

Nom du plan de leçon	Niveau et volet
Guide de programmation	

Jouez un exemple du programme complet à l'adresse suivante :  
<https://scratch.mit.edu/projects/492469291/>

Les objectifs de cette activité sont les suivants :

1. Créer des sprites cliquables qui déclenchent un résultat lorsque l'on clique dessus.
2. Développer un compteur qui accumule un point chaque fois que l'on clique sur un sprite.
3. Développer un système de multiplicateur du pointage qui augmente le nombre de points par clic lorsque certaines valeurs de pointage sont atteintes.
4. Ajouter des objets « achetables » qui augmentent ou réduisent de manière autonome le pointage au compteur (sans clic).
5. Réduire le pointage au compteur en fonction de la valeur du « coût » des achats.

Voici ce que le jeu final fait :

- La quantité de carbone commence à 10 000.
- Le carbone augmente de 1 chaque seconde.
- Cliquer sur l'arbre réduit le niveau de carbone de 1.
- Cliquer sur l'arbre augmente la quantité d'or de 1.
- Acheter une usine ajoute une augmentation automatique de l'or. Chaque achat augmente le taux de production d'or de 1 point par seconde.
- Acheter une usine augmente également le taux de production de carbone de 1 par seconde.
- Acheter une action écologique réduit le niveau de carbone de 100 et réduit le taux de production de carbone de 1 par seconde.

Le but du jeu est de réussir à réduire la quantité de carbone à zéro, mais, comme les étudiants le verront, ce n'est pas une tâche facile.

### Créer les variables

Les variables stockent des renseignements qui peuvent être utilisés dans le programme. Nous aurons besoin de quatre variables pour ce programme :

1. **Compteur de carbone** – représente la quantité totale de carbone accumulé dans l'atmosphère au cours du jeu. Le compteur sera fixé à 10 000 unités et augmentera de lui-même selon une variable fixe de carbone par seconde (voir ci-dessous). Cela imite les industries et les processus qui génèrent continuellement des émissions. Le compteur est réduit par les clics sur les arbres (planter des arbres).
2. **Carbone par seconde** – stocke les renseignements concernant les mises à niveau (les usines et les actions écologiques) que le joueur a achetées, lesquelles augmentent ou réduisent automatiquement la quantité de carbone avec le temps (sans clic).
3. **Or** – ce compteur représente la quantité totale d'or généré par les clics sur les arbres et produit de façon autonome par les usines. Acheter des mises à niveau (usines et actions écologiques) coûtera de l'or et réduira le compteur d'or.

4. **Or par seconde** – stocke les renseignements concernant le taux de production d’or autonome par seconde (selon le nombre d’usines achetées). Par exemple, pour chaque usine achetée, le taux d’or par seconde augmentera de 1 point d’or par seconde.

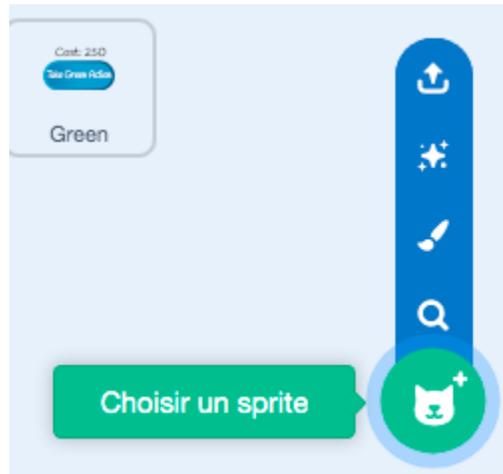
Dans le menu **Variables** orange foncé, cliquez sur le bouton **Créer une variable** pour créer une variable. Tapez le nom de la variable dans le champ **Nouveau nom de la variable** et assurez-vous que l’option **Pour tous les sprites** est sélectionnée. Cela nous permettra de nous assurer que les renseignements stockés dans la variable s’appliquent à chaque élément de notre jeu.

Chaque fois qu’une nouvelle variable est créée, un compteur correspondant apparaît sur la scène. Nous avons seulement besoin que les variables **Compteur de carbone** et **Compteur d’or** soient visibles; les deux autres variables seront en arrière-scène. Pour rendre ces deux variables invisibles, décochez les cases à côté de ces variables dans le menu **Variables**.

### Créer un sprite sur lequel cliquer

Pour l’élément interactif du jeu de clics, nous avons besoin d’un sprite cliquable. Le sprite par défaut dans Scratch est un chat; supprimez le sprite de chat en le sélectionnant dans le panneau de sprites et en cliquant sur l’icône de poubelle (supprimer) qui s’affiche.

Pour ajouter un nouveau sprite, ouvrez la bibliothèque de sprites en cliquant sur le bouton **Choisir un sprite** dans le panneau de sprites.



Au moyen de la barre de recherche de la bibliothèque Choisir un sprite, recherchez « tree » (arbre) et sélectionnez le sprite **Trees1**.

Vous pouvez glisser le sprite d'arbre dans sa position voulue sur la scène en le glissant avec votre souris.

### Faire le programme de clics de base

Le programme de clics de base a une seule fonction de base : ajouter un point au pointage du compteur de carbone chaque fois que vous cliquez sur le sprite.

Pour rendre le jeu de clics plus intéressant sur le plan visuel, nous allons également ajouter du code qui verrouille le sprite d'arbre dans sa position alors que nous jouons (afin qu'il ne se déplace pas lorsque nous cliquons) et afin que le sprite nous donne un effet visuel pour confirmer nos clics.

Dans le menu **Événements** jaune, choisissez le bloc **Quand le drapeau vert est cliqué** et glissez-le sur le canevas de programmation. Cela déclenchera le lancement du programme lorsque l'on clique sur le sprite d'arbre.

Dans le menu **Variables** orange foncé, choisissez le bloc **mettre [ma variable] à 0** et placez-le directement en dessous du bloc **Quand le drapeau vert est cliqué**. Au moyen de la liste déroulante sur ce bloc, sélectionnez la variable **Compteur de carbone**. Changez la valeur de ce bloc à 10 000. Cela réinitialisera le compteur de carbone à 10 000 chaque fois que nous lançons le programme.

Dans menu **Mouvement** bleu, choisissez le bloc **aller à x: 0 y: 0** et connectez-le en dessous du bloc **mettre Compteur de carbone à 0**. Fixez les valeurs x et y sur ce bloc à la position où vous voulez placer votre sprite d'arbre, où x est la position horizontale en pixels et y la position verticale en pixels (x : 0, y : 0 seraient le centre de la scène). Cela empêchera le sprite de se déplacer à l'écran alors que le joueur clique pendant le jeu. Dans l'exemple de code, l'arbre est fixé à x : 0 et y : -85.

Maintenant, nous programmerons ce qui arrive lorsque l'on clique sur le sprite d'arbre.

Dans le menu **Contrôle** orange, sélectionnez le bloc de boucle **répéter indéfiniment** et connectez-le en dessous du bloc **aller à x: 0 y: -85**. Le reste du programme sera bâti à l'intérieur de la boucle infinie et sera répété indéfiniment alors que le programme est exécuté.

Dans le menu **Contrôle** orange, sélectionnez le bloc conditionnel **si [espace vide] alors, sinon** et imbriquez-le à l'intérieur de la boucle **répéter indéfiniment**.



Pour décrire la condition de l'instruction conditionnelle, nous devons utiliser les blocs d'opérateur et les blocs de capteur pour définir le moment auquel on clique sur le sprite d'arbre et on relâche le bouton de souris.

Dans le menu **Opérateurs** vert, sélectionnez le bloc **[espace vide] et [espace vide]** et glissez-le sur le canevas de programmation.

Dans le menu **Capteurs** bleu pâle, sélectionnez le bloc **touche le pointeur de souris?** et glissez-le dans le premier espace du bloc d'opérateur vert. Toujours dans le menu **Capteurs**, sélectionnez le bloc **souris pressée?** et glissez-le dans le deuxième espace vide du bloc d'opérateur vert :



Cela définit le moment auquel un joueur a placé le curseur de la souris au-dessus du sprite et a cliqué sur le bouton gauche de la souris. Placez ce bloc combiné dans l'espace hexagonal foncé du bloc conditionnel **si [espace vide] alors, sinon**. L'espace foncé s'élargira pour recevoir le bloc.

Maintenant, nous indiquerons le code du résultat lorsque l'on clique sur le sprite d'arbre et placerons ce code dans la première « bouche » du bloc conditionnel **si [espace vide] alors, sinon**.

Dans le menu **Variables** orange foncé, sélectionnez le bloc **ajouter 1 à [ma variable]** et glissez-le dans la première « bouche » du bloc conditionnel **si [espace vide] alors, sinon**. Au moyen de liste

déroulante de ce bloc, assurez-vous que la variable **Or** est sélectionnée. Cela signifie que chaque fois que l'on clique sur l'arbre de sprite, le pointage du **Compteur d'or** augmentera d'un point.

Dans le menu **Variables** orange foncé, prenez un deuxième bloc **ajouter 1 à [ma variable]** et placez-le en dessous du bloc **ajouter 1 à Or**. Au moyen de la liste déroulante de ce bloc, assurez-vous que la variable **Compteur de carbone** est sélectionnée. Changez la valeur de 1 à -1. Cela signifie que chaque fois que l'on clique sur le sprite d'arbre, le compteur de carbone diminue d'un point (comme si planter un arbre absorbe une unité de carbone dans l'atmosphère).

Dans le menu **Apparence** mauve, sélectionnez le bloc **mettre la taille à 100 % de la taille initiale** et placez-le en dessous du bloc **ajouter -1 à Compteur de carbone**. Cela donnera une indication visuelle pour montrer que l'on a bien cliqué sur le sprite.

Si vous testez le programme à cette étape, vous serez en mesure de maintenir le bouton de la souris enfoncé lorsque vous cliquez sur le sprite d'arbre et le pointage continuera d'augmenter tant que le bouton ne sera pas relâché. Puisque nous voulons qu'un seul point soit ajouté par clic de souris (peu importe le temps pendant lequel le bouton de la souris est maintenu), nous devons ajouter un élément de contrôle.

Dans le menu **Contrôle** orange, sélectionnez le bloc **attendre jusqu'à ce que** et placez-le en dessous du bloc **mettre la taille à 100 % de la taille initiale**.

Nous voulons que le programme attende jusqu'à ce que le bouton de la souris soit relâché pour continuer. Pour indiquer cela, allez dans le menu **Opérateurs** vert et sélectionnez le bloc **[espace vide] = [espace vide]**. Dans le menu **Capteurs** bleu, sélectionnez le bloc **souris pressée?** et placez-le sur le premier espace vide du bloc d'opérateur vert. Dans le deuxième espace, tapez le mot « false » (faux).



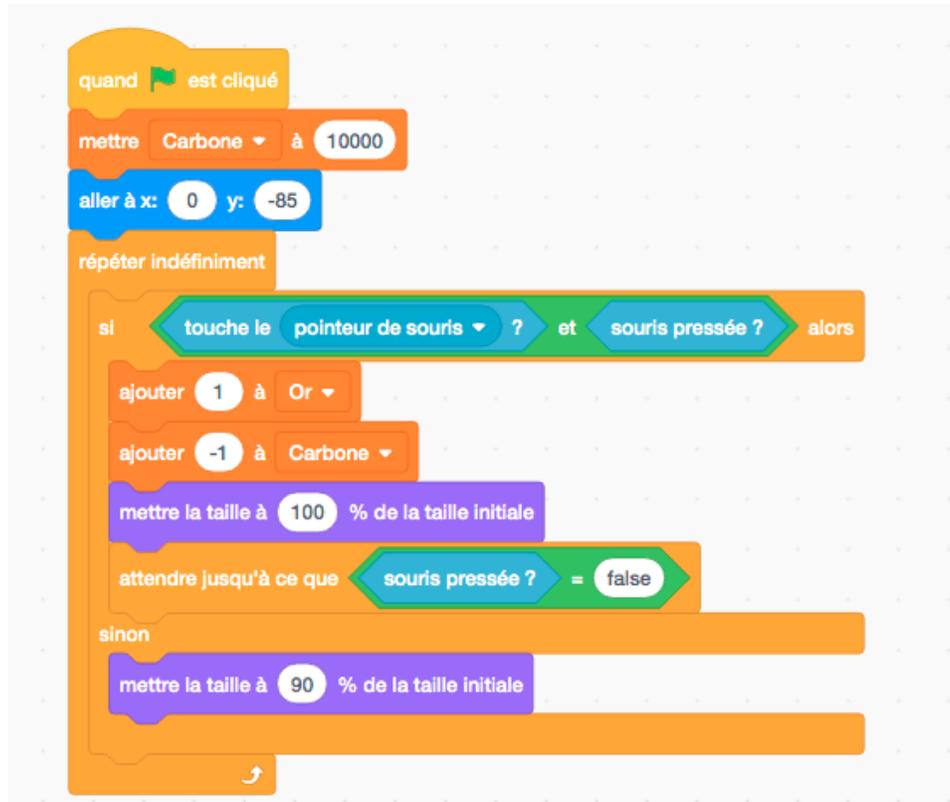
Placez ce bloc combiné sur l'hexagone foncé du bloc **attendre jusqu'à ce que**. Défini de cette façon, le programme attendra jusqu'à ce que le bouton de la souris soit relâché (bouton de la souris enfoncé = faux, alors qu'un clic de souris indique que le bouton de la souris enfoncé = vrai) pour continuer à l'étape suivante du programme.

Ensuite, nous devons programmer le résultat pour lorsque l'on clique sur le sprite d'arbre. Puisque le pointage ne change pas lorsque l'on ne clique *pas* sur le sprite, le seul résultat est de changer la taille du sprite (l'autre moitié du code de changement d'apparence décrit dans la première « bouche » du bloc d'instruction conditionnelle).

Dans le menu **Apparence** mauve, sélectionnez le bloc **mettre la taille à 100 % de la taille initiale** et placez-le dans la deuxième « bouche » (l'espace **sinon**) du bloc d'instruction conditionnelle. Changez la valeur de ce bloc à 90 %.

Si vous testez votre programme maintenant, le compteur **Carbone** devrait commencer à 10 000 et diminuer d'un point chaque fois que l'on clique sur le sprite d'arbre. Le compteur **Or** devrait commencer à 0 augmenter de 1 point chaque fois que l'on clique sur le sprite d'arbre. Lorsque l'on clique sur le sprite d'arbre et qu'on relâche le bouton, le sprite devrait brièvement changer de taille.

Le programme de clics complet devrait ressembler à ceci :



### Ajouter les « Mises à niveau » – Usines et actions écologiques

#### Pour ajouter des « usines » qui augmentent les émissions de **Carbone** tout en augmentant la production d'**Or**

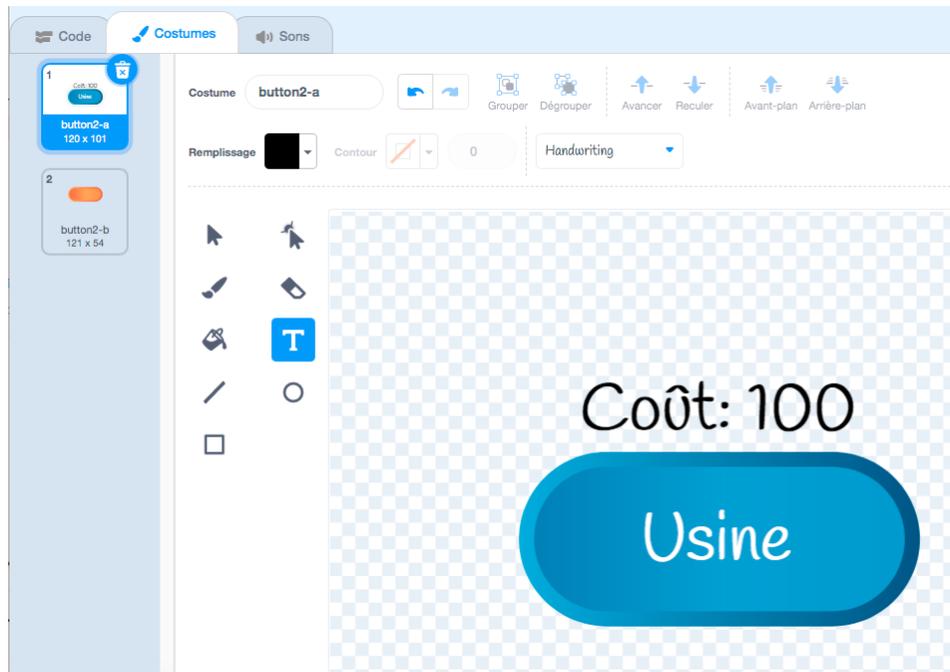
- Ajouter un sprite de bouton

Pour ajouter un nouveau sprite, ouvrez la bibliothèque de sprites en cliquant sur le bouton **Choisir un sprite** dans le panneau des sprites.

Au moyen de la barre de recherche dans la bibliothèque Choisir un sprite, recherchez « button » (bouton) et sélectionnez votre sprite de bouton préféré.

Pour étiqueter votre bouton, sélectionnez l'onglet **Costumes** dans le coin supérieur gauche de l'écran.

À l'onglet Costumes, ajoutez du texte à votre bouton au moyen de l'outil Texte. Modifiez la couleur de votre texte au moyen de l'outil de remplissage et glissez le texte en place. Dans l'exemple ci-dessous, nous avons utilisé deux boîtes de texte : une pour étiqueter le bouton et une étiquette de coût placée au-dessus du bouton. L'étiquette de coût communique le nombre de points dont aura besoin le joueur pour activer le bouton.



Remarquez qu'il y a deux costumes pour ce sprite : un costume bleu (button2-a) et un costume orange (button2-b). Nous utiliserons le deuxième costume plus loin dans notre code comme indication visuelle pour signaler que l'on clique sur le bouton.

Vous pouvez glisser le sprite de bouton dans sa position voulue sur la scène en le glissant avec votre souris.

- Programmer le bouton

\*\* Assurez-vous que le sprite de bouton est sélectionné.

Retournez à l'onglet Code.

Dans le menu **Événements** jaune, sélectionnez le bloc **quand ce sprite est cliqué** pour commencer un nouveau programme.

Dans le menu **Mouvement** bleu, sélectionnez le bloc **aller à x: 0 y: 0** et connectez-le en dessous du bloc **quand ce sprite est cliqué**. Ajustez les valeurs x et y de ce bloc à la position où vous voulez

« coller » votre sprite de bouton. Cela empêchera le bloc d'être déplacé lorsque les joueurs cliquent dessus.

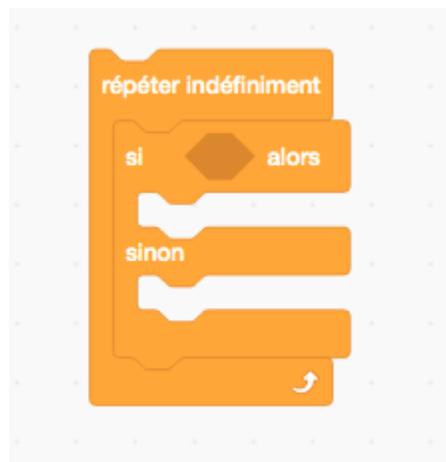
Maintenant, nous programmerons le bouton afin qu'il change de costume (de bleu à orange) lorsque l'on clique dessus.

Dans le menu **Contrôle** orange, sélectionnez la boucle **répéter indéfiniment** et connectez-la en dessous du bloc **aller à x: 0 y: 0**. Le reste de code sera placé à l'intérieur de la boucle **répéter indéfiniment**.

Nous devons ajouter une instruction conditionnelle pour produire différents résultats lorsque l'on clique sur le bouton et lorsque l'on ne clique pas dessus.

Dans le menu **Contrôle** orange, sélectionnez le bloc **répéter indéfiniment** et connectez-le en dessous du bloc **aller à x: 0 y: 0**. Le reste du programme sera inséré à l'intérieur de la boucle infinie et sera répété indéfiniment alors que le programme s'exécute.

Dans le menu **Contrôle**, sélectionnez le bloc conditionnel **si [espace vide] alors, sinon** et imbriquez-le à l'intérieur de la boucle **répéter indéfiniment**.



Pour décrire la condition de l'instruction conditionnelle, nous devons utiliser les blocs d'opérateur et les blocs de capteur pour définir le moment auquel on clique sur le sprite d'arbre et on relâche le bouton de souris.

Dans le menu **Opérateurs** vert, sélectionnez le bloc **[espace vide] et [espace vide]** et glissez-le sur le canevas de programmation.

Dans le menu **Capteurs** bleu pâle, sélectionnez le bloc **touche le pointeur de souris?** et glissez-le dans le premier espace du bloc d'opérateur vert. Toujours dans le menu **Capteurs**, sélectionnez le bloc **souris pressée?** et glissez-le dans le deuxième espace vide du bloc d'opérateur vert :



Cela définit le moment auquel un joueur a placé le curseur de la souris au-dessus du sprite et a cliqué sur le bouton gauche de la souris. Placez ce bloc combiné dans l'espace hexagonal foncé du bloc conditionnel **si [espace vide] alors, sinon**. L'espace foncé s'élargira pour recevoir le bloc.

Maintenant, nous indiquerons le code du résultat lorsque l'on clique sur le sprite d'arbre et placerons ce code dans la première « bouche » du bloc conditionnel **si [espace vide] alors, sinon**.

Dans le menu **Apparence** mauve, sélectionnez le bloc **basculer sur le costume** et placez-le à l'intérieur de la première bouche du bloc conditionnel. Utilisez la liste déroulante pour sélectionner le costume **button2-b**.

Dans le menu **Contrôle** orange, sélectionnez le bloc **attendre jusqu'à ce que** et placez-le en dessous du bloc **basculer sur le costume**.

Nous voulons que le programme attende jusqu'à ce que le bouton de la souris soit relâché pour continuer. Pour indiquer cela, allez dans le menu **Opérateurs** vert et sélectionnez le bloc **[espace vide] = [espace vide]**. Dans le menu **Capteurs** bleu, sélectionnez le bloc **souris pressée?** et placez-le sur le premier espace vide du bloc d'opérateur vert. Dans le deuxième espace, tapez le mot « false » (faux).

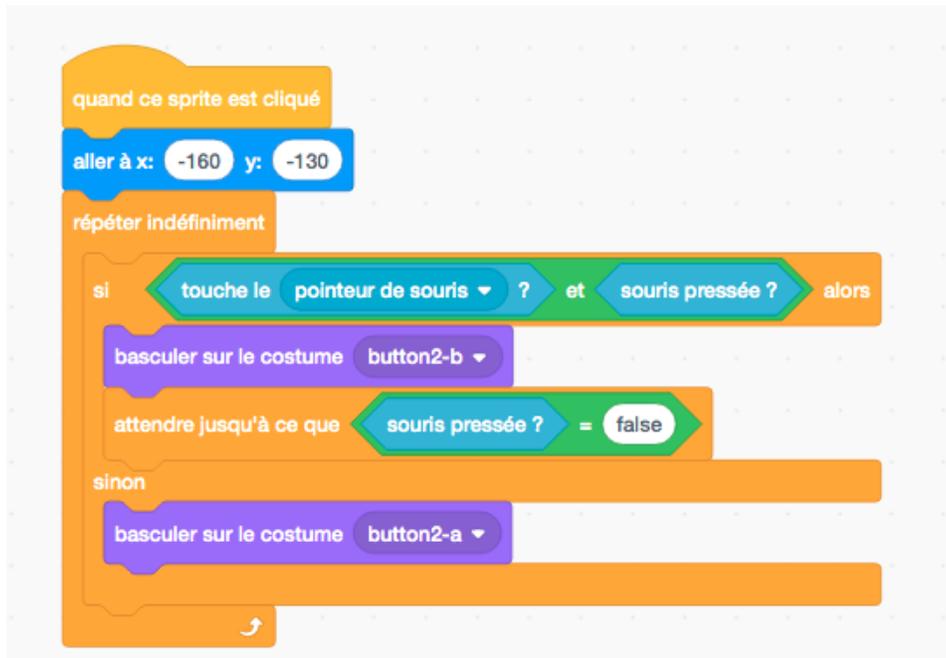


Placez ce bloc combiné sur l'hexagone foncé du bloc **attendre jusqu'à ce que**. Défini de cette façon, le programme attendra jusqu'à ce que le bouton de la souris soit relâché (bouton de la souris enfoncé = faux, alors qu'un clic de souris indique que le bouton de la souris enfoncé = vrai) pour continuer à l'étape suivante du programme.

Ensuite, nous devons programmer le résultat pour lorsque l'on clique sur le sprite d'arbre. Puisque le pointage ne change pas lorsque l'on ne clique *pas* sur le sprite, le seul résultat est de changer la taille du sprite (l'autre moitié du code de changement d'apparence décrit dans la première « bouche » du bloc d'instruction conditionnelle).

Dans le menu **Apparence** mauve, sélectionnez le bloc **basculer sur le costume** et placez-le à l'intérieur de la « bouche » sinon de l'instruction conditionnelle. Utilisez la liste déroulante pour sélectionner le costume **button2-a**.

Le programme complet devrait ressembler à ceci :



- Programmer les effets du bouton

Nous voulons que ce programme fasse trois choses :

1. Nous voulons qu'il devienne actif (utilisable par le joueur) SEULEMENT lorsque le joueur a suffisamment de points d'**Or** pour couvrir le coût d'achat d'une usine.
2. Lorsqu'une usine est achetée, nous voulons que le pointage de **Carbone** augmente de façon autonome de 1 carbone par seconde (produit automatiquement par l'usine plutôt que par les clics du joueur).
3. Lorsqu'une usine est achetée, nous voulons que le pointage d'**Or** augmente de façon autonome d'un point par seconde (produit automatiquement par l'usine plutôt que par les clics du joueur).
4. Nous voulons que le compteur du pointage d'**Or** du joueur diminue de 100 points (le coût d'achat d'une usine).

Dans le menu **Événements** jaune, sélectionnez le bloc **quand ce sprite est cliqué** pour commencer un nouveau programme.

À l'onglet **Contrôle** orange, choisissez le bloc conditionnel **si [espace vide] alors** (avec une seule « bouche » de résultat). Connectez ce bloc en dessous du bloc **quand ce sprite est cliqué**.

Pour décrire notre condition (si le pointage du compteur de carbone est supérieur à 100 points), nous devons utiliser un bloc d'opérateur vert.

Dans le menu **Opérateurs** vert, choisissez le bloc **[espace vide] > [espace vide]** et glissez-le dans l'hexagone foncé du bloc **si [espace vide] alors**.

Dans le menu **Variables** orange foncé, sélectionnez le bloc ovale **Or** et glissez-le dans le premier espace vide du bloc d'opérateur. Dans le deuxième espace vide, changez la valeur à 100.

Nous pouvons maintenant programmer le résultat de la « bouche » du bloc conditionnel.

Dans le menu **Variables** orange foncé, choisissez **ajouter 1 à [ma variable]** et placez-le à l'intérieur de la bouche du bloc conditionnel. Avec la liste déroulante, assurez-vous que la variable est fixée à **Carbone par seconde**.

Dans le menu **Variables** orangé foncé, choisissez le bloc **ajouter 1 à [ma variable]** et placez à l'intérieur de la bouche du bloc conditionnel. Au moyen de la liste déroulante, assurez-vous que la variable est fixée à **Or par seconde**.

Dans le menu **Variables** orangé foncé, choisissez le bloc **ajouter 1 à [ma variable]** et placez-le à l'intérieur de la bouche du bloc conditionnel. Utilisez la liste déroulante pour vous assurer que la variable est fixée à **Or**. Changez la valeur de ce bloc à -100.

Le programme complet devrait ressembler à ceci :



Maintenant, nous devons créer un programme qui met constamment à jour la variable **Carbone par seconde** et ajuste automatiquement le pointage du compteur **Carbone** au fur et à mesure que le temps passe.

Dans le menu **Événements** jaune, choisissez le bloc **quand le drapeau vert est cliqué** pour commencer un nouveau programme.

Dans le menu **Variables** orange foncé, choisissez le bloc **mettre [ma variable] à 0** et connectez-le en dessous du bloc **quand le drapeau vert est cliqué**. Utilisez la liste déroulante pour sélectionner la variable **Carbone par seconde**. Changez la valeur de ce bloc à 1.

Maintenant, nous devons programmer une boucle qui fera croître le pointage du compteur de carbone par le taux indiqué par la variable **Carbone par seconde**. Cette augmentation devrait se produire une seule fois par seconde.

Dans le menu **Contrôle** orange pâle, sélectionnez le bloc **répéter indéfiniment** et connectez-le en dessous du bloc **mettre Carbone par seconde à 1**. Le reste du programme se trouvera à l'intérieur de la boucle.

Dans le menu **Contrôle** orange pâle, sélectionnez le bloc **attendre 1 secondes** et placez-le à l'intérieur du bloc **répéter indéfiniment**.

Dans le menu **Variables** orangé foncé, sélectionnez le bloc **ajouter 1 à [ma variable]**. Utilisez la liste déroulante pour sélectionner la variable du compteur **Carbone**. Prenez le bloc de variable ovale **Carbone par seconde** et glissez-le dans le champ sur le bloc (remplaçant le 1).

Le programme complet devrait ressembler à ceci :



Nous devons maintenant créer un programme semblable qui met constamment à jour la variable **Or par seconde** et ajuste automatiquement le pointage du compteur d'**Or** au fur et à mesure que le temps passe.

Nous pouvons actuellement reproduire le code existant qui met à jour le compteur de **Carbone** comme base pour ce programme. Cliquez avec le bouton droit sur le bloc supérieur de ce programme et sélectionnez **Dupliquer**.

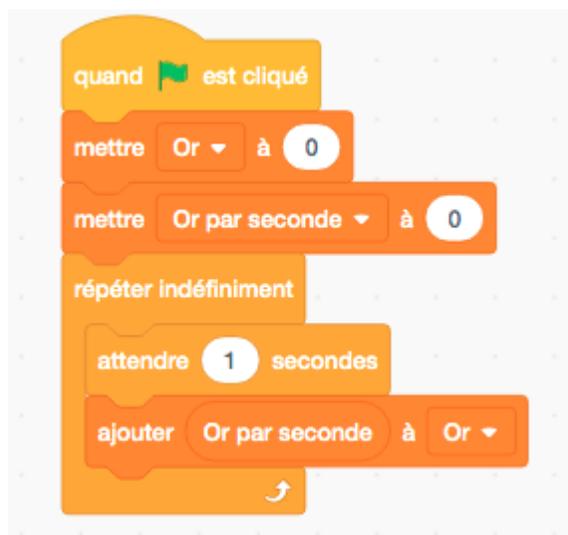


À l'aide de la liste déroulante sur le bloc, changez la variable du bloc **mettre la variable Carbone par seconde à 1** pour **Or**. Changez la valeur de ce bloc de 1 à 0.

Dans le menu **Variables** orange foncé, choisissez le bloc **mettre [ma variable] à 0** et connectez-le en dessous du bloc **mettre Or à 0**. Utilisez la liste déroulante de ce bloc pour sélectionner la variable **Or par seconde**. Ces blocs combinés signifient que lorsque nous lançons le jeu sans aucune mise à niveau, notre compteur d'**Or** est fixé à 0 point et notre taux d'**Or par seconde** est de 0, donc les seuls points d'or générés sont ceux produits en cliquant sur le sprite d'arbre.

Changez les variables dans le bloc **ajouter Carbone par seconde à Carbone** à l'intérieur de la boucle infinie. Utilisez la liste déroulante pour sélectionner **Or**. Dans le menu **Variables** orange foncé, choisissez l'ovale de variable **Or par seconde** et glissez-le sur **Carbone par seconde** dans le bloc pour le remplacer. L'ovale précédent **Carbone par seconde** sera « expulsé » du bloc et vous pouvez le supprimer complètement en le glissant dans la bibliothèque de blocs.

Le programme complet devrait ressembler à ceci :



Ajouter des « actions écologiques » qui réduisent le carbone

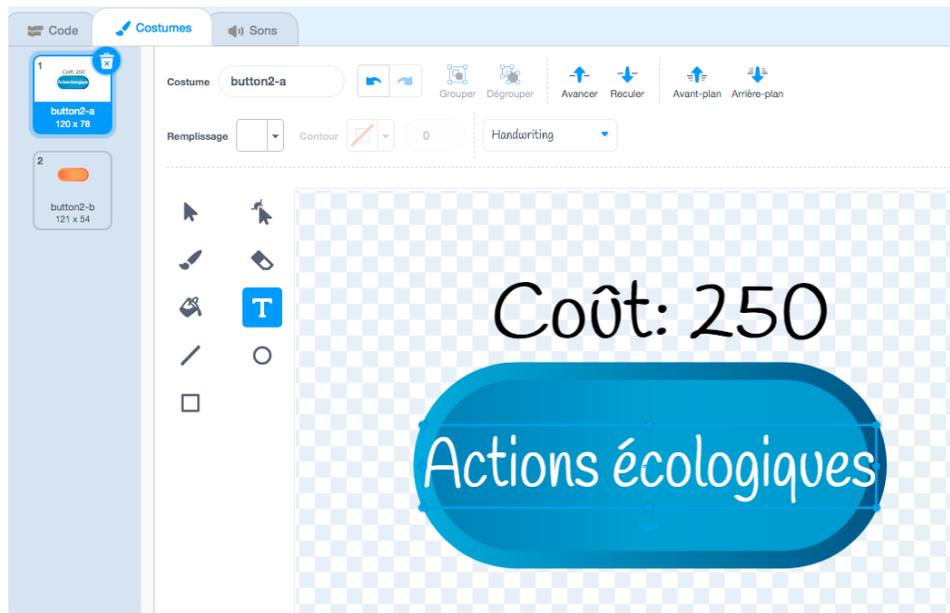
- Ajouter un sprite de bouton

Pour ajouter un nouveau sprite, ouvrez la bibliothèque de sprites en cliquant sur le bouton **Choisir un sprite** dans le panneau des sprites.

Au moyen de la barre de recherche dans la bibliothèque Choisir un sprite, recherchez « bouton » (bouton) et sélectionnez votre sprite de bouton préféré.

Pour étiqueter votre bouton, sélectionnez l'onglet **Costumes** dans le coin supérieur gauche de l'écran.

À l'onglet Costumes, ajoutez du texte à votre bouton au moyen de l'outil Texte. Modifiez la couleur de votre texte au moyen de l'outil de remplissage et glissez le texte en place. Dans l'exemple ci-dessous, nous avons utilisé deux boîtes de texte : une pour étiqueter le bouton et une étiquette de coût placée au-dessus du bouton. L'étiquette de coût communique le nombre de points dont aura besoin le joueur pour activer le bouton.



Vous pouvez glisser le sprite de bouton dans sa position voulue sur la scène en le glissant avec votre souris.

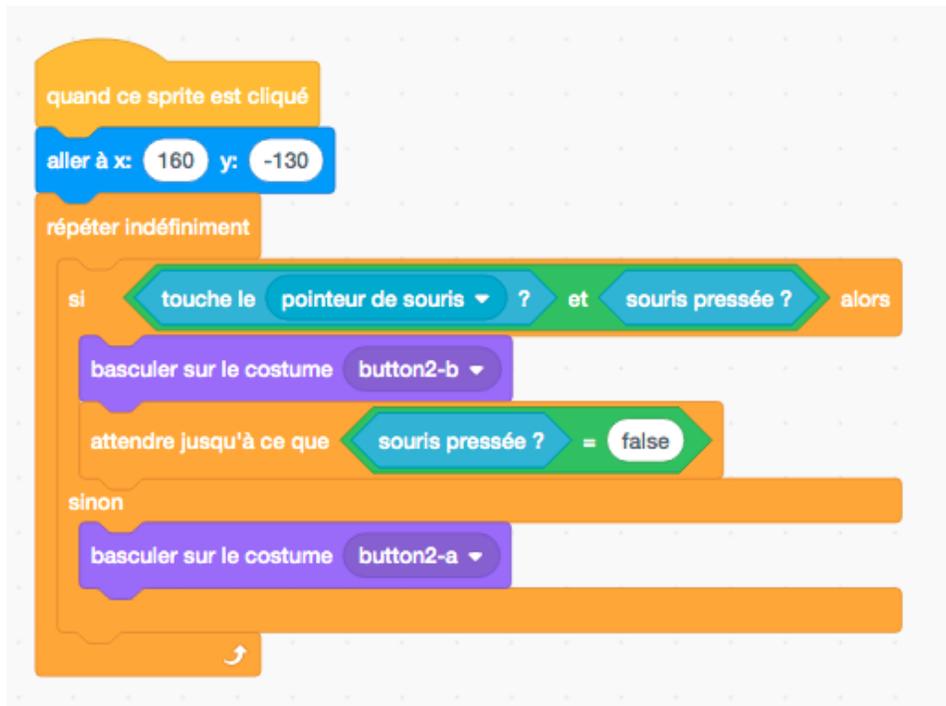
- Programmer l'apparence du bouton

Le code pour ce bouton est presque identique à celui du programme que nous avons développé pour le bouton Ajouter une usine. Vous pouvez reproduire le code existant en glissant le programme en entier que vous voulez copier sur l'icône du nouveau sprite de bouton dans le panneau des sprites.

Si vous sélectionnez le sprite du bouton d'action écologique, le code copié devrait maintenant apparaître dans le canevas de programmation. Le code original dans le sprite du bouton d'ajout d'une usine sera toujours présent.

Une fois que vous avez copié le code au sprite du bouton d'action écologique, la seule modification que vous devez apporter est de fixer la position x-y du bouton.

Le programme complet devrait ressembler à ceci :



- Programmer les effets du bouton

Nous voulons que ce programme fasse trois choses :

5. Nous voulons qu'il devienne actif (utilisable par le joueur) SEULEMENT lorsque le joueur a suffisamment de points d'**Or** pour couvrir le coût d'achat d'une action écologique.
6. Lorsqu'une action écologique est achetée, nous voulons que le pointage de **Carbone** diminue de façon autonome de 1 carbone par seconde (produit automatiquement par l'action écologique plutôt que par les clics du joueur)
7. Nous voulons que le compteur du pointage d'**Or** du joueur diminue de 250 points (le coût d'achat d'une action écologique).

Puisque ces effets sont très semblables à ceux de notre bouton Ajouter une usine, nous pouvons également reproduire le programme existant du bouton Ajouter une usine pour ce nouveau bouton.

Une fois le programme copié au sprite Action écologique, changez la valeur du bloc d'opérateur vert à 250 (pour tenir compte du fait que le bouton d'action écologique devrait être cliquable seulement lorsque le pointage du compteur d'or est de 250 points ou plus).

Changez la valeur du bloc **ajouter 1 à Carbone par seconde** pour -1, puisque l'action écologique devrait réduire la quantité de carbone avec le temps.

Changez la valeur du bloc **ajouter -100 à Or** pour -250 afin de tenir compte du coût d'achat d'une action écologique.

Le programme complet devrait ressembler à ceci :



## Possibilités d'approfondissement

### Modifier l'apparence de votre jeu

Dans l'exemple jouable du jeu (lien dans le haut de ce document), vous avez peut-être remarqué que nous avons ajouté un arrière-plan et l'avons programmé pour changer de couleurs lorsque la quantité de carbone dans l'atmosphère atteint certaines valeurs fixées.

### Ajouter un arrière-plan

Le panneau des arrière-plans se trouve à la droite du panneau des sprites. Pour ajouter un arrière-plan, cliquez sur le bouton **Choisir un arrière-plan** dans le coin inférieur droit de l'écran.

Choisissez l'arrière-plan voulu dans la bibliothèque d'arrière-plans. Dans l'exemple, nous avons sélectionné l'arrière-plan **Blue Sky (Ciel bleu)**.

### Ajouter une minuterie

En créant une variable de minuterie, nous pouvons compter le nombre de secondes nécessaire pour atteindre zéro unité de carbone (ou, au contraire, fixer un compte à rebours).

Une minuterie de base qui compte les secondes pourrait ressembler à ceci :



Encouragez les étudiants à modifier les valeurs des champs de compteur de carbone, de compteur d'or, de carbone par seconde et d'or par seconde pour changer les résultats des différents éléments du jeu.