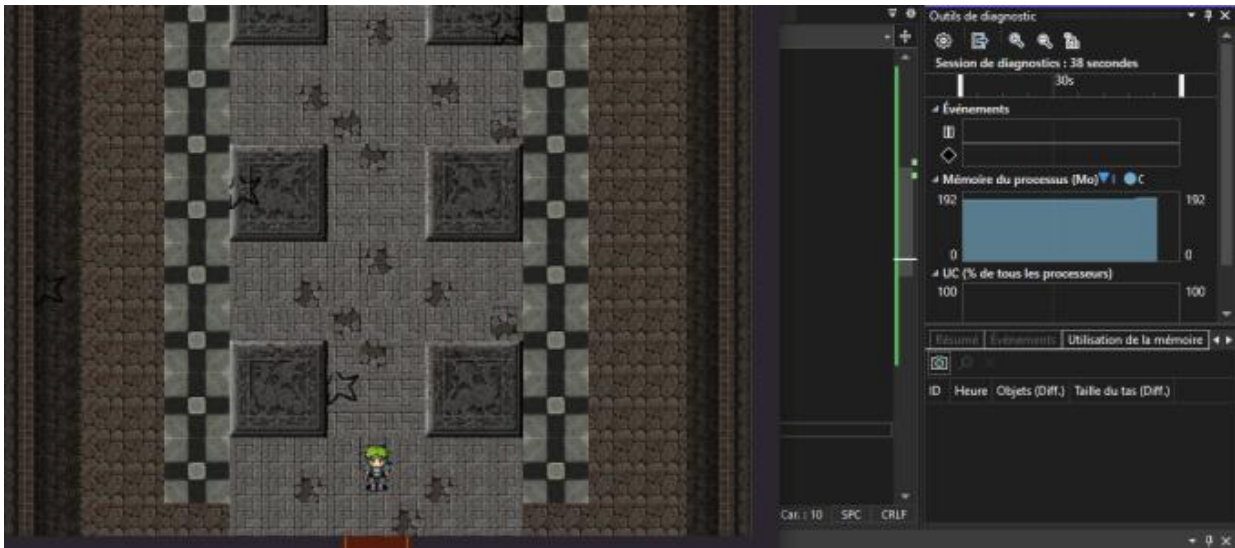
5 Cahier de recettes

5.1 Tests de validation

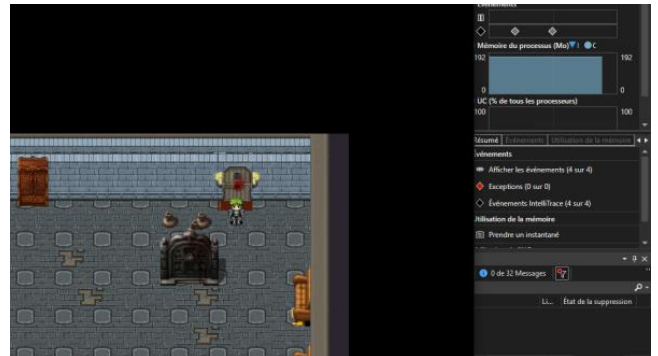
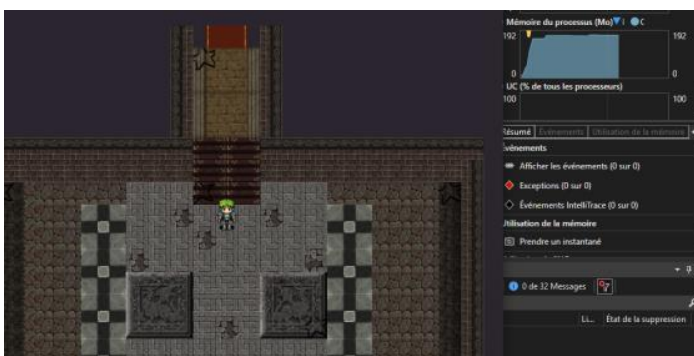
Nom	Fonctionnalité	Etat
RABEHI	Graphiste (design), Enigme, Classe Game over, Initialisation Perso, Menu, Trailer, (debut (1)) Compte rendu	Ok/ inachevé/Ok/Ok/Ok/ inachevé/Ok
BIGOT	Changement de Map, création de map(Tiled), Chargement Map 3, caméra, Classe Map3, (finalisation) Classe Map2, Gestion des Scènes, Compte rendu (2.3;5.2), (Finalisation) Classe MapExt	Ok/Ok/Ok/Ok/Ok/Ok/Ok/Ok
TSANEV	Classe Perso, Déplacements, Collisions, (début) Classe Map2, Musique, Compte rendu(1.3 ;2.1 ;2.2), (Grandes lignes) Classe MapExt, Classe Game1, Gestion des Scènes	Ok/Ok/Ok/Ok/Ok/Ok/Ok/Ok/Ok

	NOM DU GROUPE : INFO-12 RABEHI Milhane, TSANEV Pavel, BIGOT Gassien	Saé S1_01_02 Développement en C# Librairie Monogame	
	Escape Game		

5.2 Tests de performance



L'outil de diagnostic nous permet de voir l'évolution de la mémoire du processus. En effet on peut voir sur le screen ci-dessus, qu'une augmentation de la mémoire s'est effectuée (de 186 à 192). Pour sûr, plus de contenus sont présent dans la Map 2, c'est pour cela que l'augmentation se fait, de plus, plus de classes sont mobilisées c'est aussi un point essentiel à l'augmentation de celle-ci



Lorsqu'il y a un changement de scène de la Map 2 vers la Map 3 il n'y a pas de changement de mémoire du processus simplement car ces deux maps ont les même contents, mêmes images et ont été faites par les même tiles.