

# Projet TPI



Anaëlle Zahnd | SM-CA 2a



# Table des matières

Introduction.....	6
Contexte et Objectifs.....	6
Contraintes et Exigences.....	6
Déroulement du projet.....	6
<b>Analyse Marketing</b> .....	8
Objectifs SMART .....	10
Analyse du public-cible .....	11
Public-cible 1.....	11
Public-cible 2.....	12
Persona 1.....	13
Persona 2.....	13
Analyse des concurrents.....	14
Matrice de positionnement.....	16
Analyse de la Matrice de positionnement.....	17
Analyse SWOT.....	18
KPI.....	20
<b>Design</b> .....	22
Identité visuelle.....	24
Couleurs.....	24
Typographie.....	24
Logo.....	25
Map.....	26
Intérieur maison 1.....	28
Intérieur maison 2.....	29
Personnage principal.....	30
PNJ .....	31
Pain.....	32
Pièce.....	33
<b>Développement</b> .....	34
VI : SPRITE, DEPLACEMENTS ET COLLISIONS.....	38
1.Sprite du personnage principal.....	38
2. Déplacements du joueur.....	40
3. Collisions et limites du monde.....	41
V2 : INTERFACE, TRANSITIONS ET GESTION DES MAPS.....	44
1. Écran d'accueil du jeu .....	44
2. Transitions entre scènes .....	46
3. Zones de transition et de sortie .....	48
4. Logique des transitions .....	49



- 5. Maisons et gestion des collisions..... 50
- 6. Système de profondeur dynamique .....51
- V3 : INTERACTIONS ET FIN DE JEU ..... 54
  - 1. Écran de fin t réinitialisation..... 54
  - 2. PNJ (Personnages Non-Jouables) ..... 56
  - 3. Système de dialogues avancés..... 57
  - 4. Système de récolte et gestion d'objets ..... 60
  - 5. Système de bulles de dialogue dynamiques..... 62
  - 6. Cooldown et prévention des abus..... 64
  - 7. Architecture globale des états du jeu..... 65
- Validations..... 66
- Problèmes connus et améliorations possibles ..... 68
- Gestion de projet..... 70
- Conclusion Professionnelle .....72
- Conclusion personnelle.....73
- Annexes ..... 74
- Sources..... 88



# Introduction

## Contexte et Objectifs

Le projet consiste à développer un jeu vidéo en ligne de style Pokémon, Animal Crossing, jouable dans un navigateur via HTML, CSS et JavaScript (avec le framework Phaser).

Le joueur incarnera un personnage pouvant :

- se déplacer librement sur une carte, avec une bonne gestion des collisions,
- interagir avec deux PNJ (personnages non-jouables),
- entrer dans les deux maisons,
- récupérer des objets et, si possible, disposer d'un inventaire ou participer à des combats.
- réaliser une quête

Ce projet vise à combiner analyse marketing, conception graphique et développement web afin d'obtenir un produit complet et fonctionnel.

## Contraintes et Exigences

- Technologiques : le jeu doit être entièrement développé avec JavaScript, HTML et CSS.
- Ressources et temps : le projet est réalisé avec des ressources limitées, ce qui implique une planification rigoureuse.
- Recherche : une partie du temps sera dédiée à l'analyse du marché et aux tests utilisateur.

## Déroulement du projet

### 1. Analyse marketing

- Objectifs SMART
- Analyse du public cible et Persona.
- Analyse des concurrents avec une matrice de positionnement.
- Analyse SWOT
- KPI

### 2. Conception / Design

- Création d'une map du jeu (routes, et maisons).
- Création de deux intérieurs différents.
- Design du personnage principal et de deux PNJ.
- Réalisation des assets graphiques (sprites et PNG).

---

### 3. Développement

- Version 1: Le personnage peut se déplacer librement sur la carte avec un sprite qui permet son orientation. Une gestion des collisions: le personnage reste sur les chemins et ne traverse pas les obstacles.
- Version 2: Entrée dans les maisons. Changement de scène. Ajout d'un écran d'accueil avec un bouton start.
- Version 3: Interaction avec les PNJ et des objets: Possibilités de récolter un objet, indiqué par les PNJ. ainsi qu'un écran d'accueil.
- Version 4 (optionnelle): Implémentation d'un système de combat ou d'un inventaire (si le planning le permet).

Le projet est disponible sur (uniquement sur ordinateur)

**[https ://azahnd.eleves.mediamatique.ch/sakima/](https://azahnd.eleves.mediamatique.ch/sakima/)**



# Analyse Marketing



# Objectifs SMART

## Spécifique

Développer un jeu vidéo en ligne de style Pokémon, jouable dans un navigateur avec HTML, CSS et JavaScript, permettant à un joueur d'incarner un personnage qui peut se déplacer librement sur une carte, interagir avec des PNJ, entrer dans des maisons, récupérer des objets et, si possible, disposer d'un inventaire ou participer à des combats.

## Mesurable

Livrer un prototype jouable complet comportant au minimum 1 carte fonctionnelle, 2 intérieurs fonctionnels, 1 personnage jouable, 2 PNJ, et un but, ainsi qu'un rapport regroupant l'analyse marketing, la conception graphique et la documentation du développement.

## Atteignable

Réaliser le projet uniquement avec HTML, CSS et JavaScript en suivant un développement progressif (versions successives). Le tout sera codé avec l'aide d'un framework Javascript: Phaser.

## Réaliste

Limiter le périmètre aux fonctionnalités essentielles (déplacement, interactions, zones, inventaire/combats optionnels) afin de respecter les ressources et délais disponibles.

## Temporel

Finaliser le jeu complet et le rapport pour le 17 décembre 2025 (avec jalons intermédiaires pour le design et le développement).

# Analyse du public-cible

## Public-cible 1

*Adolescents et jeunes adultes joueurs de RPG*

### 1. Sociodémographique

- Âge : 10-20 ans
- Sexe : Majoritairement féminin grâce à un style graphique « girly » et coloré
- Niveau scolaire : étudiants, jeunes
- Statut socio-économique : moyen, avec accès à un ordinateur et à Internet à domicile

### 2. Géographique

- Francophones, principalement en Suisse, et d'autres régions francophones
- Accès régulier à Internet à domicile.

### 3. Comportemental

- Jouent régulièrement à des jeux vidéo en ligne ou sur console
- Préfèrent les RPG et les jeux de collection d'objets ou de créatures
- Cherchent une interface simple et intuitive avec une liberté d'exploration
- Sensibles aux contenus visuellement attractifs, avec des graphismes colorés
- Motivés par la progression et la découverte : objets à récupérer, zones à explorer, accomplissements visibles

### 5. Psychographique / Intérêts

- Passion pour la culture geek, le retrogaming et la pop culture
- Désir de socialiser et d'interagir avec des PNJ ou d'autres joueurs
- Préfèrent des expériences de jeu immersives et créatives
- Sensibles aux aspects de collection, personnalisation et accomplissement

### 6. Besoins et attentes

- Jeu gratuit et accessible directement dans le navigateur
- Possibilité d'explorer la map colorée et d'interagir avec des PNJ
- Progression motivante : objets à récupérer, zones à découvrir
- Interface simple et intuitive, facile à prendre en main

---

## **Public-cible 2**

*Adultes nostalgiques / anciens joueurs Pokémon*

### **1. Sociodémographique**

- Âge: 25-40 ans
- Sexe: mixte, avec un intérêt pour le style girly pour certains segments féminins
- Niveau scolaire: étudiants, jeunes actifs ou actifs
- Statut socio-économique: moyen à élevé, avec accès à un ordinateur et Internet

### **2. Géographique**

- Principalement urbain et connecté
- Francophone, avec accès régulier à Internet à domicile.

### **3. Comportemental**

- Ont joué à Pokémon dans leur jeunesse et recherchent un jeu rétro ou nostalgique
- Sensibles aux expériences de collection et d'interaction
- Motivés par la progression, la découverte et les accomplissements visibles
- Interagissent parfois avec d'autres joueurs en ligne ou sur forums

### **5. Psychographique / Intérêts**

- Passion pour la culture geek, les jeux rétro et Pokémon
- Attirance pour des graphismes cohérents, style kawaii / rose pour un aspect fun et léger
- Désir de socialiser et d'interagir avec des PNJ ou d'autres joueurs
- Préfèrent des expériences immersives et personnalisables
- Sensibles à la nostalgie et aux mécaniques de collection et de progression

### **6. Besoins et attentes**

- Jeu accessible sans téléchargement, jouable rapidement dans le navigateur
- Possibilité d'explorer la map et d'interagir avec des PNJ
- Style visuel très rose/girly pour créer un univers fun et accueillant
- Progression motivante et récompensante, adaptée à leur temps et disponibilité

# Persona 1



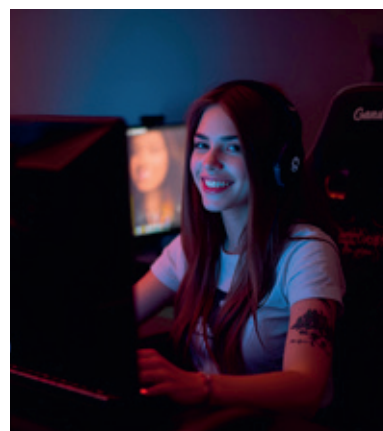
Julie, 12 ans, est la benjamine d'une fratrie de trois enfants et vit en milieu urbain avec ses parents mariés, dans un foyer à situation financière stable. Elle a facilement accès à Internet et possède un ordinateur de gaming, ce qui lui permet de jouer à des jeux vidéo.

Initiée tôt aux jeux vidéo par ses frères, Julie aime les jeux d'exploration et de quête avec des univers doux et féminins. Elle n'apprécie pas les monstres ni les contenus violents, et préfère pouvoir accéder à des jeux gratuits pour garder plus d'autonomie dans ses choix.

# Persona 2

Anna, 34 ans, vit à Morges et travaille comme secrétaire médicale. Passionnée de jeux vidéo depuis son enfance, elle aime découvrir de nouveaux univers et relever des défis à travers ses jeux. Son revenu est modeste, ce qui rend les jeux gratuits particulièrement adaptés à ses besoins.

Elle privilégie les jeux immersifs, comme les jeux d'exploration ou de stratégie, qui lui permettent de se divertir et de se détendre après le travail. Elle apprécie particulièrement les jeux bien conçus, accessibles et stimulants, qui lui offrent une expérience agréable sans frustration liée au coût ou à la complexité.



# Analyse des concurrents

**Zelda A Link to the Past** est un jeu d'action-aventure et RPG classique sorti en 1991 sur Super Nintendo (SNES), développé et édité par Nintendo. Le joueur y incarne un héros qui explore des donjons, résout des énigmes et combat des ennemis dans un univers immersif en pixel art. Le jeu est exclusivement solo et nécessite une console ou un émulateur pour être joué, ce qui le rend inaccessiblement depuis un navigateur.



**Animal Crossing** est un jeu de simulation sociale et de gestion sur mobile, sorti en 2017, développé et édité par Nintendo. Le joueur personnalise son camping, collecte des objets et interagit avec des PNJ ainsi qu'avec d'autres joueurs. L'expérience est principalement solo mais inclut des interactions sociales limitées. Très coloré et cosy, le jeu nécessite une application mobile, et n'est donc pas accessible sur navigateur.



---

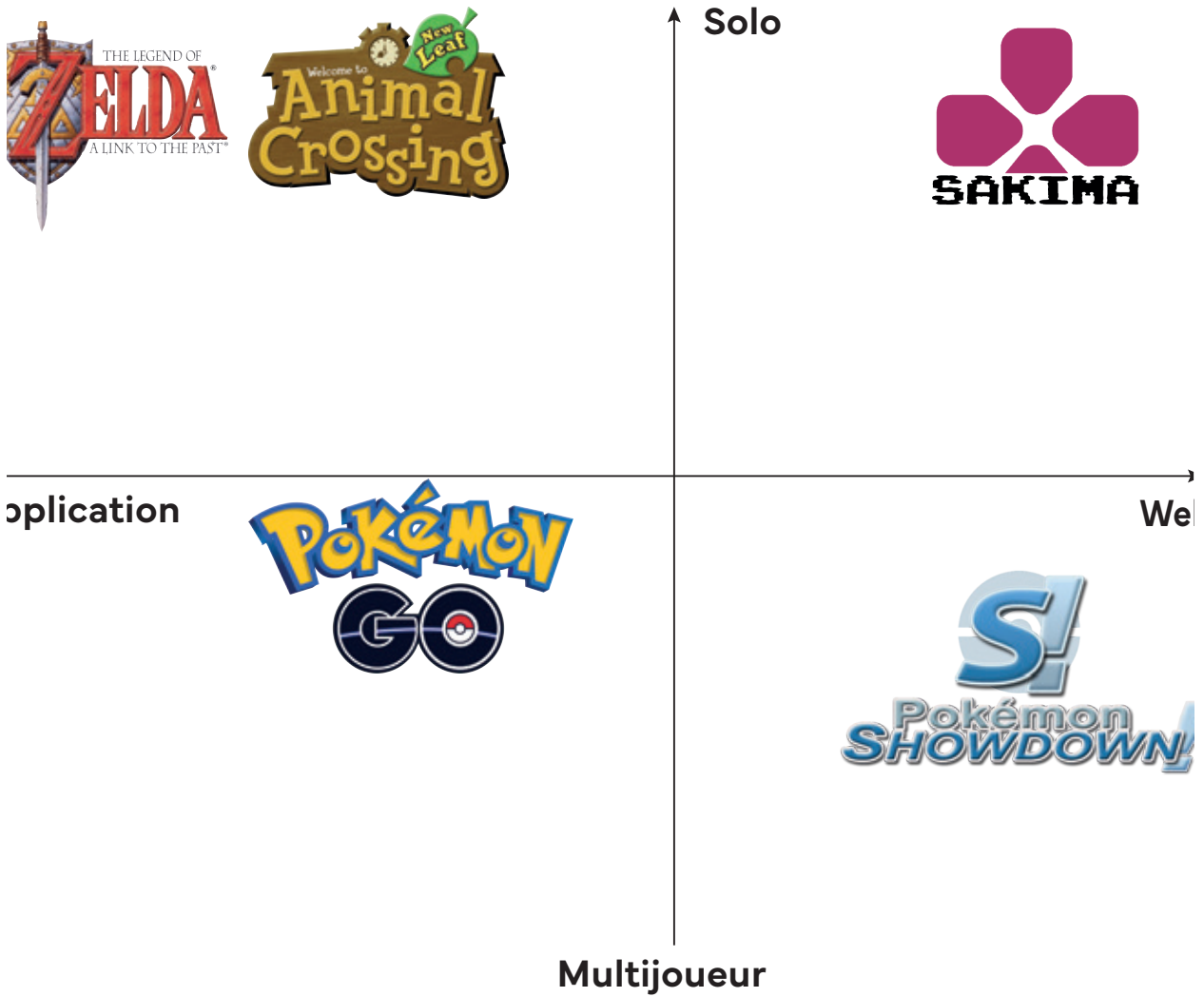
**Pokémon Go** est un RPG en réalité augmentée sur mobile, sorti en 2016. Le joueur capture des créatures dans le monde réel et peut participer à des raids ou échanges multijoueur. L'expérience est majoritairement solo, avec des éléments multijoueur pour les interactions sociales. Accessible uniquement via mobile, Pokémon Go combine exploration réelle, collection et aspects sociaux dans un univers coloré.



**Pokémon Showdown** est un simulateur de combat Pokémon sorti en 2011, accessible directement depuis un navigateur web, développé par une communauté indépendante de fans. Le jeu se concentre sur les combats stratégiques multijoueur, avec la possibilité de jouer contre des bots en solo. Les graphismes sont minimalistes mais l'accessibilité web et la profondeur stratégique des combats en font une référence pour les joueurs cherchant un gameplay rapide et compétitif.



# Matrice de positionnement



# Analyse de la Matrice de positionnement

Grâce à la matrice de positionnement, on peut constater que Sakima se démarque clairement par les axes choisis. Le premier axe distingue le gameplay solo ou multijoueur, tandis que le second prend en compte si le jeu est accessible via une application mobile ou directement en ligne. On observe que des jeux comme Zelda: A Link to the Past et Animal Crossing sont d'excellents jeux, mais ils restent uniquement disponibles via une console et se jouent exclusivement en solo. Pokémon Go, quant à lui, se situe entre le solo et le multijoueur: il permet de capturer des créatures de manière individuelle tout en offrant des fonctionnalités multijoueur comme les combats en arène ou les échanges, mais il reste accessible uniquement sur mobile. Pokémon Showdown, de son côté, est jouable directement en ligne et se concentre principalement sur le multijoueur, offrant une expérience stratégique compétitive.

Ainsi, grâce à cette matrice, Sakima se positionne très favorablement face à la concurrence, combinant une accessibilité immédiate via navigateur et un gameplay solo simple mais engageant. Initialement, une seconde matrice de positionnement était envisagée, mais il s'est avéré difficile de trouver des concurrents similaires au projet avec des caractéristiques distinctes. En effet, les jeux de grande envergure, établis depuis des années, sont difficiles à concurrencer sur d'autres axes. Il n'y a donc qu'une seule matrice, mais plus complète, permettant de mettre en valeur la force distinctive du projet.

Aide de chatGPT pour trouver les concurrents, mais avec ce prompt:

«Donne moi quatre jeux qui sont similaires à un jeu de type pokémon/Zelda ou Animal Crossing» (08.10.2025)

# Analyse SWOT



## Forces (Strengths)

- Accessibilité immédiate: Le jeu est jouable directement dans le navigateur, sans installation, ce qui le rend très accessible pour les adolescents, jeunes adultes et adultes nostalgiques.
- Gameplay solo simple et intuitif: Facile à prendre en main, il cible les joueurs casuels qui recherchent une expérience rapide et agréable.
- Univers attractif et visuel coloré: Les graphismes girly et kawaii attirent un public adolescent féminin tout en séduisant les nostalgiques de Pokémon.
- Exploration et collecte: La possibilité de récupérer des objets et interagir avec des PNJ motive la progression et renforce l'immersion.

## Faiblesses (Weaknesses)

- Contenu limité au lancement: Le nombre de zones, PNJ et objets peut être réduit par rapport à des jeux établis comme Zelda, Animal Crossing ou Pokemon.
- Absence de multijoueur: L'expérience solo peut être perçue comme moins sociale, surtout pour les joueurs recherchant des interactions en ligne.
- Ressources limitées: Le développement seul implique des contraintes techniques et graphiques par rapport aux jeux commerciaux.
- Visibilité et marketing: En tant que petit projet scolaire, le jeu peut avoir du mal à atteindre son public cible sans promotion.

## Opportunités (Opportunities)

- Marché du navigateur en ligne: Peu de RPG d'exploration accessibles directement sur Web, ce qui offre une niche intéressante.
- Public nostalgique et jeunes joueurs: Combinaison d'un style rétro et de graphismes colorés permet de toucher deux segments complémentaires.
- Facilité d'accès et viralité: Le jeu étant gratuit et accessible directement, il peut se diffuser rapidement via le bouche-à-oreille et les réseaux sociaux.

## Menaces (Threats)

- Concurrence des gros titres: Zelda, Animal Crossing, Pokémon Go ou Pokémon Showdown disposent de communautés établies et d'un contenu riche.
- Saturation du marché: Les RPG et jeux de collection en ligne sont nombreux, rendant difficile de capter l'attention.
- Évolution rapide des attentes: Les joueurs peuvent rapidement s'habituer à des mécaniques plus complexes ou à des graphismes très détaillés, réduisant l'attractivité initiale.

# KPI

Objectif 1 : Terminer le projet dans les temps

Indicateurs	Formules	Objectifs à atteindre
Taux d'avancement du projet	$(\text{Tâches réalisées} / \text{tâches totales}) * 100$	$\geq 95 \%$
Respect du planning	Jours réellement utilisés - jours prévus	Dépassement $< 30 \%$
Fonctionnalités livrées	$(\text{Fonctionnalités livrées} / \text{fonctionnalités prévues}) * 100$	$\geq 95 \%$
Respect du micro-planning hebdomadaire	% de tâches réalisées à temps chaque semaine	$\geq 90 \%$





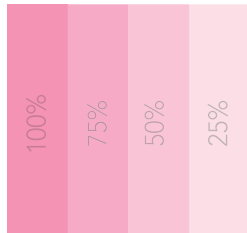
# Design



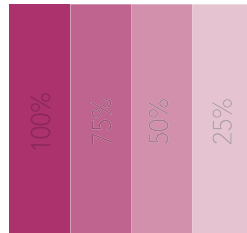
# Identité visuelle

## Couleurs

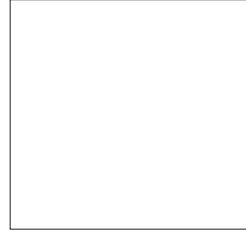
Les couleurs peuvent être utilisées avec un opacité variant de 100 % à 25 %.



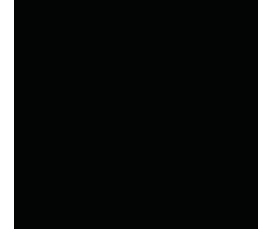
Code Hexa. : #F69ABB  
RVB : 250, 154, 187  
CMJN : 0, 50, 5, 0



Code Hexa. : #B12E66  
RVB : 177, 46, 102  
CMJN : 10, 87, 17, 23



Code Hexa. : #FFFFFF  
RVB : 255, 255, 255  
CMJN : 0, 0, 0, 0



Code Hexa. : #12120D  
RVB : 18, 18, 12  
CMJN : 71, 61, 56, 89

## Typographie

Courier New Regular pour les dialogues, les textes ainsi que les parties de code du rapport, afin de conserver un aspect rétro.

abcdefghijklmnopqrstuvwx  
yz  
ABCDEFGHIJKLMN  
OPQRSTUVWXYZ  
1234567890/()  
&%çàèèè « « \*

Courier New Bold pour les dialogues, les textes et le titre du logo.

abcdefghijklmnopqrstuvwx  
yz  
ABCDEFGHIJKLMN  
OPQRSTUVWXYZ  
1234567890/()  
&%çàèèè « « \*

Mundial Thin pour les textes du rapport et l'ensemble des contenus administratifs.

abcdefghijklmnopqrstuvwx  
yz  
ABCDEFGHIJKLMN  
OPQRSTUVWXYZ  
1234567890/()  
&%çàèèè « « \*

---

## Logo



Logo de base, Favlon



Logo avec le titre.  
Utilisé pour le rapport ainsi que la page d'accueil du site web



Logo en noir et blanc. A utiliser lors d'impression monochrome.



Logo en négatif à utiliser si l'arrière-plan est de la même couleur que le logo.

# Map

Pour créer la map de mon jeu, je me suis inspirée de l'image ci-contre. Celle-ci n'étant pas libre de droits, je l'ai uniquement utilisée comme référence afin de reprendre la disposition des chemins ainsi que le design général de la maison. J'ai ensuite modifié les couleurs ainsi que de nombreux détails pour obtenir un résultat entièrement original.

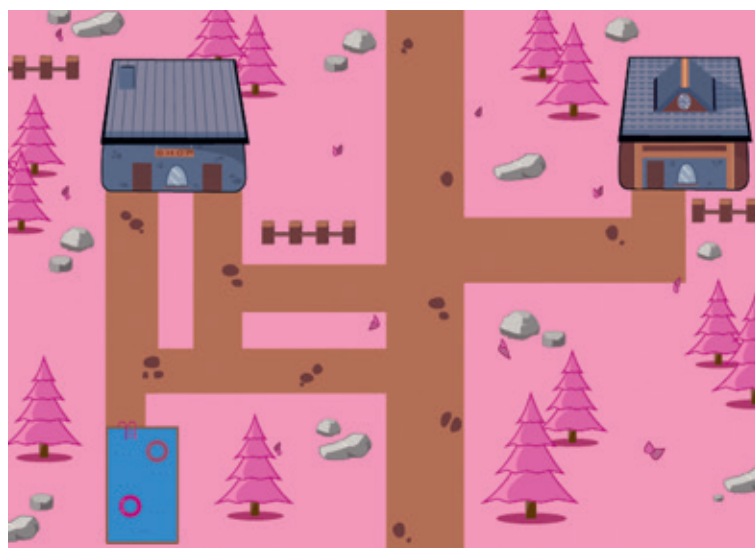
L'œuvre d'origine provient du site : [www.gamedevmarket.net](http://www.gamedevmarket.net).

J'ai choisi de rendre la map principale du jeu rose et girly. Elle est composée de deux mai-



sons. Pour apporter davantage de réalisme à cet univers très coloré, j'ai ajouté différents petits éléments tels que des cailloux, des feuilles mortes ou encore des barrières. Ces détails contribuent à rendre l'environnement plus vivant et cohérent, même si l'herbe et les sapins restent entièrement roses, ce qui renforce l'esthétique stylisée et originale que je souhaite donner au jeu.

Initialement, la map comprenait également une piscine. Toutefois, plusieurs complications techniques rencontrées durant le développement m'ont poussée à abandonner cette idée. Ci-contre se trouve la version de la map comprenant la piscine.

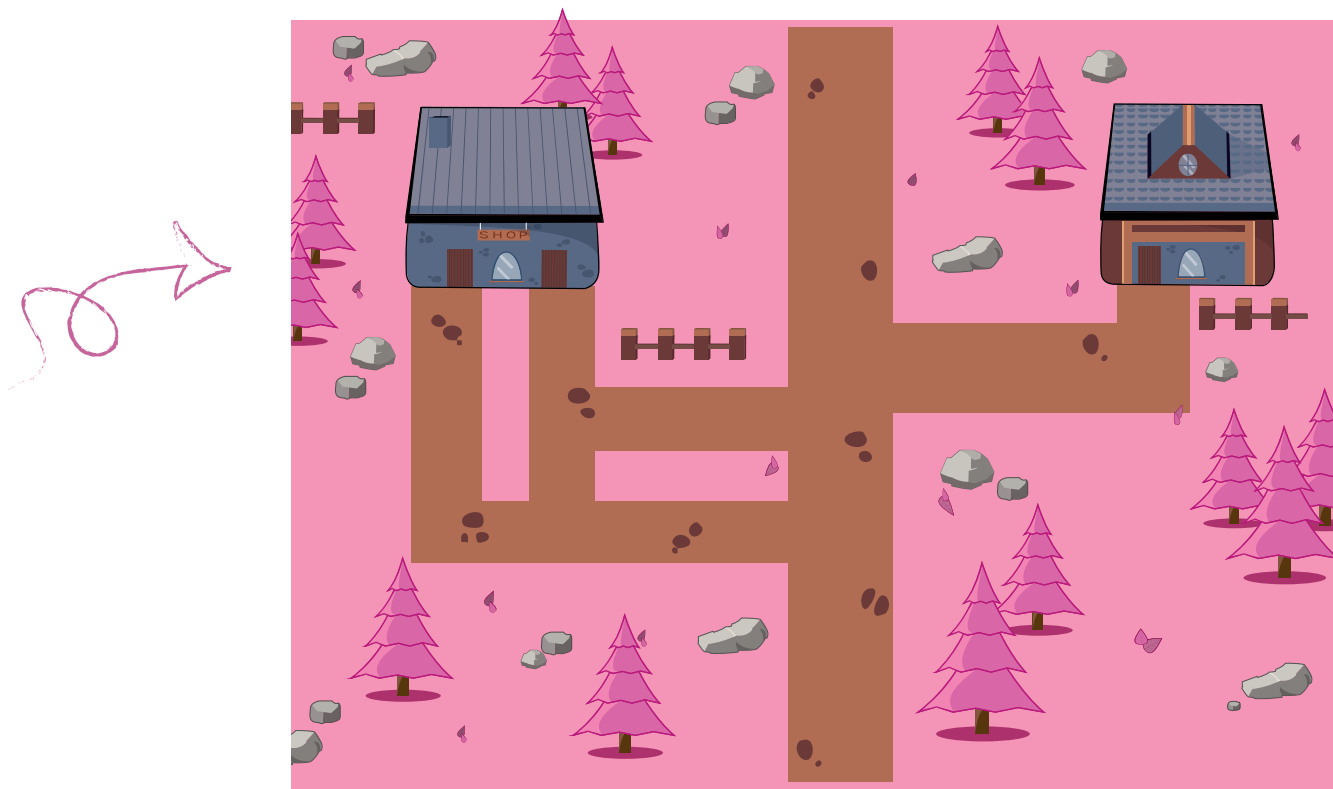


---

L'ensemble est réalisé en 2D simple, avec quelques effets d'ombre qui ajoutent de la profondeur et du dynamisme à la scène. Le choix de ne pas utiliser le pixel art est volontaire : cette technique, très complexe et chronophage, aurait également rendu plus difficile l'identification claire de mon personnage. Le pixel art nécessite en effet de gérer un très grand nombre de pixels individuellement ou d'utiliser des blocs de plusieurs pixels pour simplifier le rendu, un travail conséquent qui ne faisait pas partie de ma planification initiale. J'ai donc privilégié une approche plus rapide et mieux adaptée aux objectifs du projet, afin de concentrer mes efforts sur le développement des fonctionnalités plutôt que sur un style graphique trop technique.

Tous les éléments visuels ont été créés en vectoriel sur Adobe Illustrator, ce qui garantit une excellente qualité d'image ainsi qu'une grande lisibilité dans la conception des différents assets.

Voici la map finale utilisée pour le projet.



# Intérieur maison 1



L'intérieur de la maison située à droite a été conçu de manière simple. L'objectif principal était d'assurer le bon fonctionnement du jeu ; l'aspect esthétique passait donc au second plan. J'ai préféré consacrer davantage de temps au développement plutôt qu'au design, car un intérieur très élaboré n'aurait aucune utilité si les fonctionnalités essentielles comme le changement de scène n'étaient pas opérationnelles faute de temps. Autrement dit, un décor parfaitement détaillé ne servirait à rien s'il n'était finalement pas visible en jeu. C'est pourquoi l'aménagement actuel de cette maison reste volontairement minimaliste. Il serait toutefois tout à fait possible d'ajouter davantage d'éléments décoratifs dans une version ultérieure.

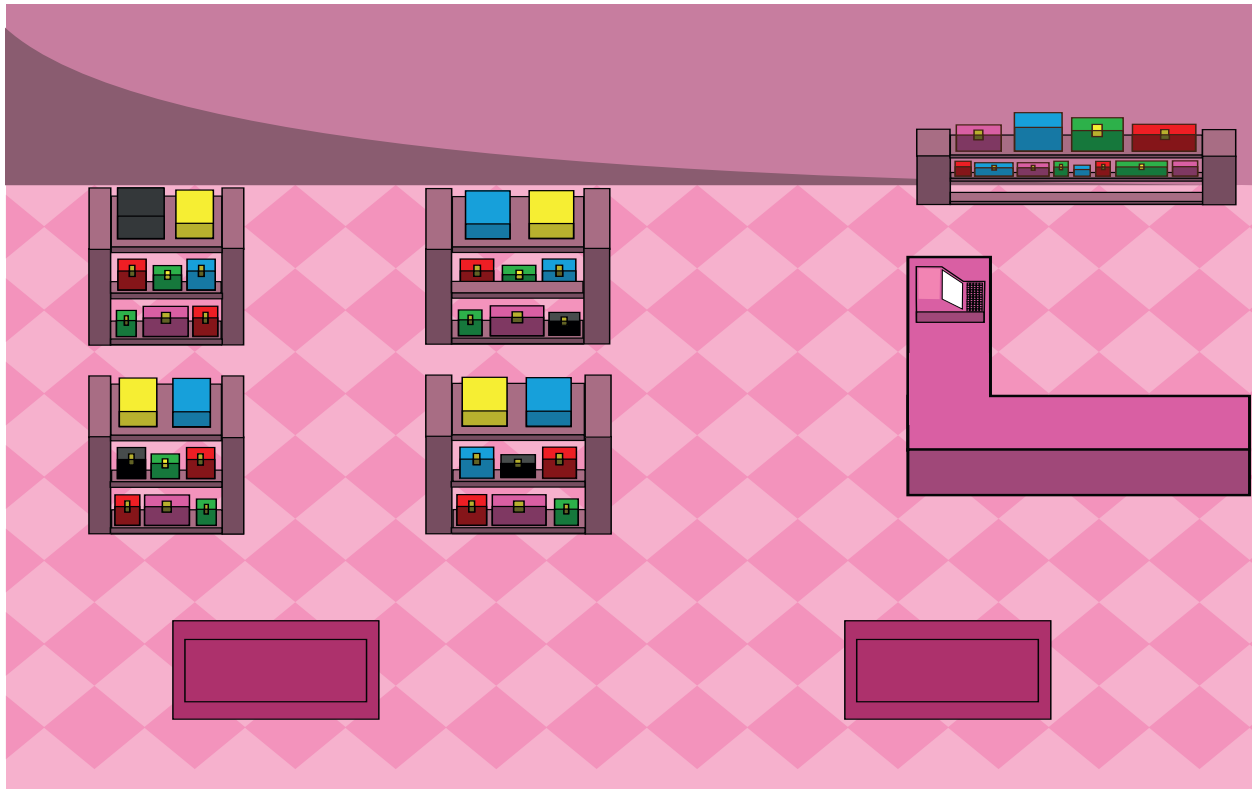
Concernant le design, j'ai créé cet intérieur sans aucune inspiration externe, uniquement à partir de mes propres idées et envies. J'ai ajouté quelques éléments essentiels :

- une bibliothèque
- une table à manger avec des sièges
- une fenêtre
- un cadre photo

afin de situer clairement le joueur dans la maison du personnage principal.

La perspective étant longue et complexe à réaliser, elle n'est pas parfaite, mais elle reste suffisante pour assurer une bonne compréhension de l'espace et s'inscrit dans la démarche globale du projet.

# Intérieur maison 2

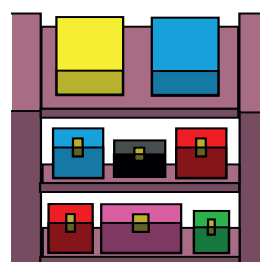


L'intérieur de la seconde maison, située sur le côté gauche de la map, est conçu pour représenter l'intérieur d'un magasin. On y retrouve différents rayons ainsi qu'une caisse, ce qui permet d'identifier clairement le lieu. Comme l'extérieur du bâtiment comporte deux portes, j'ai ajouté deux tapis à l'intérieur afin de simuler deux entrées et sorties distinctes. La couleur rose reste dominante, afin de conserver l'esprit général du jeu et l'esthétique déjà établie.

Ces éléments permettent d'ajouter de la vie au décor et de maintenir un environnement chaleureux et convivial.

La perspective n'est pas parfaitement respectée, mais comme expliqué précédemment, cet aspect n'était pas un objectif prioritaire du projet. Mon intention était avant tout de me concentrer sur le développement du jeu plutôt que sur un rendu graphique complexe et parfaitement réaliste.

Pour faciliter l'intégration dans le jeu, j'ai enregistré le fond et les quatre étals séparément. J'expliquerai plus en détail cette organisation dans la partie consacrée au développement.



# Personnage principal

Ce personnage provient d'un forum dédié à RPG Maker MV. Son utilisation et sa modification sont libres. J'ai simplement modifié les couleurs d'une sprite sheet existante afin d'obtenir un rendu visuel correspondant à mes attentes. Ce choix m'a permis de gagner un temps précieux et de me concentrer sur les aspects techniques du jeu plutôt que sur le design, qui n'était pas prioritaire dans ma planification.

Au départ, j'avais envisagé de créer le personnage moi-même en pixel art, mais cette tâche s'est révélée bien plus complexe et chronophage que prévu, notamment à cause des difficultés liées aux proportions et aux mouvements. Le personnage choisi, déjà pixelisé, s'intègre parfaitement à l'esthétique 2D inspirée de Pokémon.

Pour modifier les couleurs, j'ai d'abord vectorisé l'image, puis utilisé la baguette magique dans Adobe Illustrator, une méthode plus rapide que la sélection pixel par pixel. L'utilisation de cette sprite sheet existante garantit des animations fluides et cohérentes, tout en respectant les délais de développement.

Pour créer une sprite sheet parfaitement utilisable, j'ai dû positionner mes personnages le plus rapproché possible, afin que Phaser puisse reconnaître correctement les dimensions de chaque frame lors de la lecture de la feuille d'animation.



# PNJ

Pour ce PNJ, j'ai demandé à ChatGPT de générer un personnage à l'aide du prompt suivant :

« Crée-moi un personnage du même style, mais qu'on puisse différencier facilement grâce à d'autres couleurs. Reste quand même dans les tons rouge-rose, mais en version sorcière. »  
(01.12.2025)

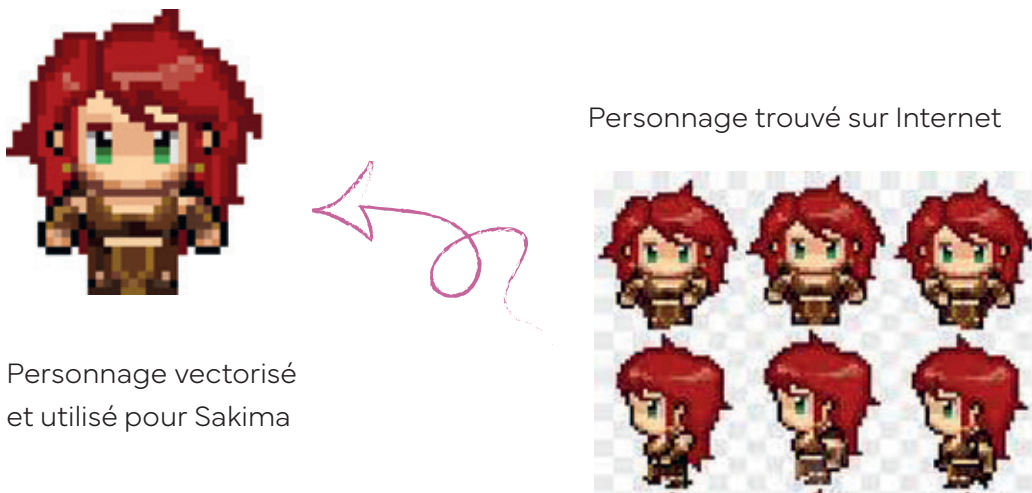
ChatGPT m'a proposé une version que j'ai ensuite améliorée en la revectorisant. Les couleurs me semblaient adaptées, et le chapeau de sorcière ouvre des possibilités intéressantes pour la suite du jeu, avec éventuellement des missions plus magiques.

À l'origine, je disposais de deux PNJ qui ne s'intégraient pas du tout à l'univers Sakima. J'ai donc décidé de remédier à ce point en ajoutant des personnages plus cohérents avec l'esthétique et l'ambiance du jeu.



Pour le second PNJ, j'ai trouvé sur Internet une image libre de droits pour une utilisation non commerciale. Il a été assez difficile de trouver un style qui ne diffère pas trop du personnage principal. Ce PNJ est parfait, car mon personnage principal est petit avec des cheveux rouges, ce qui est également le cas pour ce PNJ. De plus, il jouera le rôle de sa maman, ce qui rend cette image idéale pour le projet.

Étant donné que la qualité de l'image était médiocre, je l'ai vectorisée, puis retouchée à la main, car la vectorisation automatique n'était pas parfaite.



# Pain

Pour le pain, j'ai simplement cherché une image en pixel art représentant un pain. Sur Shutterstock, j'en ai trouvé une.

Bien qu'elle ne soit pas libre de droits, son utilisation pour un projet scolaire sans aucune commercialisation me semblait acceptable. J'ai donc pris cette image et l'ai entièrement vectorisée à la main sur Adobe Illustrator.

Je la trouvais appropriée, car on comprend immédiatement de quoi il s'agit, ce qui est idéal puisqu'elle sera affichée en petite taille dans le jeu.

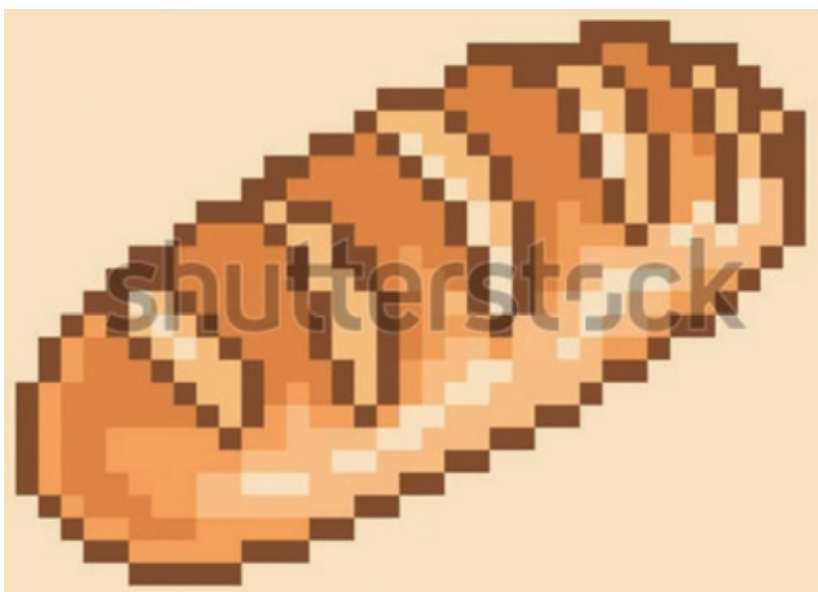
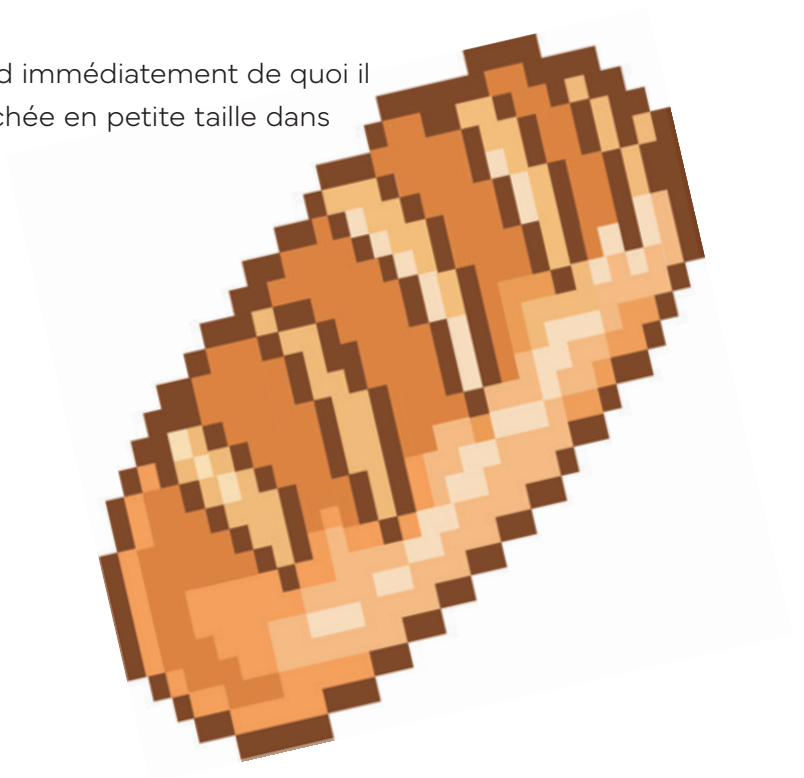
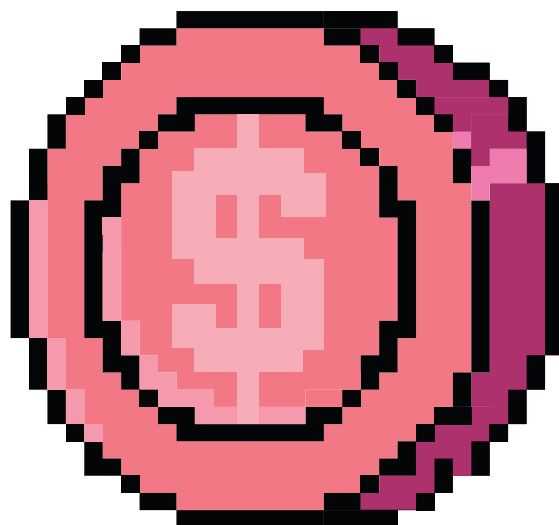
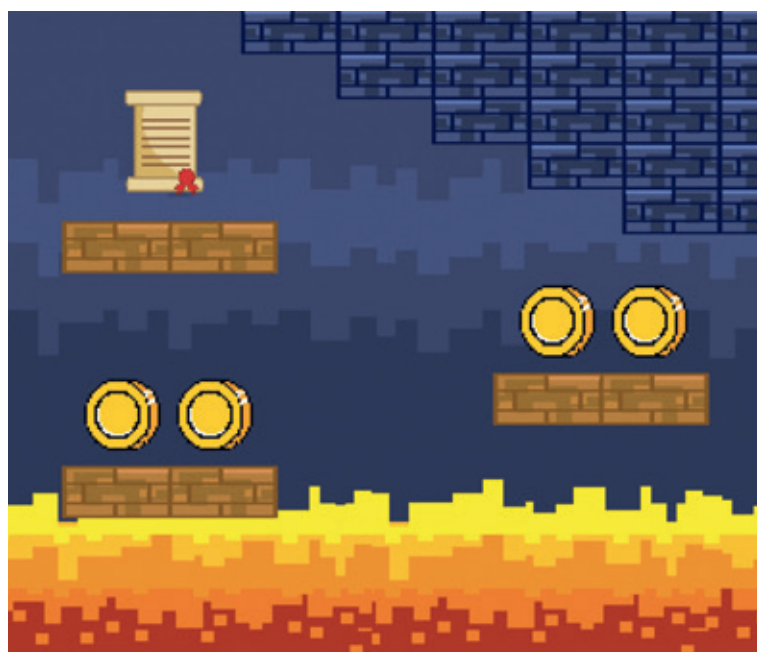


Image de Shutterstock

# Pièce

L'image de la pièce a été trouvée sur Freepik et est gratuite. Je l'ai ensuite vectorisée, puis j'y ai ajouté un signe dollar au centre afin de rendre son utilité immédiatement compréhensible. Enfin, je l'ai colorée en rose pour qu'elle s'intègre harmonieusement avec le reste du décor.



---

# Développement



# Phaser



Phaser est un framework open-source largement utilisé pour le développement de jeux vidéo 2D. Son architecture repose sur trois fonctions principales qui structurent le cycle de vie du jeu : preload, create et update. Cette organisation claire permet de séparer logiquement les différentes phases de développement.

Mon projet est composé de trois fichiers principaux :

- Page HTML : Conteneur du jeu dans le navigateur
- Fichier JavaScript : Contient tout le code Phaser et la logique du jeu
- Fichier CSS : Gère la personnalisation de l'interface et les effets visuels

Cette configuration initialise les dimensions de la scène, le moteur physique Arcade (avec débogage désactivé) et associe les trois fonctions clés au cycle de vie du jeu. L'instance du jeu est ensuite créée et rendue accessible globalement via `window.game`.

```
function initGame() {  
    var config = {  
        type : Phaser.AUTO,  
        width : 800,  
        height : 600,  
        parent : 'game-container',  
        physics : {  
            default : 'arcade',  
            arcade : { debug : false }  
        },  
        scene : {  
            preload,  
            create,  
            update  
        },  
    };  
};
```

```
window.game = new Phaser.Game(config) ;
```

---

La propriété scene indique à Phaser les trois fonctions à utiliser :

- **preload** : chargement des ressources (images, sprites, maps, PNJ...)
- **create** : création et placement des éléments (map, collisions, joueur, animations, PNJ, zones interactives...)
- **update** : boucle continue qui gère les déplacements, animations et interactions

Une fois la configuration définie, la ligne :

```
var game = new Phaser.Game(config ;
```

permet d'initialiser le jeu.

### Fonction preload

Dans preload, je charge toutes les ressources nécessaires : la map principale (mainmap), le sprite du joueur, les PNJ et différents éléments du décor. Sans ce chargement préalable, Phaser ne peut utiliser aucun asset.

### Fonction create

Dans create, j'installe tous les éléments visibles de la scène : la map, les collisions, le joueur (taille, collisions, animations), les PNJ ainsi que les zones d'interaction ou de transition. Les collisions doivent être réactivées ici, car Phaser ne les applique pas automatiquement.

### Fonction update

La fonction update est exécutée en boucle à chaque image du jeu. C'est elle qui gère les déplacements du joueur, les collisions, les transitions entre les cartes et toutes les interactions en temps réel. C'est le moteur principal qui anime le jeu et répond aux actions du joueur.

# VI: SPRITE, DEPLACEMENTS ET COLLISIONS

## 1.Sprite du personnage principal

### Chargement de la spritesheet

La spritesheet était essentielle pour mieux comprendre les déplacements de mon personnage. En effet, grâce aux 12 frames d'une spritesheet, Phaser peut afficher la frame appropriée au bon moment et effectuer un cycle d'animation, ce qui facilite la compréhension du mouvement du personnage.

La spritesheet du personnage principal est chargée dans la fonction `preload()`, avec les dimensions exactes de chaque frame:

```
this.load.spritesheet('monSprite',  
'assets/sprite1.png', {  
    frameWidth : 67,  
    frameHeight : 81  
});
```



Phaser découpe ainsi automatiquement la spritesheet en 12 frames (3 frames × 4 directions).

### Gestion du sprite physique

Le sprite du joueur est créé comme un objet physique avec une hitbox ajustée :

```
this.player = this.physics.add.sprite(420, 300, 'monSprite',  
0);
```

Les coordonnées (420, 300) sont les coordonnées x et y du point de spawn du personnage, positionné stratégiquement au centre de la map pour permettre une exploration immédiate de l'environnement.

## Création des animations

Les animations sont définies dans `create()` à partir des frames correspondantes:

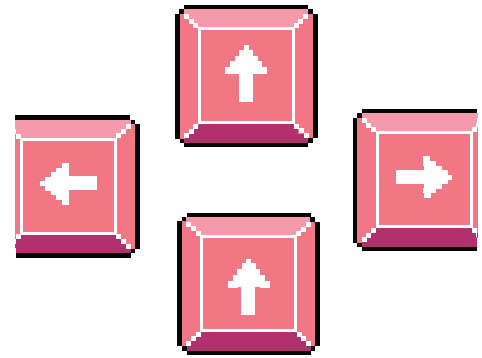
```
const anims = [  
  { key : 'down', start : 0, end : 2, rate : 10 },  
  // Bas : frames 0-2, 10 FPS  
  { key : 'left', start : 3, end : 5, rate : 10 },  
  // Gauche : frames 3-5, 10 FPS  
  { key : 'right', start : 6, end : 8, rate : 10 },  
  // Droite : frames 6-8, 10 FPS  
  { key : 'up', start : 9, end : 11, rate : 12 }  
  // Haut : frames 9-11, 12 FPS  
];  
  
anims.forEach(a => this.anims.create({  
  key : a.key,  
  frames : this.anims.generateFrameNumbers('monSprite', {  
start : a.start, end : a.end } ),  
  frameRate : a.rate, // 10-12 FPS - assez pour la fluidité,  
pas trop pour la performance  
  repeat : -1 // Répétition infinie pendant le déplacement  
}));
```

Choix des FPS: Les animations horizontales et vers le bas utilisent 10 FPS, tandis que l'animation vers le haut utilise 12 FPS pour compenser la perspective et donner une impression de mouvement plus naturel. Chaque animation contient 3 frames consécutives et se répète tant que le joueur continue son mouvement.



## 2. Déplacements du joueur

### Contrôles clavier



La configuration des contrôles est établie dans la fonction `create()` :

```
this.cursors = this.input.keyboard.createCursorKeys();
this.interactKey = this.input.keyboard.addKey(Phaser.Input.Keyboard.KeyCodes.SPACE);
```

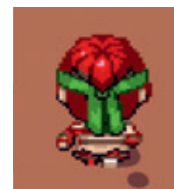
Les déplacements sont gérés dans la fonction `update()` appelée à chaque frame :

Pour réinitialiser la vitesse à chaque frame :

```
this.player.setVelocity(0);
```

La détection des touches directionnelles :

```
if (this.cursors.left.isDown) {
    this.player.setVelocityX(-150);
    this.player.anims.play('left', true);
} else if (this.cursors.right.isDown) {
    this.player.setVelocityX(150);
    this.player.anims.play('right', true);
} else if (this.cursors.up.isDown) {
    this.player.setVelocityY(-150);
    this.player.anims.play('up', true);
} else if (this.cursors.down.isDown) {
    this.player.setVelocityY(150);
    this.player.anims.play('down', true);
}
```



si aucune touche n'est pressée alors l'animation s'arrête avec :

```
} else {  
    this.player.anims.stop() ;  
}
```

## Logique de fonctionnement

- Vitesse constante:  $\pm 150$  pixels par seconde pour un déplacement rapide mais contrôlé
- Animation contextuelle : Seule l'animation correspondant au mouvement en cours s'exécute
- Arrêt immédiat: Le personnage s'arrête instantanément quand aucune touche n'est pressée

## 3. Collisions et limites du monde

La hitbox (zone invisible utilisée pour détecter les collisions) du personnage est positionnée stratégiquement pour correspondre à la perspective visuelle du jeu :

```
this.player.body.setSize(30, 20) ;  
this.player.body.setOffset(18, 57) ;
```

Ce qui permet des collisions réalistes: Seule la zone des « pieds » du personnage entre en collision

Interaction précise: Évite les collisions avec la partie supérieure du sprite



## Création des zones de collision

Les zones de collision sont créées dynamiquement pour chaque map:

```
const blocksData = [  
  [458, 306, 350, 300],  
  [458, 0, 185, 233],  
  [704, 0, 105, 233],  
  // ...  
];
```

Création de chaque bloc de collision:

```
blocksData.forEach(([x, y, width, height]) => {  
  this.collisionGroup.create(x + width / 2, y + height / 2,  
    'block')  
    .setDisplaySize(width, height)  
    .setVisible(false)  
    .refreshBody() ;  
});
```

Permet l'activation des collisions avec le joueur

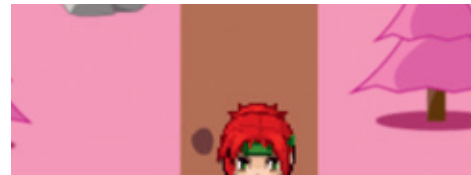
```
this.physics.add.collider(this.player, this.collisionGroup);
```

Grâce au mode debug activé (`debug : true`) dans la configuration de Phaser, il était beaucoup plus facile de visualiser les blocs de collision et de les positionner avec précision dans la scène. La destruction des groupes de collision lors des changements de map est nécessaire afin d'ajouter les collisions cohérentes avec l'environnement de la map.



## Téléportation des bords de l'écran sur la map principale

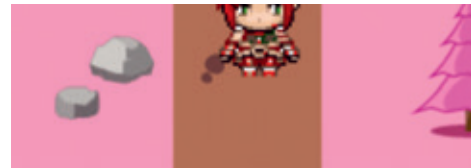
```
if (this.lastEntry === 'main') {
    if (this.player.y < 0) {
        this.player.setY(600);
    } else if (this.player.y > 600) {
        this.player.setY(0);
    }
}
```



Bas de l'écran



Désactivation des collisions mondiales: `this.player.setCollideWorldBounds(false)` permet cette et de téléportation en désactivant les collisions par défaut avec les bords du monde.



Haut de l'écran

## Intérieurs sans téléportation

```
else {
    if (this.player.y < 0) this.player.setY(0);
    if (this.player.y > 600) this.player.setY(600);
    if (this.player.x < 0) this.player.setX(0);
    if (this.player.x > 800) this.player.setX(800);
}
```

Dans les maisons, le joueur ne peut pas sortir des limites de l'écran, contrairement à la map principale où le et de téléportation crée un monde infini verticalement.

## Performance et optimisation

Groupes statiques: Utilisation de `staticGroup()` pour les éléments immobiles afin d'améliorer les performances, car cela évite des calculs physiques inutiles et rend le jeu plus fluide.

## Nettoyage systématique

Destruction des groupes de collision inutiles pour éviter des interactions dans les autres maps, chaque map ayant ses propres collisions.

# V2: INTERFACE, TRANSITIONS ET GESTION DES MAPS

## 1. Écran d'accueil du jeu

### Conception et développement

Pour créer l'écran d'accueil, j'ai utilisé DeepSeek avec le prompt spécifique :  
« Peux-tu me créer une jolie page d'accueil avec un bouton Start pour commencer le jeu ? Avec ce logo si possible. Quelque chose dans les tons roses, sympa. »

Après plusieurs itérations et ajustements visuels, l'écran final intègre :

- Une palette de couleurs roses cohérente avec l'univers féminin et chaleureux du jeu
- Le logo principal
- Des explications claires sur les interactions possibles
- Un bouton Start visible, attractif et ergonomique
- Un décor animé avec effet de particules pour dynamiser la présentation



### Système de particules animées

```
function createParticles() {  
    const particlesContainer = document.getElementById('particles') ;  
  
    const particleCount = 350 ;  
    // 350 est le nombre total de particules  
  
    for (let i = 0 ; i < particleCount ; i++) {  
        const particle = document.createElement('div') ;  
        particle.classList.add('particle') ;  
    }  
}
```

---

Intégration de propriétés aléatoires pour un effet naturel grâce à la fonction native Javascript :

`Math.random()` :

Chaque particule reçoit des valeurs aléatoires pour :

- Position horizontale (`left`): `Math.random() * 100%` place les particules à différents endroits sur l'axe X.
- Délai d'animation (`animationDelay`): `Math.random() * 15s` rend le mouvement décalé et moins uniforme.
- Taille (`width` et `height`) : `2 + Math.random() * 3px` crée des particules de tailles variées.
- Forme circulaire: `height = width` assure que chaque particule reste ronde.

Ainsi, l'usage de `Math.random()` rend le système de particules dynamique et naturel.

Initialisation au chargement de la page:

```
window.onload = createParticles ; // Initialisation au chargement de la page
```

## CSS des particules :

```
.particles {
  position : fixed ;
  width : 100 % ;
  height : 100 % ;
  z-index : -1 ;
}

.particle {
  position : absolute ;
  border-radius : 50 % ;
  background : rgba(255, 255, 255, 0.5) ;
  animation : float 10s linear infinite ;
}

@keyframes float {
  from { transform : translateY(100vh) ; }
  to { transform : translateY(-100vh) ; }
}
```



## Structure HTML et CSS

Conteneur principal: <div id= « homepage » > avec flexbox pour le centrage

- Bouton Start: Design avec un effet au survol
- Sections d'information: Typographie responsive avec clamp()
- Arrière-plan: Dégradés roses et ombres portées pour la profondeur

## Fonction de démarrage du jeu

```
function startGame() {
    document.getElementById('homepage').style.display = 'none' ;
    document.getElementById('game-container').style.display =
'block' ;
    gameStarted = true ;
    if ( !window.game) {
        initGame() ;
    }
}
```

## 2. Transitions entre scènes

### Principe général et schéma mental

Le jeu comprend trois environnements distincts :

- Map principale: Extérieur avec les deux maisons
- Maison de droite: Domicile de la maman
- Maison de gauche: Magasin Sakima (avec deux portes d'entrée)

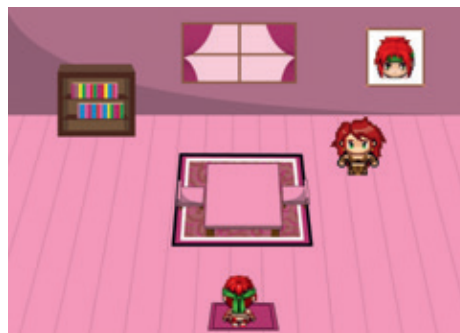
Chargement des maisons

On précharge l'image dans la fonction preload, puis on importe la fonction de la maison grâce à loadNewMapDroite() ou loadNewMapGauche().

C'est dans ces fonctions que tout sera initialisé pour les maisons.

Ensuite, on ajoute l'image avec :

```
this.currentMap = this.add.image(400,
300,
'map_maison_droite_v2')
    .setDisplaySize(800, 602)
    .setDepth(0) ;
```



Maison de droite

Cela permet de placer l'image à la taille  $800 \times 602$ , afin d'éviter les bordures noires sur les côtés. La profondeur est définie à 0 pour que tous les autres éléments s'ajoutent au-dessus.

Cet exemple concerne la maison de droite, mais pour la maison de gauche, le fonctionnement est le même.

On supprime également toutes les collisions, les dialogues et les objets provenant des autres maps.

## Variables de navigation et mémorisation

Permet de mémoriser par quelle porte l'utilisateur est entré pour faire correspondre la map et la sortie.

```
this.lastEntry
// dernière entrée utilisée
this.lastDoor
//porte spécifique utilisée

this.returnDoor
// Porte de retour vers la map principale
this.returnPoint
```

Ces variables permettent de :

- Repositionner précisément le joueur après une transition
- Conserver le contexte de navigation entre les maps
- Éviter les incohérences spatiales

## 3. Zones de transition et de sortie

### Map principale : points d'entrée

Crée une zone rectangulaires invisibles pour détecter l'entrée dans les maisons

```
this.transitionZoneDroite = new  
Phaser.Geom.Rectangle(645, 190, 90, 10) ; // Maison droite  
this.transitionZoneGauche1 = new Phaser.Geom.Rectangle(70, 190,  
60, 10) ; // Maison gauche porte de gauche  
this.transitionZoneGauche2 = new Phaser.Geom.Rectangle(159,  
196, 60, 10) ; // Maison gauche porte de droite
```

### Dimensionnement des entrées et sorties

- Largeur 50-90px: Assez large pour une détection facile
- Hauteur 10px: Fine pour éviter les déclenchements accidentels
- Position Y adaptée: Alignée avec les seuils de porte visuels

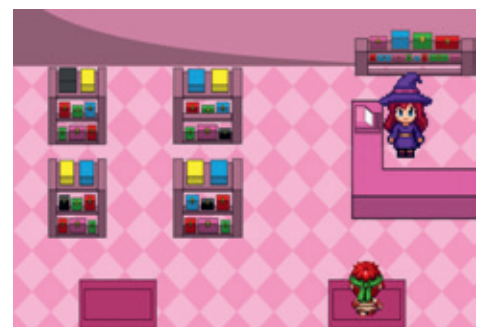
### Zones de sortie dans les maisons

Maison droite : zone en bas de l'écran

```
this.exitZone = new Phaser.Geom.  
Rectangle(380, 540, 160, 3) ;
```

Maison gauche : deux portes de sortie

```
this.exitZone2 = new Phaser.Geom.  
Rectangle(110, 595, 160, 3) ;  
// Porte gauche  
this.exitZone3 = new Phaser.Geom.  
Rectangle(550, 595, 160, 3) ;  
// Porte droite
```



Entrée maison de gauche (Shop) porte de gauche

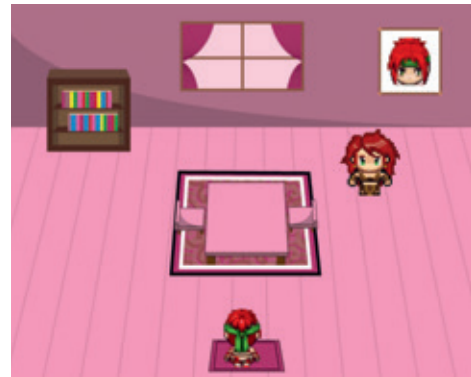
Avec une hauteur minimale de 3 px, cette zone permet une sortie fluide sans nécessiter un alignement précis du joueur parfait.

## 4. Logique des transitions

### Détection et traitement des transitions

Entrée dans la maison de droite

```
if ( !this.transitionLocked && this.transitionZoneDroite && Phaser.Geom.Rectangle.ContainsPoint(this.transitionZoneDroite, this.player)) {
```



Entrée maison de droite

Mémorisation de l'entrée :

```
    this.returnDoor = 'droite' ;  
    this.returnPoint = { x : 660, y : 195 } ;  
    loadNewMapDroite.call(this) ;  
    this.transitionLocked = true ;  
    this.time.delayedCall(1500, () => { this.transitionLocked  
= false ; }) ;  
}
```

### Positionnement précis du joueur pour l'entrée dans le shop

La position d'entrée dans le magasin varie selon la porte utilisée, et l'échelle du joueur est légèrement augmentée pour assurer une cohérence visuelle avec l'espace intérieur.

```
if (this.lastDoor === 'gauche2') {  
    this.player.body.reset(610, 595) ;  
} else if (this.lastDoor === 'gauche1') {  
    this.player.body.reset(180, 595) ;  
} else {  
    this.player.body.reset(400, 570) ;  
}  
this.player.setScale(1.2) ;
```

## Retour sur la map principale

Adaptation selon la sortie utilisée, avec le player qui reprend sa taille d'origine.

```
if (this.lastEntry === 'droite') {
    exitX = 660 ; exitY = 195 ;
// Sortie maison droite
} else if (this.lastEntry
=== 'gauche1') {
    exitX = 93 ; exitY = 186 ;
// Sortie porte gauche magasin
} else if (this.lastEntry
=== 'gauche2') {
    exitX = 170 ; exitY = 186 ;
// Sortie porte droite magasin
}
this.player.body.reset(exitX, exitY) ;
this.player.setScale(1) ;
```



Sortie droite



Sortie gauche

## 5. Maisons et gestion des collisions

### Maison de droite

```
function loadNewMapDroite() {
    this.player.body.reset(400, 570) ;
    this.player.setScale(1.2) ;

    this.player.setDepth(1) ;

    const newBlocks = [
        [0, 0, 808, 200], // Mur
        [92, 110, 130, 128], // Meuble gauche
        [340, 288, 130, 127], // Table
        [470, 320, 50, 34], // Chaise bas droite
        [290, 323, 50, 34], // Chaise gauche
    ] ;
    this.physics.add.collider(this.player, this.collisionGroup) ;
}
```



## Maison gauche (magasin Sakima)

Gestion des deux portes d'entrée/sortie

```
this.exitZone2 = new Phaser.Geom.Rectangle(110, 595, 160, 3) ;  
this.exitZone3 = new Phaser.Geom.Rectangle(550, 595, 160, 3) ;
```

## 6. Système de profondeur dynamique

### Gestion des étals dans le magasin

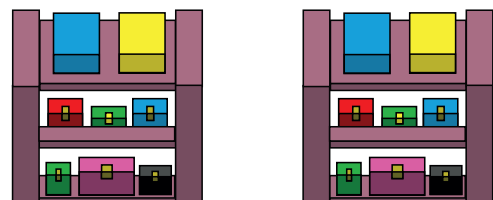
```
if (this.player.y < 248) {  
    this.player.setDepth(200) ;  
} else if (this.player.y >= 248 && this.  
player.y <= 407) {  
    this.player.setDepth(600) ;  
} else if (this.player.y > 413) {  
    this.player.setDepth(800) ;  
}
```



### Hierarchie des profondeurs

- Étals haut: setDepth(300) - toujours au second plan
- Étals bas: setDepth(700) - toujours au premier plan
- Joueur: 200-800 - variable selon la position verticale

Grâce à une gestion dynamique de la profondeur à l'aide de conditions (if), le joueur apparaît naturellement devant ou derrière les étals selon sa position, créant ainsi une illusion de profondeur plus réaliste.



---

## 7. Nettoyage et optimisation mémoire

### Destruction systématique des éléments

Lors du chargement d'une nouvelle map, tous les éléments des autres maps sont supprimés afin d'éviter toute incohérence logique, d'empêcher les résidus visuels des maps précédentes et de réduire la charge mémoire et graphique.

```
if (this.etalHautGauche) this.etalHautGauche.destroy() ;  
if (this.etalHautDroite) this.etalHautDroite.destroy() ;  
if (this.etalBasGauche) this.etalBasGauche.destroy() ;  
if (this.etalBasDroite) this.etalBasDroite.destroy() ;  
if (this.currentMap) this.currentMap.destroy() ;  
this.exitZone = null ;
```



# V3: INTERACTIONS ET FIN DE JEU

## 1. Écran de fin et réinitialisation

Structure HTML et CSS

L'écran de fin est un élément HTML/CSS intégré directement dans la page:



### HTML

```
<div id= « success-screen » >
  <h1>Mission Accomplie !</h1>
  <p>Tu as réussi à apporter le pain à ta maman !</p>
  <p>Félicitations !</p>
</div>
```

### CSS

```
#success-screen {
  display : none ;
  position : fixed ;
  background-color : rgba(0, 0, 0, 0.7) ;
  color : white ;
  z-index : 10 ;
  display : flex ;
  justify-content : center ;
  align-items : center ;
  text-align : center ;
}

#success-screen h1 {
  font-family : 'Courier New', monospace ;
  font-size : clamp(2rem, 5vw, 4rem) ;
  color : #ff69b4 ;
}
```

L'écran de fin n'est visible que lorsqu'il est explicitement appelé. Il est centré à l'écran et utilise la même police que le reste du jeu, avec une taille de police responsive afin de s'adapter à toutes les tailles d'écran.

## JavaScript

```
function showSuccessScreen() {
    document.getElementById('game-container').style.display =
'none' ;

    const successScreen = document.getElementById('suc-
cess-screen') ;
    successScreen.style.display = 'flex' ;

    setTimeout(() => {
        successScreen.style.display = 'none' ;
        document.getElementById('homepage').style.display =
'block' ;
        resetGame() ;
    }, 5000) ;
}
```

## Réinitialisation complète du jeu

```
function resetGame() {
    gameStarted = false ;
    hasMetPnj2 = false ;
    currentDialogState = 'initial' ;
    hasCoin = false ;
    hasBread = false ;
    coinSpawned = false ;
    breadSpawned = false ;
    missionCompleted = false ;
    pnj1PriceDialogCompleted = false ;
    pnj1BuyDialogCompleted = false ;
}
```

Redémarrage propre de Phaser

```
    if (window.game) {  
        window.game.destroy(true); // Destruction complète  
avec nettoyage  
        window.game = null ;  
    }  
}
```

L'écran de jeu est appelé automatiquement à la fin du dernier dialogue (`updateFinalBreadDialog()`) avec un délai de 1 seconde pour une transition fluide.

## 2. PNJ (Personnages Non-Jouables)

### PNJ de droite (la maman)

```
this.pnjDroite = this.physics.add.  
sprite(600, 350, 'pnj2') ;  
this.pnjDroite.setScale(0.17) ;  
this.pnjDroite.setOrigin(0.5, 1) ;  
this.pnjDroite.setDepth(500) ;  
this.pnjDroite.body.setImmovable(true) ;  
this.pnjDroite.body.setSize(150, 60) ;  
this.pnjDroite.body.setOffset(470, 220) ;
```



### PNJ de gauche (vendeuse)

```
this.pnj1 = this.physics.add.staticSprite(673, 300,  
'pnj1') ;  
this.pnj1.setScale(0.19) ;  
this.pnj1.setOrigin(0.5, 1) ;  
this.pnj1.setDepth(500) ;  
this.pnj1.body.setSize(30, 15) ;  
this.pnj1.body.setOffset(230, 370) ;
```

## Justification des échelles différentes

Les deux PNJ ont été redimensionnés à 17% et 19% de leur taille initiale, car ils avaient été créés trop grands sur Illustrator. Les valeurs de mise à l'échelle diffèrent volontairement, puisque les PNJ n'ont pas la même taille de base: le vendeur est plus grand, tandis que la maman est plus petite.

## Zones d'interaction des PNJ

PNJ de droite (la maman)

```
this.pnjDroiteInteractionZone = new Phaser.Geom.Circle(600, 350, 120) ;
```

PNJ de gauche (vendeuse)

```
this.pnjInteractionZone = new Phaser.Geom.Circle(673, 310, 120) ;
```

Avec un rayon de 120px la distance est optimale pour une interaction intuitive sans être trop intrusive.

# 3. Système de dialogues avancés

## Mémorisation du contexte de dialogue

```
this.lastInteractedPnj = null ;  
this.currentPnj = null ;  
this.dialogStep = 0 ;  
this.isInConversation = false ;  
Logique conditionnelle complète
```

Les dialogues évoluent selon un arbre de décision basé sur les variables d'état:

1. A rencontré la maman
2. N'a pas de pièce
3. N'a pas de pain
4. N'a pas encore demandé le prix

## Logique de décision des dialogues

PNJ1 – Vendeur du Sakima Shop

- Avant le premier contact avec la maman :  
Dialogue initial de bienvenue.
- Sans la pièce :  
Dialogue indiquant le prix du pain (1 pièce).
- Avec la pièce :  
Dialogue d'achat, avec échange de la pièce contre le pain.
- Avec le pain :  
Aucun nouveau dialogue ; la mission est considérée comme terminée

## PNJ2 (Maman)

- Premier contact :  
Demande au joueur d'aller acheter du pain.
- Sans la pièce :  
Elle donne la pièce au joueur et déclenche l'apparition de l'objet.
- Avec la pièce :  
Elle rappelle au joueur d'aller au magasin pour acheter le pain.
- Avec le pain :  
Dialogue final, menant à la fin de la mission et à l'affichage de l'écran de réussite.

```
this.time.delayedCall(2000, () => { // Délai de 2 secondes
    if (this.isInteractiveDialog && this.isInConversation) {
        this.dialogStep++; // Incrémentation de l'étape
        updateInteractiveDialog.call(this); // Passage au
message suivant
    }
});
```

## Durées des dialogues et pauses

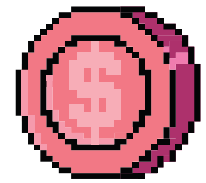
- 500 ms (0,5 seconde): Transition rapide entre deux répliques d'un même échange, pour un déroulement fluide de la conversation.
- 2000 ms (2 secondes): Temps de lecture confortable pour le joueur, permettant de comprendre chaque réplique sans précipitation.
- 3000 ms (3 secondes): Pause significative entre les tours de parole, utilisée pour marquer une réflexion, un changement de personnage ou un moment important dans le dialogue.

## Interruption automatique des dialogues

```
if (this.isInConversation) {  
    if (this.currentPnj === this.pnj1 && !isPlayerNearPNJ1) {  
        cancelDialog.call(this); // Annulation si éloignement  
du vendeur  
    } else if (this.currentPnj === this.pnjDroite && !isPlayerNear-  
PNJDroite) {  
        cancelDialog.call(this); // Annulation si éloignement  
de la maman  
    }  
}
```

Fonction `cancelDialog()` : Réinitialise proprement toutes les variables de dialogue et masque les bulles.





## 4. Système de récolte et gestion d'objets

### Conditions de récolte strictes

#### Pièce :

```
if (this.showingCoinInstruction && spaceJustPressed
&& !hasCoin) {
    collectCoin.call(this) ;
}
```

Pain:

```
if (this.showingBreadInstruction && spaceJustPressed && !has-
Bread) {
    collectBread.call(this) ;
}
```

#### Vérification :

- `showing...Instruction` : L'instruction visuelle est affichée
- `spaceJustPressed` : Appui unique sur ESPACE
- `!has...` : Le joueur ne possède pas déjà l'objet

### Instructions visuelles contextuelles

Génération du texte d'instruction pour la pièce

```
this.coinText = this.add.text(this.pnjDroite.x + 120, this.
pnjDroite.y,
    'Appuyez sur ESPACE\npour récupérer la pièce', {
    fontSize : '10px',
    fill : '#FFFFFF',
    fontFamily : 'Courier,
monospace',
    align : 'center',
    backgroundColor : '#000000'
});
```



Design cohérent: Police Courier, texte centré, fond noir, même style que les bulles de dialogue mais couleurs inversées.

## Portage visuel des objets

```
this.playerCoin = this.add.image(this.player.x + 36, this.player.y + 25, 'coin');
this.playerCoin.setScale(0.3);
this.playerCoin.setDepth(this.player.depth + 1);
```

```
if (this.playerCoin) {
    this.playerCoin.setPosition(this.player.x + 36, this.player.y + 25);
}
```

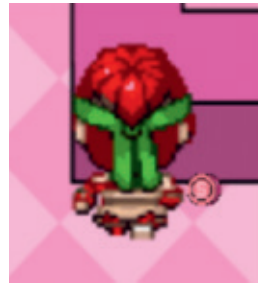
Décalage (36, 25): Position approximative de la « main » du personnage, ajustée pour un rendu acceptable.

## Système de dépôt simplifié

Au début, je pensais utiliser la touche ESPACE pour déposer les objets. Cependant, je me suis rendu compte que cela aurait permis au joueur de laisser l'objet n'importe où, ce qui aurait ajouté de la complexité au jeu en nécessitant des mécanismes supplémentaires pour le récupérer. J'ai donc opté pour une solution plus simple: l'objet disparaît automatiquement lors du dialogue, ce qui simplifie la logique tout en conservant la cohérence du jeu.

Ce qui a permis de:

- Réduire la complexité des contrôles (un seul usage d'ESPACE)
- Fluidifier la progression (disparition automatique lors des dialogues)
- Renforcer la narration (objets « donnés » plutôt que « posés »)



## Mécanisme de respawn des objets s'ils n'ont pas été ramassés

Si la pièce ou le pain n'a pas été ramassé et que le personnage sort de la maison, alors, lorsqu'il y rentre de nouveau, l'objet réapparaît après quelques secondes afin de permettre au joueur de le récupérer et de poursuivre sa quête.

```
if (coinSpawned && !hasCoin) {
    // Vérifie si la pièce a été générée mais n'a pas été
    ramassée par le joueur
    this.time.delayedCall(500, () => {
        // Planifie un appel différé de 500 ms (0,5 seconde)
        spawnCoin.call(this) ;
        // Rappelle la fonction spawnCoin pour régénérer la
    pièce
    });
}
```

Si tu veux, je peux aussi t'aider à uniformiser tout le document (titres, styles, commentaires de code) pour qu'il ressemble à une vraie documentation de projet ou à un rapport de soutenance.

## 5. Système de bulles de dialogue dynamiques

### Création et positionnement intelligent

```
function showDialogBubble(speaker,
message) {
    let bubbleX, bubbleY ;

    if (speaker === this.player) {
        bubbleY = this.player.y - 60 ;
    } else if (speaker === this.pnj-
Droite) {
        bubbleY = speaker.y - 180 ;
    } else if (speaker === this.pnj1)
    {
        bubbleY = speaker.y - 125 ;
    }
}
```



La position X de la bulle de dialogue est toujours centrée sur le locuteur

```
bubbleX = speaker.x ;
```

## Adaptation automatique de la taille

La taille de la bulle de dialogue s'adapte à la taille du texte. grâce à :

```
const tempText = this.add.text(0, 0, message, {
    fontSize : '10px',
    fill : '#000000',
    fontFamily : 'Courier, monospace',
    align : 'center',
    lineSpacing : 4
});
```

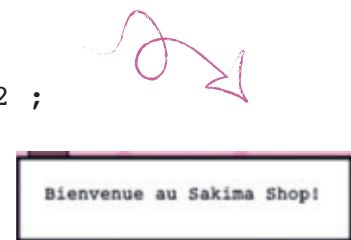
```
const textWidth = tempText.width ;
const textHeight = tempText.height ;
tempText.destroy() ; // Texte temporaire uniquement pour la mesure
```

Création de la bulle aux dimensions adaptées :

```
const padding = 15 ;
const bubbleWidth = textWidth + padding * 2 ;
const bubbleHeight = textHeight + padding * 2 ;
```

```
function updateDialogBubblePosition() {
    if (this.currentPnj === this.player) {
        bubbleX = this.player.x ;
        bubbleY = this.player.y - 60 ;
        // Suit le joueur
    }

    if (this.dialogBubble && this.dialogText) {
        this.dialogText.setPosition(bubbleX, bubbleY) ;
    }
}
```



## Gestion des profondeurs de la bulle de dialogue

```
this.dialogBubble.setDepth(1000) ;  
this.dialogText.setDepth(1001) ;
```

La bulle de dialogue est placée au premier plan, avec le texte affichée au-dessus de celle-ci, garantissant une lecture claire et une bonne visibilité pendant les interactions.

## 6. Cooldown et prévention des abus

### Système de cooldown général

Le cooldown empêche les interactions trop rapprochées pour éviter les actions involontaires. Après chaque interaction avec un PNJ, un délai de 500ms est imposé avant de pouvoir interagir à nouveau.

```
if (this.pnj1 && this.pnjInteractionZone && this.canInteract) {  
}
```

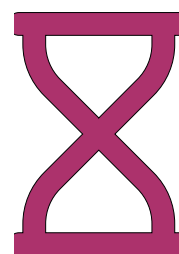
Le système utilise `this.interactionCooldown` pour bloquer temporairement les interactions. Il est activé immédiatement après une action puis désactivé après 500ms via un appel défini de Phaser.

### Cooldown spécifique pour ESPACE

```
if (spaceJustPressed) {  
    this.interactionCooldown = true ;  
    this.time.delayedCall(1000, () => { this.interactionCooldown = false ; }) ;  
}
```

### Délais définis :

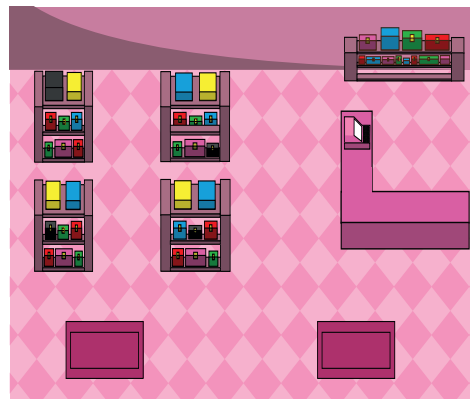
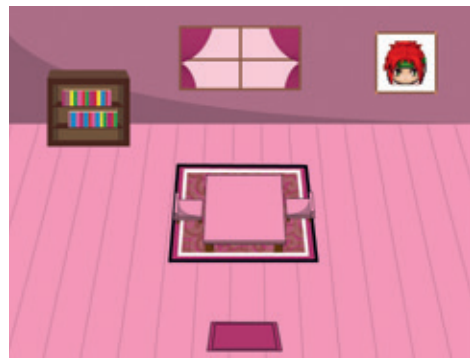
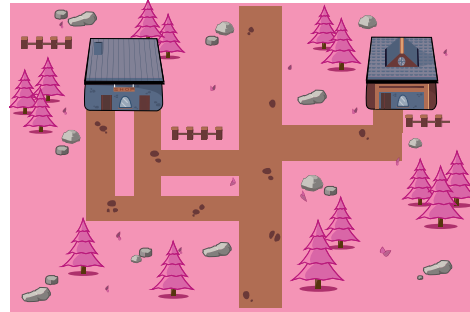
- 500ms (0,5 seconde): Interactions avec PNJ (dialogue)
- 1000ms (1 seconde): Appui sur ESPACE (récolte)
- 1500ms (1,5 seconde): Transitions entre maps



## 7. Architecture globale des états du jeu

### Variables d'état principales

```
VARIABLES GLOBALES
let gameStarted = false ;
// Si le jeu a déjà démarré ou non
let hasMetPnj2 = false ;
// Indique si le joueur a rencontré la maman (PNJ2)
let currentDialogState = 'initial' ;
// État actuel des dialogues
let hasCoin = false ;
// Indique si le joueur possède déjà la pièce
let hasBread = false ;
// Indique si le joueur possède déjà le pain
let coinSpawned = false ;
// Indique si la pièce a déjà été générée
let breadSpawned = false ;
// Indique si le pain a déjà été généré
let missionCompleted = false ;
// Indique si la mission est terminée
let pnj1PriceDialogCompleted = false ;
// Dialogue du prix avec PNJ1 terminé
let pnj1BuyDialogCompleted = false ;
// Dialogue d'achat avec PNJ1 terminé
```



Progression linéaire guidée: Le joueur est dirigé naturellement à travers les états sans possibilité de séquence alternative, garantissant une expérience narrative cohérente.

### Intégration de tous les systèmes

L'architecture globale intègre harmonieusement:

- Système de navigation (maps, transitions, collisions)
- Système d'interaction (dialogues, récolte, portage)
- Système visuel (animations, profondeur, interfaces)
- Système narratif (états, progression, n)

# Validations

À la fin de mon code, j'ai validé celui-ci sur le site [validator.w3.org](https://validator.w3.org), qui permet de détecter les erreurs majeures. Grâce aux captures d'écran ci-dessous, on peut voir que mon code HTML et CSS sont valides selon les normes W3C.

## HTML

Document checking completed. No errors or warnings to show.

## CSS

Document checking completed. No errors or warnings to show.

## JAVASCRIPT

J'ai aussi trouvé un site en ligne qui valide le JavaScript. Je ne sais pas s'il est fiable, mais il permet quand même de vérifier le code fourni.

Code is syntactically valid.



# Problèmes connus et améliorations possibles

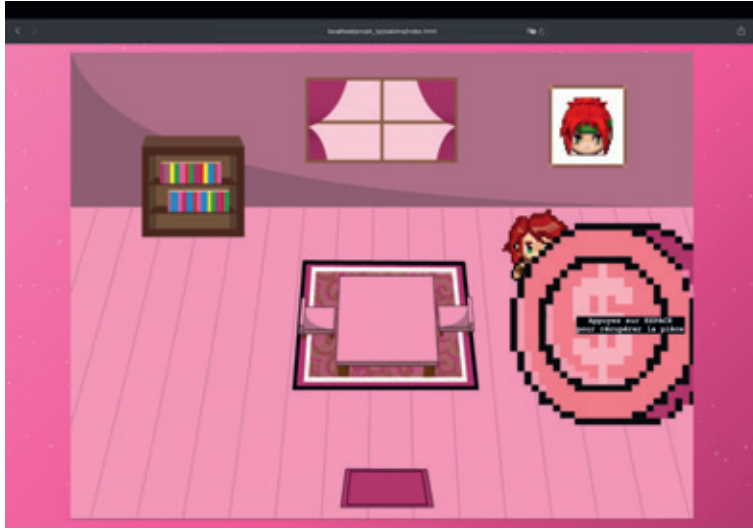
À la fin du développement de mon projet, j'ai testé mon jeu sur le navigateur Safari. C'est à ce moment-là que j'ai constaté un bug : le pain et la pièce apparaissaient beaucoup trop grands, au point que le personnage n'était plus visible à l'écran.

Je n'avais plus suffisamment de temps pour corriger ce problème et je ne souhaitais pas risquer d'introduire d'autres bugs sur Google Chrome. J'ai donc choisi de m'informer sur l'origine possible de ce dysfonctionnement. Selon un forum Stack Overflow, lorsque la propriété CSS `height: auto;` est utilisée, Safari peut ne pas l'interpréter de la même manière que Chrome, voire l'ignorer. Cela pourrait expliquer ce comportement, puisque mon game-container est défini avec `height: auto;`. Toutefois, il reste surprenant qu'aucun autre bug n'ait été détecté ailleurs dans le jeu.

La qualité graphique du jeu est également limitée. Initialement, j'avais tout créé en SVG, mais il était impossible d'intégrer les maps ou les personnages dans le jeu avec ce format, car ils n'apparaissaient pas correctement. J'ai donc dû convertir les éléments en PNG, un format compressé, ce qui a entraîné une perte de qualité visible. Ce choix a également posé problème pour les captures d'écran du rapport, qui sont de faible résolution.

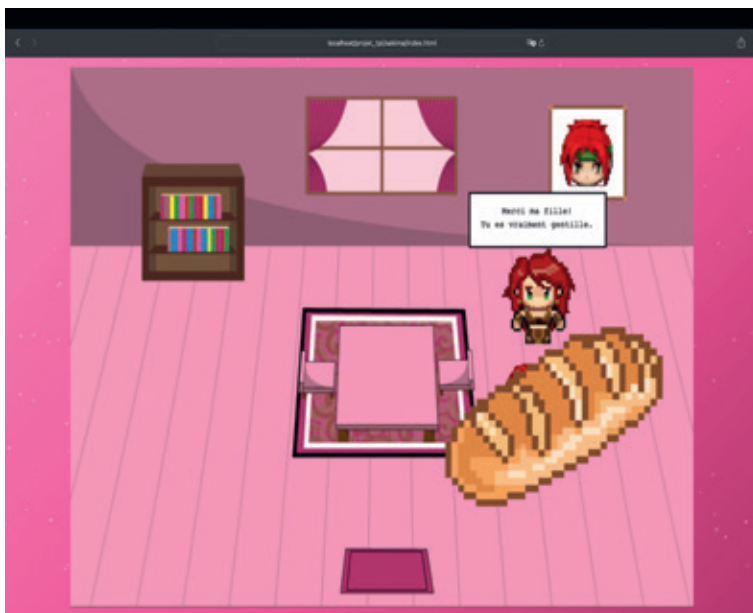
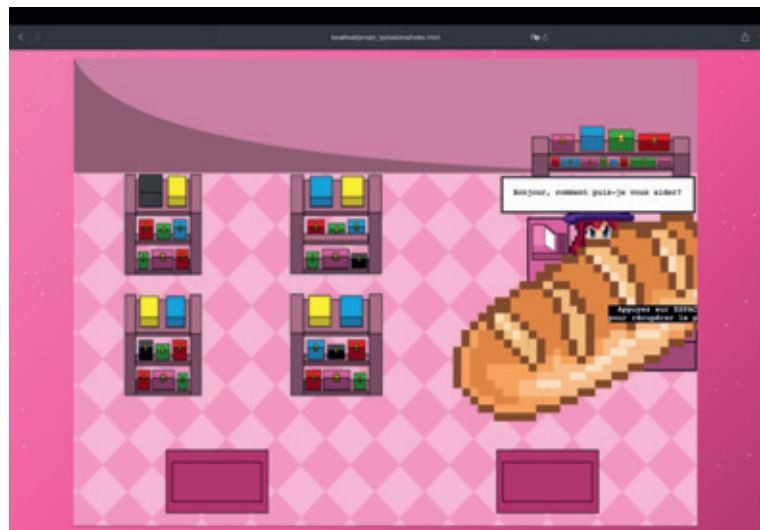
Concernant les évolutions possibles, il serait envisageable de prolonger l'histoire. Par exemple, après que la fille a ramené le pain à sa mère, une nouvelle partie de la carte pourrait se débloquer, donnant accès à un chemin situé dans la partie supérieure de la map. La fille pourrait alors explorer cette zone, rencontrer d'autres PNJ et accomplir de nouvelles quêtes.

Enfin, un léger problème de perspective est présent sur l'étal situé en haut à droite : les objets semblent passer derrière le support. Toutefois, ce détail est très discret et n'a pas d'impact significatif sur l'expérience de jeu.



Pour récupérer la pièce et sur le chemin on ne voit pas le personnage principal ce qui crée des complications avec les colisions.

Pour récupérer le pain. on ne voit même plus le personnage



Pour donner le pain à la maman.

# Gestion de projet

## Analyse de la planification

### Partie administration

Pour ce qui est de l'administration, rédaction du cahier des charges, planification des séances et rédaction des PV, j'ai passé 4 h 40, contre 8 h prévues. J'ai donc consacré un peu plus de la moitié du temps estimé. Cela m'a permis de réaliser que la partie administrative n'est pas très longue et que je l'avais largement surestimée, notamment concernant les séances prévues toutes les deux semaines : au final, je n'en ai réalisées que deux sur l'ensemble du projet.

### Partie marketing

J'ai oublié d'inclure toute la partie analyse marketing dans ma planification initiale. Elle a été directement intégrée dans le rapport. Ce manque de prévision explique le temps supplémentaire nécessaire pour compléter cette section, qui n'était pas estimé dans le planning initial.

### Partie design

Pour le design, j'avais prévu 10 h, mais j'ai finalement passé 16 h 20, soit un dépassement de plus de 60 %. La principale raison est que je n'avais pas prévu de concevoir l'intérieur des maisons, ce qui a demandé beaucoup de temps supplémentaire. Ce retard s'explique donc principalement par cet oubli dans la planification.

### Partie développement

Pour le développement, j'avais estimé 27 h 45, alors que j'ai finalement passé 33 h 45, soit 6 heures de plus (plus de 20 % de dépassement). Ce dépassement reste raisonnable : le développement est difficile à estimer et certaines difficultés ou ajustements imprévus peuvent nécessiter du temps supplémentaire.

### Partie documentation

La partie documentation est liée au rapport écrit, au support PowerPoint ainsi qu'à la présentation orale de 20 minutes. J'avais initialement prévu un total de 16 h 20 pour l'ensemble de ces éléments.

Comme expliqué dans la partie marketing, ayant oublié de planifier cette phase en amont, il est logique que le nombre d'heures attribuées à la documentation du rapport soit trop élevé. Cependant, avec du recul, je considère que 12 heures uniquement pour le rapport étaient malgré tout utopiques.

À la fin de la rédaction du rapport, j'atteins un total de 23 heures 55, et il me reste encore à réaliser la présentation PowerPoint. Celle-ci, comme initialement prévu, devrait nécessiter environ 4 heures. Il est toutefois possible que ce temps soit légèrement dépassé, d'environ 20 %, afin de garantir une présentation claire, structurée et de qualité.

---

## Observations sur la planification

J'aurais peut-être pu détailler davantage ma planification. Par exemple, dans la partie développement, seules les catégories «codage visuel du jeu» et «codage des interactions en JavaScript» étaient prévues, alors que certaines tâches concernaient les deux catégories. J'aurais pu préciser des sous-tâches telles que l'intégration de la spritesheet, l'écran d'accueil ou encore les interactions avec les PNJ. Cela m'aurait permis de mieux comprendre où j'ai passé du temps et d'analyser cette partie du projet de manière plus fine.

Actuellement, ma planification s'élève à 78 heures et 40 minutes. Afin de respecter les délais du TPI (maximum 90 heures), il me reste donc un maximum de 11 heures et 20 minutes que je peux consacrer au PowerPoint. Pour éviter d'éventuels problèmes liés à l'impression, je vais clore ma planification à la fin de la rédaction du rapport, ce qui me laissera le temps de l'imprimer et de le relier. En effet, dans le cadre du TPI réel, le temps consacré à la présentation orale ainsi qu'au rapport n'est pas inclus dans la planification du projet.

Je considère que plus de 11 heures consacrées uniquement au PowerPoint sont largement suffisantes. J'approfondirai cette partie lors de ma présentation orale en y indiquant les heures exactes.

En estimant la réalisation du PowerPoint à un maximum de 6 heures, j'atteindrai un total de 84 heures et 40 minutes. Cela représente un dépassement du nombre total d'heures d'environ 36 %, soit plus d'un tiers par rapport à la planification initiale. Ce dépassement est donc significatif. Cependant, si l'on se réfère aux exigences du TPI réel, je reste dans la fourchette d'heures attendue, soit entre 70 et 90 heures.

Sans compter le PowerPoint, le dépassement est de 27 %, soit un peu plus d'un quart, ce qui reste acceptable.

# Conclusion Professionnelle

Selon le KPI défini en début de projet, l'objectif principal était que le projet soit terminé dans les délais. Deux interprétations peuvent être retenues. D'une part, si l'on se base strictement sur le temps initialement prévu pour ce TPI blanc, celui-ci a été dépassé. D'autre part, si l'on considère le nombre d'heures réellement nécessaires à la réalisation d'un TPI « officiel », ce projet peut être considéré comme ayant été mené à bien dans les temps.

Les bugs identifiés précédemment peuvent être corrigés sans difficulté majeure. Ils nécessitent principalement un temps supplémentaire ainsi qu'une documentation plus précise concernant les différences de comportement entre les navigateurs Safari et Google Chrome. En termes de contraintes, la principale a été le temps disponible. Avec seulement quatre périodes par semaine, soit trois heures hebdomadaires sur seize semaines (48 heures au total), le projet s'est révélé ambitieux pour le cadre imposé. De plus, l'apprentissage de Phaser, un framework jusqu'alors inconnu, a constitué une contrainte technique supplémentaire. La qualité du jeu reste également perfectible et pourrait être améliorée avec davantage de temps.

Concernant les perspectives d'évolution, plusieurs améliorations pourraient être envisagées. Il serait notamment possible d'ajouter de nouvelles maps reliées entre elles par des chemins, menant à d'autres maisons où le personnage pourrait acquérir de nouveaux objets et disposer d'un inventaire. L'intégration d'un système de combat inspiré de jeux tels que Pokémon pourrait également être envisagée afin de renforcer l'engagement et la fidélisation des utilisateurs.

Enfin, l'ajout d'un écran de connexion relié à une base de données permettrait au joueur de sauvegarder sa progression grâce à un compte utilisateur, offrant ainsi une expérience plus complète et continue.

---

# Conclusion personnelle

Ce travail m'a permis d'apprendre à mieux gérer mon temps, à maîtriser les outils Illustrator, ainsi qu'à réaliser une analyse marketing complète et détaillée. Mais surtout, il m'a permis de me familiariser avec le développement d'une application et d'un jeu web.

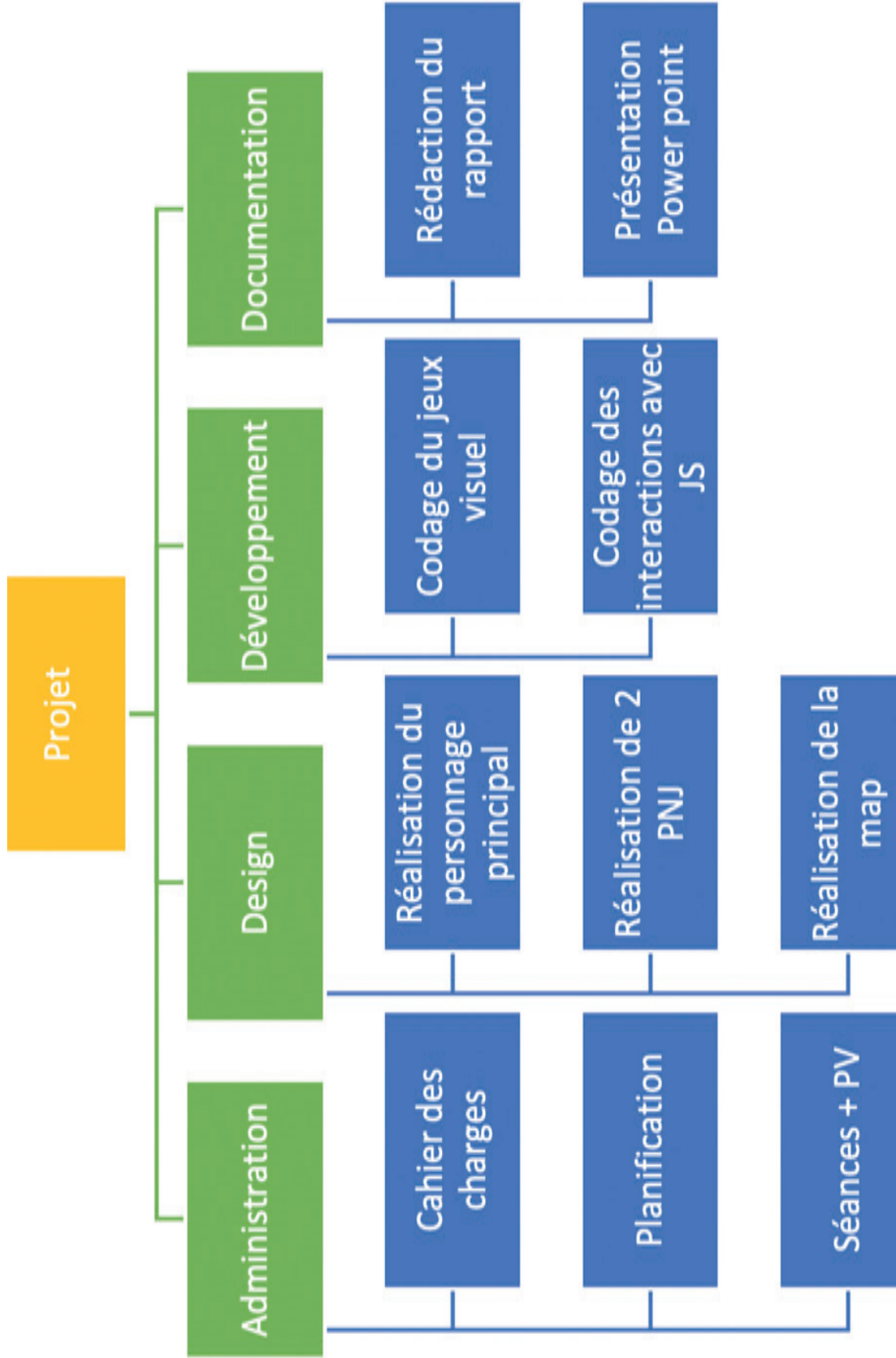
J'ai beaucoup apprécié ce projet, même s'il m'a parfois donné du fil à retordre. Je suis fière du livrable que j'ai pu rendre dans les délais. Mon jeu est fonctionnel, ce qui constituait pour moi l'objectif principal. Certes, il reste volontairement simple, mais il pourra être enrichi par la suite avec de nouvelles fonctionnalités.

Je pense avoir globalement respecté les délais. Même si j'ai passé plus de temps que prévu initialement, j'ai préféré me concentrer sur un rendu de qualité, proche des exigences d'un véritable TPI, afin de pouvoir apprendre de mes erreurs. En effet, si je n'avais développé le jeu qu'en partie ou si j'avais écourté la rédaction du rapport uniquement pour respecter le temps planifié, je n'aurais pas bénéficié d'un retour suffisamment détaillé pour m'améliorer avant le TPI officiel de mai.

En conclusion, je suis très fière de ce projet. J'ai appris à maîtriser Phaser, ce qui m'a permis de créer des changements de scènes, d'utiliser une spritesheet que je ne connaissais pas auparavant et de mettre en place des interactions avec des personnages, ainsi qu'un système de collecte d'objets. Ce projet était ambitieux, mais je suis satisfaite de l'avoir mené à terme dans les délais et d'avoir acquis de nombreuses compétences au cours de cette expérience. Grâce à ce projet, je dispose désormais d'atouts supplémentaires pour aborder mon véritable TPI de mai dans de bonnes conditions.

# Annexes

## Organigramme des tâches





# Planification

Projet :  
 Chef de projet :  
 Date planning :  
 MAJ planning :  
 Début projet :

Projet :  
 Available Zahid  
 mer 20 août 2025  
 dim 14 décembre 2025  
 mer 20 août 2025



No tâches ou Jalons	Tâches principales	Lots de tâches/lots	Charge de travail en h	Ressources - les collaborateurs par fonction et leur inscription dans le projet				Total ressources	Durées prévues en h	Remarques (dépendances, détails...)	Etat
				Accès	1	2	3				
1.	Administration										
1.1	Cahier des charges		0:30	100%			100%	0:30		TERMINÉ	1
1.2	Planification		0:30	100%			100%	0:30		TERMINÉ	1
1.3	Séances + PV		7:00	100%			100%	7:00		TERMINÉ	7
2.	Design										
2.1	Réalisation du personnage principal		3:00	100%			100%	3:00		TERMINÉ	3
2.2	Réalisation de 2 PNU		3:00	100%			100%	3:00		TERMINÉ	3
2.3	Réalisation de la map		4:00	100%			100%	4:00		TERMINÉ	4
Jalon 1 (10.08.2025)											
3	Développement										
3.3	V1		9:15	100%			100%	9:15		TERMINÉ	10
3.4	V2		9:15	100%			100%	9:15		TERMINÉ	10
3.5	V3		9:15	100%			100%	9:15		TERMINÉ	10
Jalon 2 (21.11.2025)											
4	Documentation										
4.1	Rédaction du rapport		12:00	100%			100%	12:00		TERMINÉ	12
4.2	Présentation Power Point		4:00	100%			100%	4:00		A FAIRE	4
Jalon 3 (17.12.2025)											
			<b>Total A FAIRE (8%)</b>	<b>4 h 00</b>	<b>Total A FAIRE (8%)</b>			<b>4 h 00</b>	<b>car des lots de tâches peuvent se chevaucher</b>		
			<b>Total EN COURS (7%)</b>	<b>0 h 00</b>	<b>Total EN COURS (7%)</b>			<b>0 h 00</b>			







# Journal de bord

Journal de bord

Projet : Projet

Chef de

Chargé du projet 78-40

F = facturable (noter x)

Date	Début	Fin	Durée	matin	après-midi	soir	Charge	Tâches principales	No	Lots de tâches	F	Travail effectué	
20 août 25	13:40	14:55	01:15	x			01:15	Administration	TP 1	Cahier des charges		Réalisation du cahier des charges	
	15:05	15:50	00:45	x			00:45	Administration	TP 1	Planification		Réalisation de la planification	
	16:00	16:35	00:35	x			00:35	Design	TP 2	Réalisation du personnage principal	x	Renseignement sur comment créer un design à inclure sur JavaScript	
24 août 25	12:00	13:00	01:00	x			01:00	Design	TP 2	Réalisation du personnage principal	x	Création des 6 premiers visuels de ma sprite sheet	
	13:00	14:00	01:00	x			01:00	Design	TP 2	Réalisation du personnage principal	x	Création des 6 derniers visuels de ma sprite sheet	
27 août 25	13:20	15:05	01:45	x			01:45	Design	TP 2	Réalisation du personnage principal	x	Finalisation de la création du personnage principal	
	15:15	16:40	01:25	x			01:25	Design	TP 2	Réalisation de la map	x	Commencement de la création de la map. La maison de droite.	
03 sept 25	13:30	14:00	00:30	x			00:30	Administration	TP 1	Séances + PV		Séances avec le prof.	
	14:00	14:15	00:15	x			00:15	Administration	TP 1	Séances + PV		Rédaction du PV de la séance du 03.09.2025	
	14:15	14:55	00:40	x			00:40	Administration	TP 1	Planification		Correction de la planification en fonction du PV.	
	15:05	15:45	00:40	x			00:40	Design	TP 2	Réalisation de 2 PNJ	x	Réalisation des 2 PNJ grâce à l'IA pour les changements de couleurs.	
	15:45	16:40	00:55	x			00:55	Design	TP 2	Réalisation de la map	x	Finalisation de la map avec la maison de droite, les chemins et les décors général.	
10 sept 25	13:20	14:00	00:40	x			00:40	Design	TP 2	Réalisation de la map	x	Recherches sur la manière d'intégrer des "barrières" pour que le personnage ne marche que sur les chemins et ne puisse	
	14:00	15:00	01:00	x			01:00	Design	TP 2	Réalisation de la map		Réalisation d'arbre pour la map	
	15:00	16:00	01:00	x			01:00	Design	TP 2	Réalisation de la map		Réalisation des cailloux et feuilles pour habiller un peu plus la map.	
	16:00	16:40	00:40	x			00:40	Design	TP 2	Réalisation de la map	x	Réalisation de la map en noir et blanc	
	18:00	19:00	01:00	x			01:00	Documentation	TP 4	Rédaction du rapport		Rédaction de l'introduction du rapport ainsi que la mise en page.	
	19:00	19:45	00:45	x			00:45	Documentation	TP 4	Rédaction du rapport		Rédaction de deux public-cible.	
	20:10	20:30	00:20	x			00:20	Design	TP 2	Réalisation de la map	x	Ajout d'une piscine, plus modification de la map noir et blanc.	
16 sept 25	16:00	16:40	00:40	x			00:40	Développement	TP 3	Codage des interactions JS	x	Suivi du tutoriel Phaser de la phase 1 à la phase 5	
	17:40	18:00	00:20	x			00:20	Développement	TP 3	Codage des interactions JS	x	Suivi du tutoriel Phaser de la phase 6 à la phase 10	
	18:00	18:30	00:30	x			00:30	Développement	TP 3	Codage visuel du jeu	x	Intégration de la map avec Phaser	
	18:30	19:00	00:30	x			00:30	Développement	TP 3	Codage visuel du jeu	x	Essai d'intégration du personnage sur la map	
17 sept 25	14:10	14:45	00:35	x			00:35	Développement	TP 3	Codage visuel du jeu		Séparation de la spritesheet en 12 phases distinctes	
	14:50	15:00	00:10	x			00:10	Administration	TP 1	Séances + PV	x	Séance avec le prof, pour montrer l'avancement du projet	
	15:25	16:30	01:05	x			01:05	Développement	TP 3	Codage visuel du jeu	x	Essai des collisions	
23 sept 25	11:05	12:25	01:20	x			01:20	Développement	TP 3	Codage visuel du jeu	x	Apprendre plus sur les collisions avec Phaser et l'intégration de rectangle noirs png.	
	14:05	14:55	00:50	x			00:50	Développement	TP 3	Codage visuel du jeu	x	Discussion avec le professeur sur la manière d'intégrer les sprite au code.	
	15:05	16:20	01:15	x			01:15	Design	TP 2	Réalisation du personnage principal	x	Amélioration du sprite pour pouvoir l'intégrer au jeu.	
24 sept 25	13:30	14:15	00:45	x			00:45	Développement	TP 3	Codage des interactions JS	x	Collisions	
	14:15	15:10	00:55	x			00:55	Développement	TP 3	Codage visuel du jeu	x	Intégration du Sprite correctement à mon personnage principal	
	15:10	15:20	00:10	x			00:10	Développement	TP 3	Codage des interactions JS	x	Mise à jour du carré autour du personnage	
	15:20	15:40	00:20	x			00:20	Développement	TP 3	Codage des interactions JS	x	Mise à jour des rectangle de collisions	
	15:50	16:35	00:45	x			00:45	Design	TP 2	Réalisation de la map	x	Design de la maison droite (intérieur)	
30 sept 25	11:00	11:45	00:45	x			00:45	Design	TP 2	Réalisation de la map	x	Finalisation de la map intérieure de la maison droite	
	11:45	12:30	00:45	x			00:45	Développement	TP 3	Codage des interactions JS	x	Théorie sur le changement de scènes avec Phaser	
	13:20	14:55	01:35	x			01:35	Développement	TP 3	Codage des interactions JS	x	Application du changement de scène dans mon code.	
	13:20	14:45	01:25	x			01:25	Design	TP 2	Réalisation de la map	x	Essai pour la maison de gauche.	
01 oct 25	14:45	15:30	00:45	x			00:45	Développement	TP 3	Codage des interactions JS	x	Gestion des collisions dans la nouvelle maison (en cours)	
	17:00	17:30	00:30	x			00:30	Documentation	TP 4	Rédaction du rapport	x	Amélioration du public cible	
08 oct 25	13:40	13:55	00:15	x			00:15	Documentation	TP 4	Rédaction du rapport	x	Structure du rapport.	
	13:55	14:40	00:45	x			00:45	Documentation	TP 4	Rédaction du rapport	x	Trouver les concurrents et création de la matrice de positionnement, avec des axes stratégiques.	
	14:40	15:00	00:20	x			00:20	Documentation	TP 4	Rédaction du rapport	x	Analyse de la matrice de positionnement.	
	15:00	15:30	00:30	x			00:30	Documentation	TP 4	Rédaction du rapport	x	Analyse SWOT.	
	15:30	16:20	00:50	x			00:50	Documentation	TP 4	Rédaction du rapport	x	Création de deux persona.	
	16:20	16:50	00:30	x			00:30	Documentation	TP 4	Rédaction du rapport	x	Création d'un KPI	
	16:50	17:00	00:10	x			00:10	Administration	TP 1	Planification		Mise à jour de la planification, et dépôt du rapport et de la planification sur le drive.	
04 nov 25	18:00	18:40	00:40	x			00:40	Développement	TP 3	Codage des interactions JS	x	Ajout des collisions dans la maison de gauche.	
	19:00	19:15	00:15	x			00:15	Développement	TP 3	Codage des interactions JS	x	Renseignement sur comment gérer la profondeur avec mes différents niveaux d'états	
	19:15	19:55	00:40	x			00:40	Développement	TP 3	Codage visuel du jeu	x	Essai de faire passer mon personnage devant et derrière les étals.	
05 nov 25	14:40	15:00	00:20	x			00:20	Développement	TP 3	Codage des interactions JS	x	Régler les sorties en fonction des différentes maisons. -> pas encore réussi	
	15:00	15:10	00:10	x			00:10	Design	TP 2	Réalisation de la map	x	Intérieur de la maison de gauche ajout de 2 tapis pour les 2 entrées, pour plus de réalisme.	
	15:10	16:10	01:00	x			01:00	Développement	TP 3	Codage des interactions JS	x	Finalisation des sorties des différentes maisons -> réussi	
	16:10	16:20	00:10	x			00:10	Développement	TP 3	Codage des interactions JS	x	Aggrandissement des zones de transition des portes à toute la largeur du chemin pour rendre le jeu plus agréable.	
	16:20	16:30	00:10	x			00:10	Développement	TP 3	Codage des interactions JS	x	Essai de passer le delay de milliseconde à une zone prédéfinie.	
	16:30	17:15	00:45	x			00:45	Développement	TP 3	Codage visuel du jeu	x	Amélioration du passage devant et derrière du personnages avec les étals du shop	
	17:45	18:00	00:15	x			00:15	Développement	TP 3	Codage visuel du jeu	x	Finalisation des personnages dans les étals	
	18:00	18:10	00:10	x			00:10	Développement	TP 3	Codage des interactions JS	x	Enlever les transitions des autres maps sur la map actuel pour une meilleure expérience de jeu.	
12 nov 25	18:10	18:20	00:10	x			00:10	Administration	TP 1	Planification	x	Mise à jour du journal de bord	
	13:20	16:00	02:40	x			02:40	Développement	TP 3	Codage visuel du jeu		Essai de changement de sprite pour la piscine.	
	16:00	16:30	00:30	x			00:30	Design	TP 2	Réalisation de la map	x	Retrait de la piscine sur la map.	
	16:30	17:20	00:50	x			00:50	Développement	TP 3	Codage visuel du jeu	x	Intégration du PNJ au bon endroit et avec un bon rectangle de collision dans la maison de gauche.	
13 nov 25	17:20	17:50	00:30	x			00:30	Développement	TP 3	Codage des interactions JS	x	Essai de différentes techniques pour commencer le dialogue avec le pnj.	
	17:50	18:00	00:10	x			00:10	Administration	TP 1	Planification		Mise à jour du journal de bord	
	09:05	09:50	00:45	x			00:45	Documentation	TP 4	Rédaction du rapport	x	Commencement de la rédaction de la partie code.	
19 nov 25	13:40	14:00	00:20	x			00:20	Design	TP 2	Réalisation de 2 PNJ	x	Recherches d'images libres de droit pour les 2 pnj	
	14:00	14:25	00:25	x			00:25	Développement	TP 3	Codage des interactions JS	x	Création de la 1ère interaction du PNJ1 disant "Bienvenue" quand on rentre dans une zone de 120px.	
	14:25	14:50	00:25	x			00:25	Développement	TP 3	Codage des interactions JS	x	Réification des zones de collisions au pixels près.	
	14:50	15:05	00:15	x			00:15	Développement	TP 3	Codage des interactions JS	x	Mise à jour des collisions de la maison de droite (table et chaises)	
	15:15	15:30	00:15	x			00:15	Développement	TP 3	Codage visuel du jeu	x	Ecran d'accueil intégré (aide de l'IA pour les particules)	
	15:30	15:40	00:10	x			00:10	Design	TP 2	Réalisation de la map	x	Corrections des derniers détails design de la map principale	
	15:55	16:50	00:55	x			00:55	Développement	TP 3	Codage des interactions JS	x	Interaction pnj1 sous forme de dialogue avec le personnage principal	
	16:50	17:00	00:10	x			00:10	Administration	TP 1	Planification		Mise à jour du journal de bord et dépose des fichiers sur le drive.	
26 nov 25	13:20	14:10	00:50	x			00:50	Développement	TP 3	Codage des interactions JS	x	Finalisation des dialogues entre le personnage et les PNJ	
	14:10	15:35	01:25	x			01:25	Développement	TP 3	Codage des interactions JS	x	Gestion de la récupération des objets (Pièces et pain)	
	15:35	16:00	00:25	x			00:25	Développement	TP 3	Codage des interactions JS	x	Récupération des objets	
	16:00	17:50	01:50	x			01:50	Développement	TP 3	Codage des interactions JS	x	Peaufinage des dialogues et interactions.	
	17:50	18:00	00:10	x			00:10	Administration	TP 1	Planification		Mise à jour du journal de bord et dépose des fichiers sur le drive.	
	03 déc 25	13:40	14:00	00:20	x			00:20	Développement	TP 3	Codage visuel du jeu	x	Réglage de la taille du container du jeu.
		14:00	14:30	00:30	x			00:30	Développement	TP 3	Codage des interactions JS	x	Amélioration de l'alternance des dialogues de la maison de gauche.
14:40		15:05	00:25	x			00:25	Développement	TP 3	Codage visuel du jeu	x	Garder la possibilité de récupérer le pain même si on quitte la maison.	
15:05		15:15	00:10	x			00:10	Développement	TP 3	Codage visuel du jeu	x	Sortie en haut et revenir en bas et vice versa, sur la map principale.	
15:15		15:30	00:15	x			00:15	Développement	TP 3	Codage des interactions JS	x	Dialogue possible que si le joueur est dans la zone de 120px autour du pnj.	
15:30		15:50	00:20	x			00:20	Développement	TP 3	Codage des interactions JS	x	Aggrandir les sorties et entrées des maison pour une meilleure expérience utilisateur.	
15:55		17:05	01:10	x			01:10	Développement	TP 3	Codage des interactions JS	x	Corrections des bugs des dialogues de la maison de gauche.	
18:25		19:00	00:35	x			00:35	Développement	TP 3	Codage des interactions JS	x	Debug des derniers bugs restants.	
19:55		21:20	01:25	x			01:25	Développement	TP 3	Codage des interactions JS	x	Avancement dans le nettoyage du code et division en section pour mieux s'y retrouver	
21:20		21:35	00:15	x			00:15	Administration	TP 1	Planification		Mise à jour du journal de bord et dépose des fichiers sur le drive.	
06 déc 25	10:55	11:10	00:15	x			00:15	Développement	TP 3	Codage visuel du jeu	x	Correction bug de dialogue.	
	11:10	11:15	00:05	x			00:05	Développement	TP 3	Codage visuel du jeu	x	Vérification du HTML avec W3C Validator	
	11:15	11:45	00:30	x			00:30	Développement	TP 3	Codage visuel du jeu	x	Corrections et commentaire du CSS et vérification W3C Validator.	
	11:50	12:00	00:10	x			00:10	Développement	TP 3	Codage visuel du jeu	x	Corrections de la transparence du personnage principale, sur illustrator	
	12:30	12:40	00:10	x			00:10	Développement	TP 3	Codage visuel du jeu	x	Ajout des commentaires en JS avec l'aide de l'IA pour mieux comprendre chaque lignes.	
07 déc 25	12:50	14:00	01:10	x			01:10	Développement	TP 3	Codage visuel du jeu	x	Suite de l'ajout des commentaires en JS avec l'aide de l'IA pour mieux comprendre chaque lignes.	
	17:30	19:30	02:00	x			02:00	Documentation	TP 4	Rédaction du rapport	x	Continuation de la documentation.	
	22:30	23:40	01:10	x			01:10	Documentation	TP 4	Rédaction du rapport	x	Amélioration de la partie Design.	
	11:05	11:25	00:20	x			00:20	Documentation	TP 4	Rédaction du rapport	x	Suite de l'amélioration de la partie Design	
	11:25	11:45	00:20	x			00:20	Documentation	TP 4	Rédaction du rapport	x	Modifié le design entier du rapport.	
	11:45	13:00	01:15	x			01:15	Documentation</					







## Cahier des charges

### Cahier des charges : 2526\_01\_Projet

Version : La version N°1 du 20.08.2025

<b>But du projet</b>	Le but de ce projet est de réaliser un jeu de style pokémon. Ou le personnage pourra se déplacer, interagir avec des pnj, entrer dans des maisons aller dans l'eau et récupérer des objets. Avec Javascript HTML et CSS.
<b>Objectifs supplémentaires</b>	<p><b>Spécifique</b> : Créer un mini-jeu interactif (type Pokémon simplifié ou jeu de grille) avec intégration des assets Illustrator et rapport marketing définissant les personas et public-cible principaux.</p> <p><b>Mesurable</b> : Le projet contiendra :</p> <ul style="list-style-type: none"><li>• Déplacements/interactions.</li><li>• Design du jeu.</li><li>• Étude marketing identifiant au moins deux personas et public-cibles.</li><li>•</li></ul> <p><b>Atteignable</b> : Le jeu sera réalisable en <b>64 heures</b>, avec des fonctionnalités supplémentaires (combats de créatures, inventaire) ajoutées seulement si le temps le permet.</p> <p><b>Réaliste</b> : Les compétences mobilisées sont cohérentes avec le projet : JavaScript, CSS, HTML pour le développement, Illustrator et Photoshop pour le design et des techniques marketing pour l'étude du public.</p> <p><b>Temporel</b> : Le projet sera finalisé pour le <b>17 décembre 2025</b>.</p>
<b>Livrables et délais</b>	<ul style="list-style-type: none"><li>• Toutes les unes à deux semaines, une mise au point sera faite avec le professeur en charge du projet.</li><li>• Chaque mercredis, 4 heures sont consacrées au développement de ce projet.</li><li>• Livrable 1 (12.09.2025) : Finalisation de la partie Design</li><li>• Livrable 2 (21.11.2025) : Finalisation de la partie développement</li><li>• Livrable 3 (17.12.2025) : Finalisation du projet</li></ul>
<b>Parties prenantes</b>	<ul style="list-style-type: none"><li>• <b>Coach</b> : Alain Jacobi <a href="mailto:alain.jacobi@eduvaud.ch">alain.jacobi@eduvaud.ch</a></li><li>• <b>Élèves</b> : Anaëlle Zahnd, SM-CA2a, <a href="mailto:anaelle.zahnd@eduvaud.ch">anaelle.zahnd@eduvaud.ch</a></li><li>• <b>Client</b> : Alain Jacobi <a href="mailto:alain.jacobi@eduvaud.ch">alain.jacobi@eduvaud.ch</a></li></ul>
<b>Plan de communication</b>	Tous les mercredis de 13h20 à 16h40, le coach Alain Jacobi supervisera le travail fourni. C'est à lui que sera transmis le projet final pour le livrable final le 17 décembre 2025.
<b>Contraintes</b>	<ul style="list-style-type: none"><li>• Obligation d'utiliser 2 branches en plus de l'informatiques faisant partie de la formation de médiamatique. Dans ce cas le design et le marketing.</li></ul>

<b>Risques</b>	<ul style="list-style-type: none"><li>• Manque de ressources sur JavaScript.</li><li>• Risques de ne pas réussir à réaliser tous les objectifs prévus.</li><li>• Hébergement du site/jeu.</li></ul>
<b>(Hors) périmètre</b>	
<b>Suppositions</b>	

**Date et lieu :**

**03 septembre 2025 à Ste-Croix**

**Signature du directeur projet (prof) :**

*Jacobi*

**Signature du client :**

### PV séance du 03 septembre 2025

Le professeur suggère de développer le projet en trois versions afin de préciser, dans le rapport, quelles sont ces différentes étapes et ainsi mieux définir les objectifs.

Il recommande également qu'à la fin de chaque version, un court rapport intermédiaire soit rédigé, afin d'éviter que le projet ne paraisse trop monotone.

Pour le déplacement du personnage, il serait préférable d'utiliser un système de collisions circulaires, plus simple et efficace à mettre en place.

Enfin, grâce à Kontra.js, il est possible de trouver des ressources et de l'aide pour coder un mini-jeu, ainsi que pour gérer l'insertion des sprites.

### Séance du 05 novembre 2025

Je présente mon code à monsieur Jacobi, il est content de l'avancement.

Cependant il relève plusieurs points à améliorer.

Il souhaite que :

- J'améliore les bugs avant la fin de la V2 prévue pour le 2 novembre donc à faire au plus vite avant de passer à la V3.
- Peut-être mettre 2 tapis dans la maison de gauche vu qu'il y a 2 entrées différentes, afin que ça soit plus compréhensible.
- Il me suggère un but qui est le suivant : Dans la maison de droite le personnage principal interagira avec un pnj (sa maman) qui lui demandera d'aller chercher du pain au shop. Au shop on lui répondra qu'elle n'a pas assez d'argent. Elle retournera vers sa maman qui lui donnera plus de pièces elle pourra ensuite aller acheter le pain et le redonner à sa maman = jeu fini!
- Et pour finir essayer de mettre un temps après être entré ou sortie dans une maison à la place d'un temps en milliseconde.



# Sources

## Persona 1 et persona 2

canva.com

[https://www.canva.com/design/DAG7fmcD-mA/oXDqWegCIVdO\\_HU7zOxpsQ/edit?referrer=logo-landing-page](https://www.canva.com/design/DAG7fmcD-mA/oXDqWegCIVdO_HU7zOxpsQ/edit?referrer=logo-landing-page)

Consulté le 08.10.2025

## Analyse SWOT modèle

canva.com

<https://www.canva.com/design/DAG1Nf5njc0/sS8iX0pxJUM8VcKiuoHBNA/edit>

## Logo Pokémon Go

wikipedia.org

[https://fr.wikipedia.org/wiki/Fichier:Pok%C3%A9mon\\_Go\\_Logo.png](https://fr.wikipedia.org/wiki/Fichier:Pok%C3%A9mon_Go_Logo.png)

Consulté le 08.10.2025

## Logo Pomémon Showdown

pokemonshowdown.com

<https://pokemonshowdown.com/images/>

Consulté le 08.10.2025

## Logo Animal Crossing

wikipedia.org

[https://fr.wikipedia.org/wiki/Fichier:Animal\\_Crossing\\_New\\_Leaf\\_Logo.jpg](https://fr.wikipedia.org/wiki/Fichier:Animal_Crossing_New_Leaf_Logo.jpg)

Consulté le 08.10.2025

## Loo Zelda

wikipedia.org

[https://fr.wikipedia.org/wiki/Fichier:The\\_Legend\\_of\\_Zelda\\_A\\_Link\\_to\\_the\\_Past\\_Logo.png](https://fr.wikipedia.org/wiki/Fichier:The_Legend_of_Zelda_A_Link_to_the_Past_Logo.png)

Consulté le 08.10.2025

## Map

gamedevmarket.net

<https://www.gamedevmarket.net/asset/cute-top-down-tileset>

Consulté le 27.08.2025

## Spritesheet

forums.rpgmakerweb.com

<https://forums.rpgmakerweb.com/index.php?threads/metsss-mv-materials.130785/>

Crédits graphiques : <https://metsss0221.tumblr.com/>

Consulté le 20.08.2025

## PNJ maman

hiclipart.com

<https://www.hiclipart.com/free-transparent-background-png-clipart-ootia>

Consulté le 01.12.2025

## **PNJ Vendeuse**

chatgpt.com

« Crée moi un personnage du meme style mais qu'on distingue facilement avec d'autres couleurs. Mais en version sorcière » à ChatGPT

Crée le 01.12.2025

## **Pain**

shutterstock.com

<https://www.shutterstock.com/image-vector/bread-pixel-art-vector-illustration-on-2506602621?trackingId=9a3b401d-de9f-4910-99b3-3baa3509ca>

Consulté le 26.11.2025

## **Coins**

fr.freepik.com

[https://fr.freepik.com/vecteurs-libre/monde-du-jeu-arcade-scene-pixel\\_4815132.htm#fromView=search&page=1&position=40&uuid=9e754728-b3b2-42ea-91f9-1ea7790e4b9c&query=pixel+token](https://fr.freepik.com/vecteurs-libre/monde-du-jeu-arcade-scene-pixel_4815132.htm#fromView=search&page=1&position=40&uuid=9e754728-b3b2-42ea-91f9-1ea7790e4b9c&query=pixel+token)

Consulté le 26.11.2025

## **Sablier**

fontawesome.com

<https://fontawesome.com/icons/hourglass?f=classic&s=regular>

Consulté le 14.12.2025

## **Logo Phaser**

pngwing.com

<https://www.pngwing.com/en/free-png-xrjfx>

Consulté le 13.12.2025

## **Flèches clavier**

dreamstime.com

<https://www.dreamstime.com/pixel-art-keyboard-arrow-keys-play-vector-icon-bit-game-dark-background-illustration-image390668949>

Consulté le 13.12.2025

## **W3C Validator**

validator.w3.org

[https://validator.w3.org/#validate\\_by\\_input](https://validator.w3.org/#validate_by_input)

Consulté le 06.12.2025

## **Validator JS**

esprima.org

<https://esprima.org/demo/validate.html>

Consulté le 06.12.2025

## **Correction des textes**

Textes corrigés par ChatGPT et Deepseek



# Colophon

## Projet TPI Sakima

### Classe

SM-Ca2a

### Graphisme & Maquettes

Zahnd Anaëlle

### Typographie

Mundial Thin 13pt

Courier Regular 13pt

### Format

210 mm x 297 mm

### Nombre de page

92 Pages

### Tous droits réservés.

CPNV Décembre 2025.



# Projet TPI



Anaëlle Zahnd | SM-CA 2a