

Anaïs BOUCHARDON

Corentin FELTEN

Louise PIC

---

---

# SPACE·RUSH

---

---

---

**RAPPORT FINAL**

## TABLE DES MATIÈRES

<b>INTRODUCTION</b>	2
<b>NOTE D'INTENTION</b>	3
Concept	3
Public cible	3
Objectifs	3
<b>CAHIER DES CHARGES</b>	4
Ressources médias (textes, médias, sons, vidéos)	4
Structuration et navigation	4
Forme et degrés d'interactivité	5
Choix graphiques et d'interface	5
Dimension sonore	5
Choix techniques	6
<b>SCÉNARIO DE SPACE RUSH</b>	7
Menu	7
Introduction	7
Début du jeu	8
Liste des actions à réaliser pour réparer le vaisseau	10
Action récurrente	10
Erreurs lors des mini-jeux	10
Fin du jeu	11
<b>CONCLUSION</b>	12

## INTRODUCTION

Durant le semestre A20 de SI28, nous avons choisi de travailler sur une expérience interactive sur le thème de l'espace. Il s'agit d'un escape game que nous avons appelé **Space Rush**. Nous nous sommes inspirés des jeux **Space Team** et **Among Us**.

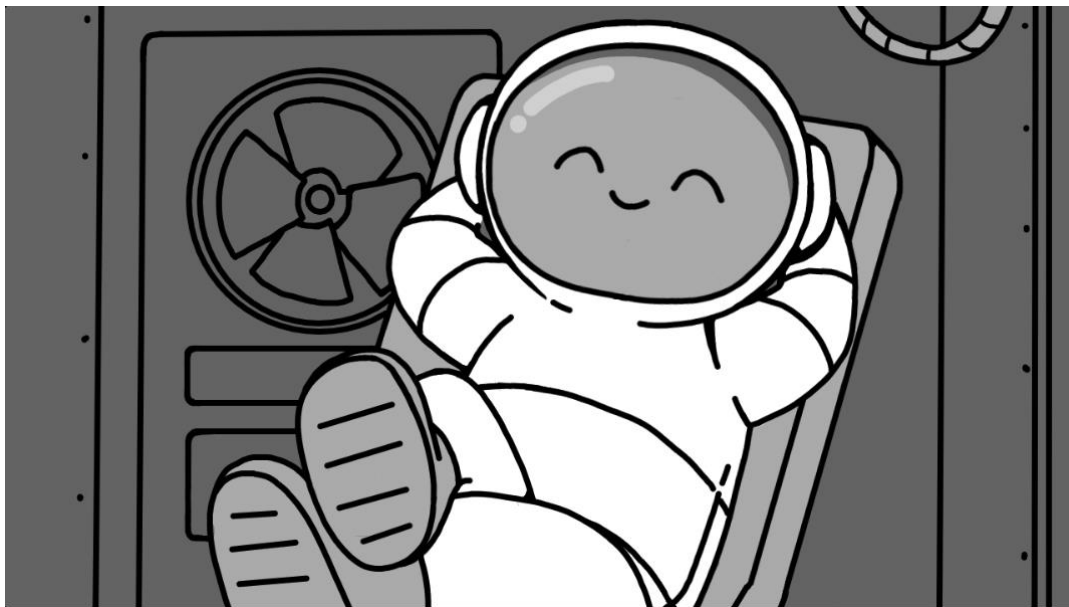


*fig. 1 : Logo de Space Team*



*fig. 2 : Logo de Among Us*

Nous allons vous présenter dans ce document le scénario du projet, mais également toute la partie réalisation technique.



*fig. 3 : Illustration de l'introduction de Space Rush*

## I. NOTE D'INTENTION

### A. Concept

Le concept de notre projet reprend la fouille d'information des **puzzles** et **énigmes** des Escape Games, mais en la transposant en jeu ordinateur en ligne. Le joueur arrive dans un vaisseau spatial via son ordinateur, et doit réussir à réparer les dégâts d'une collision avec un astéroïde grâce à des indices placés à différents endroits du vaisseau. Le point original de cette aventure est que le joueur devra également rechercher une partie des informations via son téléphone, qui représente l'appareil de communication de l'astronaute.

### B. Public cible

Nous avons fixé un public de plus de 14 ans, afin que ce dernier soit pleinement capable de naviguer le long de liens et documents sur internet afin de trouver les codes qui lui seront demandés.

### C. Objectifs

Le jeu est un **jeu chronométré**. Le joueur devra donc se dépêcher afin de trouver tous les codes et réparer un maximum de dégâts avant la fin du temps imparti. Nous avons utilisé **Twine** pour relier les différents panneaux de jeu entre eux, et les graphismes ont été réalisés à l'aide d'**Illustrator**.

## II. CAHIER DES CHARGES

### A. Ressources médias (textes, médias, sons, vidéos)

Nous n'avons pas eu beaucoup de ressources déjà existantes pour créer ce projet. Pour le texte de l'introduction, nous l'avons inventé et nous l'avons enregistré puis modifié la voix sur **Audacity**. L'alarme a également été enregistrée puis modifiée sur ce logiciel. Nous sommes encore en recherche pour le fond sonore, donc nous n'avons pas encore la source de ce média.

### B. Structuration et navigation

Nous avons fait le choix très tôt lors de la création du projet de partir sur Twine pour la réalisation, pour plusieurs raisons. Tout d'abord, nous pensions gérer l'aléatoire à l'aide de différentes branches dans l'histoire, ce qui au final ne s'est pas révélé très pratique, mais la flexibilité d'un projet Twine nous était importante. Avoir le jeu accessible de n'importe quel PC faisait partie des critères que nous avions. Le logiciel nous a permis de structurer la navigation entre les différents panneaux du jeu plus facilement qu'en se servant uniquement de pages HTML, sans compter l'aisance de l'implémentation de JavaScript native de Twine.

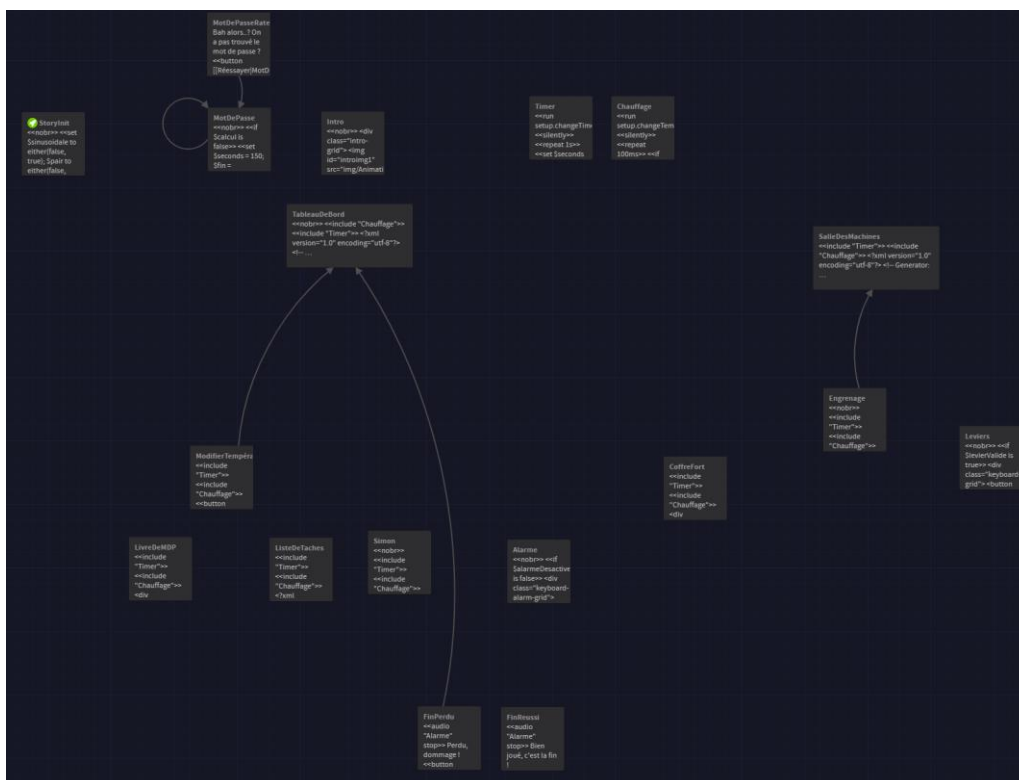


fig. 4 : Structuration de Space Rush

## C. Forme et degrés d'interactivité

Notre escape game est un projet où l'interactivité se fait entre l'utilisateur, l'ordinateur et le téléphone. L'utilisateur doit chercher des informations dans les deux supports afin de gagner la partie.

## D. Choix graphiques et d'interface

Nous avons réalisé les graphismes en fonction de notre thème :

- ambiance intérieure : vaisseau spatiale
- personnage principal : cosmonaute
- ambiance extérieure : espace

Nous avons réalisé l'interface graphique sur **Illustrator** (panneaux interactifs) et les illustrations animées de l'introduction et de la fin sur **Krita**. Nous avons choisi de faire du "flat design" car c'est une technique où les dessins sont faciles à réaliser.

Les documents convertis en documents SVG nous ont permis d'interagir avec la partie "code" du projet :

- créer des zones cliquables
- faire apparaître/disparaître des éléments du décor (par changement d'opacité)

## E. Dimension sonore

Pour poser le contexte lors de l'introduction, nous avons intégré une **voix off** et des **bruitages** qui accompagnent les illustrations animées.

Nous avons mis en place des **effets sonores** pour les mini-jeux, notamment dans le cas où l'utilisateur se trompe ou valide une énigme. Ou encore lors de la désactivation de l'alarme. En effet, pour pouvoir désactiver l'affreuse alarme qui se lance à chaque début du jeu, il faut écouter attentivement la durée des sons pour trouver le bon code à entrer.

Nous voulons également ajouter un **son d'ambiance**, dont le ton sonne **voyage spatial**, pour rendre l'expérience plus immersive.

## F. Choix techniques

Le choix d'utiliser Twine s'est révélé être à la fois un bon et un mauvais choix. Comme dit précédemment, la mise en place de la navigation ainsi que les manipulations de bas niveau (comme des comparaisons ou des affectations) est très simple et rapide à prendre en main. C'est lorsque nous avons commencé à rajouter des scripts que nous avons codés, que nous nous sommes rendu compte de la complexité de l'interaction entre Twine et JavaScript. La manipulation de variables et de données Twine directement depuis un script reste relativement simple, mais réussir à récupérer un script et le faire tourner correctement s'est révélé être un challenge important. Les documentations sur l'utilisation de JavaScript dans un environnement Twine ne sont pas très nombreuses sur le net, donc les phases de debug ont été longues et compliquées.

Les langages utilisés sont les suivants :

- HTML
- CSS
- JavaScript
- Sugarcube
- Manipulation de code SVG directement

L'apprentissage du code JavaScript aura probablement été le plus long, car aucun d'entre nous n'en avait manipulé avant.

### III. SCÉNARIO DE SPACE RUSH

L'utilisateur a, à sa disposition, une interface sur l'ordinateur sur laquelle il y a un flashcode. Lorsque l'utilisateur scanne le flashcode avec son téléphone, un document apparaît à l'écran, *Notice d'utilisation*.

#### A. Menu

Afin de s'assurer que l'utilisateur a bien compris qu'il faut utiliser son téléphone et son ordinateur, la première action à réaliser est de rentrer un mot de passe, qu'il trouve en bas du document qui s'est ouvert sur son ePhone. Si l'utilisateur ne trouve pas au bout de 50% du temps, un rappel lui indique de regarder son portable.

L'introduction se lance. Une voix grave commence à parler.

#### B. Introduction

**Narrateur :** "L'espace. Un monde vaste et infini ne demandant qu'à être découvert. Seules les planètes et les étoiles y règnent, ainsi la paix est préservée... EH ! MAIS TIENS ! c'est le vaisseau de Tom là-bas ! Tom est un petit astronaute dont la mission est de se rendre sur la planète Oméga 3, une jolie petite planète soit dit en passant ! Tom est très minutieux et appliqué dans son travail ! D'ailleurs ! Depuis qu'il était parti de la planète Terre, son voyage s'était déroulé sans aucun encombre !"

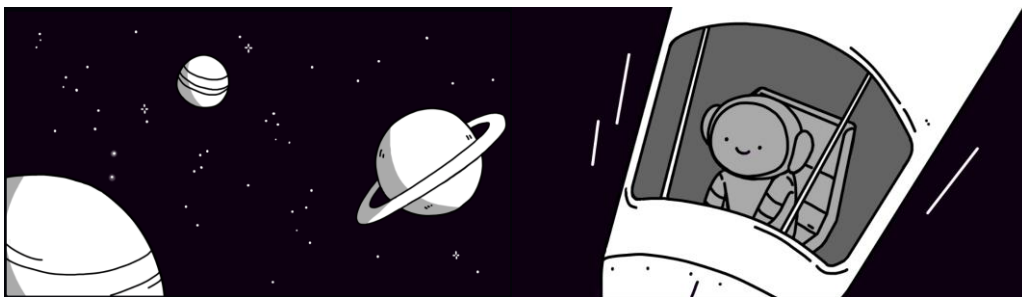


fig. 5 & 6 : Illustrations de l'introduction de Space Rush

**ePhone :** "Vous avez réalisé toutes vos tâches du jour... N'oubliez pas de contacter la station pour démarrer le plan d'atterrissage sur Oméga 3..."





fig. 7 & 8 : Illustrations de l'introduction de Space Rush

**Narrateur :** “Son ePhone contient toutes les notices d'utilisation !” \*soupire\* “mmh... tout se déroulait parfaitement bien !”

**Tom :** \*panique\*



fig. 9 & 10 : Illustrations de l'introduction de Space Rush

**Narrateur :** “ Peut-être trop bien finalement...”

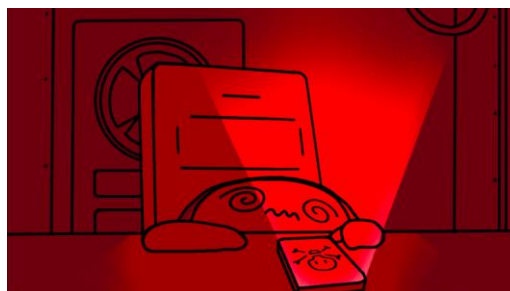
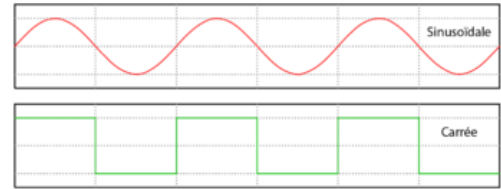


fig. 11 : Illustration de l'introduction de Space Rush

### C. Début du jeu

Le jeu est **chronométré**. L'utilisateur a quinze minutes pour réparer le vaisseau. Au début de chaque partie, 3 variables booléennes sont générées aléatoirement par Twine, et qui vont déterminer la branche de jeu dans laquelle l'utilisateur va se diriger. Ces variables déterminent les solutions des mini-jeux et les actions que l'utilisateur devra réaliser afin de compléter le jeu. Ces variables auront des impacts visuels lors du jeu :

Variable 1: Le profil de vitesse du vaisseau (soit sinusoïde (true), soit carré (false)) visible sur le tableau de bord principal.



Variable 2: Le nombre d'étoiles qu'il voit à travers la vitre du cockpit dans la salle du tableau de bord (paire ou impaire).

Variable 3: La couleur de la diode (diode rouge ou verte) présente sur le tableau de bord.

L'interface graphique du jeu apparaît. L'utilisateur peut naviguer dans deux pièces du vaisseau : le tableau de bord et la salle des machines

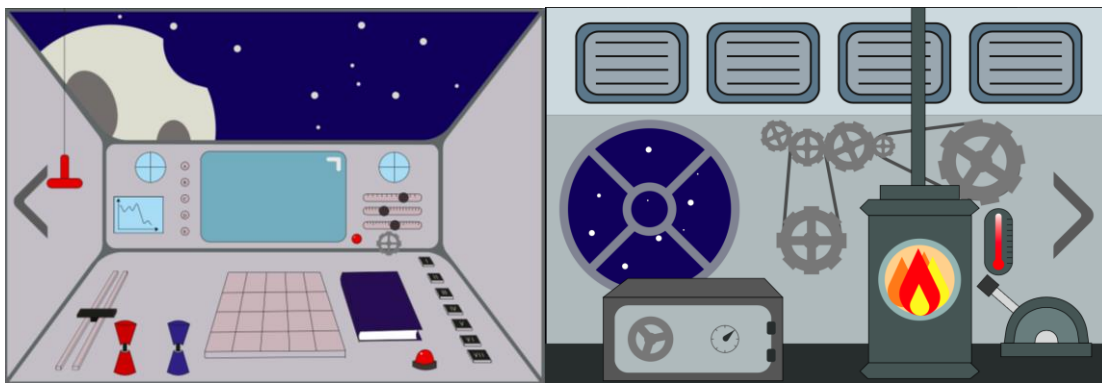


fig. 12 & 13 : Interface graphique du tableau de bord et de la salle des machines

Une alarme se déclenche. L'utilisateur doit cliquer sur le cahier bleu posé sur le tableau de bord pour désactiver l'alarme. Dans le cahier mot de passe, il y a une section "mot de passe alarme :". En dessous, il y a un tableau avec des symboles correspondant à plusieurs mots de passe en fonction du type d'alarme qui sonne. L'utilisateur doit écouter quelle alarme est actuellement en train de sonner, regarder sur son ePhone le symbole correspondant puis sur le cahier le mot de passe à rentrer pour l'éteindre. Il doit ensuite cliquer sur les boutons du tableau de bord pour éteindre l'alarme.

Lorsqu'on clique à l'écran, une liste de tâche à effectuer sur le vaisseau apparaît :

- re calibrer le moteur : "simon" sur le clavier principal sur le tableau de bord
- re positionner les capteurs infra-rouges : leviers dans la salle des machines
- contrôler la température de la chaudière
- réparer le système d'engrenage de la sortie du train d'atterrissage du vaisseau dans la salle des machines
- trouver le code du coffre-fort pour récupérer la radio de secours

## D. Liste des actions à réaliser pour réparer le vaisseau

Pour chaque mini-jeu, l'utilisateur doit se référer à sa notice d'utilisation disponible sur son ePhone afin d'avoir plus d'informations sur les actions à réaliser.

- Simon : Selon les "variables" trouvées dans l'interface graphique du tableau de bord, il devra déterminer grâce au manuel la bonne combinaison de cases de couleurs. A chaque fois que l'utilisateur clique sur une case, celle-ci change de couleur. Il gagne le mini jeu lorsque le clavier est dans la bonne configuration. Un code apparaît alors, que l'utilisateur doit impérativement noter afin de pouvoir réaliser le dernier mini jeu qui lui fera gagner l'escape game.
- Leviers : En fonction des variables de départ, l'utilisateur devra déterminer dans quelle situation il se trouve et en déduire la bonne combinaison à partir des informations du manuel. Il doit ainsi positionner les 3 leviers dans la bonne position. Idem pour le code final.
- Engrenages : La première étape est de retrouver les engrenages cachés dans le vaisseau. Dès que l'utilisateur clique sur un engrenage, celui-ci se place automatiquement au bon endroit pour combler le trou dans la chaîne et réactiver la chaîne de transmission. Idem pour le code final.
- Coffre-fort : A l'issue de chaque jeu, on trouve des indices pour trouver le code du coffre-fort. L'utilisateur doit regarder dans le manuel mot de passe dans quel ordre rentrer les codes.

## E. Action récurrente

L'utilisateur doit en permanence vérifier la température du vaisseau sur l'écran principal. Il doit allumer ou éteindre la chaudière afin de réguler la température du vaisseau (lorsque la température est trop chaude ou trop froide, le joueur perd la partie). Pour allumer/éteindre la chaudière, l'utilisateur doit lever/baisser le slider du tableau de bord.

## F. Erreurs lors des mini-jeux

Si l'utilisateur entre un mot de passe erroné / se trompe lors d'un mini-jeux, il recevra un malus au niveau du temps restant (-30s au temps total). L'intérêt de ce malus est d'empêcher l'utilisateur de brute-forcer le mot de passe / la solution du puzzle, et l'inciter à trouver la bonne solution par lui-même.

## G. Fin du jeu

Plusieurs fins sont possibles pour notre escape game :

- si la température est supérieure à (ex : 40 degrés) : le vaisseau explose => **PERDU**
- si la température est inférieure à (ex : -10 degrés) : l'utilisateur meurt de froid => **PERDU**
- si l'utilisateur met plus de 15mn à trouver le code du coffre-fort, une pluie d'astéroïdes s'abat sur le vaisseau => **PERDU**

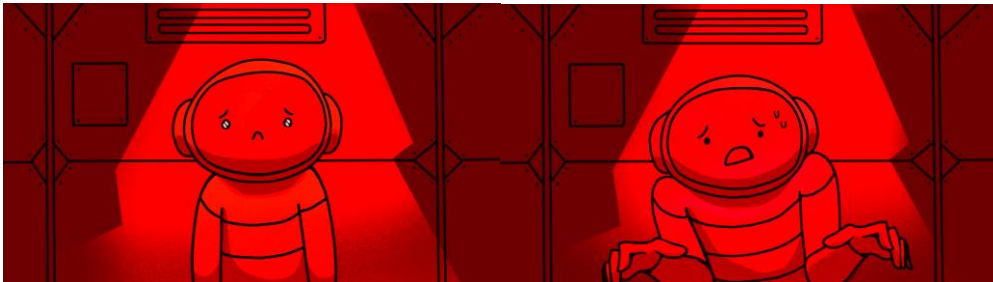


fig. 14 & 15 : Illustrations de game over de Space Rush

- sinon, si l'utilisateur rentre le bon code du coffre-fort : il récupère sa radio de secours et peut contacter la station spatiale, et avoir les bonnes indications pour rentrer en sécurité => **GAGNÉ**

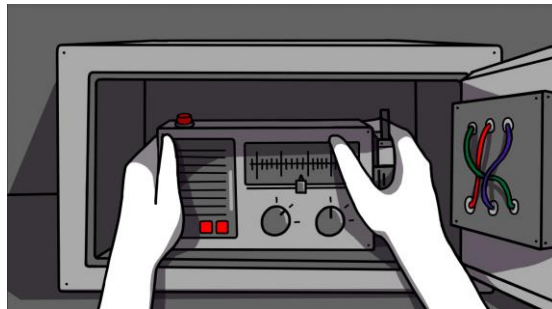
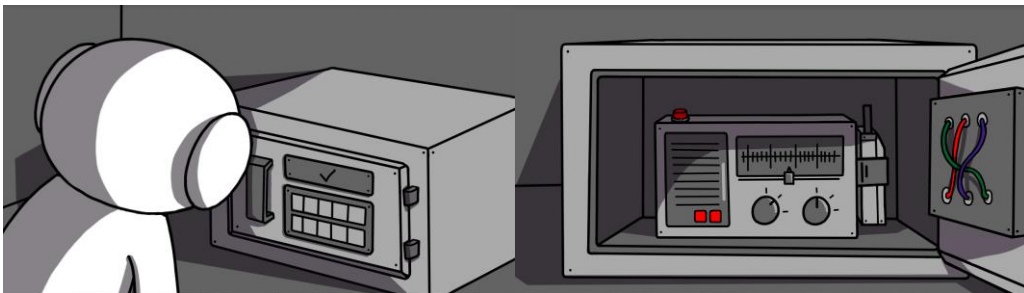


fig. 16 & 17 & 18 : Illustrations de fin de Space Rush

## **CONCLUSION**

Nous avons beaucoup apprécié réaliser ce projet. En effet, le choix très large du sujet nous a permis d'être très créatif du point de vue de l'histoire. De plus, notre groupe était composé d'un TC05, d'une IM01 et d'une IM05. La diversité de nos formations nous a permis de voir un éventail de techniques différentes pour notre projet. La partie implémentation nous a pris beaucoup de temps, mais le résultat final est là et nous sommes très satisfait !