

Le poisson, le chat et le chien

Cette activité se déroule sur une ou plusieurs séances.



Situation à programmer

Ce jeu propose une interaction entre 3 lutins :

- un poisson qui navigue dans le plan graphique;
- un chien qui fait des aller-retours sur un segment horizontal;
- un chat qui se place comme bon lui semble pour attraper le poisson. Mais le chien peut attraper le chat sur son segment.

Objectifs

L'intérêt de cette activité est

- d'assurer l'utilisation des commandes "Déplacement" - "Motion" de base de Scratch;
- comprendre l'idée de programmation en parallèle;
- donner des notions de programmation objet.

Notions algorithmiques visées

- Les variables : score
- L'instruction conditionnelle : Si ... alors ... sinon
- Les boucles : Tant que

Compétences/capacités en lien avec le programme de Maths

Les actions scratch donnent du sens au concept mathématiques

Action scratch	Programme Maths
Déplacement du chien	Translation
Taille du chat	Pourcentage
Déplacement du chien	Vitesse
Déplacement du poisson	Random et l'aléatoire
Lutins	Repérage

Déroulement

Ce programme est basée sur la programmation objet. Chaque objet, nommé lutin, possède ses propres propriétés. Pour parvenir à créer ce programme, il faut procéder comme toujours en informatique par étapes. Les actions de ces 3 lutins ne sont pas d'un même niveau de création. On peut considérer l'ordre suivant de difficulté croissante :

1. Création du lutin Chien
2. Création du lutin Chat
3. Création du lutin Poisson

Ce jeu peut donc être proposé aux élèves sous au moins 2 formes :

Par élève : l'élève fait tout le programme. Il programme donc les 3 lutins.

Par trinôme d'un niveau hétérogène : chaque élève programme un lutin et mutualise ensuite son lutin.

Plusieurs élèves sur un même projet

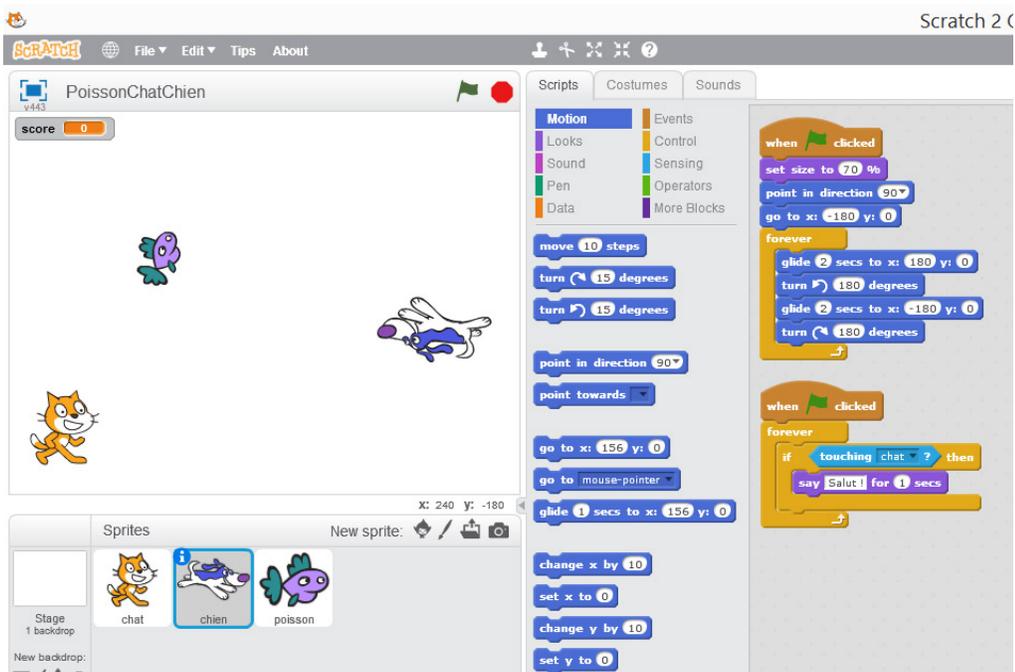
Lors de cette activité, il est intéressant de faire travailler les élèves en trinôme. Dans un premier temps, chaque élève crée un programme pour travailler son lutin. Une fois les lutins créés avec leurs propriétés, Les 3 lutins sont fusionnés sur chaque projet Scratch.

La procédure est donnée sur la feuille élève.

Création du chat



Création du chien



Création du poisson

The screenshot shows the Scratch IDE interface for a project named "PoissonChatChien" (version v443). The stage displays a score of 0 and three sprites: a fish, a dog, and a cat. The "Scripts" and "Motion" block palettes are visible on the right. The "Scripts" palette includes blocks for "when green flag clicked", "forever", "if touching chat?", "play sound", "set heure to timer", "hide", "wait until timer > heure + 2", and "show". The "Motion" palette includes blocks for "move 10 steps", "turn 15 degrees", "point in direction 90", "point towards", "go to x: -101 y: 43", "go to mouse-pointer", "glide 1 secs to x: -101 y: 43", "change x by 10", "set x to 0", "change y by 10", "set y to 0", and "if on edge, bounce". The "Sprites" panel shows three sprites: "chat", "chien", and "poisson".

? Situation à programmer

Ce jeu propose une interaction entre 3 lutins :

un poisson navigue sur le plateau de jeu aléatoirement,

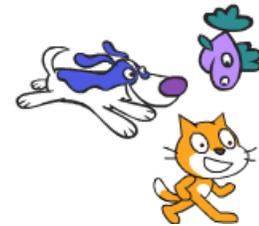
un chien fait des aller-retours sur un segment horizontal.

Un chat se place comme bon lui semble. Le joueur est le chat.

Le chat a 5 vies au départ du jeu. Il n'est pas possible de gagner des vies.

Le chat attrape le poisson et marque 1 point à chaque contact avec le poisson.

Le chien attrape le chat et enlève une vie au chat à chaque contact.



Création du chien

Consignes.

Créer un lutin "chien" qui va d'un coté à l'autre du plateau en un déplacement horizontal de façon ininterrompue et répétitive jusqu'à la fin du jeu.

Le chien doit parcourir la fenêtre en son milieu et de façon horizontale. Tu placeras le chien initialement en $(-180;0)$.

Le mouvement doit démarrer lorsque le drapeau vert est cliqué.



Création du chat

Consignes.

Créer un lutin "chat" qui se déplace à l'aide des touches de direction du clavier.

Le chat doit mesurer 70% de sa taille initiale. Tu placeras le chat initialement en $(-188; -117)$.

Le mouvement du chat doit démarrer lorsque le drapeau vert est cliqué.



Création du poisson ★

Consignes.

Créer un lutin "poisson" qui se déplace de façon ininterrompue et répétitive pendant 4 secondes dans une direction rectiligne. Toutes les 4 secondes, la direction change de façon aléatoire.

Le poisson doit mesurer 40% de sa taille initiale. La position du poisson est aléatoire sur le plateau de jeu.

Le mouvement du chat doit démarrer lorsque le drapeau vert est cliqué.



Création des variables score et vie

La variable score

La variable score augmente d'un point si le poisson touche le chat.

A quel lutin doit-on attacher cette action ?

La variable vie

La variable vie perd un point si le chat touche le chien.

A quel lutin doit-on attacher cette action ?

Action supplémentaire après la création des 3 lutins

Lorsque le poisson touche le chat :

1. un son retentit.
2. le poisson disparaît pendant 2 secondes et réapparaît à un emplacement aléatoire du plateau (x varie dans l'intervalle $(-200;200)$ et y varie dans l'intervalle $(-160;160)$).

Création d'un fond

Il est possible de mettre en fond un arrière plan pour simuler le fond de la mer.



Aide

Action	Brique
Déplacer vers le haut	
Créer la variable score	
Ajouter 1 à i	
Ajuster à la taille	
Jouer une action tout au long du jeu	
Si un lutin est touché alors	
Glisser vers une position	
Faire une rotation de 15 degrés	
Choisir un nombre aléatoire entre deux valeurs	
Attendre	

Travail par groupe : mutualisation des lutins

Partage

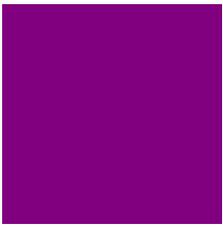
1. Ouvrir un compte sur scratch : <https://scratch.mit.edu/>. Lors de l'inscription, un mail est envoyé et il est important de confirmer l'inscription.
2. Créer un projet dans lequel on définit un lutin (avec ses actions) et enregistrer le projet.
3. Partager son projet à l'aide du bouton Partage, puis envoyer l'adresse internet de son projet par courriel à son équipe.

Récupération

1. Ouvrir son compte courriel et copie le lien scratch reçu [CTRL+C].
2. Coller dans la barre d'adresse du navigateur **sans se déconnecter de son compte**.
3. A l'ouverture du programme, "voir à l'intérieur".
4. Ouvrir le sac à dos en bas de la page. Ce sac à dos permet de déposer des morceaux de codes réutilisables. Glisser-copier le lutin. Ses actions sont copiées en même temps.

Utiliser

1. Ouvrir son propre programme.
2. Ouvrir le sac à dos.
3. A l'ouverture du programme, "voir à l'intérieur".
4. Ouvrir le sac à dos en bas de la page. Glisser-copier le lutin.



Le jeu de pong

Cette activité se déroule sur une ou plusieurs séances.

? Situation à programmer

Ce jeu, ô combien célèbre, est sans doute un des premiers jeux Atari! Il met aussi en cohésion 3 lutins. Une balle qui interagit avec deux raquettes. Les raquettes ne se déplacent que verticalement alors que la balle rebondit sur les raquettes ou les murs. Si la balle dépasse la raquette, $x \in [-208; 208]$, la joueur a perdu. C'est le principe du tennis.

Objectifs

L'intérêt de cette activité est

- d'assurer l'utilisation des commandes "Déplacement" - "Motion" de base de Scratch;
- comprendre l'idée de programmation en parallèle;
- donner des notions de programmation objet.
- créer un lutin : un segment.
- utiliser une boucle infinie.

Notions algorithmiques visées

- Les variables : score
- L'instruction conditionnelle : Si ... alors ... sinon
- Les boucles : Tant que

Compétences/capacités en lien avec le programme de Maths

Les actions scratch donnent du sens au concept mathématiques

Action scratch	Programme Maths
Déplacement de la raquette	Translation
Taille de la balle	Pourcentage
Déplacement de la balle	mesure d'angle
Départ de la balle	Random et l'aléatoire
Position de la balle par rapport aux murs droit et gauche	Repérage

Déroulement

Ce programme est basée sur la programmation objet. Chaque objet, nommé lutin, possède ses propres propriétés. Les deux raquettes fonctionnent de manière similaires. Les actions de ces 3 lutins ne sont pas d'un même niveau de création. On peut considérer l'ordre suivant de difficulté croissante :

1. Création des raquettes
2. Création de la balle

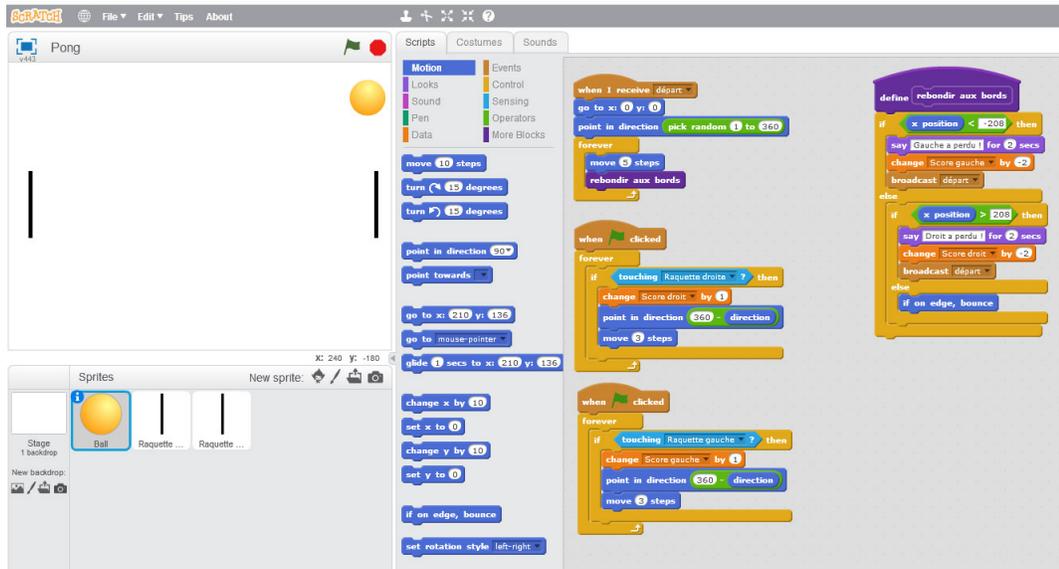
Ce jeu peut donc être proposé aux élèves sous au moins 2 formes :

Par élève : l'élève fait tout le programme. Il programme donc les 3 lutins.

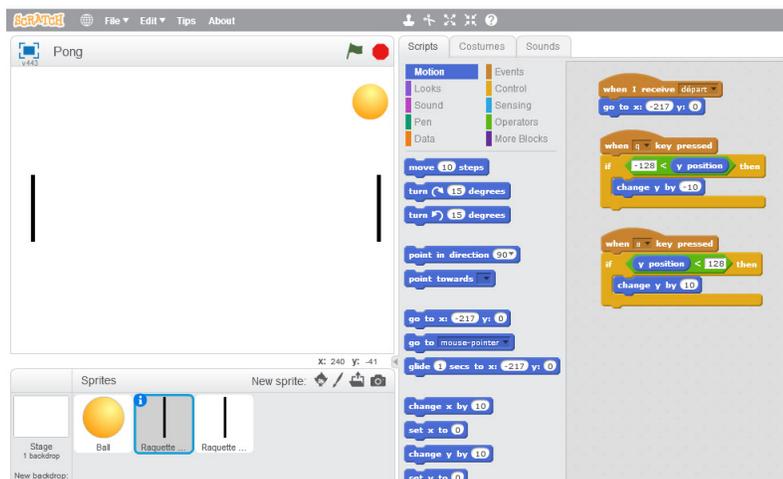
Par trinôme d'un niveau hétérogène : chaque élève programme un lutin et mutualise ensuite son lutin.

Une difficulté est quand même à signaler sur la création des raquettes. Lorsque la raquette arrive sur le bord du plateau de jeu, elle ne doit pas disparaître. La raquette n'a qu'un déplacement vertical. Pour rappel, y appartient à l'intervalle $[-128; 128]$.

Déplacements de la balle



Déplacements de raquette



On comprend assez bien le déplacement de la raquette droite en utilisant les touche **p** et **m**.

Prolongements

Pour les élèves les plus rapides, on peut proposer de coder le jeu sans utiliser la brique : **rebondir si le bord est atteint** ou faire augmenter la vitesse de la balle tous les 10 points.

Pong

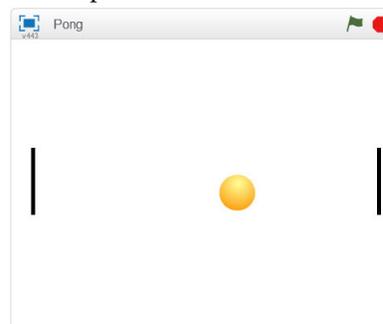
Fiche Élève

Cette activité repose sur la création du jeu de Pong, diminutif de ping pong, nom familier pour tennis de table.

Consignes

Sur chacun des 2 cotés du jeu, il faut créer une raquette, simulée par un segment. Une balle va d'un coté à un autre du plateau de jeu. La balle rebondit sur les murs haut et bas du plateau.

Si avec la raquette, le joueur rate la balle, le joueur adverse marque un point. Le premier des deux joueurs qui atteint 10 points gagne la partie.



Création des raquettes

Création de la raquette gauche

Consignes.

1. En utilisant le crayon de création de lutin, dessiner un segment de 80 px de haut et d'épaisseur 5 px centré **sur la croix**. Zoomer le quadrillage si besoin. Pour ouvrir la palette de dessin, cliquer sur cette partie de l'écran 
2. Cliquer sur l'onglet **Scripts** pour associer le script à la raquette.
3. Créer un déplacement vertical de la raquette gauche pour $x = -208$ et $-128 \leq y \leq 128$
4. Associer les touches **A** et **Q** du clavier à ces déplacements. **A** pour simuler la montée, **Q** la descente.

Exemple



Création de la raquette droite

Consignes.

1. Dupliquer la raquette gauche.
2. Modifier le script pour utiliser la raquette droite. Penser symétrie.
3. Associer les touches **P** et **M** du clavier aux déplacements. **P** pour simuler la montée, **M** la descente.



Création de la balle ★

Consignes.

1. Choisir le lutin balle.
2. Commencer par créer un programme qui
 - (a) démarre lorsque le drapeau vert est cliqué.
 - (b) place la balle au centre du plateau puis la met en mouvement selon une direction aléatoire rectiligne.
 - (c) impose à la balle de rebondir lorsqu'elle touche un mur



Aide. On pourra utiliser une brique pour définir le rebond sur les murs.

Interaction des lutins

Faire interagir la balle avec les raquettes.

- Lorsque la balle touche une raquette, elle rebondit dessus.
- Lorsque la balle dépasse une raquette du joueur B, le joueur A marque un point, le jeu dit "A marque 1 point" et au bout d'une seconde, le jeu reprend.

Création des variables score Gauche et score Droit

Pour créer une variable, il faut se placer dans la section **Données**

Les variables score Droit

Lorsque la position de la balle en abscisse est inférieure à -208 ($x \leq -208$) le joueur de droite marque 1 point.

Comment coder le score du joueur de gauche?



Pour les plus rapides

Quelques idées non exhaustives pour améliorer le jeu

- tous les 10 échanges, la balle accélère
- tous les 10 échanges, la balle diminue de 5% de sa taille.

Création de la variable échange

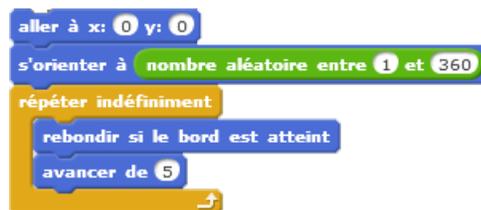
La variable échange compte le nombre d'échanges.

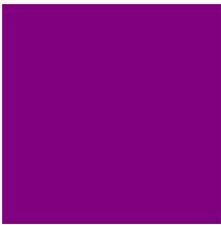
Création de la variable taille

La variable taille donne la taille de la balle.

Aide pour la balle

Deux petits blocs pour se lancer.





Casse briques

Cette activité se déroule sur une ou plusieurs séances.

? Situation à programmer

Ce jeu met en actions plusieurs lutins mais principalement 3 : Une balle, une raquette et une brique reproduite en plusieurs exemplaires. Le but du jeu est de casser toutes les briques avec la balle qui rebondit sur la raquette. Le jeu s'arrête lorsque toutes les briques sont cassées (disparues) ou lorsque les vies du joueur sont épuisées.

Objectifs

L'intérêt de cette activité est de

- élaborer un projet étapes par étapes
- travailler sur les angles avec le rebond.
- Créer une variable compteur pour établir la fin du jeu (plus de briques restantes)

Déroulement

Ce jeu peut être proposé aux élèves sous au moins 2 formes :

Par élève : l'élève fait tout le programme. Il programme donc la raquette, les briques et la balle.

Par binôme d'un niveau hétérogène : un élève programme la balle et un autre la raquette et les briques. L'intérêt de cette disposition est moindre. Seule la balle a un codage intéressant.

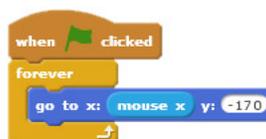
Il est préférable de commencer par le codage de la raquette, puis de la balle, puis d'une brique. Lorsque une brique réagit bien aux actions de la balle, il suffit de dupliquer la brique et de positionner les briques. Pour ce positionnement, la taille de la brique initiale est importante. On peut considérer seulement 20 briques sur le plateau : 4 rangées de 5 briques.

Codage de la raquette

La position de la raquette au départ est (0; -170).

Une petite précision est quand même à signaler sur la création de la raquette : Lorsque la raquette arrive sur le bord du plateau de jeu, elle ne doit pas disparaître. La raquette n'a qu'un déplacement horizontal.

Il est possible d'actionner la raquette avec les touches du clavier ou avec la souris. voir ci dessous.



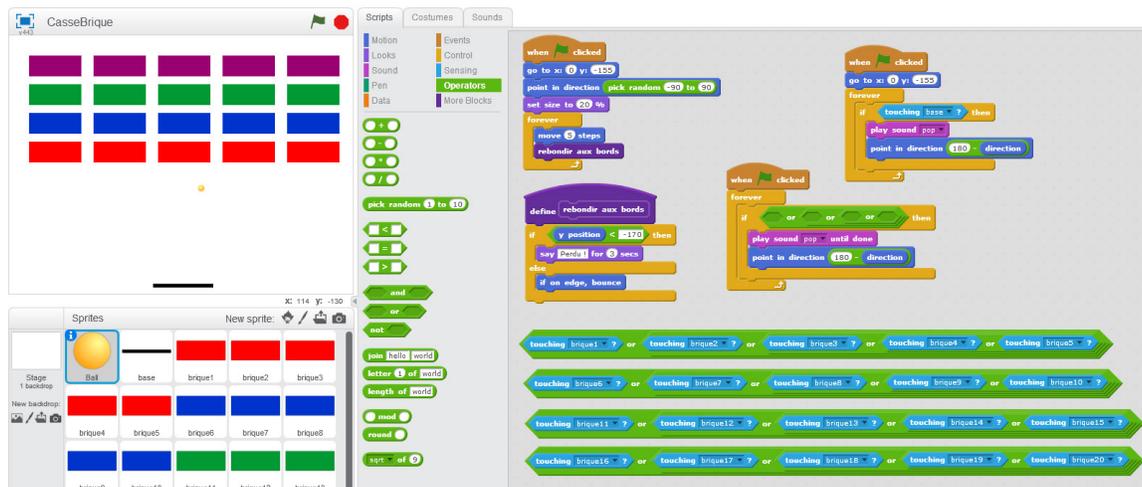
Codage de la brique

La balle doit rebondir sur la brique. **Après** rebond, la brique doit disparaître. Plusieurs possibilités sont à étudier dont

attendre secondes

Codage de la balle

La balle rebondit sur chaque objet ou mur. Une étude intéressante est de proposer aux élèves de coder le rebond. Un travail sur les angles doit alors être mené car Scratch ne gère que les angles géométriques. Il convient de mettre les briques vertes dans les emplacements de la conditions si - if.



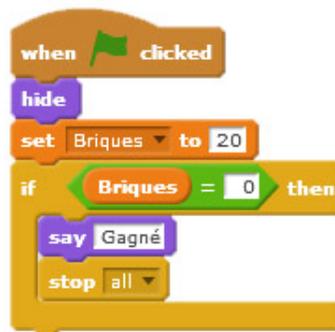
Plusieurs élèves sur un même projet

Voir l'activité précédente.

Codage de la variable de compteur

Lorsque il n'y a plus de brique sur le plateau, le jeu s'arrête. Pour cela, il faut créer une variable qui compte le nombre de briques restantes et lorsque la variable arrive à 0, un message écrit "Gagné!". On peut aussi créer une variable qui compte le nombre de briques cassées et lorsque la variable arrive au nombre de briques initial, un message écrit "Gagné!".

Une idée de script ci contre.



Variante ou étude des angles

Un problème intéressant est de travailler sur les angles en parlant des rebonds. Lorsque la balle touche un mur ou une brique ou la raquette, elle rebondit avec un angle de réfraction opposé à son angle d'incidence par rapport à sa verticale. On pourra évaluer cet angle par rapport aux angles géométriques de Scratch.

Demander aux élèves de coder un rebond sans utiliser la brique `rebondir si le bord est atteint`

On peut alors remplacer la brique `rebondir si le bord est atteint` par la brique `s'orienter à 180 - direction`

Prolongements

Pour les élèves les plus rapides, on peut proposer de coder plusieurs niveaux de jeux. Dans ce cas, c'est la vitesse de la balle qui augmente. Lorsque toutes les briques sont cassées, il faut les faire réapparaître toutes en même temps. Il est intéressant de commencer par une balle qui rebondit sur les bords du jeu.

? Situation à programmer

Ce jeu, ô combien célèbre, est sans doute un des premiers jeux Atari! Il met aussi en cohésion 3 lutins. Une balle qui interagit avec deux raquettes. Les raquettes ne se déplacent que verticalement alors que la balle rebondit sur les raquettes ou les murs. Si la balle dépasse la raquette, $x \in]-208; 208]$, la joueur a perdu. C'est le principe du tennis.

Codage de la raquette

La position de la raquette au départ est $(0; -170)$. Elle ne peut se pas sortir du plateau de jeu et doit être toujours visible. Déterminer un déplacement horizontal sur un intervalle approprié.

Actionner la raquette avec les touches directionnelles du clavier ou avec la souris, au choix.

Codage de la brique

La balle doit rebondir sur la brique. **Après** rebond, la brique doit disparaître. Plusieurs possibilités sont à étudier dont la commande 

Les briques doivent être espacées régulièrement. Lors de la création des briques, on peut lire leurs dimensions dans la vignette au centre de l'écran entourée en bleu. Pour dessiner les briques, utiliser l'outil rectangle du mode bitmap.

Codage de la balle

Choisir le lutin "Ball" et lui attribuer une taille de 20%.

La balle démarre sur la raquette. Lorsque la balle touche un mur ou une brique ou la raquette, elle rebondit.

Si la balle dépasse la raquette, le joueur perd une vie. Lorsque ses vies sont épuisées, le jeu écrit "Perdu!".

Sans utiliser la brique de rebond

Lorsque la balle touche un mur ou une brique ou la raquette, elle rebondit avec un angle de réfraction opposé à son angle d'incidence par rapport à sa verticale. On pourra évaluer cet angle par rapport aux angles géométriques de Scratch.

L'angle de départ de la balle est appartient à l'intervalle $[20^\circ; 160^\circ]$.

Codage de la variable compteur

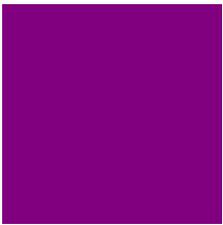
Lorsque il n'y a plus de brique sur le plateau, le jeu s'arrête.

Pour cela, il faut créer une variable qui compte le nombre de briques restantes et lorsque la variable arrive à 0, un message écrit "Gagné!"

On peut aussi créer une variable qui compte le nombre de briques cassées et lorsque la variable arrive au nombre de briques initial, un message écrit "Gagné!".

Pour aller plus loin

Lorsque le niveau 1 est terminé, le niveau 2 s'enclenche avec une vitesse de balle plus rapide. Prévoir au moins 4 niveaux de difficultés.



Space Invader

Cette activité en interdisciplinarité : Cycle 4- cycle 2.

? Situation à programmer

Le jeu de Space invader est basé sur un graphisme simple qui permet de travailler sur 2 cycles. Les élèves de cycle 2, en cours préparatoire, dessinent les costumes et les élèves de cycle 4 vont créer la programmation du jeu.

Objectifs

Cette activité permet de travailler

cycle 2 le repérage sur un quadrillage ;

cycle 2 la reproduction de figure selon un modèle donné ;

cycle 4 Expliquer, formuler les déplacements de la souris, le repérage dans le plan ;

Les compétences visées

cycle 2 Reproduire une figure sur un quadrillage selon un modèle donné ;

cycle 4 Représenter l'espace, se repérer dans un plan

cycle 4 Communiquer :

- Expliquer à l'oral son raisonnement (déplacements de la souris, le repérage dans le plan)
- Vérifier la validité d'une information (ce que je dis est-ce clair pour un pair?)

Les attendus de fin de cycle

cycle 2

cycle 4 Représenter l'espace, se repérer dans un plan

Déroulement

Phase 1

Cycle 2 Les élèves travaillent sur papier le repérage et le dessin sur quadrillage par coloriage de mosaïques.

Cycle 4 Les costumes sont proposés aux élèves de cycle 4. Ils doivent ouvrir un projet sur Scratch et les reproduire en temps libre. Le projets sont envoyés à l'enseignant par ENT ou mail (selon l'établissement) pour validation.

Phase 2

Les élèves de cycle 2 et cycle 4 sont regroupés en salle informatique. Les élèves travaillent en binôme. Un élève de cycle 2 avec un de cycle 4.

Chaque élève de cycle 2 ouvre un compte et dessine les 2 premiers costumes avec l'aide de son binôme de cycle 4.

Cette séance peut être dissociée avec pour les élèves rapides, un plus grand nombre de costumes dessinés. Pour les plus en difficulté, les costumes de la balle, puis du blocs et du défenseur peuvent être un moyen de commencer le dessin sur quadrillage plus facilement.

Phase 3

Cycle 2 Les élèves de cycle 2 dessinent les costumes 3 et 4.

Cycle 4 Les élèves de cycle 4 se lancent dans la programmation du jeu. Les séances sont détaillées dans les pages suivantes.

Phase 4

Cycle 2 Les élèves de cycle 2 dessinent la balle, le bloc et le défenseur.

Cycle 4 Les élèves programment les actions des lutins.

Phase 5

Regroupement pour tester le jeu.



Attention

Le plateau mesure 480 px de large sur 380 pixel de hauteur. Il est subdivisé en carrés de 4 × 4 pixels. Le repérage se fait par ces carrés. Le zoom maximal est de 1600%. Cela signifie qu'on grossit la figure 16 fois. Pour réaliser un dessin précis, il faudra utiliser le zoom maximal. L'épaisseur du pinceau se règle en bas à gauche du plateau de dessin avec un curseur.

Applications

Consigne : Réaliser les costumes. Il faut positionner le centre du costume sur la croix.

Conseil : Utiliser le pinceau avec la taille 2 × 2 pixels.



Remarque

Les costumes 2 et 4 sont deux costumes d'un même lutin.

<p>costume 1</p>	<p>costume 2</p>	
<p>costume 3</p>	<p>costume 4</p>	
<p>Balle</p>	<p>Bloc</p>	<p>Défenseur</p>

Cette activité se déroule sur un certain nombre de séances ou sur un EPI. Certaines actions sont délicates et peuvent être traitées dans le cours de mathématiques.

Attendus visés

connaissances et compétences associées travaillées

- Elaborer un projet étapes par étapes
- Travailler les nombres premiers et le pgcd (tirs séquencés)
- Se repérer dans une grille (création des lutins)
- Utiliser des coordonnées (placements des lutins sur le plateau de jeu)
- Introduire les équations de droites (défenseur)
- Introduire la notions de suites pour les plus performants (**Hors programme**)
- Travailler le calcul d'aire (dessin des blocs)
- Revoir les grandeurs quotient avec la vitesse des envahisseurs synchronisée lors du premier déplacement de droite à gauche puis de gauche à droite.
- Penser à un lutin variable pour gérer la hauteur

Déroulement en séance

1. Création des lutins

- Envahisseur 1 : 2 costumes
- Envahisseur 2 : 2 costumes
- Envahisseur 3 : 2 costumes
- Bloc : 1 costume
- Balle : 1 costume
- Défenseur : 1 costume

2. Positionnement d'un envahisseur

- Commencer par un mouvement horizontal sur un même axe
- Introduire une variable pour permettre au lutin de descendre de 10 px lorsque le bord est touché

3. Positionnement un second lutin et synchroniser leurs mouvements.

4. Positionnement les autres envahisseurs et synchroniser leurs mouvements.

5. Mouvement du défenseur, de la balle et d'un envahisseur sur chaque ligne

Les fiches élèves proposent la création de tous les lutins mais il est possible de dissocier les travaux.

Création des lutins

Pour créer les lutins, il faut prendre la croix comme référence car c'est le point de coordonnées (0;0) dans le plan de création des costumes.

Positionnement

Blocs et du défenseur

On peut positionner les éléments :

les blocs les coordonnées $(-135; -130)$, $(-45; -130)$, $(45; -130)$, $(135; -130)$

le défenseur est mobile sur un axe horizontal. Son ordonnée peut être $y = -160$. On pourra parler de droite d'équation $y = -160$.

Envahisseurs - Invaders

On peut jouer avec 5 lignes comprenant sur chacune 8 envahisseurs – 3 formes d’envahisseurs. Le premier envahisseur de la cohorte a pour coordonnées $(-150;30)$. Les 40 autres ont pour coordonnées $(-150 + 40n; 30 + 30k)$, k et n deux entiers tels que $n \in [0; 7]$ et $k \in [0; 4]$.

Blocs, défenseur et envahisseurs

On peut positionner les éléments comme le montre l’image



On peut jouer avec 5 lignes comprenant sur chacune 8 envahisseurs – 3 formes d’envahisseurs. Le premier envahisseur de la cohorte a pour coordonnées $(-150;30)$. Les 40 autres ont pour coordonnées $(-150 + 40n; 30 + 30k)$, k et n deux entiers tels que $n \in [0; 7]$ et $k \in [0; 4]$.

Prolongements

On peut proposer de coder plusieurs niveaux de jeux. Dans ce cas, c’est la vitesse de la balle qui augmente. Lorsque toutes les briques sont cassées, il faut les faire réapparaître toutes en même temps.

Il est intéressant de commencer par une balle qui rebondit sur les bords du jeu.

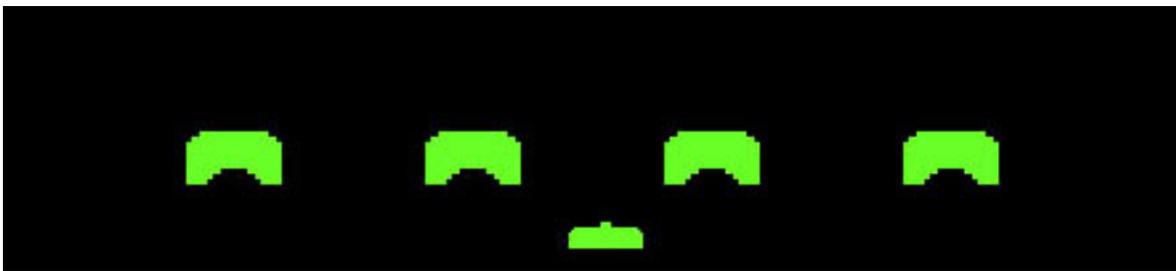
Le jeu est ici : **space_invader.sb2**. Il est inachevé dans la conception des lutins. Mais il fonctionne pour les premiers lutins. Un bon exercice est de le finir pour se l’approprier!

Position des blocs et du défenseur

Consigne :

1. Ouvrir le projet;
2. Créer 5 costume d'un même lutin bloc. Chaque costume doit avoir un bout en moins pour donner l'impression qu'il se détruit;
3. Dupliquer le bloc pour obtenir 4 blocs;
4. Positionner les blocs en respectant des distances égales entre chaque bloc;
5. Positionner le défenseur.

Résultat attendu.



Remarque

Un bloc a une largeur de 36 px. Le plateau mesure 480 px de largeur.
Les blocs sont posés sur une droite déterminée par $y = -135$.

Notion mathématique. Équation de droite

On dit que $y = -135$ est une équation de la droite.
Les équations de droites sont de la forme $y = ax + b$ ou $x = k$ (droites "verticales").



Aide

Ce jeu, ô combien célèbre, est sans doute un des premiers jeux vidéo!

Les envahisseurs se déplacent horizontalement en gardant le même espacement et la même vitesse. Lorsque la colonne du bord touche le bord, les lignes d'envahisseurs descendent de 10 px. lorsque au moins un envahisseur arrive proche des blocs, le joueur a perdu. On pourra placer la limite à -100 px, et placer le premier envahisseur à $(-150; 30)$. Le placement des autres se déduit.

Les envahisseurs tirent des obus. On pourra penser à tirer un obus guidé au début du tir par la position du défenseur. Si l'obus touche le défenseur, la partie s'arrête. Si l'obus touche un bloc, le bloc change de costume. Au bout de 5 touches, le bloc disparaît.

Le défenseur tire sur les envahisseurs pour les empêcher de descendre. Le défenseur gagne lorsque que tous les envahisseurs ont disparu. Lorsque la balle du défenseur touche un envahisseur, il disparaît.

Space invader : Séance 2

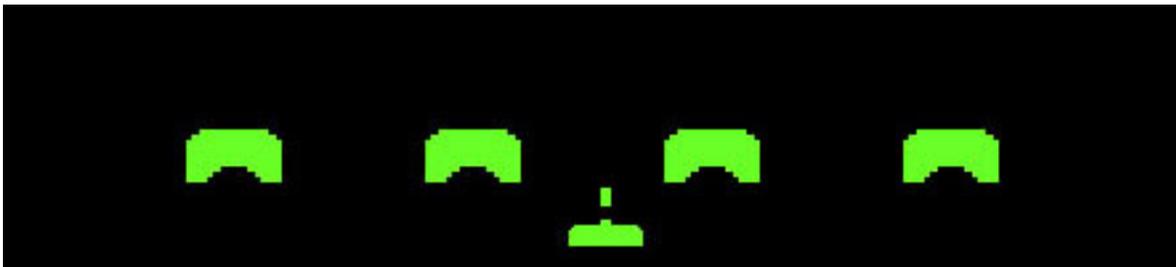
Déplacement du défenseur

Consigne :

1. Ouvrir le projet;
2. Actionner le déplacement du défenseur avec la souris **ou** avec les touches du clavier;
3. Actionner la balle avec la barre Espace

Le tir doit toujours démarrer du canon du défenseur.

Résultat attendu.



Bloc utiles

Suivre le pointeur de la souris	
Aller à	
Répéter indéfiniment	
Déplacer vers le haut	 

Étudier ce bloc

```
quand espace est pressé
  s'orienter à 90
  aller à x: souris x y: -155
  glisser en 1 secondes à x: souris x y: 200
```

Position et action d'un envahisseur

Consigne :

1. Positionner l'envahisseur du coin bas à gauche. (voir les positions sur la fiche suivante)
2. Actionner le déplacement de cet invader.
Lorsque l'invader touche le bord du plateau, il part en sens inverse et descend de 5 px.
Chaque seconde, le lutin doit changer de costume.



Aide

- Il doit avoir 8 invaders par ligne.
- Les intervalles sont de même longueur.
- La largeur du plateau est 480 px. Un invader mesure 20px.



Situation à programmer

- | Quelle est la position d'un Invader sachant que son ordonnée est $y = 20$?

Notion mathématique. Repère

Un repère est une grille qui possède une origine et qui permet de déterminer la position de tous les points du plan. Ici, l'origine est au centre du plateau.

Notion mathématique. Coordonnées

Les coordonnées d'un point A dans un repère sont un couple de nombres (a, b) qui permettent de positionner le point A . a est appelé **l'abscisse** du point A et b est **l'ordonnée** du point A .

Tester le code

```
when clicked
  go to x: -30 y: hauteur
  point in direction -90
  glide 2 secs to x: -110 y: hauteur
  repeat until hauteur < -100
    go to x: -110 y: hauteur
    point in direction 90
    glide 4 secs to x: 60 y: hauteur
    go to x: 60 y: hauteur
    point in direction -90
    glide 4 secs to x: -110 y: hauteur
```

Aide

Rappel : Le lutin des 2 lignes les plus basses sont constitués de deux costumes  et 

- Créer la variable hauteur
- Assigner à hauteur la valeur 20.
- Modifier les valeur de x pour le déplacement du lutin.
- Lorsque le lutin arrive au bord du plateau, il doit repartir dans le sens opposé avec une ordonnée de -5 par rapport à la ligne précédente.
- hauteur prend la valeur hauteur -5 . On utilisera la brique  avec la variable hauteur et -5 comme valeur.
- Pour changer de costume à un lutin, on utilise la brique 

Action de la balle défenseur des invaders

Consigne :

1. Lorsque la balle touche un bloc
le bloc change de costume.
la balle disparaît. La balle doit réapparaître pour un nouveau tir.
2. Lorsque la balle touche un invader
l'invader disparaît.
la balle disparaît. La balle doit réapparaître pour un nouveau tir.

Aide

```
si Sprite5 touché? alors
  jouer le son pop
  cacher
  ajouter à bloc4 1
  attendre 1 secondes
  montrer
```

il convient de modifier le nom du lutin.

Position des invaders

L'Invader de base est créé.

Consigne.

1. Dupliquer le **en modifiant les coordonnées de placement de son clone**.
2. Faire déplacer les deux invaders de la même façon.
3. Lorsque les 2 invaders se déplacent en escadrille, dupliquer autant de fois que nécessaire le lutin et changer ses coordonnées de placement.

Le déplacement de tous les envahisseurs est identique. Cela donne l'impression de cohorte.



Aide

Pour modifier les coordonnées du clone, aller dans le script du clone.

Les invaders se positionnent comme ci-dessous.



Aide

Pour passer un script d'un lutin à un autre, on peut utiliser le sac à dos. Le sac à dos est en dessous de la partie programmation.

Pour ouvrir le sac à dos, cliquer sur la flèche pointant vers le haut.

Pour placer un script dans le sac à dos, cliquer sur le script et glisser le dans le sac à dos.

Sélectionner un autre lutin.

Prendre le script dans le sac à dos et le déposer dans la partie script.

Space invader : Séance 6

Tir de balle ennemi

Fiche Élève

Consigne :

Passage de la navette

Toutes les 15 secondes, une navette passe tout en haut du plateau de jeu et lâche une bombe à une position aléatoire.

Consigne :

1. Dessiner la navette
2. Programmer le lâcher de la bombe



Étymologie

aléatoire vient du mot latin *alea* qui signifie hasard. On retrouve ce mot dans *alea jacta est!*, une expression dite par Jules Cesar et qui se traduit par *le hasard est jeté* mais que l'on entend par les dés sont jetés.