

GAME DESIGN A BACHELOR 1

Juin 2020

SIMOES LUNA



Projet de production Bachelor 1 Game Design, ETPA
Rennes 2019-2020

Sommaire

MICRO-JEU 1	2
MICRO-JEU 2	7
MICRO-JEU 3	12
MICRO-JEU 4	17
MICRO-JEU 5	22
MICRO-JEU 6	26
MICRO-JEU 7	30
MICRO-JEU 8	35
MICRO-JEU 9	40
MICRO-JEU 10	44

Micro-Jeu 1

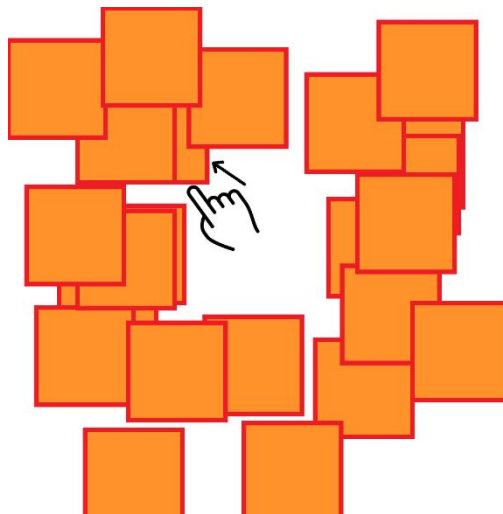
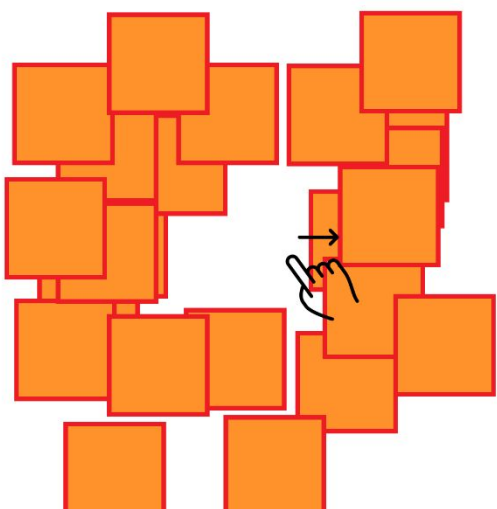
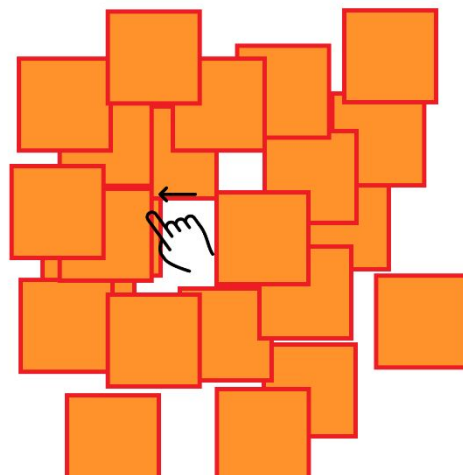
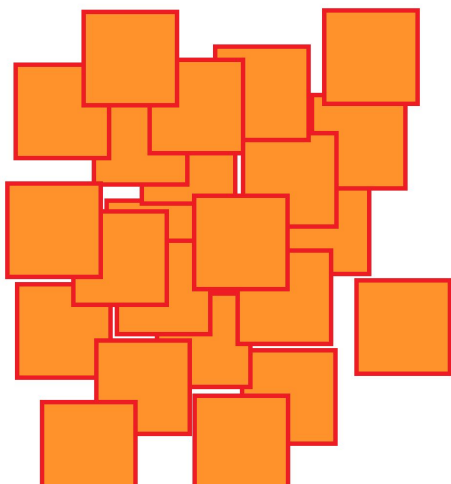
Vue d'ensemble

« Du fil à retordre »

Ce micro-jeu nécessite de faire glisser son curseur sur l'écran afin de pousser les éléments qui obstruent la voie. Le challenge est de libérer la voie dans le laps de temps imparti, en faisant attention à ne pas faire rebondir les obstacles qui peuvent revenir se placer devant nous.

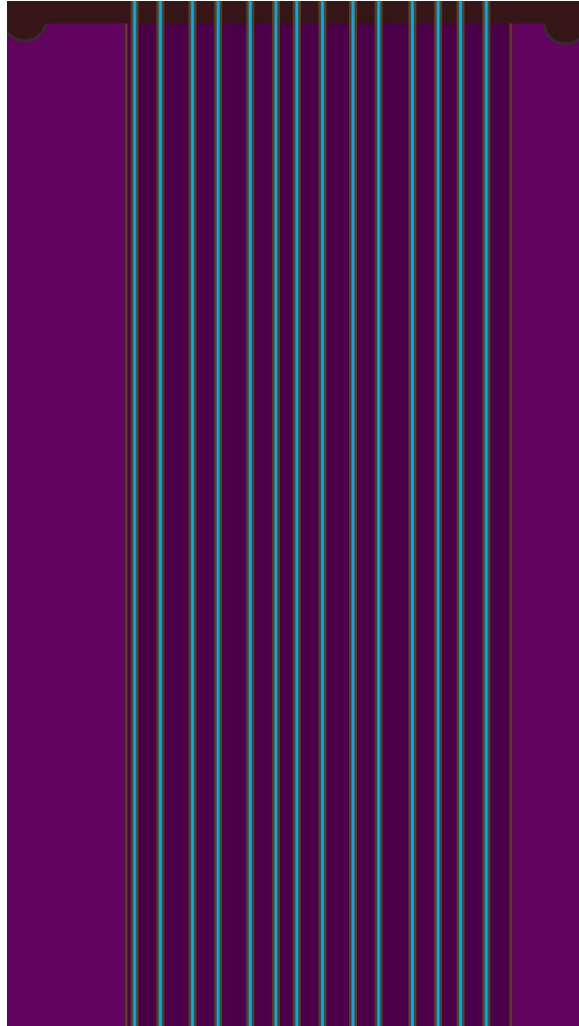
Gameplay

Ce micro-jeu joue sur le moteur physique de Phaser 3. Le joueur doit balayer l'écran afin d'écarter les obstacles qui se trouvent devant lui. S'il y va trop fort, les obstacles ont possibilité de rebondir et revenir obstruer la voie.



Look&Feel

Le micro-jeu se déroule dans une boîte de nuit. On y retrouve les couleurs mauve et bleu de la retrowave, notamment avec le rideau qui se compose de fils rappelant les néons. Le jeu se déroule en 1280 x 720, ce qui incite le joueur à tourner son écran.



Le micro-jeu se déroule à la première personne pour apporter un sentiment d'oppression suite au champ de vision réduit. Ce sentiment a été précédemment instauré par la la phase narrative.

Données

Nous déclarons une variable “Obstacle” qui nous permettra plus tard de faire paraître les fils qui sont censés obstruer le chemin.

Nous aurons deux assets, un pour le fond appelé “background” et un autre pour les fils.

Lesdits fils seront des éléments physiques, donc apparaîtront tels que “this.physics.add.image”, contrairement au fond qui ne sera pas interactive et apparaîtra en “this.add.image(x,y,'fond');”

Il y aura nécessité d’ajouter un input pour cliquer sur l’écran afin de déplacer les obstacles.

```
this.input.setDraggable(viseur.setInteractive());  
  
viseur.on('drag', function (pointer, dragX, dragY) {  
    this.x = dragX;  
    this.y = dragY;  
});
```

Afin d’assurer les collisions, nous ajouterons “.setCollideWorldBounds(true);”

Nous déclarons une autre variable “chrono” qui sera nécessaire pour faire apparaître du texte. Ce texte permettra de faire paraître le compte à rebours du micro-jeu via la fonction “this.time.addEvent”. Par le biais du “delay” nous pouvons choisir à quel moment apparaît le texte.

```
// chronomètre

this.time.addEvent({
  delay: 1000,
  callback: ()=>{
    chrono = this.add.text(50, 50, '5', {fontSize: '100px', fill:'#FFF'});
  },
  loop: false
});
```

Micro-Jeu 2

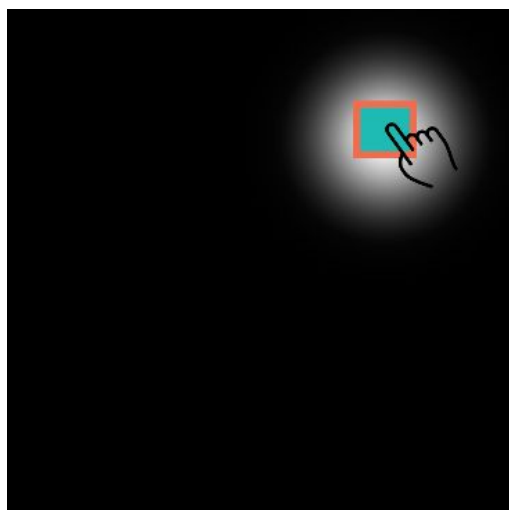
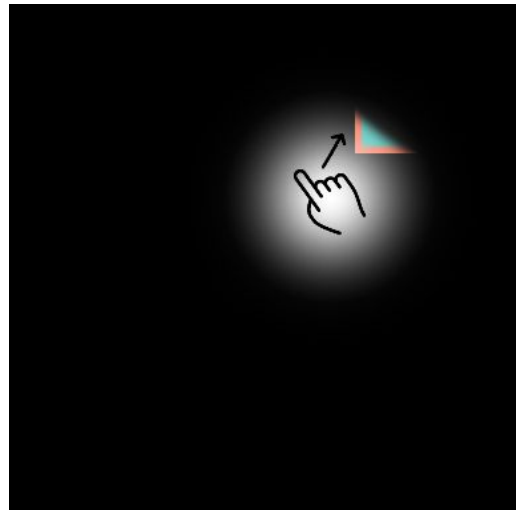
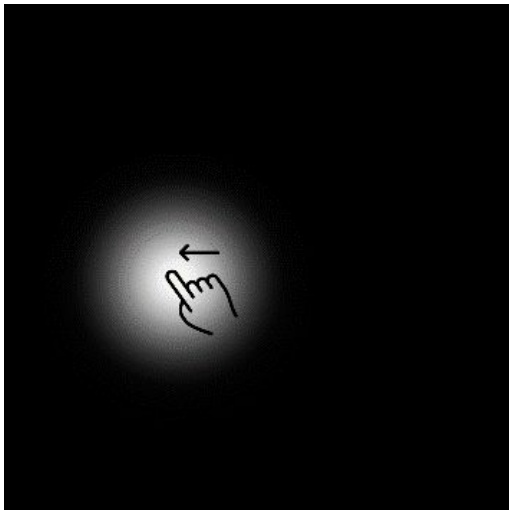
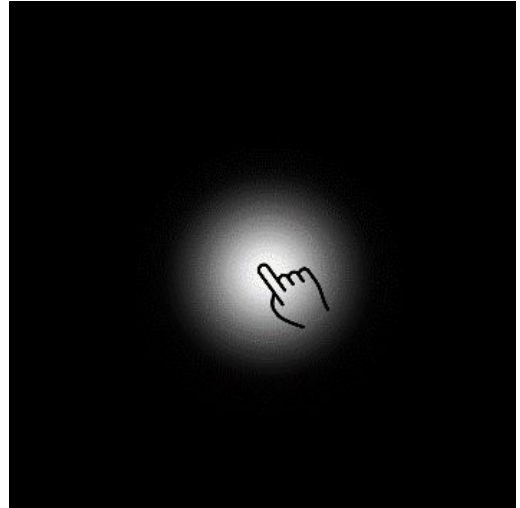
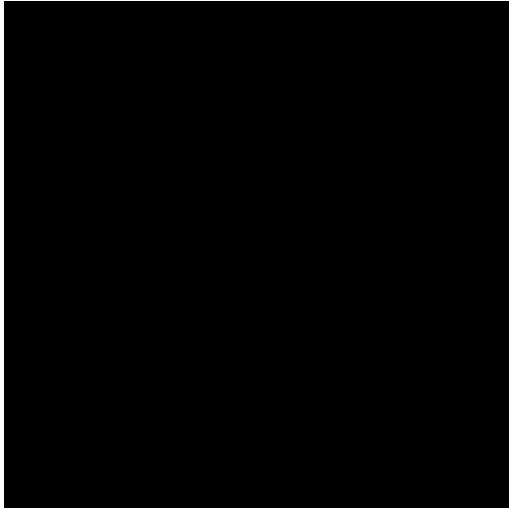
Vue d'ensemble

« Dans l'Obscurité »

Ce micro-jeu nécessite de faire glisser son curseur sur l'écran afin de révéler une petite partie de l'environnement. Le challenge est de retrouver un élément dans l'obscurité dans le laps de temps imparti.

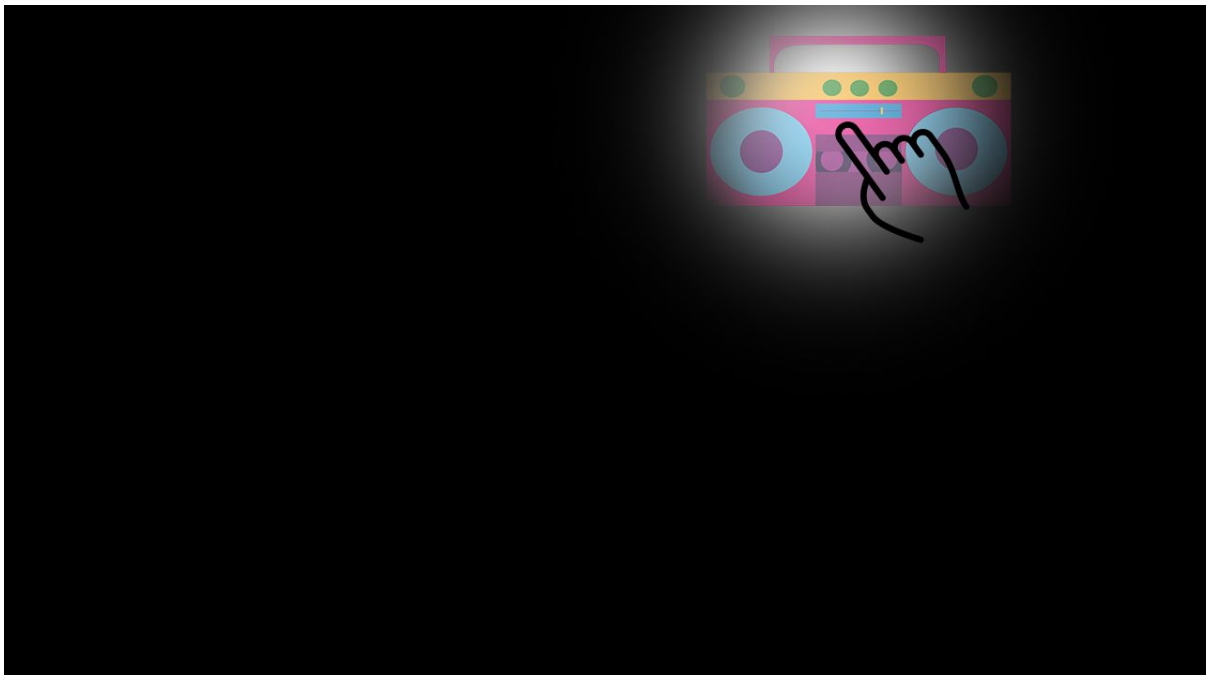
Gameplay

Le jeu use du “Spotlight” de Phaser 3, soit la capacité à donner l’illusion d’un faisceau lumineux. Le joueur glisse le curseur sur l’écran pour générer de la lumière et rechercher un élément qui se dissimule dans le noir. Il faudra cliquer sur l’élément une fois qu’il sera trouvé.



Look&Feel

Le micro-jeu se déroule dans le noir pour apporter un sentiment d'oppression et d'inquiétude, sentiment mis en place auparavant par la phase narrative. La première personne sert à nous mettre à la place du personnage et nous faire sentir vulnérable. Le champ de vision se trouve davantage réduit qu'à la troisième personne.



Données

Nous aurons une variable “radiocassette” pour l’élément sur lequel nous devons cliquer, mais également une variable pour le background ainsi qu’une “spotlight” pour l’effet de lumière. De plus nous aurons une variable “score” qui permettra la condition de victoire, ainsi qu’une variable “Text” pour afficher la consigne ainsi que les mots “Gagner” ou “Perdu”.

Nous mettons en place deux inputs. Le premier permet de glisser le curseur sur l’écran afin de révéler une partie de l’image.

```
this.input.on('pointermove', function (pointer) {  
    spotlight.x = pointer.x;  
    spotlight.y = pointer.y;  
});
```

Le second permet de cliquer sur l’objet qu’on recherche afin d’enclencher la condition de victoire.

```
var radiocassette = this.add.image(200,200,'radiocassette').setInteractive()  
radiocassette.on('pointerdown', function (pointer){  
    this.destroy();  
    score += 1;  
}); //on clique sur la radio pour valider
```

Nous déclarons une autre variable “chrono” qui sera nécessaire pour faire apparaître du texte. Ce texte permettra de faire paraître le compte à rebours du micro-jeu via la fonction “this.time.addEvent”. Par le biais du “delay” nous pouvons choisir à quel moment apparaît le texte. De plus ce chronomètre permet de mettre en place la condition de défaite.

```
// chronomètre

this.time.addEvent({
  delay: 1000,
  callback: ()=>{
    chrono = this.add.text(50, 50, '5', {fontSize: '100px', fill:'#FFF'});
  },
  loop: false
});
```

Micro-Jeu 3

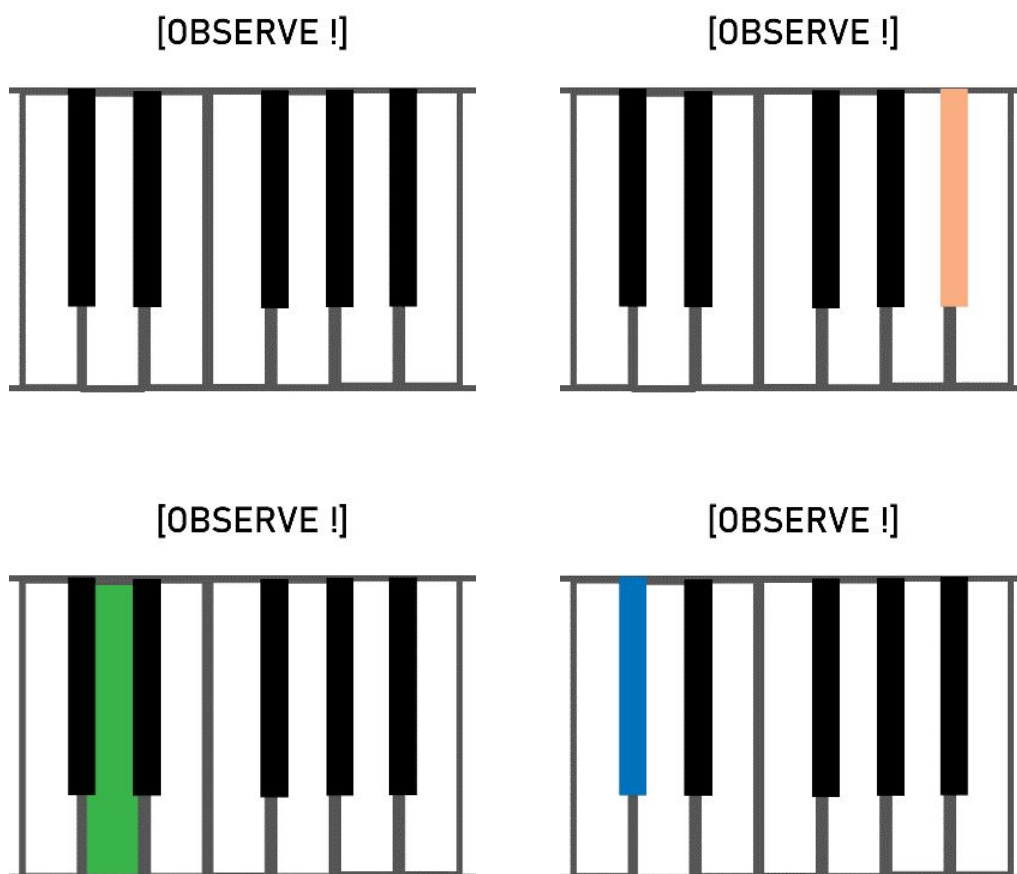
Vue d'ensemble

« Mémoire visuelle »

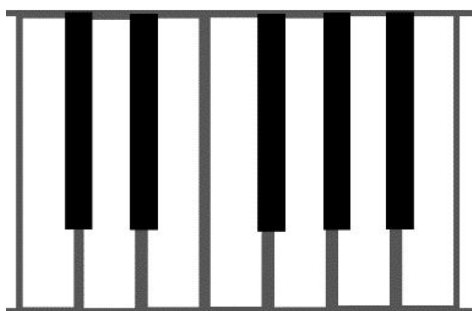
Ce micro-jeu fait appel à notre observation et notre mémoire. Il faut regarder ce que fait le jeu et pouvoir le reproduire ensuite. Nous utiliserons notre curseur pour cliquer sur les éléments nécessaires.

Gameplay

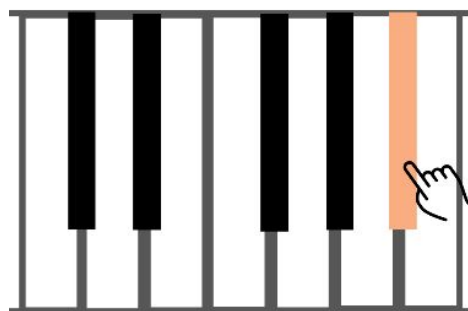
Ce micro-jeu se déroule principalement à l'aide du timer. Dans un premier temps, le joueur doit observer dans quel ordre les éléments s'allument. Il doit ensuite reproduire ce qu'il a vu.



[REPRODUIS !]



[REPRODUIS !]



[REPRODUIS !]

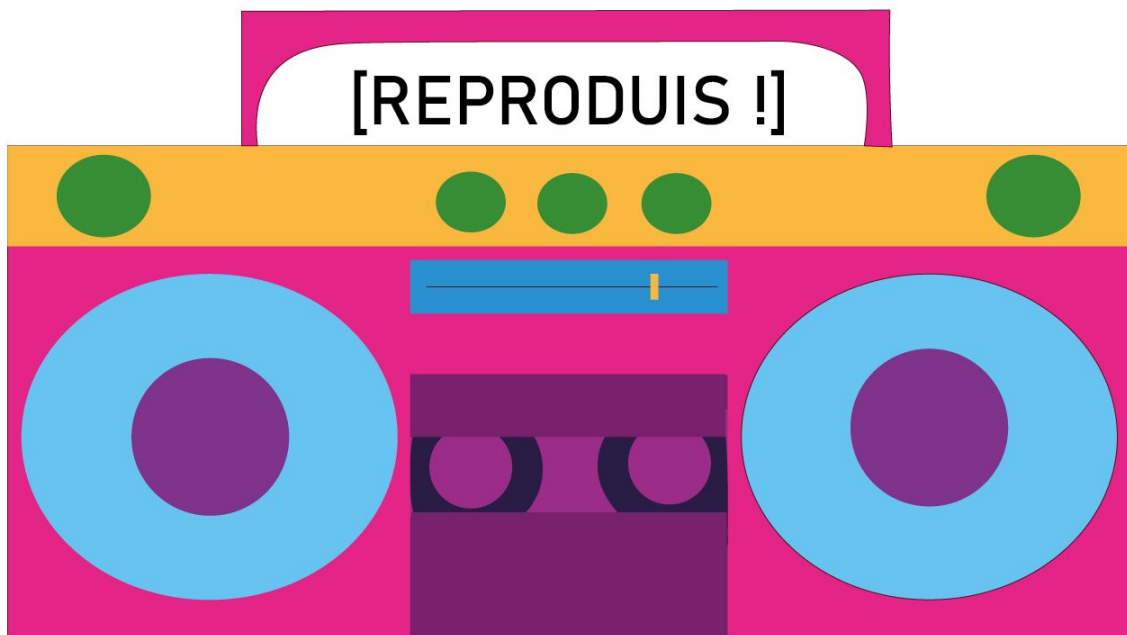


[REPRODUIS !]



Look&Feel

Pour se rapprocher davantage de l'univers Retrowave, le micro-jeu ne se déroula pas sur un piano mais une radiocassette.



Données

Nous déclarons une variable pour le background, constitué de la radiocassette, ainsi qu’une variable pour chaque bouton sur lesquels nous pouvons cliquer, soit six au total en comptant celui pour régler le volume.

Le jeu nécessite 4 sprites différents puisque certains boutons possèdent le même.

Le jeu nécessite un input permettant de cliquer.

Nous déclarons une autre variable “chrono” qui sera nécessaire pour faire apparaître du texte. Ce texte permettra de faire paraître le compte à rebours via la fonction “this.time.addEvent”. Par le biais du “delay” nous pouvons choisir à quel moment apparaît le texte.

```
// chronomètre  
  
this.time.addEvent({  
  delay: 1000,  
  callback: ()=>{  
    chrono = this.add.text(50, 50, '5', {fontSize: '100px', fill:'#FFF'});  
  },  
  loop: false  
});
```

“chrono” est important car il permet d’indiquer au joueur à quel moment le jeu presse certains boutons. Lorsque le joueur devra reproduire, il devra se baser sur le ‘chrono’ pour savoir quand il faut appuyer sur telle touche.

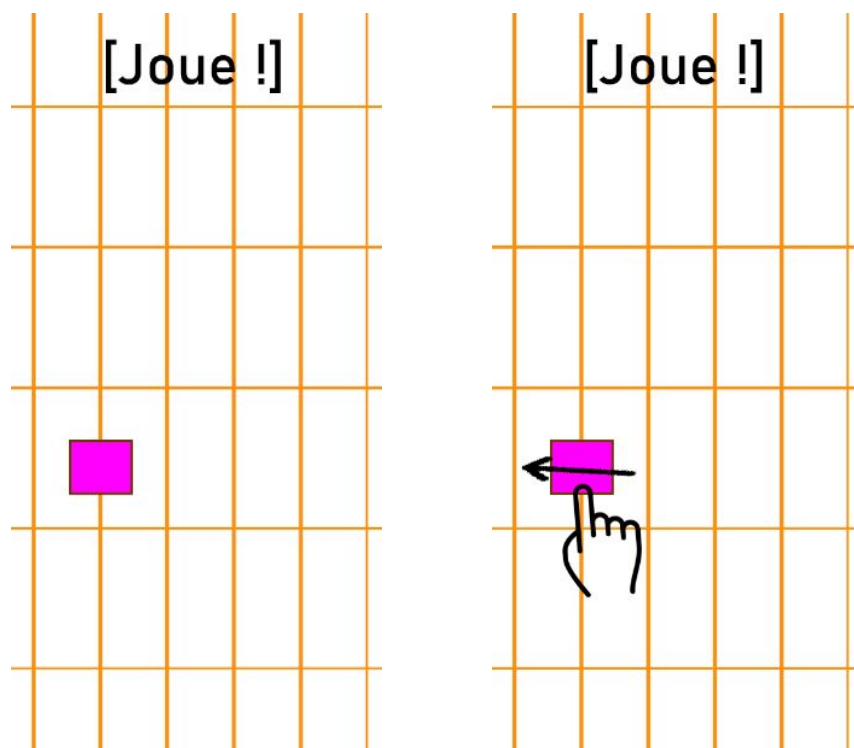
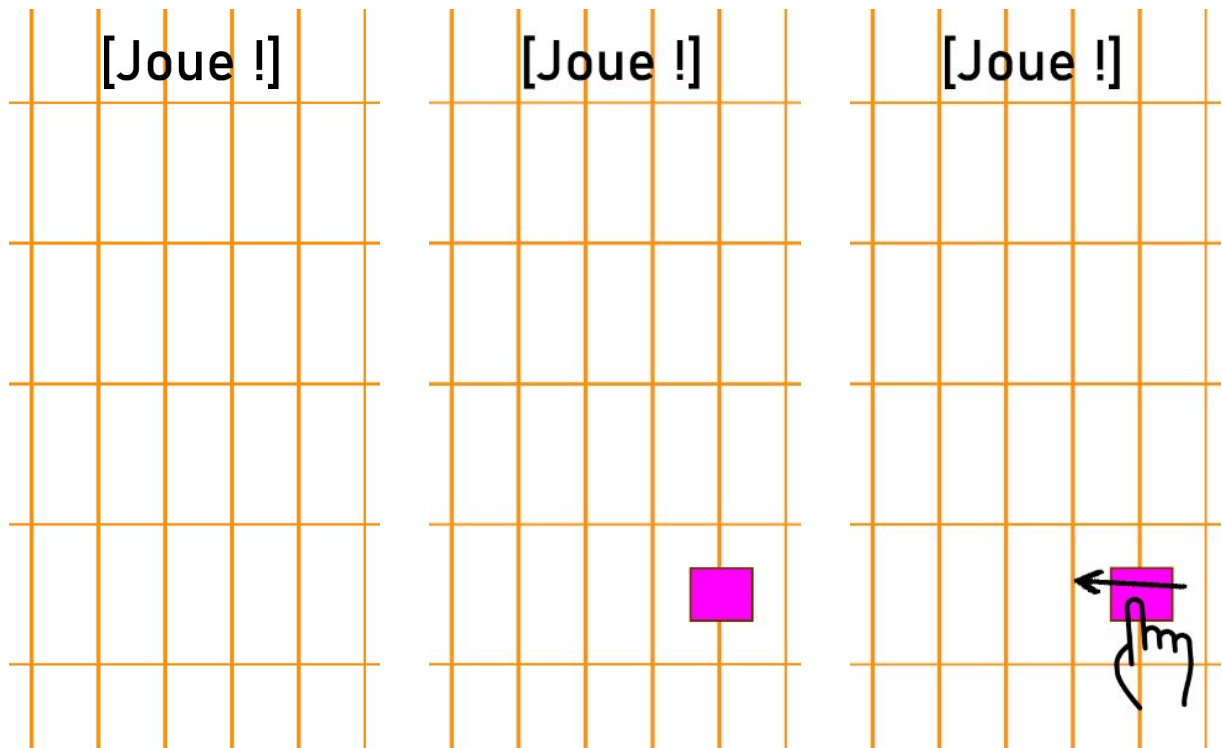
Micro-Jeu 4

Vue d'ensemble

« Solo de guitare »

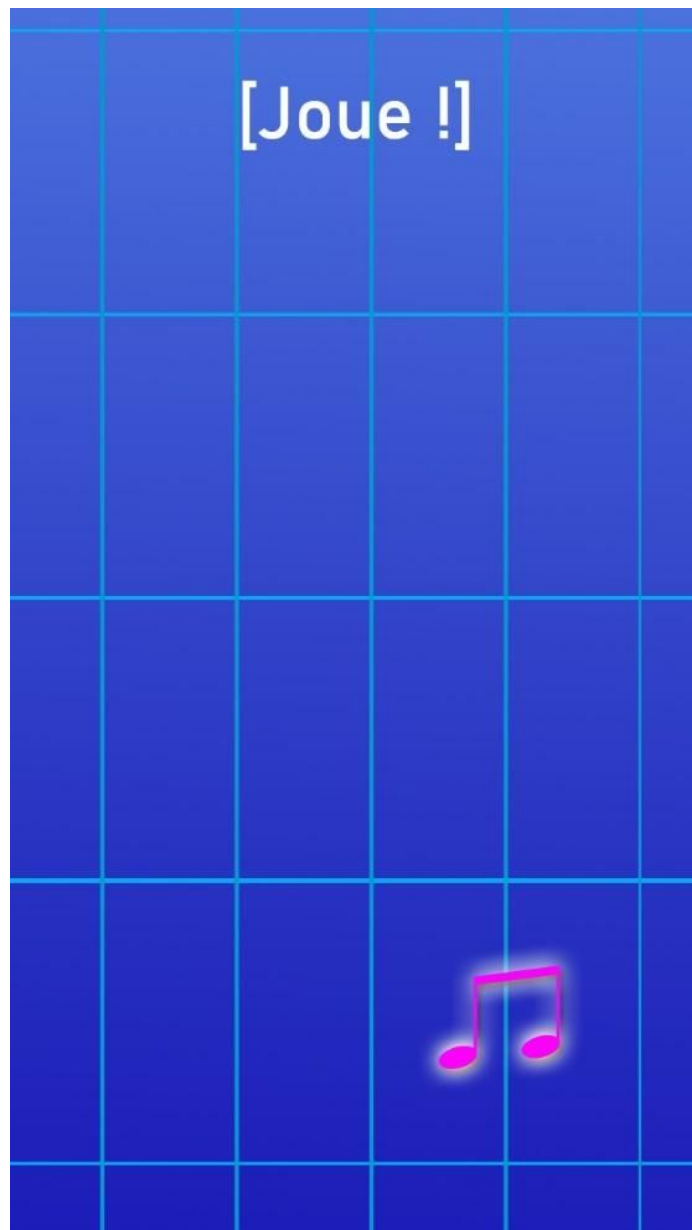
Ce micro-jeu nécessite de l'attention et du réflexe. Des notes de musique apparaissent chacune leur tour au fur et à mesure que le temps avance et il faut cliquer dessus avant qu'elles ne disparaissent, autrement le micro-jeu est perdu.

Gameplay



Look&Feel

Le micro-jeu représente une guitare. On retrouve le bleu et le mauve du retrowave, respectivement pour le fond de la guitare et pour le sprite des notes de musiques. Les cordes de la guitare sont plus claires et rappellent l'effet néon de la retrowave. Le jeu se déroule en 1280 x 720, ce qui incite le joueur à tourner son écran.



Données

Nous déclarons des variables comme indiqués ci-dessous :

```
var Note1;  
var Note2;  
var Note3;  
  
var chrono;  
var pointer;  
  
var score = 0;  
var gameText;  
var winText;  
  
var consigneText;
```

Le jeu ne nécessite que deux assets. L'un pour le background et l'autre concerne le sprite pour les notes de musiques sur lesquelles nous devons cliquer.

```
function preload(){  
    this.load.image('fond','assets/guitare.png');  
    this.load.image('note','assets/note.png');  
}
```

Les notes sont au début cachées aux yeux du joueurs, elles se révéleront au fur et à mesure que le temps s'écoule par le biais de la fonction "addEvent".

```
//notes

Note1 = this.add.image(180,300,'note').setInteractive().setVisible(false);
Note2 = this.add.image(600,640,'note').setInteractive().setVisible(false);
Note3 = this.add.image(200,1100,'note').setInteractive().setVisible(false);

this.time.addEvent({
  delay: 2000,
  callback: ()=>{
    chrono.destroy();
    chrono = this.add.text(50, 50, '4', {fontSize: '100px', fill:'#FFF'});

    Note1.setVisible(true); //La note apparaît
    Note1.on('pointerdown', function (pointer) { //on clique dessus pour la faire disparaître

    this.destroy();
    score += 1;
  })
})
```

Lorsque nous cliquons sur une note, nous obtenons “+ 1” à la variable “Score”. Ce score permet d’obtenir les conditions de victoire et de défaite. Effectivement, si “Score == 3” à la fin du timer, alors le joueur gagne. Si le score est inférieure à 3 alors le joueur perd.

La variable “chrono” sera nécessaire pour faire apparaître du texte. Ce texte permettra de faire paraître le compte à rebours du micro-jeu via la fonction “this.time.addEvent”. Par le biais du “delay” nous pouvons choisir à quel moment apparaît le texte.

```
// chronomètre

this.time.addEvent({
  delay: 1000,
  callback: ()=>{
    chrono = this.add.text(50, 50, '5', {fontSize: '100px', fill:'#FFF'});
  },
  loop: false
});
```

Micro-Jeu 5

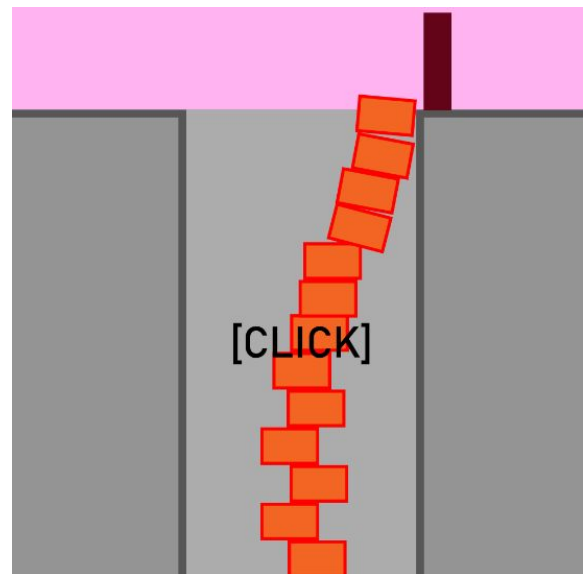
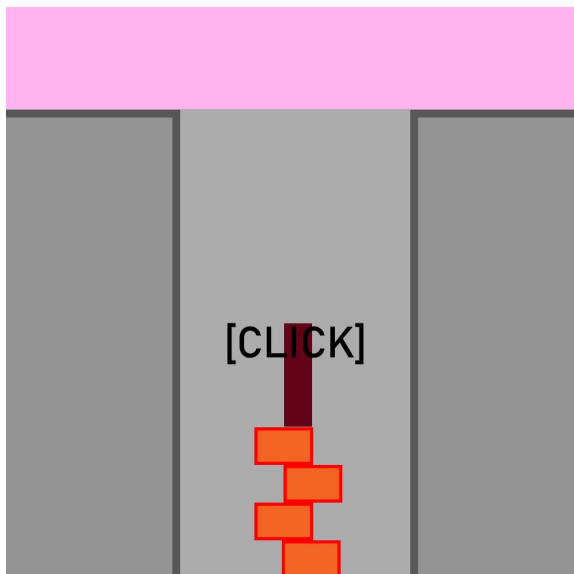
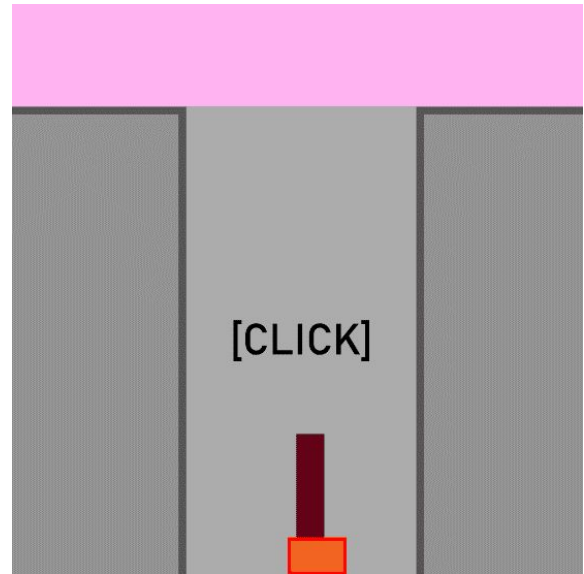
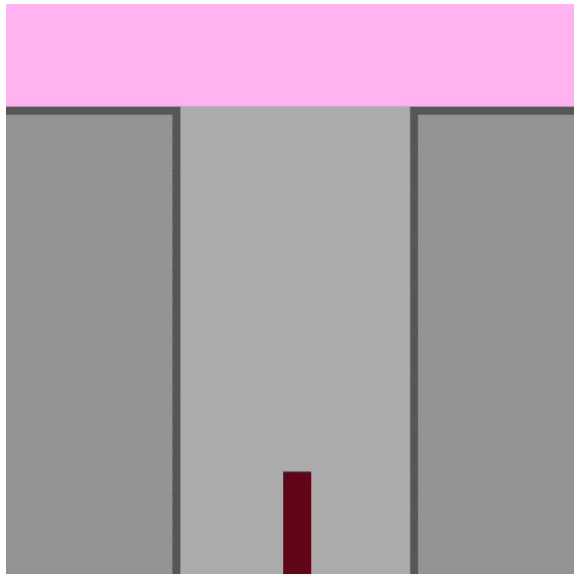
Vue d'ensemble

« S'élever plus haut »

C'est un micro-jeu de type 'clicker', dans lequel il faut faire apparaître un certain nombre d'objets dans un temps imparti en cliquant le plus vite possible sur l'écran. Le challenge est d'être plus rapide que le compte à rebours. Le sens du toucher est mis à contribution, il faut savoir rester concentrer pour ne pas perdre de temps.

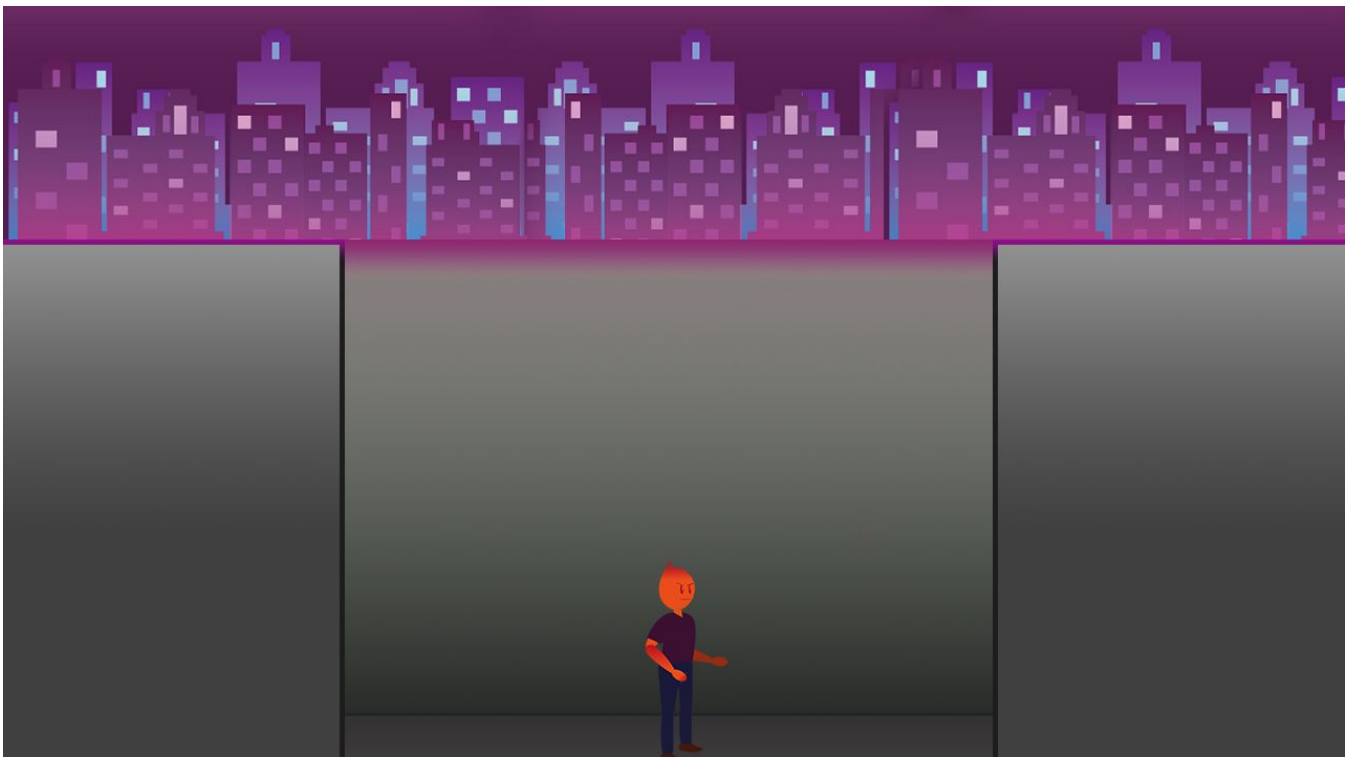
Gameplay

Ce micro-jeu joue sur le moteur physique. Le joueur doit appuyer pour générer des objets qui le porteront vers le haut. Il faut cliquer le plus rapidement possible afin de générer un nombre suffisant d'objets avant que le temps ne soit écoulé.



Look&Feel

Le champ de vision est plus vaste, on voit autour de nous, on voit également la ville en arrière fond. Cependant, impossible de réellement en profiter car notre temps est limité. Nous avons un contraste entre la ville en arrière plan, colorée et éclairée au néon, tandis que notre personnage au premier plan est coincé dans une impasse dans des nuances de gris. L'idée est de donner la volonté au joueur de "s'échapper", aider le protagoniste à s'élever pour sortir de cette impasse.



Données

Ce micro-jeu nécessite une variable “player” ainsi qu’une variable “objet”.

Nous aurons besoin d’un sprite pour les objets ainsi que de sprites pour l’animation du personnage.

“Objet” est un élément physique, ce qui permettra “d’empiler” les objets au fil du jeu. “player” est également un élément physique et pourra ainsi se tenir au dessus des “objets”.

Nous aurons besoin d’un input qui nous permettra de cliquer afin de faire apparaître les objets.

Nous déclarons une autre variable “chrono” qui sera nécessaire pour faire apparaître du texte. Ce texte permettra de faire paraître le compte à rebours du micro-jeu via la fonction “this.time.addEvent”. Par le biais du “delay” nous pouvons choisir à quel moment apparaît le texte.

```
// chronomètre

this.time.addEvent({
  delay: 1000,
  callback: ()=>{
    chrono = this.add.text(50, 50, '5', {fontSize: '100px', fill:'#FFF'});
  },
  loop: false
});
```

Micro-Jeu 6

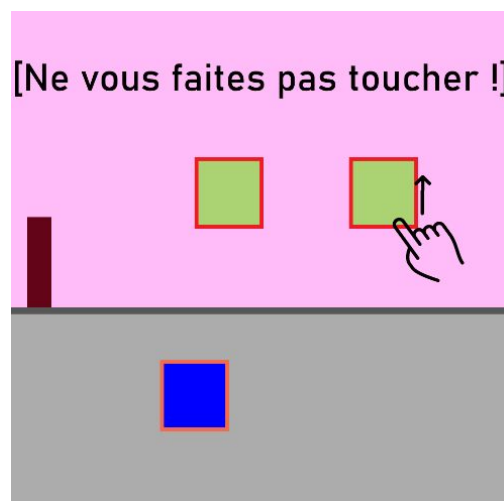
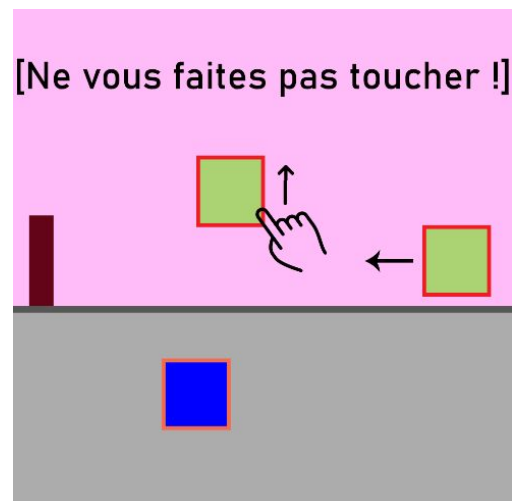
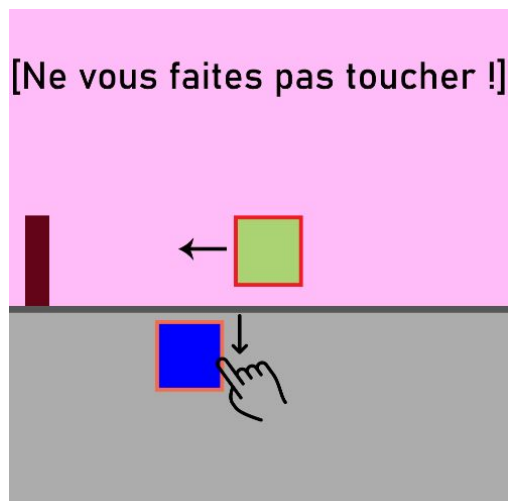
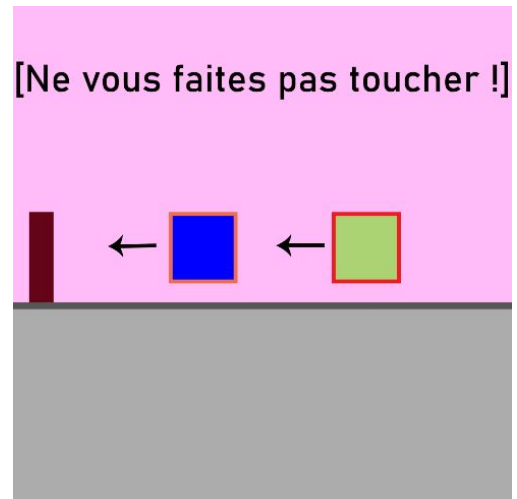
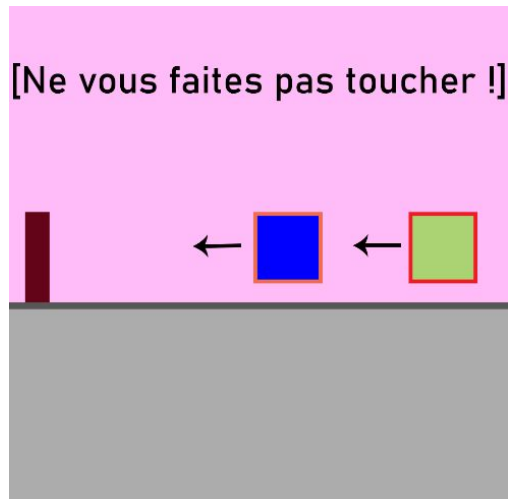
Vue d'ensemble

« Esquive »

Ce micro-jeu nécessite de l'observation et de la vitesse. Il faut repérer les obstacles arriver, voir de quels couleurs ils sont et les glisser vers le haut ou le bas en fonction de leur couleur, avant qu'ils ne touchent notre personnage.

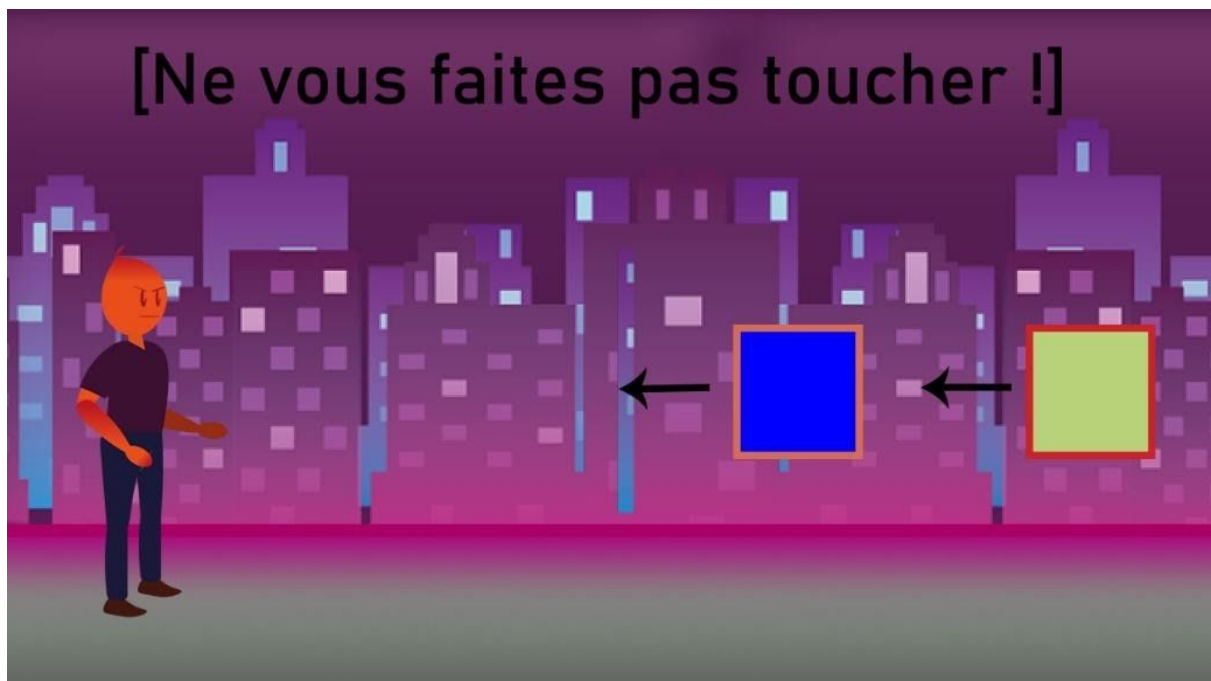
Gameplay

Notre personnage se trouve à gauche de l'écran et ne bouge pas. Des éléments arrivent par la droite pour le toucher. Le joueur doit empêcher le personnage de se faire toucher en venant faire glisser les obstacles en haut ou en bas de l'écran, en fonction de la couleur de l'obstacles.



Look&Feel

Ce micro-jeu se déroule à l'extérieur, de nuit. On perçoit en arrière fond la ville illuminée au néon, toujours dans les teints mauves et bleus. Notre personnage se démarque au premier plan par sa couleur orangée, tout comme les obstacles qui sont de deux couleurs distinctes de façon à percevoir lesquels peuvent être glissés vers le haut, et lesquels peuvent glisser vers le bas.



Données

Ce micro-jeu nécessite une variable “player” ainsi qu’une variable “obstacle1” et une variable “obstacle2”.

La variable “player” représente notre joueur et sera configuré pour rester sur la gauche. Ce sera un élément physique de façon à pouvoir effectuer une collision avec “obstacle1” et “obstacle2”.

“obstacle1” et “obstacle2” sont également des éléments physiques qui, contrairement à “player”, peuvent se mouvoir de la droite vers la gauche.

Nous aurons besoin d’un sprite pour “obstacle1” et “obstacle2” ainsi qu’un sprite pour “player”.

Nous mettons en place un input qui nous permettra de cliquer sur “obstacle1” et “obstacle2” pour les faire glisser vers le bas (pour “obstacle 1”) et vers le haut (“pour obstacle2”).

Nous déclarons une autre variable “chrono” qui sera nécessaire pour faire apparaître du texte. Ce texte permettra de faire paraître le compte à rebours du micro-jeu via la fonction “this.time.addEvent”. Par le biais du “delay” nous pouvons choisir à quel moment apparaît le texte.

```
// chronomètre
this.time.addEvent({
  delay: 1000,
  callback: ()=>{
    chrono = this.add.text(50, 50, '5', {fontSize: '100px', fill:'#FFF'});
  },
  loop: false
});
```

La fonction “addEvent” permettra également de faire apparaître les obstacles à des temps précis.

Micro-Jeu 7

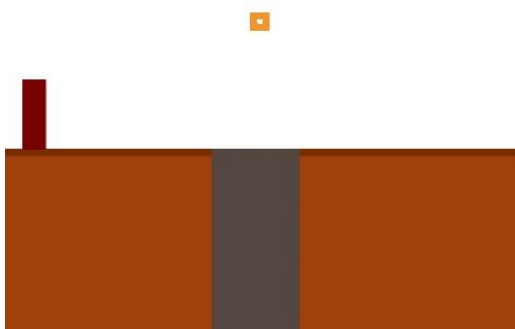
Vue d'ensemble

« Reculer pour mieux sauter »

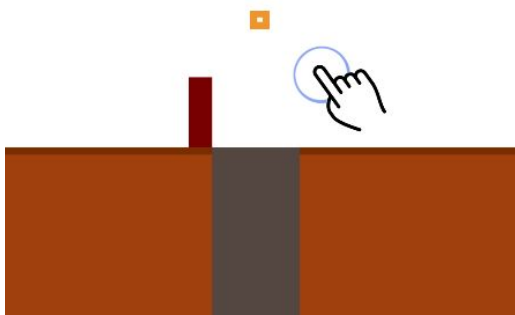
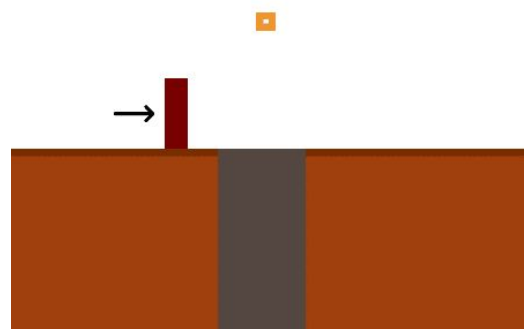
Ce micro-jeu nécessite de l'attention. La consigne n'apparaît qu'à un bref moment et le joueur doit avoir le bon timing. Si le joueur n'appuie pas durant l'apparition de la consigne, le micro-jeu est perdu.

Gameplay

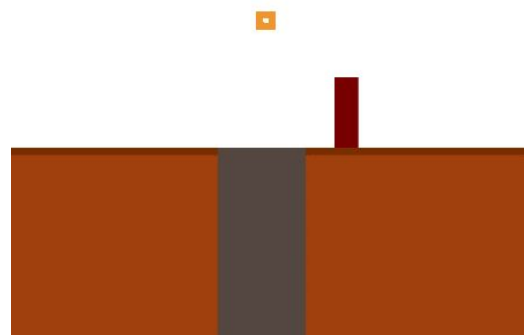
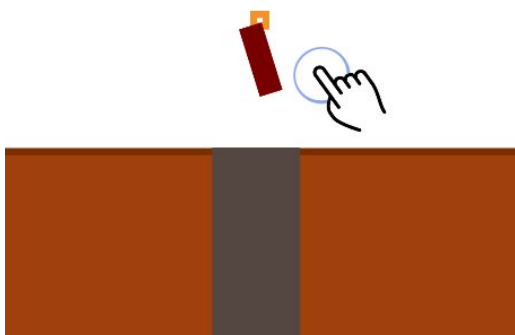
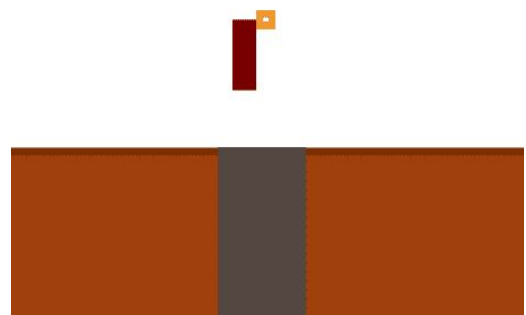
Le personnage débute de la gauche de l'écran et se dirige automatiquement vers la droite. Le joueur doit appuyer lorsqu'on lui indique afin de faire sauter le protagoniste pour lui permettre d'esquiver le trou qui se situe devant lui.



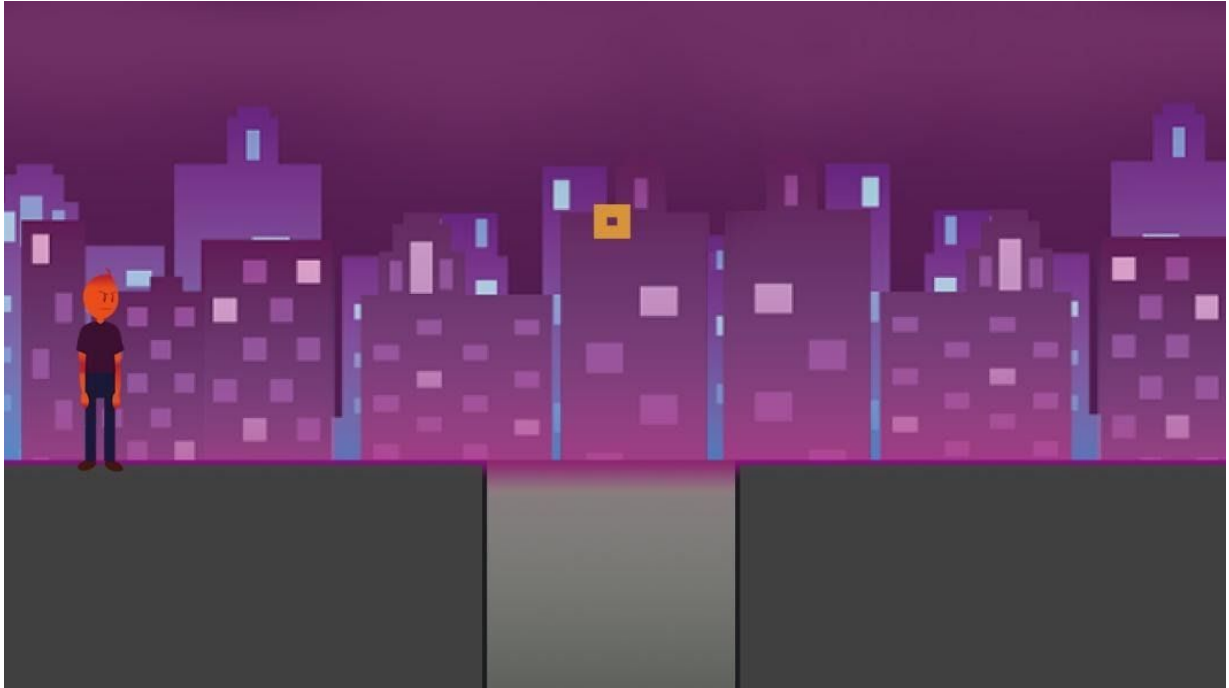
[SAUTE !]



[SAUTE !]



Look&Feel



Ce micro-jeu se déroule à l'extérieur, de nuit. On perçoit en arrière fond la ville illuminée au néon, toujours dans les teints mauves et bleus. Notre personnage se démarque au premier plan par sa couleur orangée.





Données

Nous déclarons une variable “Player” ainsi qu’une variable “branche”.

La variable “Player” sera notre protagoniste et bénéficiera d’une sprite pour courir et sauter. La variable branche aura également un sprite et sera un élément physique avec lequel le “Player” pourra interagir en s’y accrochant.

Nous mettrons en place un input qui permettra au joueur de cliquer afin de faire sauter le personnage.

Nous déclarons une autre variable “chrono” qui sera nécessaire pour faire apparaître du texte. Ce texte permettra de faire paraître le compte à rebours du micro-jeu via la fonction “this.time.addEvent”. Par le biais du “delay” nous pouvons choisir à quel moment apparaît le texte.

```
// chronomètre
this.time.addEvent({
  delay: 1000,
  callback: ()=>{
    chrono = this.add.text(50, 50, '5', {fontSize: '100px', fill:'#FFF'});
  },
  loop: false
});
```

Nous utiliserons également “this.time.addEvent” pour timer la course de notre personnage, choisir à quel moment il arrive près du trou. Cela permettra de faire apparaître la consigne au bon moment.

Micro-Jeu 8

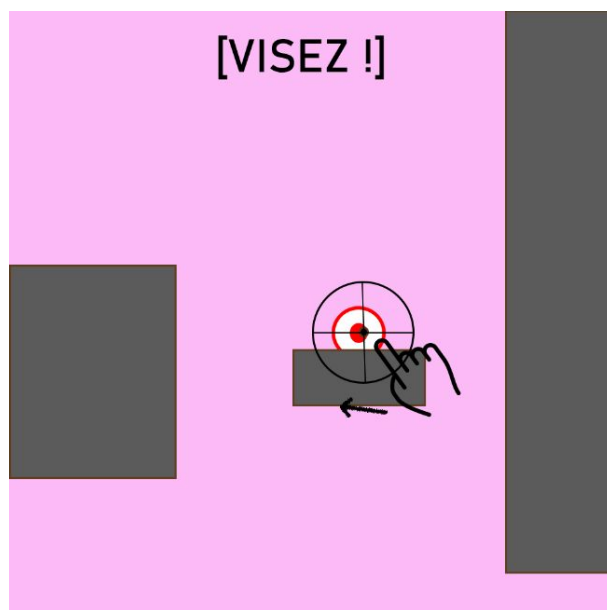
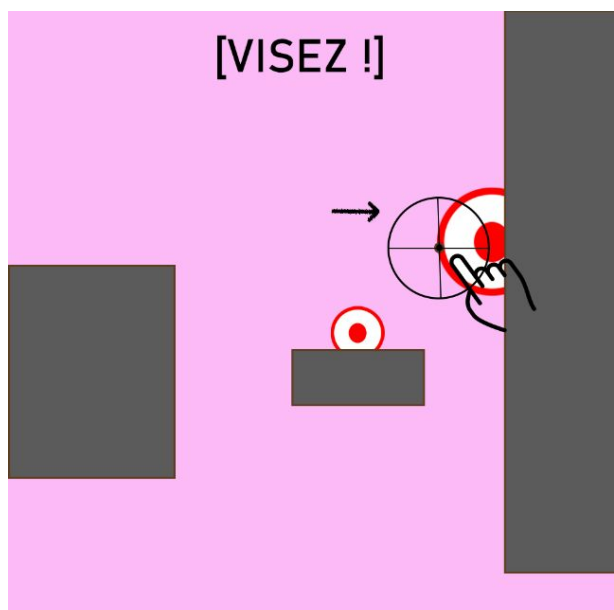
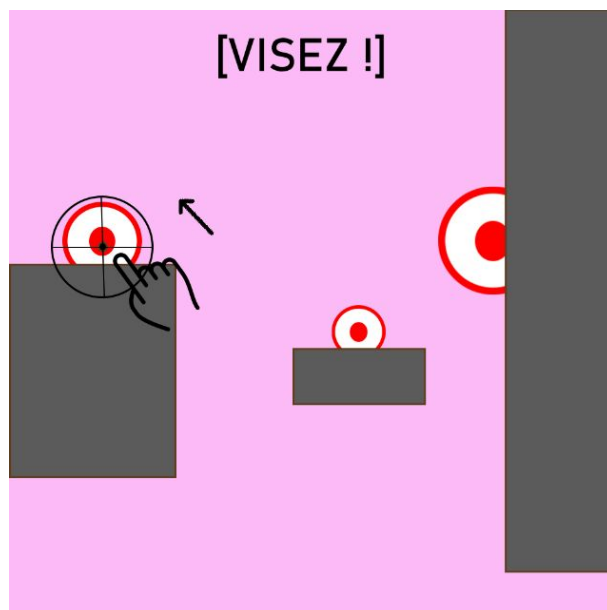
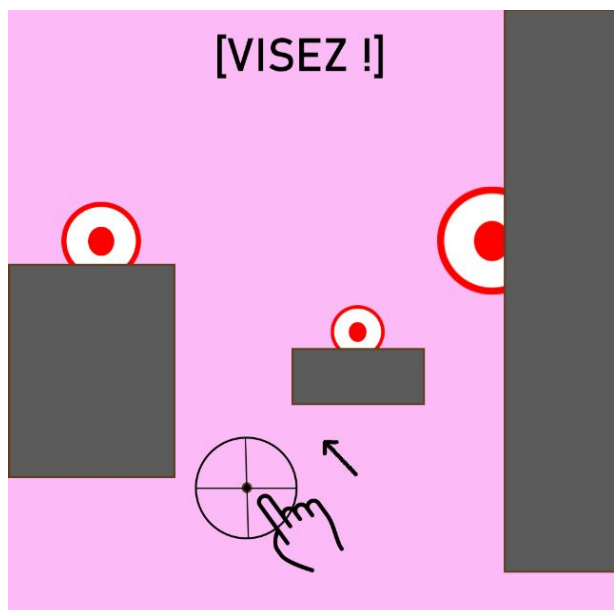
Vue d'ensemble

« Tirons notre révérence »

Ce micro-jeu pousse le joueur à détruire toutes les cibles visible à l'écran dans un lapse de temps imparti. Le sens du toucher et de l'observation sont nécessaires pour déplacer le viseur et repérer les cibles.

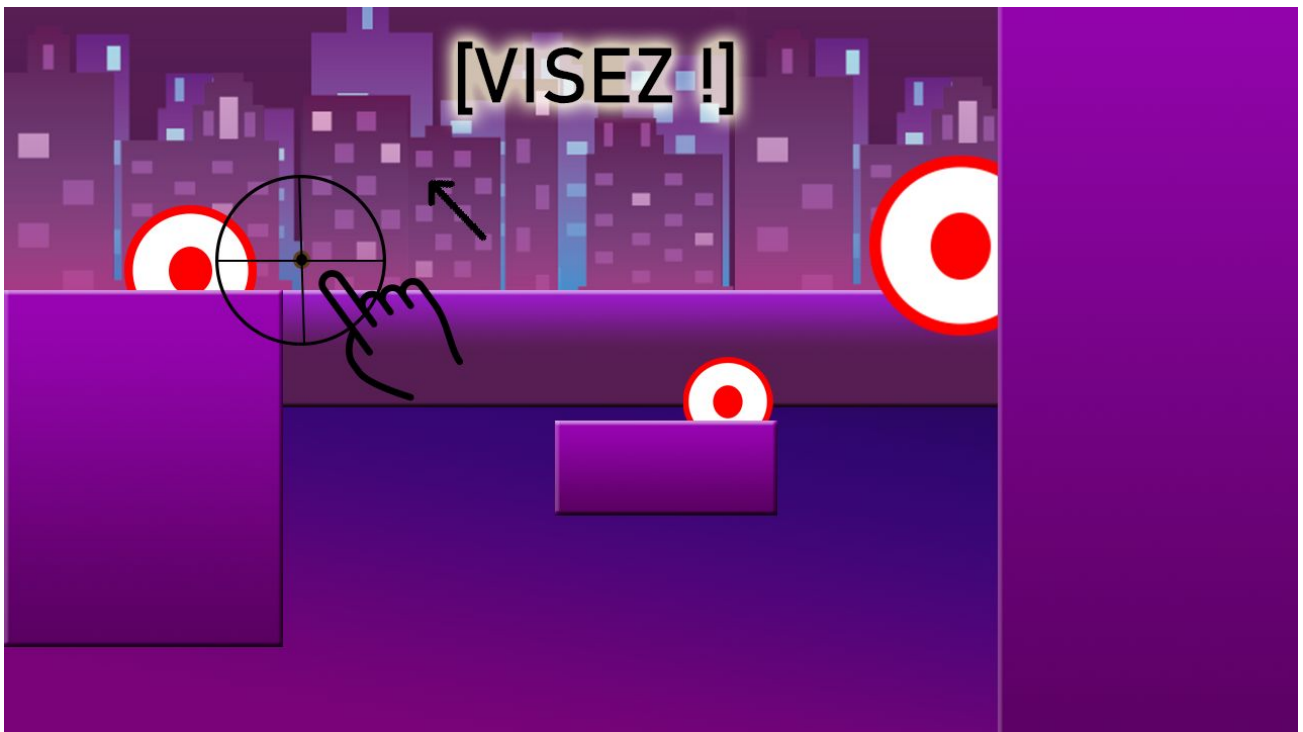
Gameplay

Le joueur possède un viseur qu'il doit faire glisser sur les cibles. Lorsque le viseur touche une cible celle-ci est détruite.



Look&Feel

Ce micro-jeu se déroule à l'extérieur, de nuit. On perçoit en arrière fond la ville illuminée au néon, toujours dans les teints mauves et bleus.



Les cibles, qui représentent nos ennemis, sont dissimulées derrière des murs. Le viseur est toujours visible sur l'écran mais il ne bougera pas tant que le joueur ne le fera pas glisser.

Données

Nous déclarons les variables comme montré ci-dessous :

```
var cible1;  
var cible2;  
var cible3;  
  
var viseur;  
var cursor;  
  
var gameOver = false;  
  
var c = 0;  
var timedEvent;  
var timertext;  
var overtext;
```

Le jeu nécessite quatre assets. Le 'fond' et les 'murs' constituent le décor avec lequel nous ne pouvons pas interagir.

```
function preload() {  
    this.load.image('murs', 'assets/murs.png');  
    this.load.image('viseur', 'assets/viseur.png');  
    this.load.image('fond', 'assets/minijeu1fond.png');  
    this.load.image('cible', 'assets/cible.png');  
}
```

Les cibles sont des éléments physiques car nous devons les viser afin de les détruire :

```
var cible1 = this.physics.add.image(180,250,'cible');
var cible2 = this.physics.add.image(1000,200,'cible');
var cible3 = this.physics.add.image(640,350,'cible');
```

Nous mettons en place un input qui nous permettra de cliquer sur le ‘viseur’ pour le faire glisser où nous le souhaitons. Lorsque le viseur entre en contact avec une cible, cette dernière se détruit.

```
//Le joueur possède un viseur qu'il doit déplacer pour détruire les cibles
```

```
var viseur = this.physics.add.image(600, 600, 'viseur')
    .setCollideWorldBounds(true);

this.input.setDraggable(viseur.setInteractive());

viseur.on('drag', function (pointer, dragX, dragY) {
    this.x = dragX;
    this.y = dragY;
});
```

Nous déclarons une autre variable “chrono” qui sera nécessaire pour faire apparaître du texte. Ce texte permettra de faire paraître le compte à rebours du micro-jeu via la fonction “this.time.addEvent”. Par le biais du “delay” nous pouvons choisir à quel moment apparaît le texte.

```
// chronomètre

this.time.addEvent({
    delay: 1000,
    callback: ()=>{
        chrono = this.add.text(50, 50, '5', {fontSize: '100px', fill:'#FFF'});
    },
    loop: false
});
```


Micro-Jeu 9

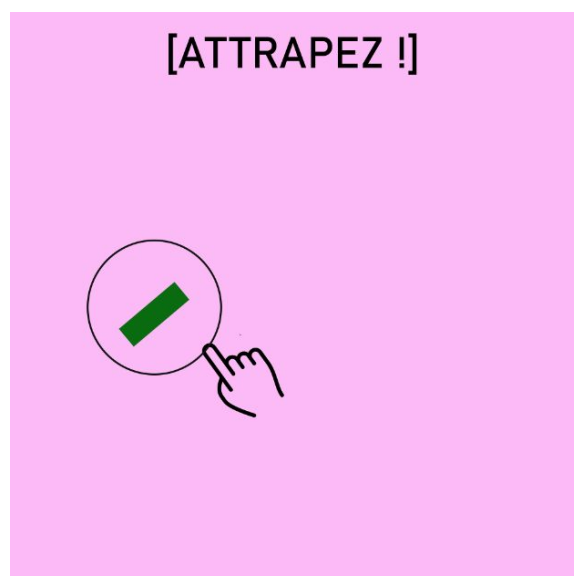
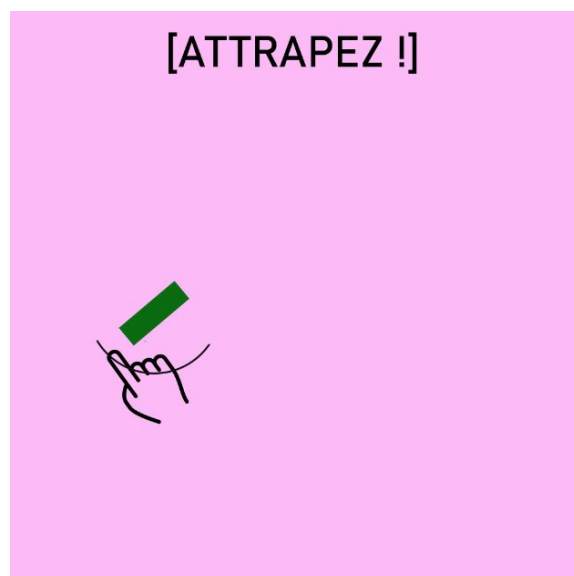
Vue d'ensemble

« L'argent par les fenêtre »

Ce micro-jeu nécessite principalement de la vitesse. Des billets apparaissent à l'écran et il faut les attraper avant que le temps ne soit écoulé.

Gameplay

Des éléments apparaissent à l'écran. Le joueur doit former un cercle autour d'un élément afin de l'attraper. Il n'est possible d'attraper qu'un élément à la fois.



Look&Feel

Nous retrouvons un ciel semblable à celui de la ville et de l'interface pour le menu du jeu. Ce ciel est nuancé, allant d'un rose fushia jusqu'à un mauve plus foncé. Démonstration que nous sommes encore à l'extérieur, de nuit au vue de la présence des étoiles.

Les billets sont d'une couleur verte qui contraste avec le fond et permet mieux de les repérer.



Données

Nous déclarons la variable “billet” qui sera nécessaire pour faire apparaître les billets à l’écran.

Ce micro-jeu a besoin d’un sprite pour “billet” ainsi qu’un background.

Il nous faudra un input qui nous permet de glisser le curseur pour former un cercle autour des “billet”

Nous déclarons une autre variable “chrono” qui sera nécessaire pour faire apparaître du texte. Ce texte permettra de faire paraître le compte à rebours du micro-jeu via la fonction “this.time.addEvent”. Par le biais du “delay” nous pouvons choisir à quel moment apparaît le texte.

```
// chronomètre

this.time.addEvent({
  delay: 1000,
  callback: ()=>{
    chrono = this.add.text(50, 50, '5', {fontSize: '100px', fill:'#FFF'});
  },
  loop: false
});
```

Nous pouvons choisir à quel moment apparaissent les billets, toujours par le biais de “this.time.addEvent.”

Micro-Jeu 10

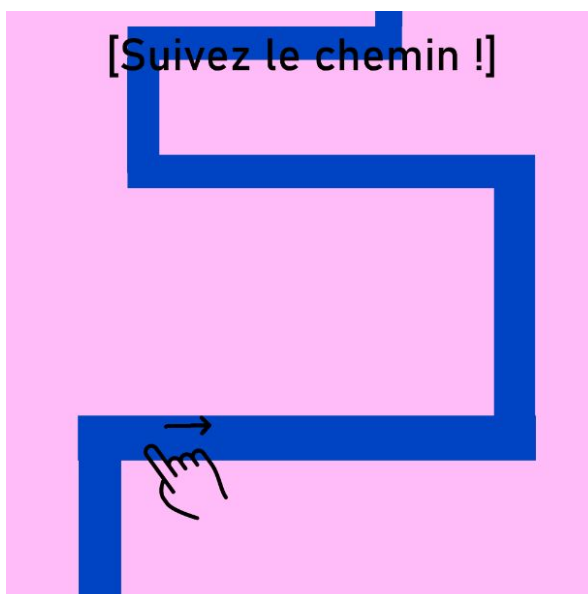
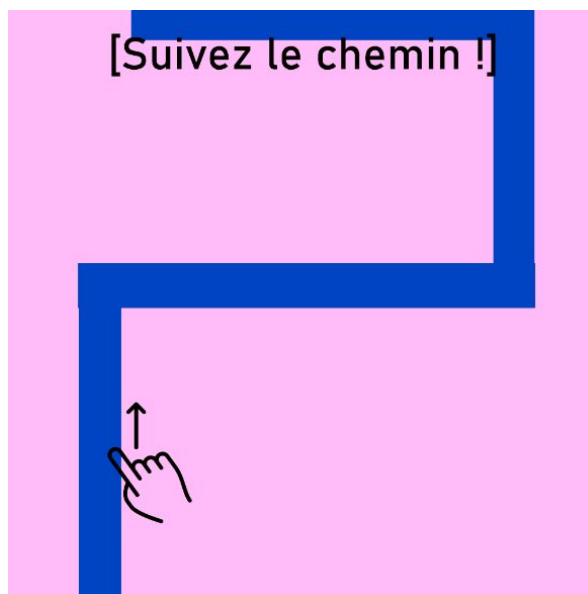
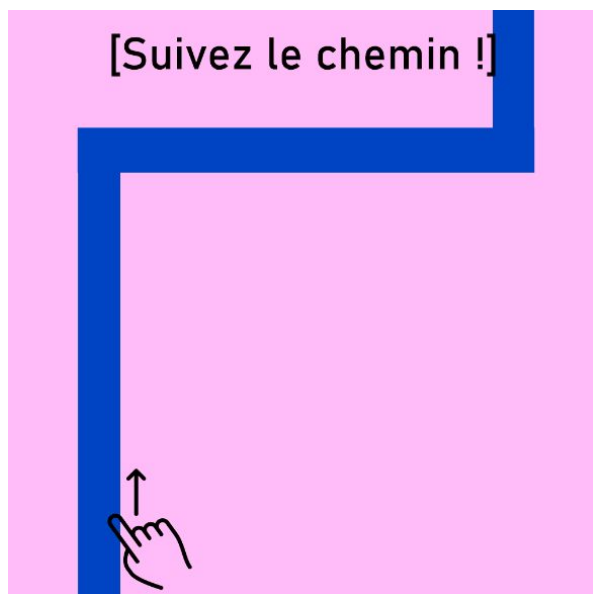
Vue d'ensemble

« Dernière ligne droite »

Ce dernier micro-jeu nécessite de la concentration et de l'observation. Il faut prendre garde au chemin qui défile, faire attention aux tournants, tout en restant sans cesse appuyer dessus. Cesser d'appuyer ou bien mal suivre le chemin mènent à la défaite.

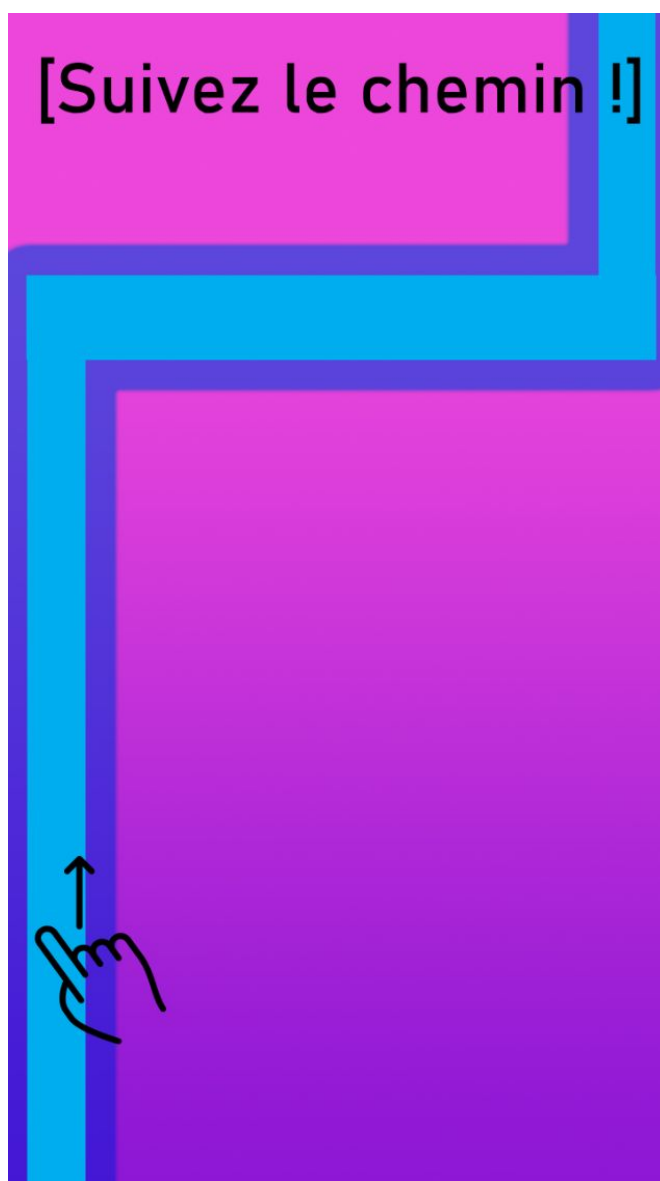
Gameplay

Le joueur doit rester appuyer sur le chemin qui se révèle à lui. Le chemin défile au fur et à mesure, de haut en bas.



Look&Feel

Le fond rose représente la ville tandis que le chemin se révèle en bleu. On retrouve une nouvelle fois les couleurs bleu et mauve de la retrowave, toujours avec l'effet de néon. Le jeu se déroule en 1280 x 720, ce qui incite le joueur à tourner son écran.



Données

Nous déclarons une variable “chemin”. Cette variable se verra attribuer un sprite qui défilera de haut en bas.

Nous mettrons en place un input pour cliquer sur ce chemin.

Ajout des conditions de victoire et de défaite = si le joueur n’est plus en train de cliquer sur le chemin, il perd. Autrement, si le joueur maintient le clique sur le chemin jusqu’à la fin du temps imparti, il gagne.

Nous déclarons une autre variable “chrono” qui sera nécessaire pour faire apparaître du texte. Ce texte permettra de faire paraître le compte à rebours du micro-jeu via la fonction “this.time.addEvent”. Par le biais du “delay” nous pouvons choisir à quel moment apparaît le texte.

```
// chronomètre

this.time.addEvent({
  delay: 1000,
  callback: ()=>{
    chrono = this.add.text(50, 50, '5', {fontSize: '100px', fill:'#FFF'});
  },
  loop: false
});
```