

# FIHIT

**Antonin LAINE / Arthur WEENS / Remi BORDES**

Rapport de projet SI28 (UTC)

# SOMMAIRE

NOTE D'INTENTION	2
SCÉNARISATION	3
DIRECTION ARTISTIQUE	5
VISUEL	5
SONORE	7
CONCEPTION TECHNIQUE	8
RETROSPECTIVE	10
ANNEXES	11
ASSETS FIGMA	11
TRAILER	11

# NOTE D'INTENTION

D'abord parti d'une idée de Point and click, Fix It est finalement une collection de mini-jeux rétro 2D architecturée de façon non linéaire et immersive.

La dimension scénaristique est limitée au presque minimum. Un vaisseau spatial est hacké et vous devez en reprendre les commandes en réparant plusieurs modules du panneau de contrôle. Ces réparations faites, vous pouvez donc prendre le vaisseau en main au travers d'une phase de shoot'em up.

## Inspirations

- Please Don't touch anything - Four Quarters
- Still There - GhostShark
- You Have 10 Seconds - Tamationgames (précisément pour le premier mini-jeu)

Comme évoqué plus haut, le jeu a évolué au cours de sa conception pour au final prendre les aspects moins puzzle et réflexion. L'idée d'un manuel à la Still There contenant des instructions pour réparer la machine au cœur du jeu a notamment été évincée en faveur d'une expérience plus "straight forward".

Les choix techniques, visuels et sonores seront présentés plus tard dans des parties dédiées.

## Public Cible

Le jeu n'a pas pour but d'être difficile et ne vise pas un public de joueurs confirmés. L'absence de dimension référentielle va également dans ce sens, le jeu ne s'adresse pas à une communauté ou une niche particulière. Même l'esthétique reprend des codes rétro ce n'est pas tant le cas du jeu, qui lui reste avant tout une mini-expérience interactive plus qu'un jeu rétro.

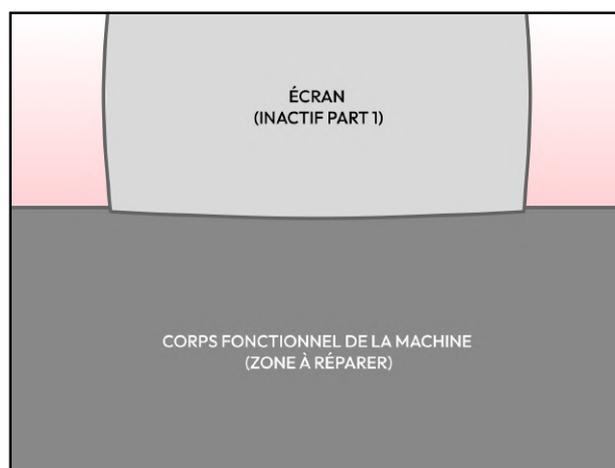
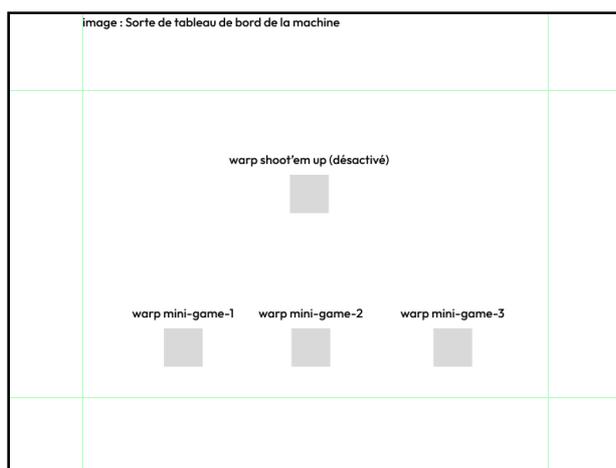
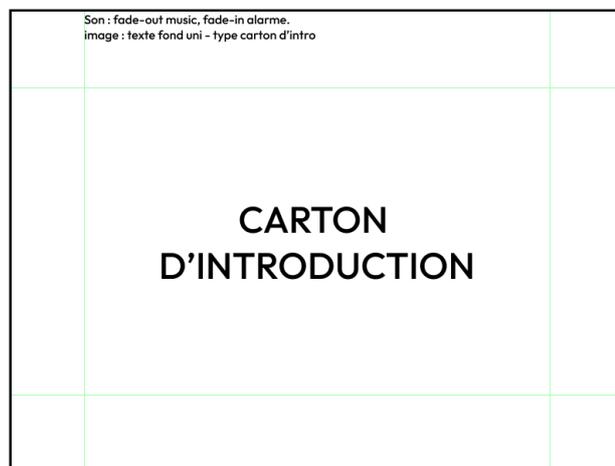
On peut cependant admettre que la nature des mini-jeux et cette variation régulière du gameplay sera plus simple à admettre et prendre en main pour des joueurs que des non-joueurs.

# SCÉNARISATION

## Progression en deux grandes étapes :

1. Alerte générale, le vaisseau est hacké, vous devez vite le réactiver !
  - Remettre le vaisseau en marche, réparer les 3 modules qui dysfonctionnent.
2. Vous avez réparé le vaisseau ! Maintenant prenez en le contrôle et repoussez les envahisseurs.
  - Une fois réparé, l'écran de la machine s'allumera et donnera accès à un shoot'em up.

## Storyboard basique de l'introduction et aperçu de l'écran de sélection :



## Séquençage détaillé :

### 1. Menu principal : "Start" et "Quit".

Mettre dans l'ambiance et ne pas lancer le jeu immédiatement, que le joueur ait l'initiative.

### 2. Carton d'introduction pour poser le contexte.

Textes du carton :

<<

Oh no !

The spaceship has been hacked !

Quick ! You need to Fix It !

You have the biggest brain here.

You are our only hope !

>>

Le joueur n'aura pas plus d'information textuel que celle-ci, le reste de la narration étant réservé à des éléments visuels, sonores et interactifs.

### 3. Écran de sélection des 3 mini-jeux. Cette phase est à temps limité.

Les mini-jeux peuvent être lancés et finis dans n'importe quel ordre.

Il est possible de revenir à l'écran d'accueil depuis ceux-ci pour ne pas resté bloqué, la progression cependant, n'est pas sauvegardée si le joueur quitte avant d'avoir fini.

Pour chaque mini-jeu complété le menu de sélection se colore peu à peu en vert. Une fois tout fini, l'écran principal, cassé depuis le début, se met en marche permettant alors d'accéder au shoot'em up.

Les éléments interactifs profitent d'effets de hover (curseur, couleur et freeze)

### 4. Shoot'em Up

Il est le jeu le plus complet. Nous l'imaginons comme un accomplissement ainsi qu'une surprise pour le joueur une fois le reste complété.

# DIRECTION ARTISTIQUE

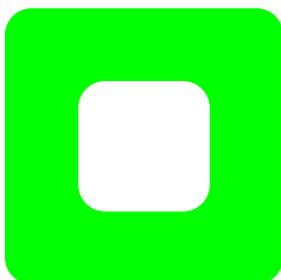
Très tôt dans le projet il a été fait le choix de donner à l'expérience un aspect rétro et 8 bits. Attention cependant, le jeu ne recrée pas un style rétro spécifique, il accumule des artefacts "retro" majoritairement 8 bit destiné à donner une "vibe" plus qu'un pastiche.

L'avantage principale est de pouvoir obtenir des résultats convaincants malgré le manque de compétences spécialisées du groupe de travail. De tourner des faiblesses en force.

## VISUEL

### Couleurs et polices

Retro Green



#00ff2a  
rgb(0, 255, 42)

Retro Red



#ee1c24  
rgb(238, 28, 36)

Black



#000  
rgb(0, 0, 00)

White



#ffffff  
rgb(255, 255, 255)

**Awkward / Awkward Tall / Awkward Ext**

**abcdefghijklmnopqrstuv**

**ABCDEFGHIJKLMNPOQRSTUVWXYZ**

**0123456789**

## Logotype



Travaux préparatoires plus complets (et désordonnés) :

- [figma.com/file/6ogrdXMfJpBqY7JzRrUWIn/FixIt?node-id=301%3A140&t=bAdBGO6HuU9CCmOZ-1](https://figma.com/file/6ogrdXMfJpBqY7JzRrUWIn/FixIt?node-id=301%3A140&t=bAdBGO6HuU9CCmOZ-1)

## Ratio d'image / définition / Pixel art et minimalisme / Animations hachées

Nous avons opté pour un ratio d'image 4:3 pour son look résolument rétro. Son aspect plus vertical s'adaptant aussi très bien à la nature des mini-jeux qui ne profitent de toute façon pas d'un ratio plus panoramique, comme le classique 16:9.

La définition choisie se limite à un très bas 400x300 assurant un niveau de pixellisation natif de l'image en complément à des assets dans un style minimaliste et pixel art.

Tous les assets du jeu nous appartiennent à l'exception de certains utilisés dans le shoot'em up construit sur la base du travail de : insérer lien. Les travaux graphiques pour ce projet sont faits sur Figma, Photoshop ainsi que dans l'éditeur de sprites intégré à GameMaker.

Enfin pour continuer de convoier le look rétro souhaité, nous avons fait le choix de limiter le framerate interne du jeu et le nombre de frames utilisés dans les animations.

## Filtre CRT

Enfin une part importante de la texture de l'image est apportée par un filtre CRT (Cathode-ray tube). L'imitation en jeu de l'artefact est relativement simple : il s'agit d'une bande de 3 pixels avec différents niveaux d'opacité. Ce Sprite de 1x3 est ensuite répété sur toute la surface de l'écran.

Sans filtre CRT



Avec filtre CRT



## SONORE

Nous n'avons pas créé de son, mais avons simplement utilisé des bibliothèques et assets sonores gratuits. Il s'agissait avant tout d'un travail de sélection.

Pour le traitement cela dépendait du son d'origine, mais l'échantillonnage était réduit et la dynamique écrasée. Dans l'ensemble le son était perceptiblement dégradé et rapproché des aigus.

Dans des cas particuliers, pour ancrer un son dans le jeu, comme le bruit de l'alarme par exemple, nous utilisons un soupçon de reverb.

L'ensemble du traitement sonore était fait via Audacity puis les options d'importation de GameMaker Studio 2.

Les transitions des sons de fond dans le jeu sont assurées par des fade-in et fade-out, et quelques coupures nettes dans des cas particuliers pour des effets de styles (cf retour menu principal depuis un mini-jeu).

# CONCEPTION TECHNIQUE

Nous avons déjà abordé dans la partie précédente, sans trop de précisions, les outils de conception utilisés en dehors du moteur de jeu. Il est maintenant temps de nous concentrer sur celui-ci.

Pour la création du jeu à proprement parler, nous avons donc opté pour GameMaker Studio 2 ; un moteur accessible, mais versatile, pensé pour la conception 2D.

Aucun de nous n'avait une expérience notable avec ce moteur ou un autre du genre, ce projet était donc l'occasion de se faire la main et d'expérimenter.

De la nature complexe d'un jeu, dans notre cas composite, avec plusieurs mini-jeux, il est peine perdue d'essayer de retracer ne serait-ce qu'un dixième de comment et pourquoi, il se produit ce qu'il se produit à l'écran.

Voici à titre d'illustration, deux extraits de code : les sons d'ambiance pour la séquence d'introduction, ainsi que le saut pour le personnage dans le mini-jeu de plateforme. Ces extraits utilisent GML (le langage propriétaire utilisé dans le moteur), qui dans l'idée, se rapproche pas mal de JS. Notez aussi qu'il existe une version visuelle du langage, à la "blueprint" de Unreal Engine, que nous n'avons cependant pas utilisé.

Extrait de : l'évènement "Room Start" de l'objet "obj\_son"

```
if room==rm_start1 {
    audio_sound_gain(mainMenu_sound, 0.8, 10000);}

if room==rm_start2 {
    audio_sound_gain(alarm_sound, 0.12, 10000);
    audio_sound_gain(mainMenu_sound, 0, 10000);}

if room==rm_start3 {
    audio_sound_gain(alarm_sound, 0.1, 3000);
    audio_sound_gain(mainMenu_sound, 0, 0);}
```

Extrait de : l'évènement "Step" de l'objet "obj\_play" (sous-dossier "Virus", mini-jeu 1)

```
//Saut, déplacement vertical
if oPlayer.room_number == 4 || oPlayer.room_number == 6 {
    image_yscale = -1;
    vsp -= grv;
    if (place_meeting(x,y-1,oTrampoline)) {
        vsp = 20;
        audio_play_sound(snd_trampo,10,false,0.4);
    }

    if (place_meeting(x,y-1,oMur)) && (key_space) {
        vsp = 15;
        audio_play_sound(snd_jump,9,false,1);
    }
}
else{
    image_yscale = 1;
    vsp += grv
    if (place_meeting(x,y+1,oTrampoline)) {
        audio_play_sound(snd_trampo,10,false,0.4);
        vsp = -20;
    }

    if (place_meeting(x,y+1,oMur)) && (key_space) {
        vsp = -15;
        audio_play_sound(snd_jump,9,false,1);
    }
}
```

Vous trouverez la documentation "complète" de GameMaker à cette adresse ainsi qu'une sélection de quelques autres sources consultées pour la création du jeu :

- [manual.yoyogames.com/#t=Content.htm](http://manual.yoyogames.com/#t=Content.htm)
- [reddit.com/r/gamemaker/](http://reddit.com/r/gamemaker/)
- [youtube.com/@GameMakerEngine](http://youtube.com/@GameMakerEngine)

# RETROSPECTIVE

## ANTONIN - MASTER DESIGN D'EXPÉRIENCE

**Principales contributions : Interface et progression, Audio, Dossier, Trailer**

Comme les autres ont déjà pu le dire, nous n'avions pas d'expérience sur GameMaker et le projet a donc été une bonne occasion pour se faire la main. Je pense que ça a été une bonne première initiation. Le travail de groupe aussi, c'est notablement bien passé et au finale la nature ludique du projet a plutôt motivé tout le monde j'ai l'impression.

Même si Fix It c'est finalement formalisé comme un jeu simple, je pense que ça a aussi permis de rendre quelque chose de satisfaisant, tout comme le choix du retro pour arrondir les angles, donnant au final de la personnalité au jeu. C'était agréable de travailler dessus avec Rémi et Arthur.

## ARTHUR - GÉNIE INFORMATIQUE

**Principales contributions : Mini-jeu Labyrinthe, Shoot'em Up**

Le projet Fix It a été une expérience enrichissante. Premièrement j'ai découvert GameMaker, nous partions tous les trois de zéro et les débuts ont donc été compliqués. Le manque d'option pour du travail collaboratif a également été un problème, on se rend alors facilement compte de l'intérêt d'un Git. Après des débuts difficiles nous avons tout de même réussi à produire un rendu que je trouve très satisfaisant, et conforme à notre idée originale. Nous avons appris le travail collaboratif numérique asynchrone, nous obligeant à respecter des structures de notations et à tenir les autres au courant de notre avancée.

## RÉMI - GÉNIE INFORMATIQUE

**Principales contributions : Mini-jeu de plateforme, Mini-jeu Casse-briques, Audio, Correction de bugs**

Le projet Fix It fut une très bonne expérience dans l'ensemble. J'ai beaucoup apprécié voir le projet évoluer petit à petit alors que notre idée de base du projet était assez vague. J'ai trouvé plaisant la bonne répartition du travail dans le groupe, et les idées de chacun n'ont fait qu'enrichir le projet et le faire évoluer.

Enfin, ce projet, et SI28 en général m'ont beaucoup apporté, j'ai découvert et appris à me servir de beaucoup de nouveaux logiciels, et j'ai maintenant une meilleure appréhension des enjeux du numérique. Je trouve très gratifiant de pouvoir affirmer d'avoir co-créer une expérience interactive/un jeu nous-même en étant parti avec quasiment aucune base.

# ANNEXES

## ASSETS FIGMA

<https://www.figma.com/file/6ogrdXMfJpBqY7JzRrUWln/FixIt?node-id=126%3A2&t=AaVvMkVjbHettx8o-1>

## TRAILER

Réaliser avec Adobe Premiere Pro

<https://youtu.be/LeWn-hH18E8>

