

*Phaser 3*

**etpa**

*Lardon's Adventure*

École du  
Jeu Vidéo

*Game Design Document*



# *Sommaire*

Sommaire	2
Vue d'ensemble du jeu	4
High concept	4
Compétences développées	4
Synopsis	5
Objectif de jeu	5
Règles du jeu	6
Structure du jeu	6
Gameplay	7
Contrôle (PC)	8
Contrôle (Mobile)	8
Caméra du jeu	8
HUD	8
Tétrade élémentaire	9
Technologie	10
Mon jeu	10
SMB	10
Mécanisme	10
Mon jeu	10
SMB	10
Histoire	10
Mon jeu	10
SMB	10
Esthétique	11
Mon jeu	11
SMB	11
Joueur	12
Situation et histoire	12
Etats	13
Capacité	13
Animation	14
Ennemis et obstacles	15
2 monstres	15
Etats	15
Obstacles	16
Bugosaur	17

Art & graphisme	18
Level Design	18
Audio	18
Wishlist	18
Analyse du jeu Super Mario Bros	19
Apprentissage	19
Confirmation	20
Ajout des différents éléments	20
Finalité	20
Liens	21
Remerciements	21

# *Vue d'ensemble du jeu*

Titre : Lardon's Adventure - Les aventures de Lardon

Plateformes : PC (Web) + Mobile (version dégradé)

Genre : Plateforme conceptuelle.

Cible : Les joueurs en quête d'une expérience inédite..

Date de sortie du jeu : 07 Juin 2021

## *High concept*

Le joueur a la possibilité de débloquer ses propres compétences à l'aide d'un menu de programmation. Pour accéder à ce menu, il peut alors mettre en pause le jeu et accéder à sa matrice.

## *Compétences développées*

Quelques notions très basiques de programmation



## *Synopsis*

Lardon, un jeune programmeur va essayer de contrer un ancien programmeur qui a décidé d'introduire des bugs dans tous les jeux de la planète. Pour ce faire, notre héros devra débugger les jeux et affronter des bugs majeurs dans ceci pour enfin rencontrer son opposant : Bugosaur. Lardon s'immisce alors dans son propre jeu pour combattre son ennemi !



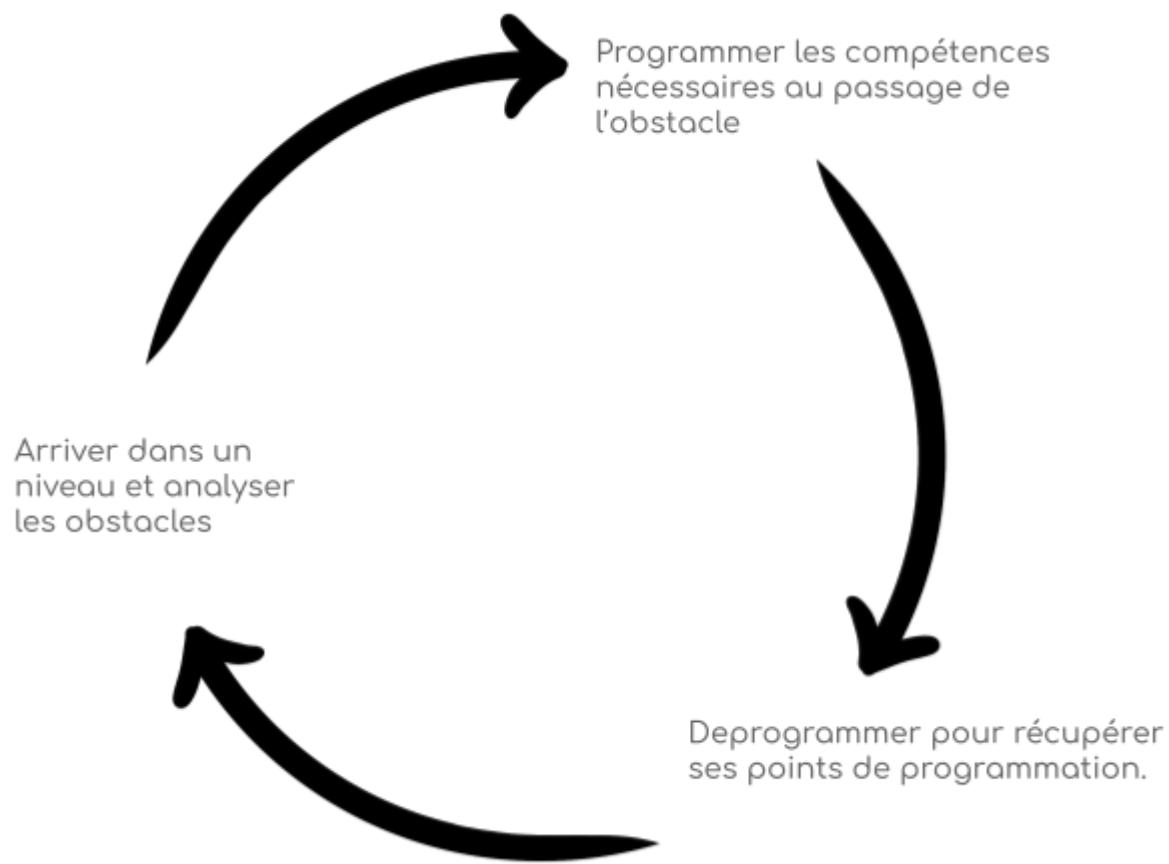
## *Objectif de jeu*

L'objectif est de parcourir le jeu en utilisant ses points de programmation pour débloquer certaines capacités. Le joueur a la possibilité de déprogrammer ses compétences pour regagner des points de programmation.

## Règles du jeu

Le joueur peut se déplacer de gauche à droite dans le niveau. Tout au long de sa progression, il sera stoppé par des monstres incorporés dans le jeu par Bugosaur et devra traverser des précipices. Il peut à tout moment passer en mode programmeur et s'octroyer les capacités nécessaires pour franchir les différents obstacles. Il a le choix entre un saut, un dash ou une attaque de mêlée. Actuellement seul 1 point de programmation est disponible dans le niveau.

## Structure du jeu



# Gameplay

Le héros est déplacé par le joueur à travers le niveau de la même manière que Mario. Si Lardon ne possède pas les compétences nécessaires pour le franchissement de l'obstacle, le joueur peut mettre en suspens le niveau et programmer ses propres compétences : création d'un dash, d'une attaque au corps à corps. Attention, le joueur ne pourra pas programmer toutes les compétences quand il le souhaite. Pour ce faire, il devra récupérer des points de programmation au cours du franchissement des divers niveaux. Il pourra également utiliser ses compétences pour améliorer les compétences déjà obtenues (dash + loin/+ rapide, armes lançant plus loin). Le joueur pourra choisir l'ordre dans lequel il décide d'acquérir ses compétences.

## POUR LA CRÉATION DES COMPÉTENCES :

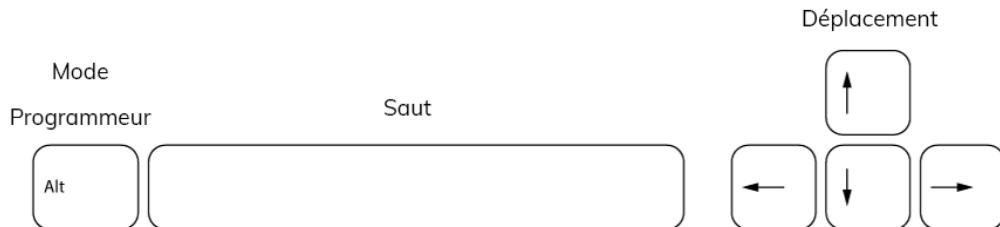
Le joueur aura la possibilité de mettre en pause sur le niveau : apparaîtront alors diverses lignes de codes. Le joueur devra faire glisser différents blocs de codes pour essayer de coder, sans erreur, une compétence.

- Arrivée dans un niveau
- Essai de traverser le niveau
- Réussite -> niveau suivant
- Echec -> Programmation d'une compétence avec la possibilité de se tromper
- Nouvel essai avec la compétence acquise



## *Contrôle (PC)*

Les flèches GAUCHE et DROITE pour se déplacer de gauche à droite.  
La touche ALT pour passer en mode programmation.  
La touche Espace pour utiliser la compétence programmée.  
La souris dans le mode programmation pour déplacer les différents blocs de programmation.



## *Contrôle (Mobile)*

Les boutons de la partie de droite de l'écran permettent de déplacer le joueur.  
Le bouton vert permet de passer en mode programmation.  
Le bouton rouge pour utiliser la compétence programmée.  
Les doigts pour déplacer les blocs de programmation.



## *Caméra du jeu*

La caméra est posée sur la tranche du niveau de jeu. Le jeu est en 2D.

## *HUD*

Le HUD est composé du nom du niveau en haut à droite, du compteur de points de programmation en haut à gauche. Dans le mode programmeur : le HUD se transforme en faisant apparaître les différents blocs possibles de programmation ainsi que les deux conteneurs pouvant accueillir les blocs.

PTS DE PROGRAMMATION: 1

NIVEAU  
NIVEAU

DASH

JUMP

MELEE

TRUE

FALSE



# Tétrade élémentaire

## Technologie

### Mon jeu

Lordon's Adventure a été créé en JavaScript à l'aide de la librairie Phaser 3 qui est un framework de jeu 2D

La librairie a été créée par Proton Storm disponible ici : [github.com/photonstorm/phaser](https://github.com/photonstorm/phaser).

La librairie gère les images, les spritesheets, la physique de base du jeu (gravité, collision) et les inputs du joueur.



Number	Register name
0	%eax
1	%ecx
2	%edx
3	%ebx
4	%esp
5	%ebp
6	%esi
7	%edi
F	No register

### SMB

Super Mario Bros a été possible grâce au système Hardware poussé de la Famicom. Développé en C et en assembleur avec une équipe de 7 programmeurs. La force du jeu provient de l'utilisation optimale de l'espace mémoire de la cartouche rendu possible grâce au choix de langage.

## Mécanisme

### Mon jeu

Lordon's Adventure tire son originalité et son gameplay de la phase de programmation propre aux différents obstacles et ennemis.

### SMB

Les quelques mécaniques poussées proviennent du changement de joueur à l'aide de la fleur ou de l'étoile qui peuvent être apparenté à une cascade de booléen/de l'héritage. L'ajout de niveaux sous-marins permettent une diversité de Gameplay.

## Histoire

### Mon jeu

Lordon et Bugosaur sont deux protagonistes qui partagent une passion commune : le code et la création de jeu. L'histoire est donc une histoire de vengeance et une quête de liberté d'expression.

## SMB

L'histoire est liée entre Mario, Bowser et Peach, une histoire chevaleresque d'un héros qui sauvera sa princesse des griffes d'un méchant cracheur de feu dans les divers donjons.

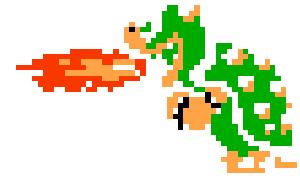
## *Esthétique*

### Mon jeu

La technologie ayant évolué le jeu affiche notamment des personnages ayant une résolution allant jusqu'à 772\*2665.

### SMB

Le jeu affiche très peu de pixels car la quantité de pixel disponible dans la cartouche ainsi que la vitesse de lecture mémoire n'était pas la même qu'aujourd'hui. Bowser étant le plus gros personnage et il comptaibilise un total de 32\*32 pixels.



# Joueur

## *Situation et histoire*

Lardon est un jeune homme qui travaille dans une société de création de jeux vidéo en tant que développeur. Tous les jours, il prend le métro pour aller au travail et c'est là que commence sa journée. Il maîtrise la programmation mais il est surtout expert en déboggage. Tout au long de ses journées, il ne pense qu'aux lignes de codes et à la manière dont il pourrait réparer les jeux. Il programme également durant son temps libre chez lui, pour ses multiples amis qui le sollicitent quand leurs jeux sont buggés.

Il rêve de créer son propre studio mais manque de compétence d'un point de vue graphisme et d'un point de vie gestion d'entreprise pour tenter cette aventure en solo. Son travail est compliqué car il vient tout juste de sortir de ses études et ne maîtrise pas encore les affres rencontrées en entreprise.

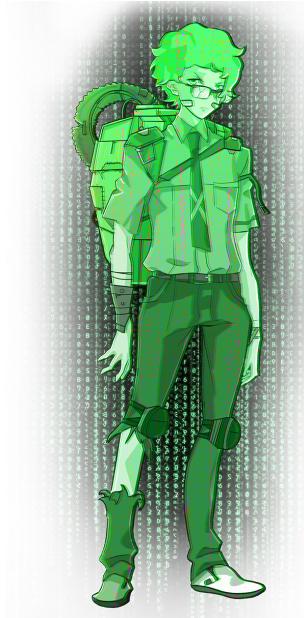
C'est un jeune homme qui aime la musique notamment lorsqu'il travaille. Ses langages préférés sont le C# et l'OCaml. A contrario, il déteste l'HTML et le PHP. Il s'habille tous les jours en costume sauf lorsqu'il debug son jeu dans ce cas il enfile sa tenue de chasse aux bugs.

Son objectif est de créer ses propres jeux. Pour cela, il doit progresser dans son métier car il n'a pas encore les savoirs et connaissances pour être indépendant.



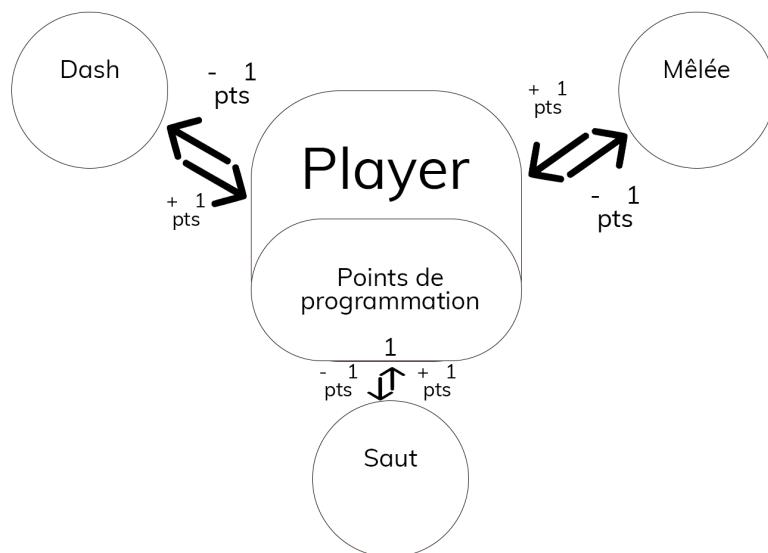
## *Etats*

Il possède deux états primordiaux : la phase active et la phase programmeur. Au début du jeu, il porte sa tenue de bureau traditionnelle tandis que dans le jeu il est en tenue de chasse aux bugs.



## *Capacité*

Il peut arrêter le temps pour se donner des compétences spéciales : le saut, le dash ainsi que l'attaque de mêlée pour détruire les monstre.



## *Animation*

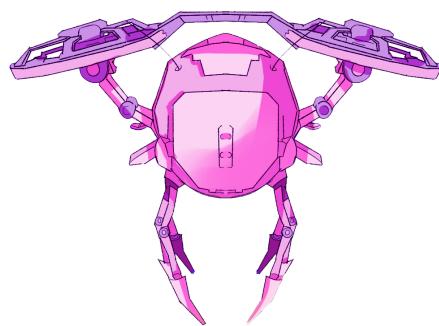
L'animation du personnage a été faite à partir de 2 keyframes dessinées par le graphiste en charge ses visuels.



# *Ennemis et obstacles*

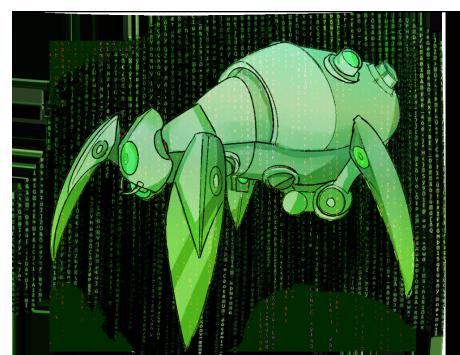
## **2 monstres**

L'Hélico-byte et le Web-Crawler sont deux virus créés par Bugosaur. Ils possèdent tous les deux des patterns de déplacement différents. L'un se déplace au sol de gauche à droite tandis que le second vole dans les airs en se déplaçant.



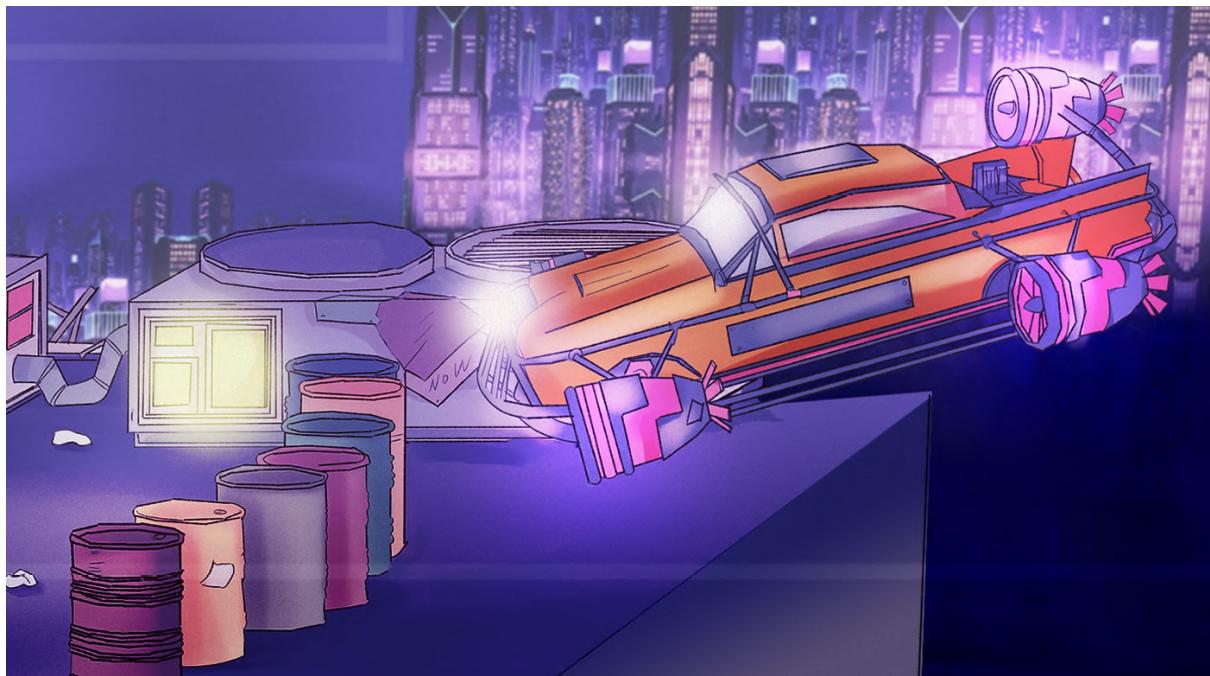
## **Etats**

Les monstres possèdent 3 états distincts : en vie et en mouvement, en vie mais inerte du fait d'être bloqué par le mode programmation et mort.



## Obstacles

Le joueur peut être bloqué par certains éléments du décor, les espacements entre les bâtiments ainsi que les différents objets éparpillés à travers le niveau : barils, voitures, échafaudages.





## Bugosaur

Pionnier dans la création de jeux, c'était un grand nom dans ce milieu jusqu'au jour où il eu l'idée folle de créer un jeu où le joueur devra déplacer un personnage pour esquiver des obstacles et tuer des monstres, avec la possibilité de coder ses powers up. La presse n'avait pas du tout compris le message caché derrière le jeu et les articles incriminants Bugosaur ne cessèrent d'apparaître.

Depuis toujours, Bugosaur créait des jeux profonds avec des gameplays innovants. Il avait tout appris de lui même et c'était un très grand développeur. Un véritable artiste lorsqu'il était devant son clavier et qu'il écrivait ses lignes de codes. Au fur et à mesure des années qui suivent sa chute, il développe une certaine haine envers l'industrie et surtout la presse dotée d'un pouvoir dont elle n'est pas digne. Il décide de nuire à l'industrie pour enfin pouvoir sortir son jeu et enfin être récompensé de la manière dont il estimer l'être à l'époque.

Pour ce faire, il crée un virus, qu'il répand à travers tous les jeux de la planète, ayant pour effet de faire bugger les jeux. Il trouve le moyen de contaminer les anciens jeux ainsi que ceux en cours de création, quelque soit l'ampleur de chaque projet : petits projets entre amis ou sortie nationale.

Bugosaur n'est pas affrontable dans ce niveau mais il apparaît pour menacer Lardon.

Son objectif en tant que renégat est de totalement anéantir l'économie du jeux vidéo en détruisant tous les jeux de la planète et ainsi se venger du traitement qu'on lui a infligé.

## *Art & graphisme*

Tout l'aspect graphique est un travail en collaboration avec Ahmad MAKKÉ, élève en ICA2. Son côté graphique, proche des codes de l'animation japonaise, ainsi que sa créativité sont parfaitement en accord avec les idées de bases de la direction artistique du jeu. Les couleurs principales du jeu sont le violet, le rose et le bleu. Les différentes textures de glitch ainsi que les lignes de code du film MATRIX sont là pour donner du volume et appuyer le côté programmeur.

## *Level Design*

Level design sommaire pour montrer les différentes capacités ainsi que leur utilité. Le jeu se passe sur des toits dans un univers rétro-futuriste. L'utilisation de toiture permet de jouer avec les vides ainsi que divers décors tel que des panneaux publicitaires. Le niveau se termine sur une confrontation avec Bugosaur.

Le niveau à été crée en deux exemplaires, une version normale et une version programmeur.

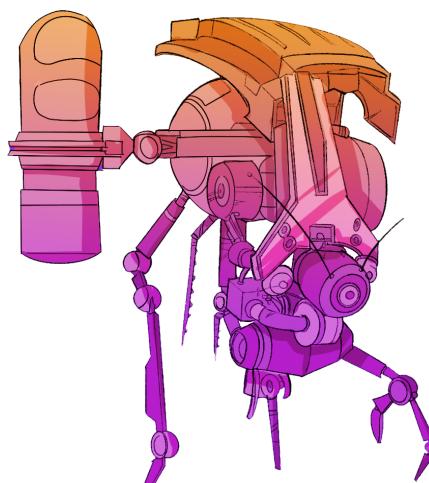


## *Audio*

Les trois pistes audios du jeu sont inspirées des années 80, un peu électro. La première musique est la musique dans les bureaux de Lardon. Suivie ensuite de deux musiques quasi similaires, l'une pour le mode normal du jeu tandis que la seconde ne s'active que lorsque le joueur passe en mode programmation, il s'agit ici de la même musique que la précédente mais enrichie de fausses notes pour marquer le côté buggé du jeu et joué sur des instruments différents.

## *Wishlist*

A terme, le jeu pourra être agrémenté de nouveaux niveaux ainsi que plusieurs autres monstres. De plus, le jeu pourra être ponctué de phases de boss qui demanderont au joueur plus de difficulté. J'aimerais aussi que la phase de programmation soit plus poussée en proposant aux joueurs de combiner plusieurs éléments (mélée + dash) permettant aux joueurs de tuer les monstres dans lesquels il dash. J'aurais aussi aimé pouvoir créer des assets spécifiques aux mobiles, pouvant s'afficher sans perte de qualité.



# Analyse du jeu Super Mario Bros

Le titre du niveau de 1 de SMB est : World 1-1. Cela signifie que le jeu se divise en plusieurs mondes ayant chacun une dynamique et un thème particulier. Ses différents mondes sont eux-même divisés en plusieurs niveaux de quelques minutes maximum, permettant au joueur des pauses et permettant de donner une dynamique particulière au jeu lorsque les joueurs peuvent choisir le niveau exact sur lequel ils veulent jouer à condition d'avoir réussi les précédents. De plus, la difficulté est croissante tout au long des niveaux pour terminer sur un niveau accueillant une phase de boss.



## Apprentissage

La première partie du niveau 1 est une zone d'apprentissage, en effet le joueur va être confronté à des éléments importants qu'il rencontrera tout au long du jeu. Il apprendra d'abord à se déplacer le long de l'écran. Puis il fera la connaissance des premiers ennemis très simples. En fonction de ses connaissances, il pourra essayer plusieurs actions : sauter par-dessus le monstre ou bien lui sauter dessus pour tenter de le tuer. Il va aussi faire la découverte de blocs placés dans le niveau et pouvant être détruits en sautant par en dessous. S'il a eu la curiosité de tous les détruire il en aura détruit un qui permet de faire apparaître un champignon qui le fera grandir.



## *Confirmation*

La seconde partie est une partie de confirmation. Le joueur aura affaire aux monstres et il devra gérer ses sauts pour parcourir le niveau. Il y aura à certains moment des blocs destructibles qui lui permettront de récupérer des points ou des champignons.



## *Ajout des différents éléments*

Le joueur ayant compris qu'il faut casser les blocs qu'il rencontre, tombera sur deux types de bonus différents qu'il ne connaît auparavant : les fleurs de feu ainsi que l'étoile. En fonction de ces bonus il pourra tenter différentes actions pour comprendre leur mécanisme.



## *Finalité*

La fin du niveau se termine sur un escalier qui monte quasi tout en haut de l'écran et l'apparition d'un drapeau sur lequel le joueur peut essayer de sauter, la présence de l'escalier laisse penser qu'il faut sauter le plus haut et le plus loin possible. Le joueur aura donc appris quasi toutes les mécaniques les plus importantes du jeu pour la suite.

## *Lien avec mon jeu*

Le déplacement de gauche à droite ainsi que la facilité de Gameplay après apprentissage lie mon jeu à SMB. La différence réside dans le fait de devoir programmer ses compétences actives. ATTENTION : on ne peut pas sauter sur la tête des ennemis.

# Liens

[Jeu GitHub](#)

Jeu Itch.io

[Repo](#)

[Moodboard](#)

# Remerciements

Je tiens tout d'abord à remercier Chloé : ma copine pour son soutien durant les phases de production ainsi que pendant toute l'année scolaire.

Je remercie aussi ma famille pour m'avoir motivé et soutenu tout au long de l'année.

Je remercie aussi le corps enseignant ainsi que l'équipe pédagogique pour m'avoir guidé ainsi que conseillé.

Merci à Ahmad pour sa collaboration, sa créativité et son talent. Il a su retranscrire parfaitement le mood visuel du jeu que j'imaginais.

Merci à mes camarades de classe pour cette année ainsi que pour les différents conseils et idées.

