

Simulation des forces de vol avec le micro:bit		6e année, vol position	
Plan de leçon	Outil de programmation	Makecode et micro:bits	
	Compétences transversales	Programmation et vol	
Idées générales Les étudiants utiliseront leurs compétences en codage (compréhension des variables et des instructions conditionnelles) pour coder un simulateur de vol simple qui modélise les forces de base du vol : la poussée, la traînée, la portance et le poids. Les étudiants expérimenteront avec leur simulateur en modifiant le poids et la poussée de leur avion pour voir quel impact cela a sur le fait que l'avion monte, descend ou reste stable.	Attentes précises A2.1 écrire et exécuter des codes lors de l'exploration et de la modélisation de concepts, notamment pour obtenir des données d'entrée de différentes façons à des fins diverses. D2.2 décrire les relations qui existent entre les quatre forces du vol : la portance, le poids, la traînée et la poussée, et qui rendent le vol possible. D2.4 décrire diverses méthodes utilisées pour modifier les quatre forces du vol.		
Matériel • Chromebook ou ordinateur portable pour chaque élève • Optionnel : micro:bits (le code peut être conçu et testé uniquement en utilisant le site web micro:bit, cependant l'apprentissage est plus efficace si vous avez des micro:bits physiques)			

Introduction

Dans cette leçon, nous explorerons les quatre forces du vol et verrons comment elles interagissent pour maintenir un aéronef dans les airs. Ces forces agissent par paires qui s'opposent, et comprendre leur équilibre nous aide à expliquer comment les avions montent, descendent, accélèrent ou ralentissent.

La portance et le poids sont des forces opposées qui déterminent si un avion gagne ou perd de l'altitude.

- Portance : La force ascendante créée par le mouvement de l'air autour des ailes. La portance doit être supérieure au poids pour qu'un avion puisse s'élever.
- Poids : La force descendante causée par la gravité qui attire l'avion vers la Terre.

La traînée et la poussée sont une autre paire de forces opposées qui influencent la vitesse de l'avion, et affectent donc également la portance.

- Traînée : La force qui résiste au mouvement lorsque l'avion se déplace dans l'air.
- Poussée : La force propulsive produite par les moteurs ou les hélices qui pousse l'avion vers l'avant.

Tout au long de l'activité, les élèves verront comment les quatre forces doivent rester en équilibre pour un vol stable.

Dans la partie codage de cette leçon, nous nous concentrerons sur deux concepts essentiels de la programmation :

- Variables : Valeurs qui peuvent changer au cours d'un programme et qui peuvent être utilisées pour stocker des informations telles que la vitesse, le poids et la portance.
- Instructions conditionnelles : Règles « si... alors... » qui permettent au programme de prendre des décisions. Par exemple, montrer l'avion qui monte si la portance est supérieure à la force du poids.

Les étudiants utiliseront ces concepts pour simuler les forces du vol et observer comment la modification d'une variable affecte les autres, renforçant ainsi à la fois les compétences en science et en codage.

Action

Faites visiter à chaque élève le *site de codage Micro:bit*, « <https://makecode.microbit.org/> », et commencer un *nouveau projet*.

Étape 1 : Identification et création de nos variables

Expliquez que les étudiants construiront un simulateur de vol simple qui modélise les quatre forces importantes impliquées dans le vol.

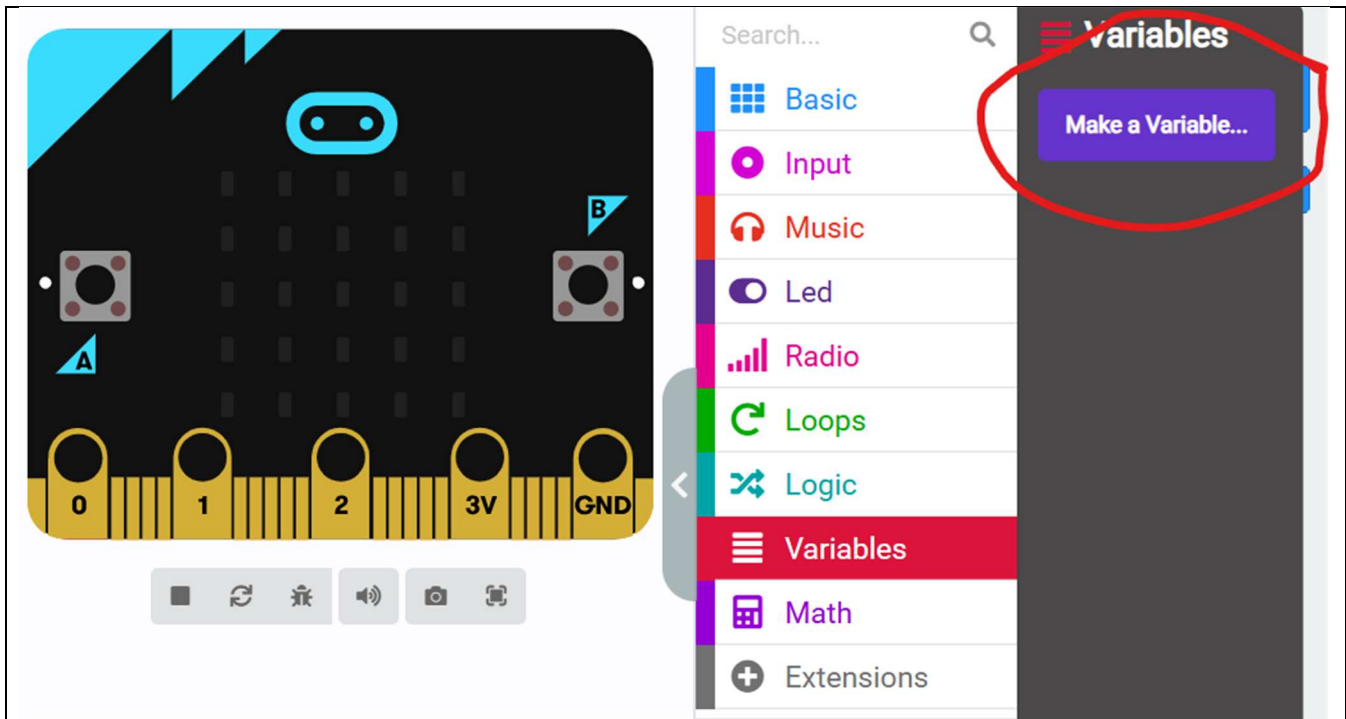
Si nécessaire, passez brièvement en revue les quatre forces opposées – le poids et la portance, ainsi que la poussée et la traînée (explication trouvée dans la section d'introduction).

Pour construire ce simulateur, les étudiants créeront des variables qui changeront tout au long du programme et détermineront si l'avion monte, descend ou reste à niveau.

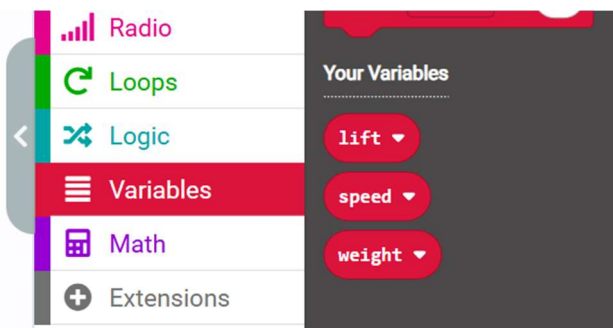
Les variables que les étudiants coderont sont la *portance*, le *poids* et la *vitesse*.

Remarque : Pour simplifier, la « vitesse » représentera à la fois la poussée et la traînée. Augmenter la vitesse implique une poussée, tandis que diminuer la vitesse implique une traînée.

Cliquez sur l'*onglet Variables rouge* et sélectionnez *créer une variable*. Vous devrez créer trois variables : « vitesse » (cela représentera à la fois la *poussée* et la *traînée*), « poids » et « portance ».



Une fois les variables créées, vous devriez les voir listées dans le *menu Variables* sous « Vos variables ».



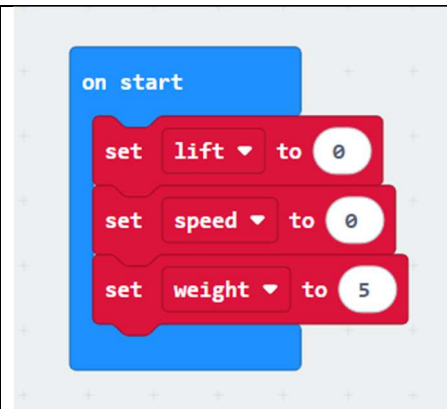
Étape 2 : Définir les valeurs de départ

Expliquez que nous devons définir nos variables au début du programme; sinon, l'ordinateur ne connaîtra pas leurs valeurs. Les ordinateurs ne suivent que les instructions que nous leur donnons, donc des valeurs de départ claires sont essentielles.

S'il n'y en a pas déjà un dans l'espace de travail, allez à l'*onglet Basic bleu* et faites glisser un bloc « au démarrage » dans l'espace de travail.

Ensuite, faites glisser trois blocs « *définir variable à* » et placez-les à l'intérieur du bloc « au démarrage » – un pour chacune de nos variables : *portance*, *poids* et *vitesse*.

- Réglez la vitesse à 0
- Réglez la portance à 0
- Réglez le poids à 5 (une valeur arbitraire; vous pouvez attribuer une unité telle que « avion modèle de 5 kg » si vous préférez)



Note aux étudiants : Cela signifie que lorsque notre code commence, l'avion est immobile parce que la vitesse est nulle, l'avion a un poids, et l'avion n'a pas de portance puisqu'il n'a pas encore de puissance ni de poussée.

Étape 3 : Coder le simulateur des principales forces de vol

Si un bloc « pour toujours » ne se trouve pas déjà dans votre espace de travail, faites glisser un bloc « pour toujours » depuis l'onglet *bleu de base* dans votre espace de travail.

3A : Ajout de la poussée et de la traînée

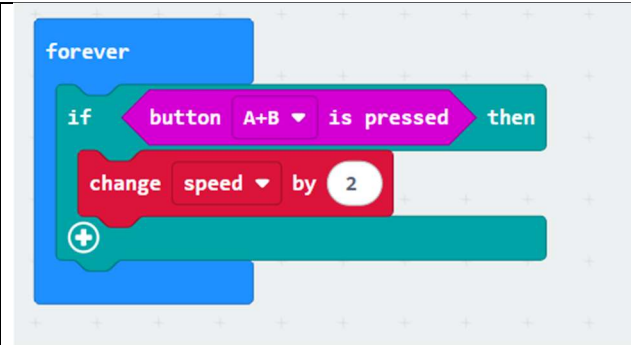
Commencez par placer un bloc « si » de l'onglet *Logique turquoise* à l'intérieur de la boucle « pour toujours ». Ce bloc nous permettra de coder ce qui se passe lorsque la poussée est ajoutée et ce qui se passe lorsque la poussée est retirée.

Ensuite, ajoutez le bloc hexagonal « si le bouton est pressé », que l'on trouve également dans l'onglet *Logique sarcelle*. Utilisez le menu déroulant sur le bloc pour sélectionner « A+B ».

Enfin, faites glisser un bloc « changer la variable de » depuis l'onglet *rouge Variables*, sélectionnez « vitesse » comme variable, et changez la valeur à 2. Cela augmentera la

vitesse de l'avion chaque fois que les boutons A+B seront pressés, simulant une poussée supplémentaire.

Votre code devrait ressembler à ceci :

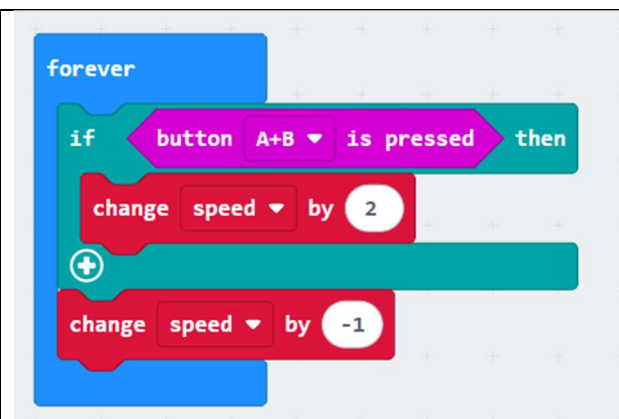


Cela signifie que lorsque nous *maintenons enfoncés les boutons A et B*, nous ajoutons de la *vitesse* à l'avion. Cela représente la *force de poussée*, qui augmente la vitesse de l'avion et contribue à la *portance* (nous coderons cette interaction sous peu).

Pour ajouter les effets de la *traînée* sur notre avion, placez un bloc « changer la variable de » directement *en dessous* de l'instruction conditionnelle « si ». Assurez-vous de sélectionner à nouveau « vitesse » comme variable et de changer la valeur à -1 .

Cela simule la traînée en diminuant lentement la vitesse chaque fois que la poussée n'est pas ajoutée.

Notre boucle *pour toujours* devrait maintenant ressembler à ceci :



Cela signifie que *la traînée agit toujours* sur notre avion, ce qui fait diminuer la *vitesse*, ce qui à son tour affecte la *portance* de l'avion.

Cela aide à démontrer aux étudiants pourquoi les avions nécessitent une *poussée continue* pour rester en l'air.

3B : Prévenir la vitesse négative

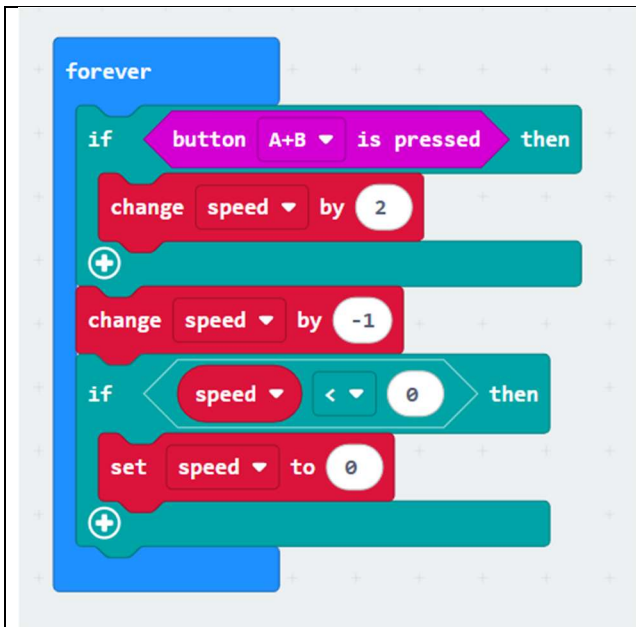
Nous devons maintenant ajouter une autre instruction conditionnelle pour empêcher notre avion d'avoir une vitesse négative, puisque la traînée agit toujours sur l'avion et réduit sa vitesse.

Faites glisser un autre bloc « si/alors » de l'onglet *Logique* et placez-le sous le bloc précédent, toujours à l'intérieur de la *boucle pour toujours*. Placez ensuite un *bloc de comparaison* hexagonal à l'intérieur du bloc « si » et sélectionnez le symbole *inférieur à*.

Ajoutez la variable « vitesse » de l'onglet *Variables rouge* du côté gauche de la comparaison, et tapez 0 du côté droit. Votre comparaison devrait être : « Si la vitesse < 0 ».

Enfin, placez un bloc « définir la variable à » à l'intérieur de ce bloc « si », sélectionnez *vitesse* et réglez la valeur à 0.

Votre code devrait maintenant ressembler à ceci :



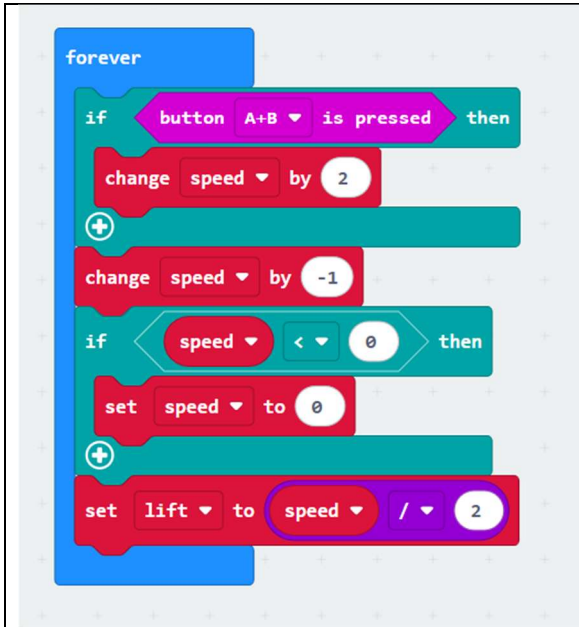
Cela signifie que notre vitesse ne peut pas descendre en dessous de zéro, ce qui serait arrivé autrement puisque nous venons de coder notre force de traînée pour agir toujours sur l'avion.

3C : Ajouter de la portance.

La force de portance est directement affectée par la vitesse de notre avion, donc nous devons coder cette relation. Pour ce faire, placez un autre bloc « définir variable à » provenant de l'onglet *Variables rouge* sous le bloc « si » précédent, toujours à l'intérieur de la *boucle pour toujours*. Assurez-vous de sélectionner la variable « portance ».

Ensuite, allez à l'onglet *Math violet* et faites glisser un *bloc de division ovale*. Placez la variable « *vitesse* » (trouvée dans l'onglet *Variables rouge*) du côté gauche de l'équation, et tapez 2 du côté droit. Votre équation devrait indiquer « *vitesse divisée par 2* ».

Maintenant, votre code devrait ressembler à ceci :



Cela signifie que si nous augmentons la vitesse en appuyant sur les boutons A+B (représentant la poussée), notre portance augmentera de la moitié de l'augmentation de la vitesse.

3D : Comparer la portance et le poids

L'étape suivante consiste à coder ce qui nous indique si notre avion *monte*, *descend* ou *reste à niveau*. La *portance* fait monter l'avion, tandis que la force du *poids* (gravité) le fait descendre.

Si la *force de portance* est supérieure au *poids*, l'avion devrait monter.

Si la *force de portance* est inférieure au *poids*, l'avion devrait descendre.

Si la *force de portance* est égale au *poids*, l'avion devrait rester à *niveau*.

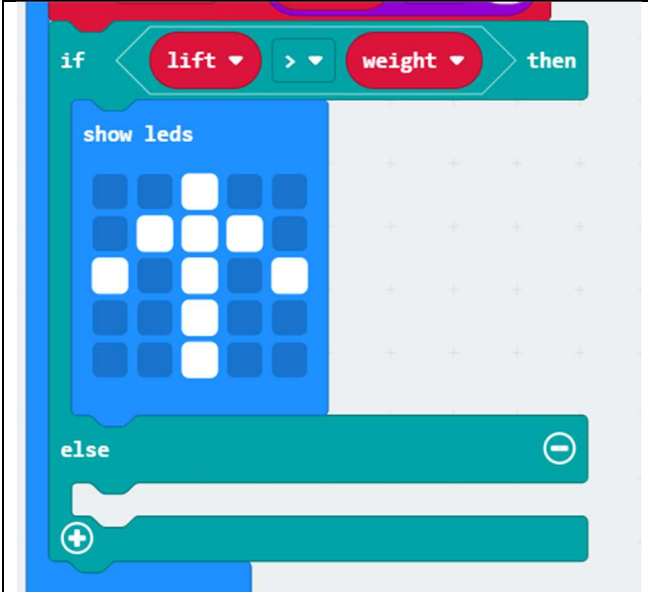
Vous pouvez probablement deviner que nous avons besoin de plus de *structures conditionnelles*. Cette fois, prenez un *bloc « si/sinon »* de l'onglet *Logique turquoise* et placez-le à l'intérieur de la *boucle pour toujours*, juste en dessous du dernier bloc que vous avez ajouté.

Ensuite, faites glisser un bloc de comparaison hexagonal depuis l'onglet *Logiciel sarcelle*, sélectionnez le symbole *supérieur à*, et placez-le à l'intérieur de la partie « *si* » du bloc. Du

côté gauche de la comparaison, ajoutez la *variable* « *portance* »; du côté droit, ajoutez la *variable* « *poids* » (toutes deux provenant de l'*onglet Variables rouge*). Votre condition devrait indiquer : « Si la portance est supérieure au poids ».

Puisqu'une force de portance supérieure au poids signifie que l'avion devrait monter, nous devons trouver un moyen de le montrer sur le micro:bit. Ajoutez un bloc « *afficher les DEL* » de l'*onglet Basic bleu* et concevez une icône de flèche vers le haut en sélectionnant les DEL que vous souhaitez allumer.

La nouvelle partie de votre code devrait ressembler à ceci :

	Maintenant, si la portance est supérieure au poids, une flèche vers le haut s'affichera, indiquant que notre avion prend de l'altitude.
--	--

Nous devons maintenant remplir la déclaration conditionnelle. Si la portance n'est pas supérieure au poids, il y a deux possibilités : elle peut être inférieure au poids, ou elle peut être égale au poids. Nous devons coder pour les deux scénarios.

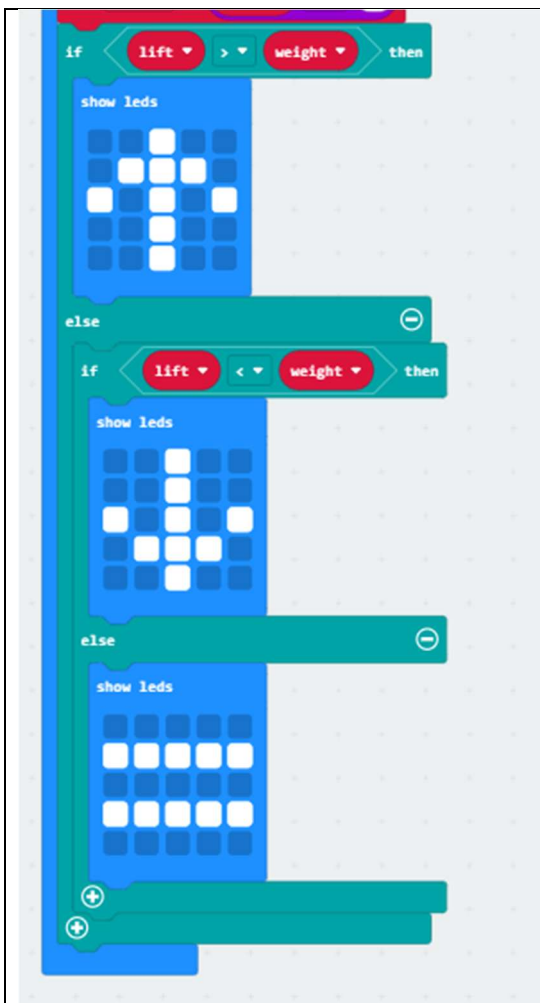
Pour ce faire, placez un autre *bloc* « *si/sinon* » de l'*onglet Logique turquoise* à l'intérieur de la section *sinon* du premier bloc.

Ensuite, faites glisser un autre *bloc de comparaison* hexagonal depuis l'*onglet Logique sarcelle* et sélectionnez le symbole *inférieur à*. Placez la variable « *portance* » du côté

gauche de la comparaison et la variable « *poids* » du côté droit. Votre condition devrait être : « si la portance est inférieure au poids ».

Pour cette condition, suivez les mêmes étapes qu'auparavant : ajoutez un bloc « *afficher les DEL* » et concevez une icône *flèche vers le bas* pour indiquer que l'avion descend.

La partie *sinon* de ce deuxième bloc représente la dernière possibilité : la portance est égale au poids. Nous allons donc faire glisser un autre bloc « *afficher les DEL* » dans la section *sinon* et sélectionner les DEL pour former un *signe égal*, indiquant un vol en palier.



Cela signifie que si la force de notre portance est supérieure à celle du poids, notre simulateur indiquera que notre avion prend de l'altitude avec une « flèche vers le haut ».

Si la force de notre portance est inférieure à celle du poids, notre simulateur indiquera que notre avion perd de l'altitude avec une « flèche vers le bas ».

Sinon, la force de la portance doit être égale à celle du poids, et notre avion ne gagnera ni ne perdra d'altitude, il volera à niveau, ce qui nous est indiqué par un « signe égal ».

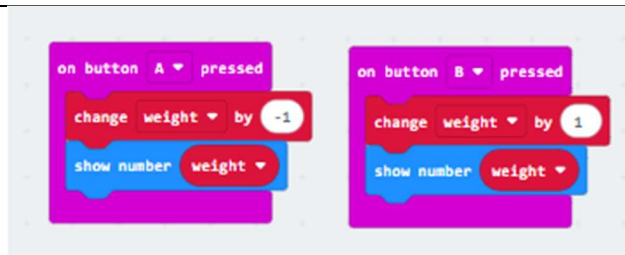
Étape 4 : Contrôler le poids

Faites glisser un bloc « *lors de l'appui sur un bouton* » depuis l'onglet rose *Entrée* et sélectionnez « Bouton A ». À l'intérieur de ce bloc, placez un bloc « *changer variable de* » provenant de l'onglet rouge *Variables*, et sélectionnez la variable « *poids* ». Changez la valeur à -1.

En dessous, toujours à l'intérieur du même bloc « *lorsque le bouton est pressé* », ajoutez un bloc « *afficher le nombre* » provenant de l'onglet bleu de base. Faites glisser la variable ovale « *poids* » dans le bloc « *afficher le nombre* » pour compléter le code.

Répétez les mêmes étapes, sauf que cette fois, sélectionnez *Bouton B* et modifiez la valeur de la variable « *poids* » de 1 au lieu de -1.

Votre code supplémentaire devrait ressembler à ceci :



Cela signifie que lorsque nous appuyons sur le Bouton A, le *poids* augmente de 1 et la valeur mise à jour est affichée à l'écran, et lorsque nous appuyons sur le Bouton B, le *poids* diminue de 1 et la valeur est affichée à l'écran.

Étape 5 : Codes optionnels

Chacun des ajouts de code suivants peut améliorer la simulation, mais ils ne sont pas nécessaires pour la fonctionnalité de base. Ils peuvent être inclus pour ajouter de la clarté ou de la profondeur, ou omis dans un souci de simplicité et/ou de temps.

5A : Indicateur de valeur variable

Il peut être utile d'avoir un moyen rapide de voir les valeurs actuelles de vos variables à tout moment pendant la simulation.

Faites glisser un bloc « *lors du secouement* » depuis l'onglet rose *Entrée*. À l'intérieur de ce bloc, placez trois blocs « *afficher nombre* » de l'onglet bleu *Basique*, chacun contenant une des variables.

Assurez-vous de vous souvenir de l'ordre dans lequel vous placez les variables, cela vous aidera à interpréter correctement les valeurs lorsque le micro:bit les affichera.

Le code devrait ressembler à ceci :



Cela signifie que lorsque nous secouons le micro:bit, la valeur de la vitesse, de la portance et du poids sera affichée successivement à l'écran.

5B : Bouton de réinitialisation

Si vous avez la version 2 de micro:bit ou si vous utilisez uniquement le logiciel en ligne, vous avez également un *logo* qui peut être programmé comme un bouton. Nous pouvons utiliser cette fonction comme un bouton de réinitialisation pour ramener nos variables à leurs valeurs initiales.

Ajoutez un bloc « *lors de l'appui sur le logo* » depuis l'*onglet rose Entrée*. À l'intérieur, placez deux blocs « *définir variable à* » de l'*onglet rouge Variables* et sélectionnez les variables « *vitesse* » et « *poids* ». Réglez la *vitesse* à 0 et le *poids* à 5. Ensuite, ajoutez un bloc « *afficher l'icône* » depuis l'*onglet bleu Basique* et choisissez n'importe quelle icône que vous souhaitez pour représenter l'action de réinitialisation.

Votre code devrait ressembler à ceci :



Cela signifie que lorsque vous appuyez sur le logo, vos variables seront réinitialisées à leur état initial et afficheront une icône de votre choix pour vous indiquer qu'elles ont été réinitialisées.

Étape 6 : Expérience!

Téléchargez le code sur vos micro:bit ou utilisez le simulateur fourni sur le site web à côté de votre espace de travail de codage. Laissez les élèves expérimenter en appuyant sur la poussée (boutons A+B) et observer combien de temps il faut pour obtenir une flèche vers le haut, indiquant que l'avion prendra de l'altitude. Faites-les expérimenter en maintenant les boutons, en les relâchant, etc. Voyez ce qui se passe lorsque vous augmentez le poids (bouton B) et diminuez le poids de l'avion (bouton A).

Consolidation et approfondissement

Vous pouvez demander aux élèves de créer un tableau pour comparer combien de secondes il faut ajouter de poussée pour un poids donné afin que l'avion décolle et augmente en altitude.