

## Cellule végétale ou animale?



Consultez une version jouable de ce code ici : <https://scratch.mit.edu/projects/459073461>

Ce jeu peut être facilement réécrit pour être un outil d'étude pour l'apprentissage des organites et identifier lesquels peuvent être trouvés dans les cellules végétales, les cellules animales, ou les deux.

Voici ce que nous voulons que notre jeu fasse :

- Trois options (cellule végétale, cellule animale, les deux) seront assignées aux touches fléchées du clavier (ou boutons tactiles sur iPad);
- Le jeu affichera aléatoirement un nom d'organite à la fois sur l'écran;
- Le joueur appuie sur la touche de flèche pour le type de cellule qui correspond à cet organite (végétale, animale, ou les deux);

- Si le joueur devine correctement, le score augmente d'un point;
- Si le joueur est incorrect, le score diminue d'un point.

### Créer des images-objets non jouables

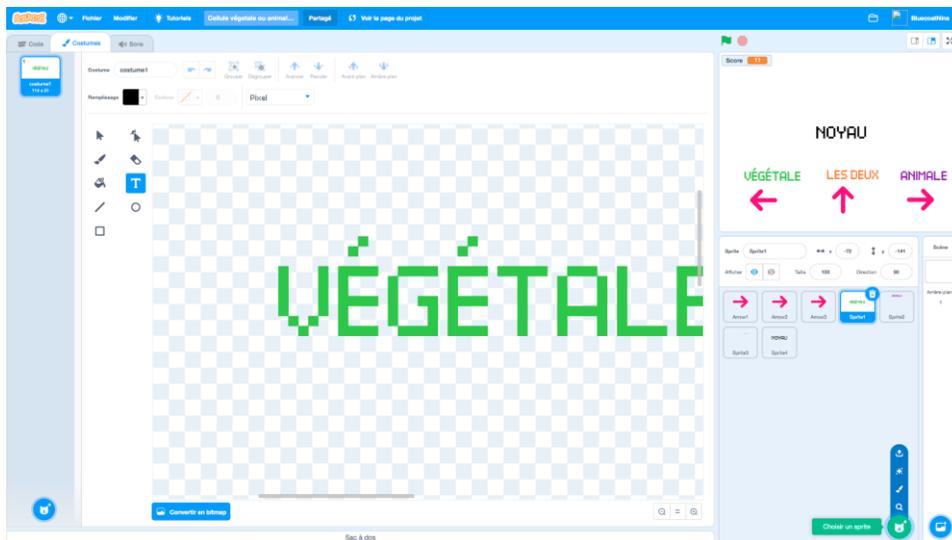
Ce programme utilise sept images-objets. Six de ces images-objets n'ont pas besoin de codage; ils sont affichés pour établir les règles du jeu.

Pour créer les images-objets de flèche :

- Appuyez sur le bouton Choisir un image-objet au bas du panneau images-objets;
- Créez trois images-objets de flèche. Pour modifier la direction dans laquelle la flèche pointe, vous pouvez faire pivoter les images-objets en sélectionnant l'image-objet que vous souhaitez faire pivoter et en modifiant la valeur (en degrés) dans le champ Direction;
- Placez les images-objets en les faisant glisser sur la toile du programme avec votre souris ou votre écran tactile.

Pour créer les images-objets d'étiquette :

- Appuyez sur le bouton Choisir un image-objet au bas du panneau images-objets et sélectionnez « peindre image-objet »;
- Utilisez l'outil Texte pour créer trois images-objets d'étiquette : « végétale », « les deux » et « animale ». Vous pouvez également modifier les couleurs du texte pour chaque étiquette.
- Faites glisser les images-objets en place sur la toile du programme avec votre souris ou votre écran tactile.

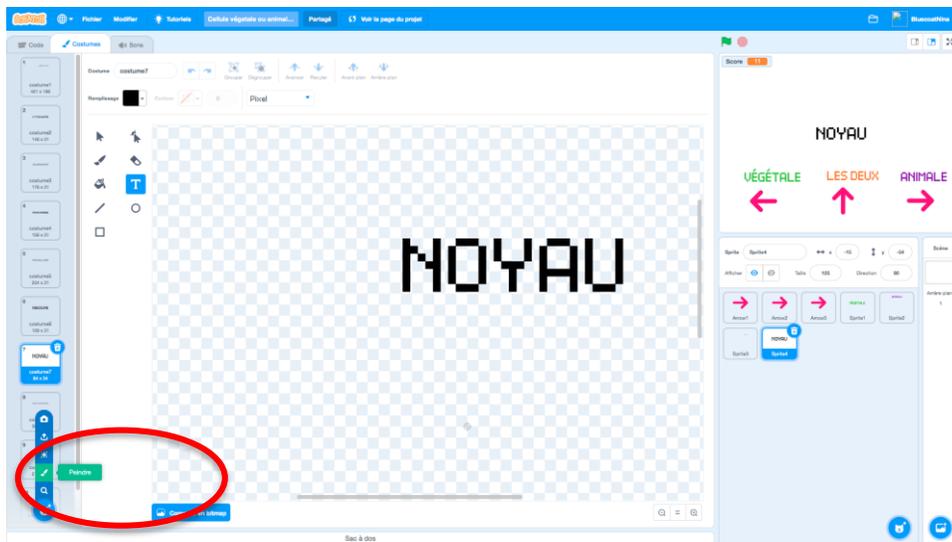


### Créer un image-objet « randomisateur »

- Appuyez sur le bouton Choisir un image-objet au bas du panneau images-objets et sélectionnez « peindre image-objet »;
- Utilisez l’outil Texte pour créer un image-objet textuel « Membrane de cellule ».

Cet image-objet utilisera une fonction Scratch appelée Costumes, qui permettra à l’apparence de l’image-objet de changer selon différentes conditions.

- Allez à l’onglet Costumes. Vous remarquerez que « Membrane cellulaire » est le costume par défaut pour cet image-objet. Nous devons ajouter des costumes pour chaque autre partie de cellule que nous voulons inclure dans notre jeu (j’ai inclus 10 au total dans cet exemple : membrane cellulaire, paroi cellulaire, ribosome, noyau, mitochondrie, chloroplaste, lysosome, appareil de Golgi, réticulum endoplasmique, cytoplasme);
- Cliquez sur le bouton Choisir un costume au bas du panneau Costumes pour ajouter les costumes restants.



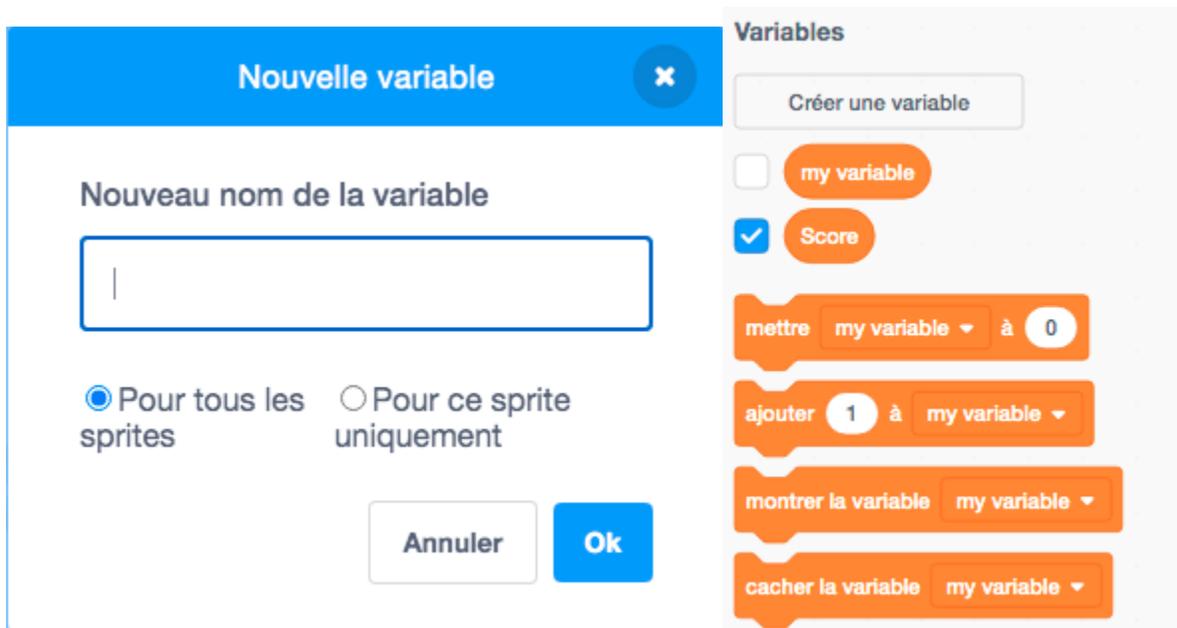
### Créer un compteur de score

Pour créer un compteur de score, nous devons créer une variable pour représenter la valeur de score.

Pour créer une variable :

- Allez à l’onglet Code;
- Accédez au menu Variables (orange foncé);
- Cliquez sur le bouton Créer une variable;
- Nommer la variable « Score » (sans guillemets);

- Dans ce cas, peu importe que vous choisissiez « pour tous les images-objets » ou « pour cet image-objet seulement », car nous ne programmons que pour un image-objet dans ce jeu.
- Une fois que votre variable est créée, assurez-vous que la case à côté de la variable Score est sélectionnée.



### Créer un randomisateur

Ce programme fixera le compteur de score à zéro au début du jeu et montrera un costume aléatoire pour notre randomisateur d'image-objet à l'écran pendant un certain temps avant de passer à un nouveau costume.

- Allez dans le menu Événements (jaune) et choisissez le bloc **Quand l'indicateur vert est cliqué** et faites-le glisser dans la zone de codage. C'est le bloc que nous utiliserons pour démarrer tous nos programmes pour ce jeu (de sorte qu'ils commencent tous au début du jeu);
- Allez dans le menu Variables (orange foncé) et choisissez le bloc **Définir ma variable à 0**. Connectez-le directement sous le bloc d'indicateur vert. À l'aide de la liste déroulante de ce bloc, changez « ma variable » en « Score ».

Maintenant, nous devons créer le randomisateur qui boucle à l'infini (ce jeu n'a pas de point de terminaison défini).

- Allez dans le menu Contrôle (orange pâle). Choisissez le bloc de boucle **infinie** et connectez-le directement en dessous du bloc **Mettre le score à 0**. Tout code que nous mettons à l'intérieur de ce bloc sera en boucle à l'infini.

- Allez dans le menu Apparence (violet). Choisissez le bloc **changer le costume pour** et le placer à l'intérieur de la boucle.
- Allez dans le menu Opérateurs (vert). Choisissez le bloc **choisir aléatoirement de 1 à 10** et faites-le glisser sur la tache noire sur le bloc de **changer le costume pour**. Les blocs doivent se combiner pour créer un bloc violet et vert qui lit **changer le costume pour choisir au hasard de 1 à 10**.
- Comme nous avons dix costumes, nous n'avons pas à changer les valeurs dans le bloc **aléatoire 1 à 10**.
- Allez dans le menu Contrôle (orange pâle). Choisissez le bloc **Attendre 1 seconde**. Placez ce bloc directement sous le bloc changer le costume pour **choisir aléatoirement de 1 à 10**, mais toujours à l'intérieur de la boucle infinie. Ce bloc contrôle la durée pendant laquelle une lettre restera à l'écran (en secondes) avant de passer à une nouvelle lettre aléatoire. Une seconde peut être trop rapide. Cela peut être ajusté au goût personnel pour être plus rapide ou plus lent.



### Programmer le jeu

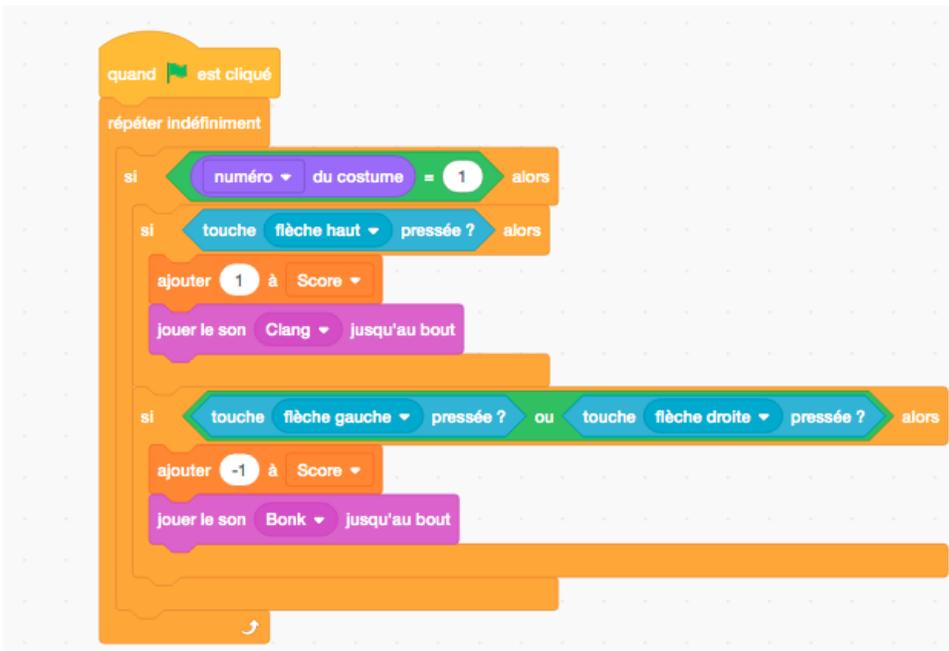
Maintenant, il ne reste plus qu'à créer des programmes qui reconnaîtront quand un joueur appuie sur la touche appropriée lorsqu'une partie de cellule spécifique est affichée. Nous devons créer un programme pour chaque partie de cellule que nous avons inclus dans le jeu (donc, 10 au total).

- Tout d'abord, nous allons placer un **quand on clique sur l'indicateur vert** pour lancer notre programme.

- Dans le menu Contrôle (orange clair), choisissez un bloc de boucle **infinie** et connectez-le directement au bloc de démarrage. Nous allons bâtir les prochaines étapes à l'intérieur de ce bloc de boucle infinie.
- Pour créer le premier énoncé conditionnel, qui exécute le programme si le costume A est affiché :
  - o Dans le menu Contrôle (orange pâle), choisissez le bloc **si [vide] alors** et imbriquez-le dans la boucle infinie.
  - o Dans le menu Opérateurs (vert), choisissez le bloc **[vide] = 50** et faites-le glisser sur le dessus de l'hexagone vide sur le bloc **si [vide] alors**.
  - o Dans le menu Apparences (violet), choisissez le bloc de **numéro de costume** et faites-le glisser sur l'ovale vide sur le bloc vert **[vide] = 50**. L'énoncé si devrait maintenant lire **si numéro de costume = 50 alors**.
  - o Les numéros de costume ont été automatiquement attribués dans l'onglet Costumes sur Scratch lors de leur création. Le numéro du costume de la membrane cellulaire (s'il s'agissait du premier costume que vous avez créé) est 1. Éditez **50** sur le bloc vert pour qu'il soit maintenant **1**. (Si votre numéro de costume pour A n'est pas 1, assurez-vous que le numéro de costume de votre programme correspond au numéro de costume pour A dans votre onglet Costumes).
- Pour créer la condition imbriquée pour lorsque le joueur appuie sur la touche appropriée :
  - o Dans le menu Contrôle (orange pâle), choisissez un autre bloc **si [vide] alors** et imbriquez-le dans le bloc conditionnel existant.
  - o Dans le menu Senseur (bleu clair), choisissez le bloc **touche espace enfoncée?** et faites-le glisser vers le dessus de l'hexagone vierge du bloc **si [vide] alors**. La bonne réponse pour quand la membrane cellulaire est à l'écran est d'appuyer sur la flèche pointant vers le haut (pour « les deux »), alors utilisez la liste déroulante de ce bloc pour sélectionner la **flèche pointant vers le haut**.
- Maintenant, nous devons coder ce qui se passe lorsque la touche correcte est enfoncée.
  - o Dans le menu Variables (orange foncé), choisissez le bloc **changer ma variable par 1** et placez-le dans la condition imbriquée. Utilisez la liste déroulante du bloc pour sélectionner **Score**. Cela garantira que le compteur score augmente d'un point.
  - o Facultatif : vous pouvez choisir d'ajouter un effet sonore comme retour pour obtenir la bonne réponse, au goût. Pour ce faire, allez dans le menu Son (rose) et choisissez **jouer le son Clang jusqu'à la fin** et placez-le directement sous le bloc **changer le score par 1** à l'intérieur du même énoncé conditionnel. Utilisez la liste déroulante pour choisir le son que vous préférez.

- Pour créer la condition imbriquée pour lorsque le joueur appuie sur les touches incorrectes :
  - Dans le menu Contrôle (orange pâle), choisissez un autre bloc **si [vide] alors** et imbriquez-le dans le premier bloc conditionnel (**si le numéro de costume = 1 alors**).
  - Dans l'onglet Opérateurs (vert), choisissez le bloc **[vide] ou [vide]** et faites-le glisser sur le dessus de l'hexagone vide sur le bloc **si [vide] alors**.
  - Ce bloc **[vide] ou [vide]** nous permet de créer le même programme de pointage pour si le joueur appuie sur l'une des deux clés incorrectes.
  - Dans le menu Détection (bleu clair), faites glisser un bloc **touche espace enfoncée?** dans chacun des espaces du bloc vert **[vide] ou [vide] ou [vide]**. Utilisez la liste déroulante pour modifier chacune de ces touches afin qu'elles représentent chacune des touches de réponse incorrectes (dans ce cas, flèche vers la droite et flèche vers la gauche).
- Maintenant, nous devons coder ce qui se passe quand l'une de ces touches incorrectes est enfoncée.
  - Dans le menu Variables (orange foncé), choisissez le bloc **changer ma variable par 1** et placez-le dans la condition imbriquée. Remplacez la valeur par -1. Utilisez la liste déroulante du bloc pour sélectionner **Score**. Cela garantira que le compteur score diminue d'un point.
  - Facultatif : vous pouvez choisir d'ajouter un effet sonore comme retour pour obtenir la bonne réponse, au goût. Pour ce faire, allez dans le menu Son (rose) et choisissez **jouer le son Bonk jusqu'à la fin** et placez-le directement sous le bloc **changer le score par -1** à l'intérieur du même énoncé conditionnel. Utilisez la liste déroulante pour choisir le son que vous préférez.

Le programme terminé devrait ressembler à ceci :



Créer un programme pour chaque costume

Maintenant, nous devons créer le même programme pour les costumes de parties cellulaires restantes. Les programmes sont tous identiques, à l'exception de la valeur du nombre des costumes, et quelles touches sont attribuées à des réponses correctes ou incorrectes.

Conseil : Puisque nous réutilisons de grandes parties de notre programme pour coder les costumes restants, vous pouvez copier le code existant (cliquez avec le bouton droit > Copier) plutôt que de reconstruire le code autant de fois que nécessaire. Ensuite, il vous suffit d'ajuster les paramètres de code dans chaque nouvelle copie du code.

Encouragez les élèves à jouer leur jeu et à apporter des changements comme ajuster le défi du jeu en changeant la valeur dans le bloc d'attente ou les valeurs de score pour différentes conditions ou en ajustant l'esthétique du jeu en changeant les sons joués ou l'apparence du jeu.