

Premiers Pas

- Premiers Pas
- Partie 1
- Partie 2
- Partie 3

Lore

> Dans un désert infini, sous un ciel pixelisé, un petit dinosaure court sans fin, sautant pour éviter les cactus et les ptérodactyles. Mais que se passerait-il si vous pouviez briser les règles du jeu ? Et si, au lieu d'obéir aux limites imposées, vous pouviez les réécrire à votre guise ?

La méthode Epitech : Learn different

- Vous allez acquérir des nouvelles compétences en pratiquant et en vous amusant.
- Votre premier réflexe lorsque vous vous posez une question doit être de demander de l'aide à vos voisins. Si vous ne trouvez pas la solution demandez de l'aide à un accompagnateur.

- Pour avancer sereinement dans la réalisation de votre projet, pensez à faire valider chaque partie au fur et à mesure par un animateur Epitech qui pourra vous donner des conseils.
- Nous vous encourageons à utiliser les resources disponibles sur internet pour trouver des réponses à vos questions et pour vous renseigner.
- Vous pouvez ajouter des fonctionnalités de votre choix et personnaliser votre projet : c'est ce que nous appelons un bonus à Epitech et c'est possible pour tous les projets.
- Les IA génératives peuvent sûrement faire cet atelier, mais si vous les utilisez vous n'apprendrez rien.



Pré-requis

Pour réaliser cette activité, vous devez utiliser le navigateur Chrome : le jeu Chrome Dino est intégré directement dans ce navigateur et n'est pas accessible autrement.

Introduction : Découverte du jeu Chrome Dino

Le jeu Chrome Dino est un jeu caché dans le navigateur Google Chrome. Il apparaît automatiquement lorsque vous perdez votre connexion Internet, mais vous pouvez aussi y jouer à tout moment.

attachment:abd973da-d751-4216-8dec-ef3d9c3b72de:Screen_Recording_2025-02-17_095713.mp4

Pour commencer, ouvrez la page du jeu Chrome Dino : copiez-collez l'URL **chrome://dino/** dans la barre de recherche en haut de la page web.

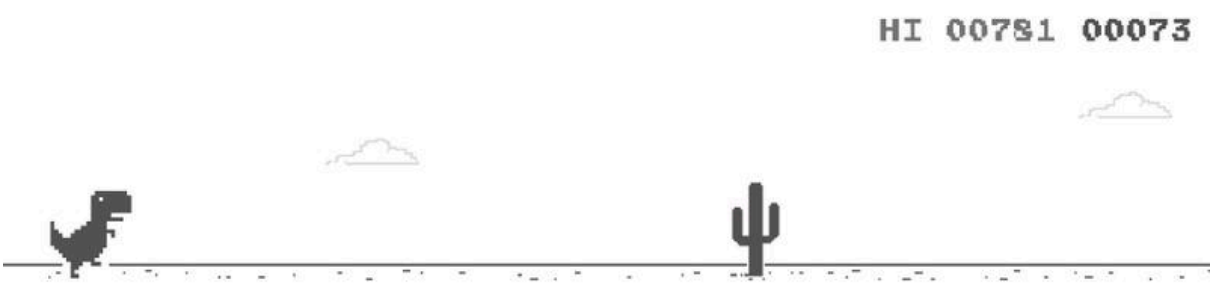


L'objectif est simple : courir le plus loin possible sans heurter d'obstacles. Prenez quelques minutes pour tester le jeu :

- Appuyez sur la barre `Espace` ou la `flèche haut (↑)` pour sauter.
- Maintenez la `flèche bas (↓)` pour éviter les ptérodactyles quand ils volent bas.
- Plus vous avancez, plus la vitesse augmente !

> 🔥 Si vous voulez tricher ou modifier le jeu, il est possible d'accéder à son code et d'en changer les règles. Voulez-vous apprendre comment hacker le jeu Chrome Dino ? Rendez-vous dans la 🦖 Partie 1 !

> 💙 **Crédits :** Cet atelier a été écrit et testé à Epitech Montpellier pour le Coding Club.



Partie 1

- Premiers Pas
- Partie 1
- Partie 2
- Partie 3

- Etape 1 : Découvrez l'inspecteur Chrome
  - Tutoriel
  - Exercice 1 : Affichez "Hello World"
- Etape 2 : Modifiez le jeu
  - Tutoriel
  - Exercice 1 : Changez la position du dinosaure
  - Exercice 2 : Manipulez les fonctions du jeu
  - Exercice 3 : Devenez invincible
  - Exercice 4 : Changez la touche de saut

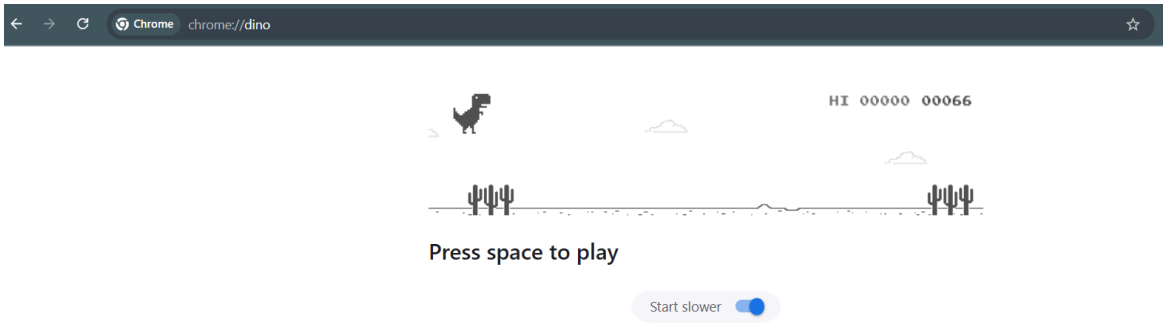
Dans cette partie, vous allez prendre en main l'**inspecteur Chrome** pour modifier le jeu : vous voulez sauter plus haut ou devenir invincible ? C'est possible !

Etape 1 : Découvrez l'inspecteur Chrome

Objectif : Apprendre à utiliser l'**inspecteur Chrome** pour coder en **JavaScript**.

Tutoriel

Pour commencer, ouvrez la page du jeu Chrome Dino : copiez-collez l'URL **chrome://dino/** dans la barre de recherche en haut de la page web.

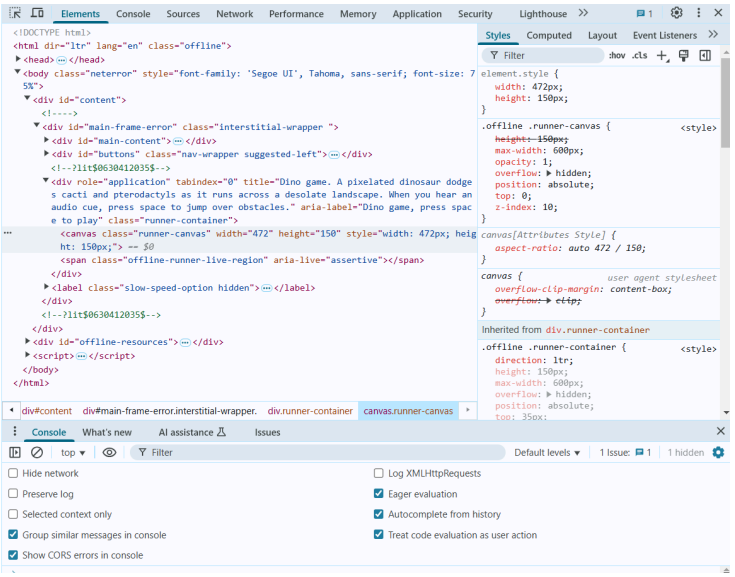


Ouvrez l'**inspecteur Chrome** : pour cela, appuyez sur `F12` ou `Ctrl + Shift + I` (`Cmd + Option + I` sur Mac) ou faites un clic droit sur la page web et sélectionnez "Inspecter" ou "Inspecter l'élément".

> L'**inspecteur Chrome**, aussi appelé **DevTools**, est un outil intégré à Google Chrome permettant d'analyser et de modifier le code **HTML**, **CSS** et **JavaScript** d'une page web en temps réel.

Voici à quoi ressemble l'**inspecteur Chrome** :





Celui-ci a plusieurs onglets dont :

- **Elements** : Permet de voir et modifier le **HTML et CSS** en direct.



L'**HTML (HyperText Markup Language)** est le langage de base pour structurer le contenu d'une page web. Il utilise des balises pour définir différents types de contenu comme des titres, des paragraphes, des images, des liens, etc.

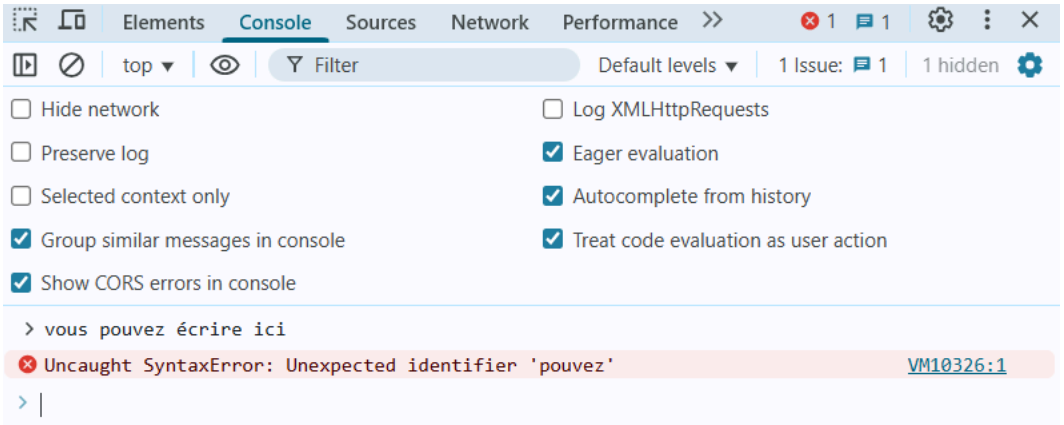
- **Console** : Sert à exécuter du **JavaScript** et voir les erreurs.



Le **JavaScript** est un langage de programmation utilisé principalement pour rendre les pages web interactives. Alors que l'**HTML** structure le contenu, le **JavaScript** ajoute de la dynamique : clics, animations, mises à jour en temps réel, etc.

- **Sources** : Permet de debugger du code **JavaScript**.
- **Network** : Analyse les requêtes **HTTP** envoyées et reçues par le site.
- **Performance** : Vérifie la rapidité du site.
- **Application** : Accède au **LocalStorage, cookies, cache**...

Allez dans l'onglet **Console**. Vous pouvez écrire du code **Javascript** ici pour interagir avec le jeu. Pour exécuter une ligne de code, appuyez sur la touche `Entrée`.



💡 Si jamais vous voulez recharger la page parce-que vous avez trop modifié le code de la page, appuyez sur `Ctrl + Shift + R` (`Cmd + Option + R` sur Mac) ou faites un clic droit et appuyez sur “Recharger”.

Maintenant, testons un script simple !

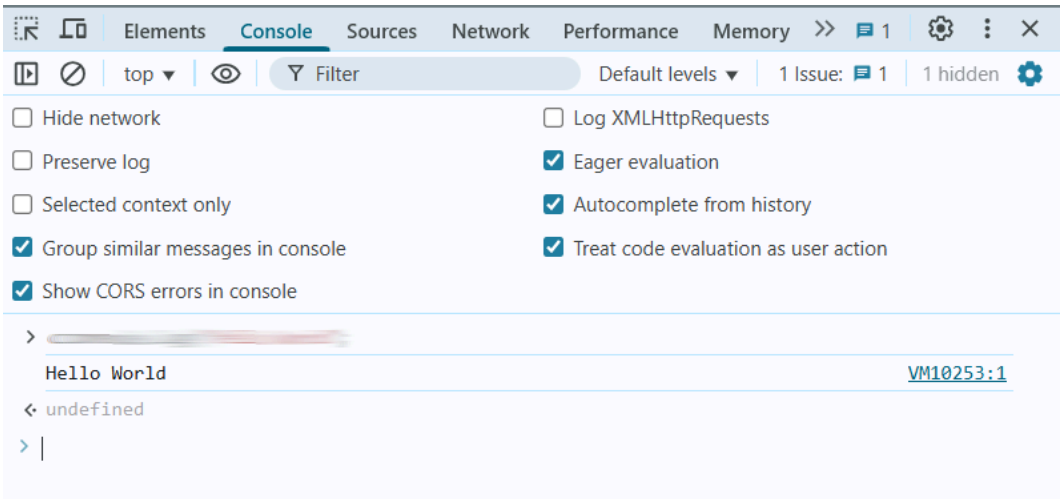
💻 Exercice 1 : Affichez “Hello World”

Dans ce premier exercice, votre but est d’afficher “Hello World” dans la **console**. Pour cela, utilisez la **fonction** `console.log()` (https://www.geeksforgeeks.org/javascript-console-log-method/) qui permet d'afficher un message dans la **console**.

📕 Une **fonction** est un **bloc de code réutilisable** qui effectue une tâche spécifique. Elle permet de regrouper des instructions pour les exécuter facilement à plusieurs endroits dans un programme. Une **fonction** est appelée comme ceci :

```
nomDeLaFonction(paramètres);
```

Résultat type :



🦖 Etape 2 : Modifiez le jeu

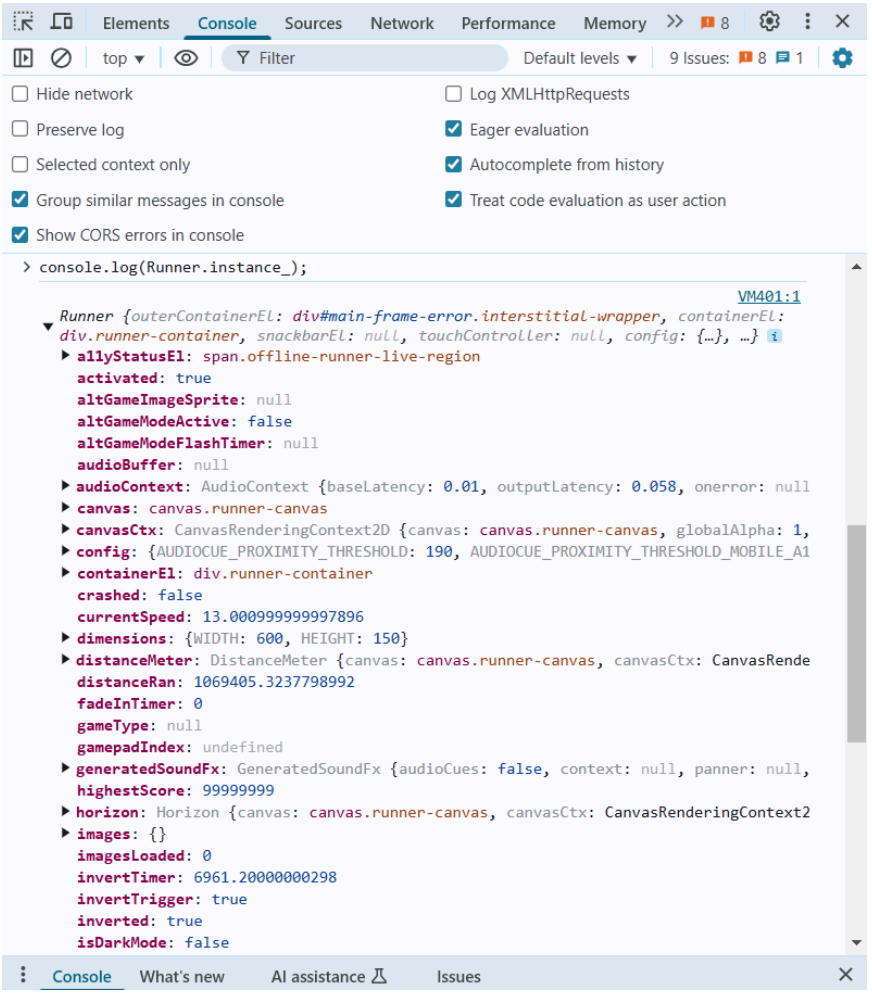
Objectif : Modifier les règles du jeu Chrome Dino à votre avantage.

📖 Tutoriel

L'**objet** `Runner.instance_` contient toutes les informations nécessaires au fonctionnement du jeu. Pour afficher toutes les **variables** de cet **objet** dans la **console**, utilisez cette commande :

```
console.log(Runner.instance_);
```

Résultat type :



Vous pouvez explorer les différentes propriétés, telles que :

- `config` : Contient des paramètres de jeu comme la vitesse, la gravité, etc.
- `tRex` : Représente le dinosaure, avec des propriétés comme sa position et sa vitesse.

📕 Une **variable** permet de stocker une valeur et de la réutiliser dans le code, en évitant de répéter constamment la valeur. Vous pouvez annoncer une variable comme ceci :

```
let x;
```

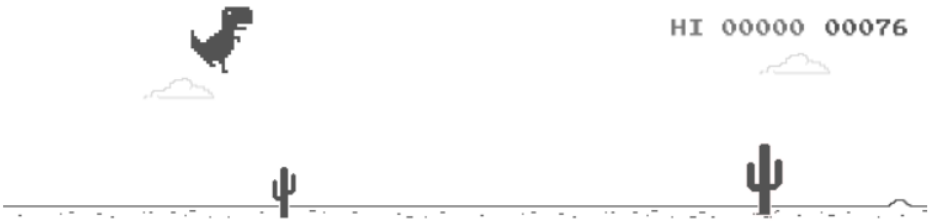
On peut lui attribuer une valeur comme ceci :

```
x = 10;
```

💻 Exercice 1 : Changez la position du dinosaure

Dans cet exercice, vous devez changer la position du dinosaure dans le jeu. Aidez-vous des **variables** présentes dans l'**objet** `Runner.instance_.tRex`.

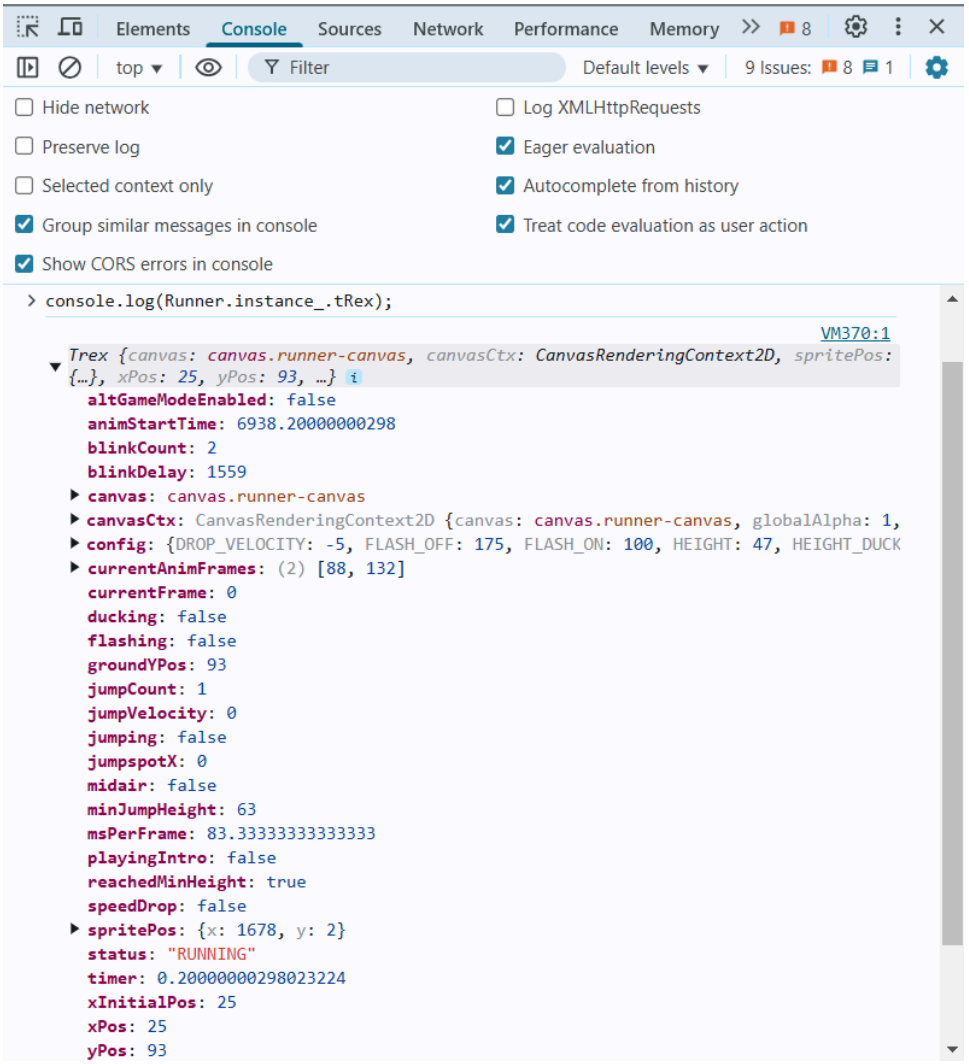
Résultat type :



Pour visualiser les **variables**, vous pouvez utiliser la commande :

```
console.log(Runner.instance_.tRex);
```

Résultat type du `console.log()` :



🔥 Essayez de trouver et de modifier la **variable** qui détermine la fréquence des changements entre le mode jour et le mode nuit.

💻 Exercice 2 : Manipulez les fonctions du jeu

Dans cet exercice, le but est d'utiliser les **fonctions** du jeu au lieu de **variables** pour modifier le fonctionnement du jeu. Vous allez explorer comment interagir directement avec les **fonctions** du jeu pour contrôler le comportement du dinosaure, modifier l'affichage, et plus encore.

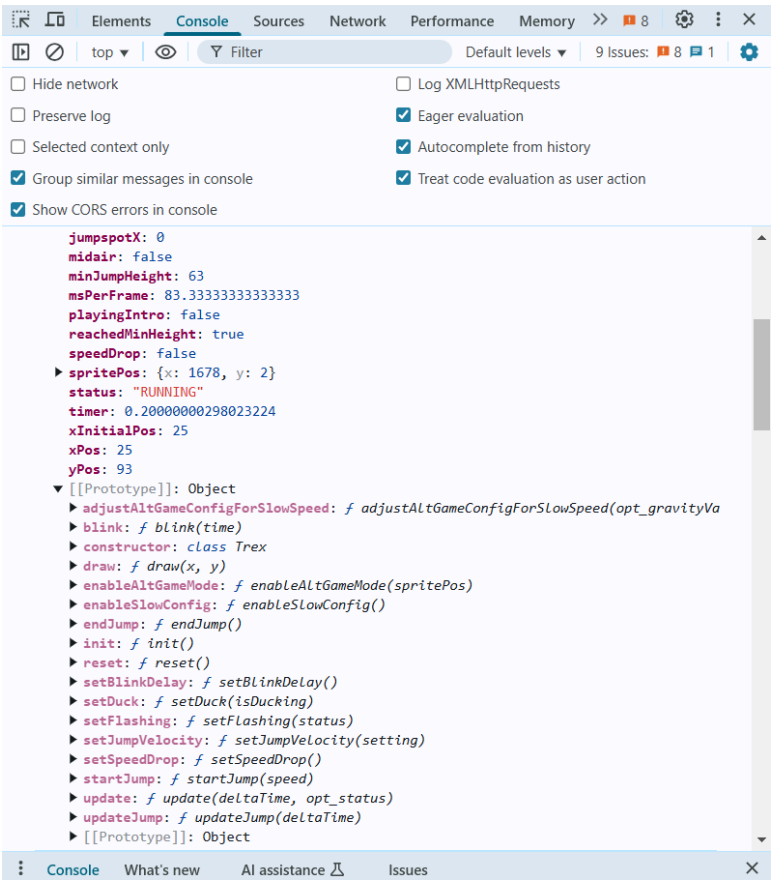


En **JavaScript**, les **objets** ont des **prototypes**, qui sont des modèles contenant des **fonctions** pouvant être modifiées ou étendues. Le jeu Chrome Dino repose sur des **objets** et des **prototypes**, ce qui nous permet d'intervenir directement sur ses fonctionnalités.

Les **fonctions** sont visibles à la fin de chaque **objet**, lorsque vous faites ceci par exemple :

```
console.log(Runner.instance_);
```

Résultat type :



Les **fonctions** apparaissent en rose dans le menu déroulant appelé `[[Prototype]] : Object]]` et ont un indicateur `f`, avec un exemple d'utilisation et les **paramètres** nécessaires.

Pour utiliser la **fonction** `endJump()` par exemple, vous pouvez mettre ce code dans la **console** :

```
Runner.instance_.tRex.endJump();
```

Vos objectifs sont les suivants :

- Augmenter la hauteur du saut du dinosaure
- Faire sauter le dinosaure
- Baisser le dinosaure
- Changer le mode du jeu (jour / nuit)
- Changer le score actuel
- Sauvegarder un nouveau highscore

> 💡 Il est possible qu'un objectif ait plusieurs solutions. Fouillez bien dans les **fonctions** du jeu pour les trouver !

💻 Exercice 3 : Devenez invincible

Le but de cet exercice est de rendre le dinosaure invincible en modifiant le comportement du jeu, notamment la **fonction** qui termine le jeu lorsque le dinosaure touche un obstacle. Vous devrez sauvegarder la **fonction** originale et la remplacer pour empêcher le jeu de se terminer.

Essayez de trouver et désactiver cette **fonction** pour éviter la fin du jeu et assurez-vous de pouvoir la restaurer lorsque vous le souhaitez.

▼ Indice

Pour désactiver une **fonction**, il suffit d'assigner son **prototype** à une **fonction** vide :

```
Runner.prototype.invert = function() {};
```

La **fonction** `invert()` permet de changer entre les modes jour / nuit, la désactiver comme ci-dessus bloque le jeu dans le mode actuel. Vous pouvez donc rester en mode jour ou en mode nuit aussi longtemps que vous le voulez !

> 💡 N’oubliez pas de sauvegarder la **fonction** dans une **variable** avant de la désactiver si vous voulez pouvoir restaurer la **fonction** plus tard sans recharger la page.

💻 Exercice 4 : Changez la touche de saut

Pour cet exercice, changez la touche qui permet de faire sauter le dinosaure. Par défaut, le jeu utilise la barre d'espace, mais vous pouvez ajouter n'importe quelle autre touche de votre choix. Utilisez les événements du clavier pour capturer une nouvelle touche (par exemple "Z") et déclencher le saut du dinosaure.

Pour cela, vous pouvez utiliser la **fonction** `addEventListener("keydown")` (en français “ajouter un dispostif d’écoute sur l’événement touche enfoncée”) (https://www.w3schools.com/jsref/met_document_addeventlistener.asp) qui permet de déclencher une action lorsque l’on appuie sur une touche.

▼ Indice

Voici un code qui permet d’écouter les événements clavier et afficher les informations de la touche appuyée :

```
document.addEventListener("keydown", function(event) {
    console.log(event);

    // Codez ici : Appuyez sur 'Z' pour sauter
});
```

Rendez-vous dans la 🦖 Partie 2 pour jouer de manière automatiser et même changer l’apparence du jeu !

> 💙 **Crédits :** Cet atelier a été écrit et testé à Epitech Montpellier pour le Coding Club.



Partie 2

- Premiers Pas
- Partie 1
- Partie 2
- Partie 3

- Etape 3 : Jouez automatiquement
  - Tutoriel
  - Exercice 1 : Sautez à l'infini
  - Exercice 2 : Détectez les obstacles
  - Exercice 3 : Faites sauter le dinosaure
- Etape 4 : Changez l'apparence
  - Tutoriel
  - Exercice 1 : Créez vos propres sprites
- Pour aller plus loin

Avant de commencer cette partie, réalisez la Partie 1 pour comprendre comment faire des modifications simples sur le jeu Chrome Dino. Dans cette deuxième partie, vous allez faire des manipulations plus avancées pour automatiser le jeu et changer son apparence.

Etape 3 : Jouez automatiquement

Objectif : Automatiser pour que le jeu se déroule de manière autonome.

Tutoriel

La **fonction** `setInterval()` permet d'exécuter de façon répétée un morceau de code à un intervalle de temps défini (en millisecondes).

Vous pouvez l'utiliser comme ceci :

```
setInterval(function () {
    console.log("Je m'affiche toutes les 2 secondes !");
}, 2000);
```

Dans les prochains exercices, vous allez l'utiliser pour exécuter du code à chaque `Runner.instance_.msPerFrame`, ce qui correspond à chaque mise à jour du jeu.

💻 Exercice 1 : Sautez à l'infini

Le but de cet exercice est de faire sauter le dinosaure répétitivement, en utilisant la **fonction** `setInterval()`. Pour faire sauter le dinosaure, vous avez plusieurs méthodes à votre disposition :

- Directement utiliser la **fonction** pour faire sauter le dinosaure.
- Simuler d'appuyer sur la touche `Espace` (en utilisant la **fonction** `dispatchEvent()`).

▼ Indice

Pour la deuxième méthode avec la **fonction** `dispatchEvent()`, si vous souhaitez obtenir le bon `keyCode` pour la touche `Espace` , faites :

```
document.addEventListener("keydown", function(event) {
        console.log(event);
});
```

Ensuite, appuyez sur la touche `Espace` et retenez le `keyCode` affiché dans la console.

💻 Exercice 2 : Détectez les obstacles

Dans cet exercice, votre objectif est de repérer les obstacles à l'écran et d'identifier quand ils approchent du dinosaure. En analysant la position des obstacles, vous pourrez savoir à quel moment il faut réagir.

Affichez un message lorsqu'un obstacle est proche du dinosaure et celui-ci devrait sauter pour l'éviter. Vous devez pouvoir jouer sans regarder l'écran, en

se servant uniquement des messages affichés dans la console.

Pour cela, vous aurez besoin de comprendre ce qu'est une **condition** (`if`) :

📕 Les conditions

Mais aussi le fonctionnement d'un tableau :

📕 Les tableaux

▼ Indice

Vous pouvez accéder au prochain obstacle dans le **tableau** d'obstacles affichés à l'écran grâce à cette **variable** :

```
Runner.instance_.horizon.obstacles[0]
```

Un obstacle est retiré de ce **tableau** une fois qu'il n'est plus visible à l'écran.

💻 Exercice 3 : Faites sauter le dinosaure

Maintenant que vous pouvez détecter les obstacles, il est temps de réagir automatiquement !

Vous devez programmer le dinosaure pour qu'il saute lorsqu'un obstacle est proche, et se baisse si nécessaire (contre les pterodactyles).

Pour cela, aidez-vous des derniers exercices et combinez-les pour obtenir votre solution.

🦖 Etape 4 : Changez l'apparence

Objectif : Modifier l'apparence du jeu Chrome Dino en changeant les **sprites** du dinosaure et des obstacles.

📖 Tutoriel

Le jeu Chrome Dino utilise des images appelées **sprites** pour afficher le dinosaure et les obstacles. Ces images sont chargées depuis des fichiers internes du navigateur.

Selon la taille de votre écran, Chrome charge l'un de ces fichiers pour les **sprites** du jeu, d'`id` :

- Petit écran : `offline-resources-1x`



- Grand écran : `offline-resources-2x`



Pour modifier les sprites, vous pouvez charger une nouvelle image directement dans le jeu avec cette commande :

```
document.getElementById('offline-resources-1x').src='URL_DE_VOTRE_IMAGE';
document.getElementById('offline-resources-2x').src='URL_DE_VOTRE_IMAGE_HD';
```

Si vous voulez utiliser un skin **Minecraft** par exemple, utilisez :

```
document.getElementById('offline-resources-1x').src='https://i.imgur.com/vhUjTZL.png';
document.getElementById('offline-resources-2x').src='https://i.imgur.com/y27zIvC.png';
```



Voici à quoi ressemblent les images :

- Petit écran



- Grand écran



💻 Exercice 1 : Créez vos propres sprites

Dans cet exercice, vous devez créer une nouvelle apparence pour le jeu en dessinant vos propres **sprites**. Pour cela, vous pouvez utiliser les éditeurs d'image suivants :

- https://www.pixilart.com/draw?ref=home-page
- https://pixlr.com/fr/editor/
- https://www.gimp.org/downloads/

Téléchargez les **sprites** du jeu pour votre résolution d'écran (petit ou grand écran) et apportez les modifications que vous souhaitez. Pour :

- Petit écran



- Grand écran



Après vos modifications, hébergez votre image sur un site comme https://imgur.com/upload ou https://postimages.org/fr/, et testez en remplaçant l'URL par le vôtre dans cette commande :

```
document.getElementById('offline-resources-1x').src='URL_DE_VOTRE_IMAGE';
document.getElementById('offline-resources-2x').src='URL_DE_VOTRE_IMAGE_HD';
```

Vous pouvez vous inspirer de personnages et univers déjà existants (Sonic, Mario, Cuphead) ou même inventer le vôtre !

Bien-sûr, n'hésitez pas à partager vos créations avec les autres participants !

Pour aller plus loin

Grâce à tout ce que vous avez appris, ajoutez plein de nouvelles fonctionnalités !

Vous pouvez par exemple :

- Ajouter un double saut
- Permettre au dinosaure de sprinter
- Changer le mode de jeu dynamiquement
- Relancer la partie automatiquement

N'hésitez pas à être créatif, et surtout, amusez-vous ! Merci d'avoir participé à ce Coding Club !

💙 **Crédits :** Cet atelier a été écrit et testé à Epitech Montpellier pour le Coding Club.



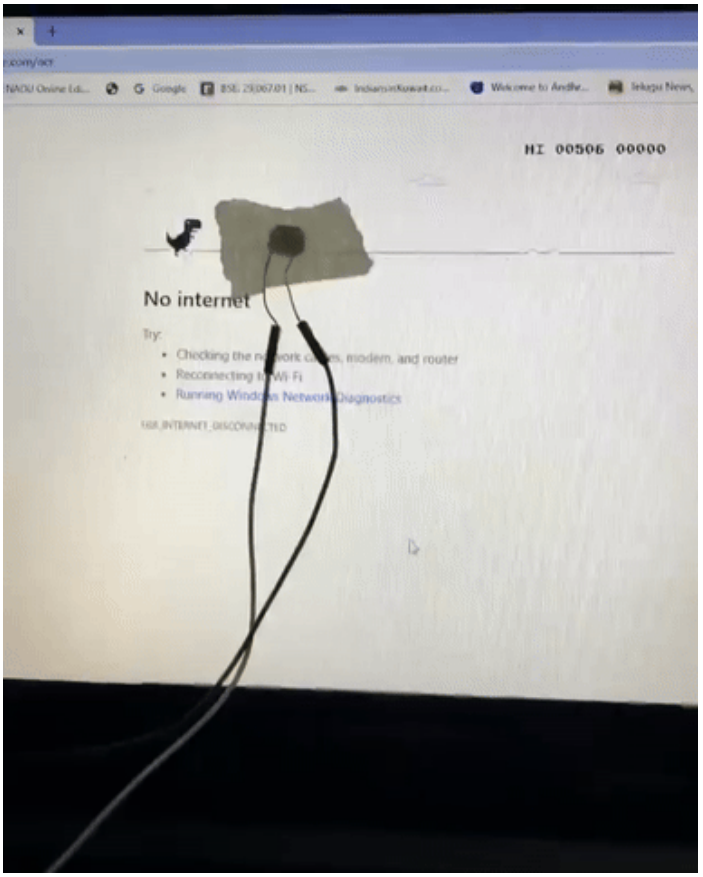
Partie 3

- Premiers Pas
- Partie 1
- Partie 2
- Partie 3

- Etape 5 : Hardware Hack
  - Pré-requis
    - Hardware
    - Software
  - Exercice 1 : Câblage
  - Exercice 2 : Implémentation
  - Exercice 3 : Mise en place

Etape 5 : Hardware Hack

Objectif : Automatiser pour que le jeu se déroule de manière autonome avec un hack hardware



📖 Pré-requis

Hardware

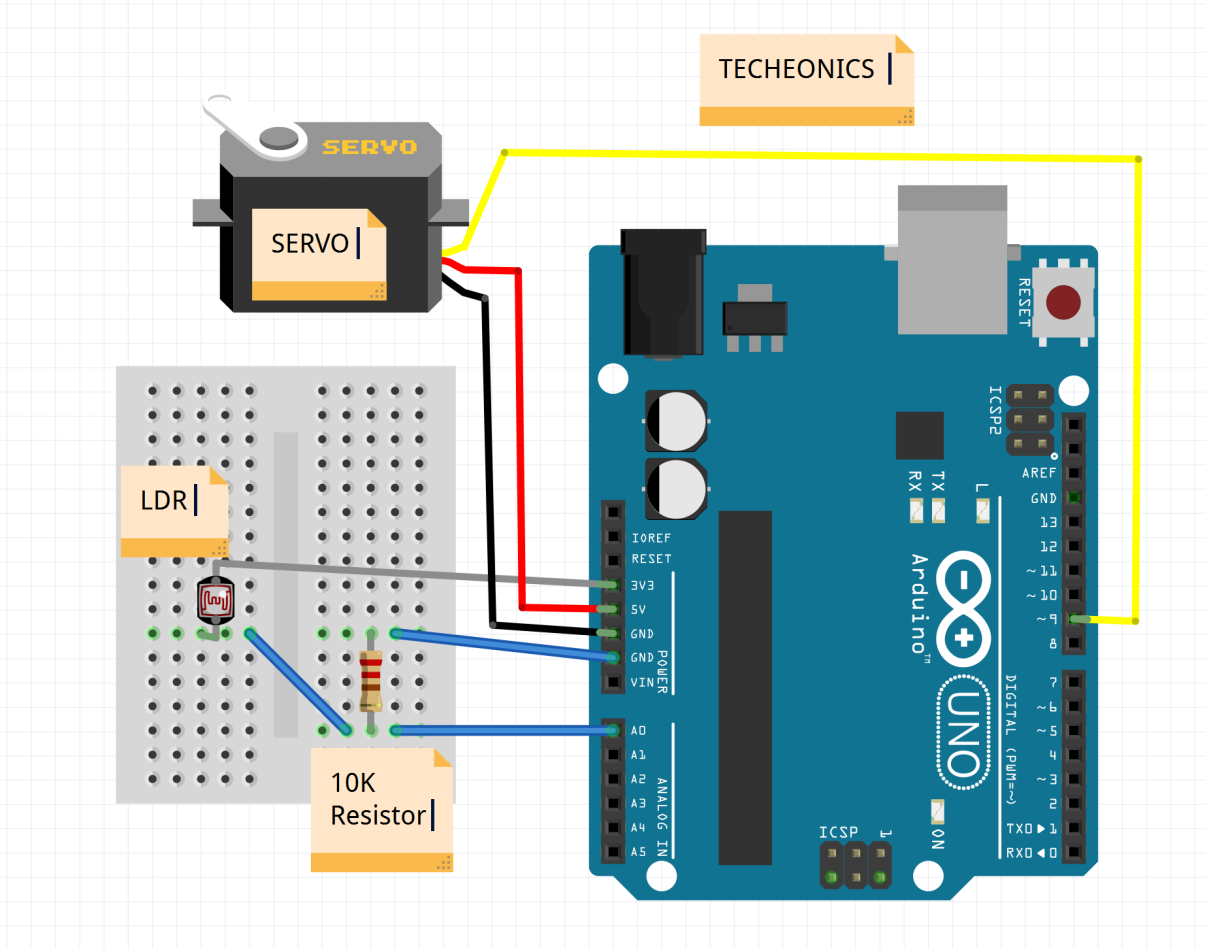
- Un kit Arduino Uno : carte + cable USB + breadboard
- Un servo-moteur
- Une photorésistance (aussi appelée LDR) GL5528
- Une résistance 10K
- Environ 10 câbles de liaison
- Du scotch classique, du scotch double face

Software

- L'IDE adruino : https://www.arduino.cc/en/software/

💻 Exercice 1 : Câblage

à partir des schémas ci-dessous, réalisez le câblage de votre installation qui doit prendre en charge le servo-moteur et la LDR



💻 Exercice 2 : Implémentation

```
#include <Servo.h>
#define threshold 600
#define unpress_angle 70
#define press_angle 36


Servo myservo;
bool trig=true;
int ldr_value=0;

void setup() {
  Serial.begin(9600);
  myservo.attach(10);
```

```
    myservo.write(unpress_angle);
}

void loop() {
 ldr_value = analogRead(A0);
 Serial.println(ldr_value);

 myservo.write(unpress_angle);
 delay(1);

// codez ici la boucle qui va permettre d'activer le servo moteur au bon mome
}
```

💻 Exercice 3 : Mise en place

Grâce aux deux étapes précédente et beaucoup de scotch, vous avez tout pour mettre en place le setup qui va jouer à ta place.

💙 **Crédits :** Cet atelier a été écrit et testé à Epitech Montpellier pour le Coding Club.