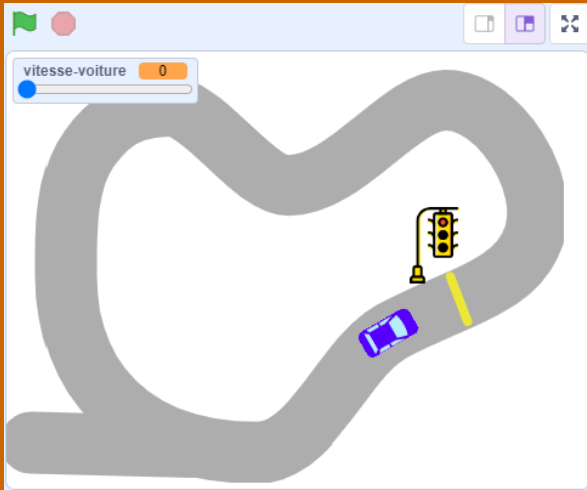




# 6 Une voiture autonome



Elle roule toute seule en suivant le tracé de la route. Elle s'arrête au feu rouge et redémarre quand il passe au vert...

Tu vas découvrir de nouvelles techniques de programmation avec cette simulation.



▼ Vidéo

<https://jeunesingenieurs.be/vid-s2-6>



▼ Proposition de solution

<https://scratch.mit.edu/projects/752207224>

Et une autre pour ceux qui veulent aller plus loin :

<https://scratch.mit.edu/projects/1046315590>



# Une voiture autonome

## ▼ Tu te lances en mode Défi ?

Regarde la vidéo et essaie d'atteindre le résultat proposé. Voici une proposition d'étapes :

1. Prépare un circuit et une ébauche de voiture dotée de deux capteurs qui détecteront les bords de la route pour éviter les sorties de route.
2. Programme la voiture pour qu'elle reste effectivement sur la route.
3. Crash test. Prévois un moyen de modifier la vitesse. À partir de quelle vitesse la voiture a-t-elle tendance à sortir de la route ? Pourquoi ?
4. Ajoute un feu de circulation qui passe automatiquement du vert ou rouge toutes les X secondes.
5. Re-programme ta voiture pour qu'elle s'arrête au feu et redémarre quand il revient au vert.

Si tu veux aller plus loin...

6. Programme la voiture de manière à éviter les sorties de route quelle que soit la vitesse.
7. Utilise la technique de la HITBOX pour que la voiture garde son comportement mais avec un costume qui ressemble plus à une voiture...



## 6.1 UN CIRCUIT ET UNE PETITE VOITURE

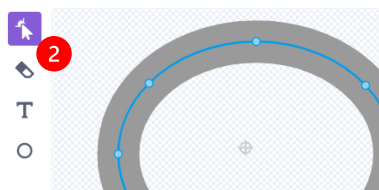
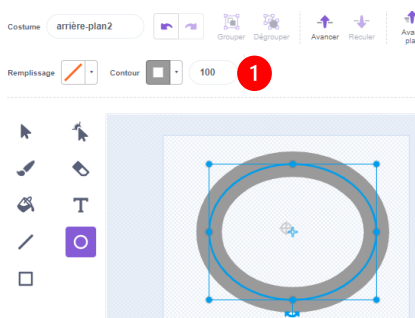
### 🎯 Objectifs

Préparer un circuit simple en arrière-plan et une petite voiture (version très simplifiée).

### 📌 Dessine le circuit sur l'arrière-plan

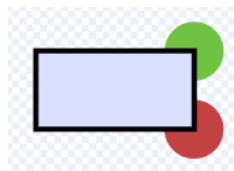
1 Dessine une ellipse de couleur grise avec un contour de 100px.

À l'aide de l'outil 2 *redessiner*, ajoute de points nœuds et déforme le circuit.



### 📌 Prépare un sprite *voiture*

Pour commencer, on se contente d'un simple rectangle doté de deux capteurs : un vert et un rouge. Ceux-ci serviront à détecter les bords de la route pour ajuster la direction du véhicule.





## 6.2 PROGRAMME LA VOITURE

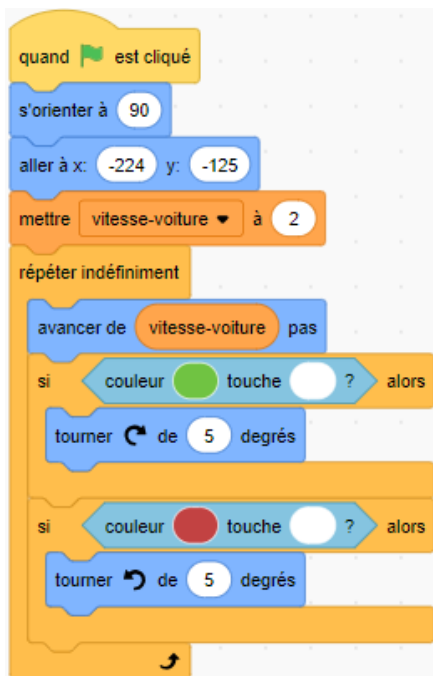
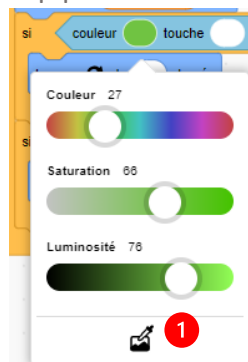
### 🎯 Objectif

Ecrire un programme qui fait avancer la voiture sur le circuit sans qu'elle quitte la route. À l'aide des capteurs rouge et vert, modifie la trajectoire de la voiture quand elle touche le bord de la route.

### ➤ Ajoute ce code à la voiture

Crée une variable **vitesse-voiture**.

Attention. Quand tu testes si le capteur touche la couleur du fond (blanc), il faut aller chercher exactement la bonne couleur. Utilise la **1** pipette !





## 6.3 CRASH TEST

### 🎯 Objectif

Teste le comportement de la voiture lorsqu'elle accélère.

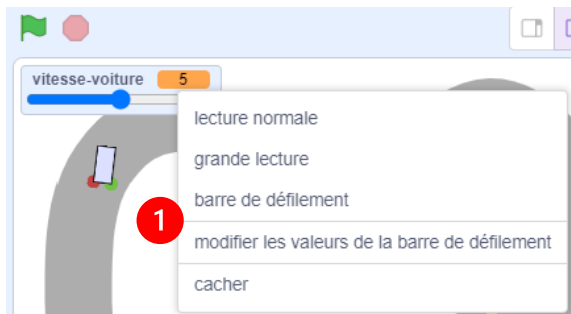
#### ▼ Modifie la vitesse de la voiture

Affiche la variable vitesse-voiture en mode **barre de défilement** et affecte-lui les valeurs minimales et maximales à 0 et 10.

- Affiche la variable
- Clique-droit sur la variable affichée sur la scène
- Choisis **barre de défilement**
- ❶ Clique-droit à nouveau,

choisis

**modifier les valeurs de la barre de défilement** et tape ensuite les valeurs 0 et 10.



#### ▼ Teste le comportement de la voiture

Lance ton programme et teste les différentes vitesses à l'aide du curseur.



# ON S'EXPLIQUE ?

## 🎯 Objectif : répondre à deux questions

1. Que signifie exactement **avancer de ... pas** ?
2. Pourquoi la voiture quitte la piste quand on augmente la vitesse ?



### ▾ 1. Vitesse ou distance parcourue ?

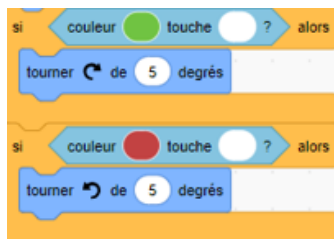
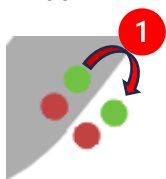
Le bloc **avancer de ... pas** renseigne une distance à parcourir. Mais la boucle **répéter indéfiniment** est programmée par les concepteurs de Scratch pour s'exécuter 30 fois par seconde. La combinaison des deux blocs doit donc se lire : « **avancer de deux pas tous les 1/30<sup>ème</sup> de seconde** » ou encore « **avancer de 60 pas par seconde** ». On peut donc considérer que c'est bien une vitesse qui est exprimée ici.

### ▾ 2. Pourquoi la voiture sort de la piste ?

On a l'impression que la voiture avance de façon continue. Mais en réalité, le bloc **avancer de 10 pas** devrait être lu

1 **apparaître 10 pas plus loin**. Quand on

1 augmente la vitesse, il y a donc un risque que les deux capteurs se retrouvent simultanément dans la zone blanche et exécutent ensemble tourner à gauche et tourner à droite, ce qui revient à aller tout droit...





## 6.4 UN FEU DE CIRCULATION

### 🎯 Objectif

Ajouter un feu de circulation qui passe du vert au rouge toutes les 5 secondes.

### ▾ Dessiner ou importer un sprite *feu de circulation*

Tu peux trouver des « traffic lights » ici : <https://www.svgrepo.com/vectors/traffic-lights/>. Il te restera à créer deux costumes : un feu rouge et un feu vert.

### ▾ Ajoute ce code pour le feu de circulation

Il changera toutes les 5 secondes mais tu pourras forcer le basculement avec la touche espace.





## 6.5 LA VOITURE S'ARRÊTE AU FEU

### 🎯 Objectif

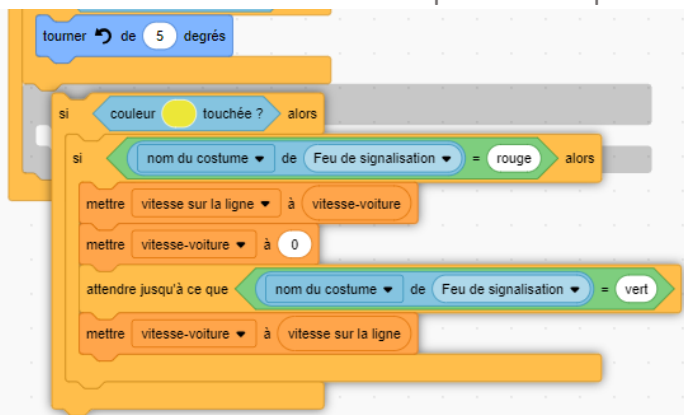
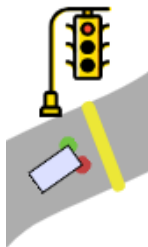
Faire en sorte que la voiture s'arrête au feu rouge et redémarre quand le feu passe au vert.

#### ▾ Ajoute une ligne jaune

Sur l'arrière-plan, ajoute une ligne de couleur à proximité du feu.

#### ▾ Ajoute ce code au script à la voiture

Crée une nouvelle variable **vitesse sur la ligne** dont le rôle sera de mémoriser temporairement la valeur de la vitesse-voiture avant qu'elle s'arrête et de restituer celle-ci quand le feu passe au vert.



Si tu es débutant, tu peux t'arrêter ici ! Les cartes suivantes nécessitent un minimum de pratique des **blocs personnalisés**.



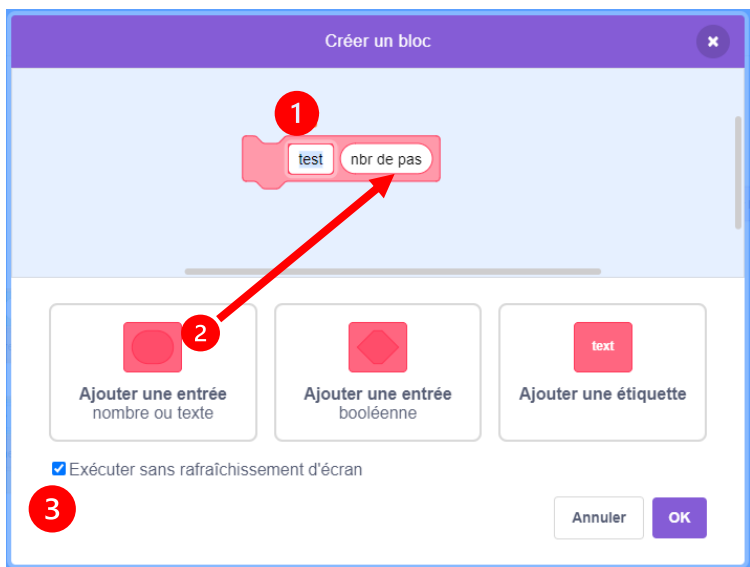


## 6.6 CRASH TEST : RÉUSSI !

🎯 Objectif : éviter les sorties de route

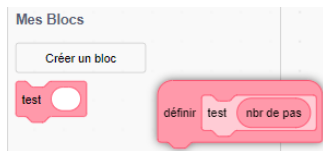
### ▼ Crée un bloc personnalisé

Dans la catégorie **Mes Blocs**, crée un bloc personnalisé que tu appelles **1 test**.



**2** Clique ensuite sur **Ajouter une entrée** et nomme-la **nombre de pas**.

**3** Coche enfin la case **Exécuter sans rafraîchissement d'écran**. Ces deux nouveaux blocs apparaissent.

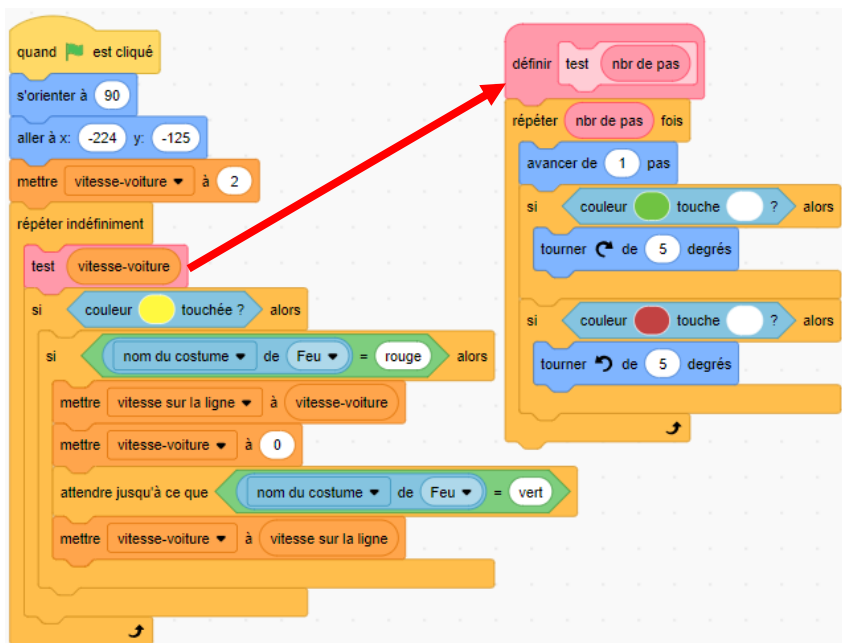




# CRASH TEST : RÉUSSI !

## Modifie le script de la voiture

Modifie le script de la voiture pour obtenir celui-ci.



Si la vitesse-voiture vaut 6, le bloc **test vitesse voiture** va exécuter 6 fois le test en avançant chaque fois d'un pas. Mais les 6 tests du bloc personnalisé s'exécuteront sans rafraîchissement d'écran c'est-à-dire en arrière-plan et de façon instantanée.



## 6.7 UNE VRAIE VOITURE

### 🎯 Objectif : expérimenter une hitbox

Remplacer la voiture avec ces deux capteurs un peu moches par une jolie petite voiture...

#### 📌 Ajoute un costume à ta voiture

Tu peux aussi aller le chercher ici :

<https://www.svgrepo.com/vectors/car-top/>

Nomme ton premier costume avec ces deux capteurs **hitbox** et le nouveau **voiture**. Une hitbox est un costume qui sera invisible mais utilisé pour effectuer des opérations en arrière-plan, sans rafraîchissement d'écran. **Attention**, il s'agit bien d'un nouveau costume pour le même sprite.



#### 📌 1 Modifie ce code et teste-le

```
quand cliqué
  basculer sur le costume voiture
  s'orienter à 90
  aller à x: -224 y: -125
  mettre vitesse-voiture à 2
  répéter indéfiniment
    test vitesse-voiture
    si couleur touchée ? alors
      si nom du costume de Feu = rouge alors
        mettre vitesse sur la ligne à vitesse-voiture
        mettre vitesse-voiture à 0
        attendre jusqu'à ce que nom du costume de Feu = vert
      sinon
        si couleur touche ? alors
          répéter nbr de pas fois
            avancer de 1 pas
            si couleur touche ? alors
              tourner de 5 degrés
            si couleur touche ? alors
              tourner de 5 degrés
          basculer sur le costume voiture
```



# À TOI DE JOUER !

Il y a 1000 façons d'enrichir cette application !

Par exemple :

- Régler la taille des costumes de la voiture pour éviter qu'elle donne l'impression de frotter sur les bords.
- Ajouter des obstacles, des piétons qui traversent la route et les éviter
- Ajouter plusieurs véhicules et les détecter de façon à adapter la vitesse de la voiture et éviter les accidents
- Améliorer les mouvements avec des accélérations/décélérations
- Transformer cette application en jeu du style « Tower defense »
- Etc...

