

| | | | | | | | | | | | | |
|---|--|---|---|------|--|------|---|------|--|------|---|--|
|  | <h1>Séance pratique</h1> <p>Année 2023-2024</p> |  | | | | | | | | | | |
| <p>Cycle 4</p> | <h2>Un premier jeu de « Pong »</h2> | <p>Matériel : PC, Scratch</p> | | | | | | | | | | |
| <p>Niveau 2/3</p> | <p>Notions travaillées : algorithmique, tests conditionnels</p> | <p>Pré-requis : bases de Scratch</p> | | | | | | | | | | |
| | <p>Compétences du socle associées s :</p> <table border="1" style="width: 100%;"> <tr> <td>TECH</td> <td>1.4 : Participer à l'organisation et au déroulement de projets.</td> </tr> <tr> <td>TECH</td> <td>3.3 : Présenter à l'oral et à l'aide de supports numériques multimédia des solutions techniques au moment de leur réalisation.</td> </tr> <tr> <td>TECH</td> <td>4.2 : Appliquer les principes élémentaires de l'algorithmique et du codage à la résolution d'un problème.</td> </tr> <tr> <td>TECH</td> <td>5.1 : Simuler numériquement la structure et/ou le comportement d'un objet.</td> </tr> <tr> <td>TECH</td> <td>5.2 : Organiser, structurer et stocker des ressources numériques.</td> </tr> </table> | TECH | 1.4 : Participer à l'organisation et au déroulement de projets. | TECH | 3.3 : Présenter à l'oral et à l'aide de supports numériques multimédia des solutions techniques au moment de leur réalisation. | TECH | 4.2 : Appliquer les principes élémentaires de l'algorithmique et du codage à la résolution d'un problème. | TECH | 5.1 : Simuler numériquement la structure et/ou le comportement d'un objet. | TECH | 5.2 : Organiser, structurer et stocker des ressources numériques. | |
| TECH | 1.4 : Participer à l'organisation et au déroulement de projets. | | | | | | | | | | | |
| TECH | 3.3 : Présenter à l'oral et à l'aide de supports numériques multimédia des solutions techniques au moment de leur réalisation. | | | | | | | | | | | |
| TECH | 4.2 : Appliquer les principes élémentaires de l'algorithmique et du codage à la résolution d'un problème. | | | | | | | | | | | |
| TECH | 5.1 : Simuler numériquement la structure et/ou le comportement d'un objet. | | | | | | | | | | | |
| TECH | 5.2 : Organiser, structurer et stocker des ressources numériques. | | | | | | | | | | | |

But du jeu :

Le but du jeu est de relancer une balle rebondissante avec la raquette. Chaque fois que la balle touche la raquette, on gagne un point ; si on rate le rebond de la balle...on a perdu !

C'est un programme pour débuter où l'on va se simplifier la vie en laissant Scratch gérer les rebonds grâce à la fonction « **Rebondir si le bord est atteint** », y compris sur la raquette ! On évite ainsi de gérer le calcul de l'angle de redépart après le rebond...ça va beaucoup nous simplifier la tâche. Nous utiliserons mBlock pour programmer, éventuellement Scratch...mais les noms de commandes peuvent varier un peu



Démonstration vidéo sur <https://bit.ly/3yVX3F6>

1-Crée ton arrière-plan et tes lutins

- Arrière-plan : Tu peux choisir ton arrière-plan comme moi parmi les fonds proposés par Scratch, c'est le plus rapide.
- Lutin « balle » : supprime le lutin initial (chat ou panda), puis clique sur « nouveau lutin » et choisis une balle en bibliothèque. Nomme ce lutin « balle ».
- Réduis ta balle celle de bibliothèque étant beaucoup trop grosse
- Dessine un autre lutin qui sera le lutin « raquette ».
 - Tu peux simplement dessiner un rectangle, dimensionne-le puis nomme ce nouveau lutin « raquette ».

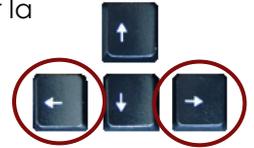
Voilà, tu as créé ton fond d'écran, et tes deux lutins (balle et raquette), il ne reste plus qu'à écrire les scripts qui leur correspondent.

2-Crée le script du lutin « raquette »

Nous allons utiliser les deux flèches de direction droite et gauche pour déplacer la raquette.

On voudrait aussi que la raquette soit au centre de la scène, en bas, au démarrage.

Quand le drapeau vert est cliqué, remets la raquette à sa position initiale (au centre de l'écran, tout en bas). Tu utiliseras cette commande (à toi de définir x et y)



allez à x: 0 y: 10

- On utilisera les flèches de droite et de gauche pour déplacer la raquette, par exemple comme ça (x est l'axe horizontal) :



- Pour les autres directions...c'est exactement le même principe

PS : N'oublie pas qu'il faut répéter tout ça infiniment...

3-Crée le script du lutin « balle » et fais-la rebondir

Clique sur le lutin « balle » pour ouvrir le script de celui-ci.

- Quand le drapeau vert est cliqué, remets-la balle au centre en haut de l'écran grâce à la commande (...choisis les valeurs de x et y pour ça).

allez à x: 0 y: 10

- On la fait avancer dans la direction du moment en utilisant la commande :

bouger de 10 pas

- Pour que ta balle continue d'avancer toujours, il te faudra inclure ton code dans une boucle « Répéter infiniment » ou « Pour toujours » avec mBlock :



- Pour que la balle rebondisse sur les bords, il te suffira d'inclure dans ta boucle la commande :

rebondir si le bord est atteint

Tu as maintenant une raquette que tu peux déplacer de gauche à droite, tu as une balle qui part du haut de l'écran et rebondit sur les bord ...il ne reste plus qu'à détecter si ta balle rebondit sur la raquette ou pas..

Remarque : tu dois constater que la balle rebondit toujours de la même façon et dans la même direction...pas drôle 😞. Pour éviter ça, donne à ta balle un petit angle au départ (avant la boucle « Pour toujours »...)

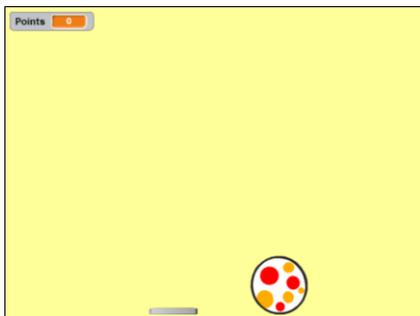
tourner de 23 degrés

Tu as terminé cette partie ? Demande la suite au prof...

3 – Détecter si la balle touche la raquette ou pas

La raquette se déplace, la balle rebondit, mais que tu la touches ou pas cela ne change rien pour l'instant. Tu vas donc maintenant chercher à détecter si, quand la balle est en bas, elle touche ou pas la raquette.

- Si la balle est en bas et que raquette est touchée, la balle repart vers le haut, le jeu continue ;
- Si la balle est en bas et que la raquette n'est pas touchée, le jeu s'arrête car tu n'as pas relancé la balle



C'est perdu 😞



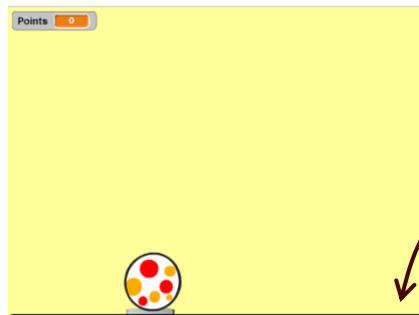
C'est gagné 😊

Il faudra donc vérifier **deux** conditions pour que la balle reparte :

- que tu touches la balle avec la raquette **ET** que la balle soit à ce moment-là en bas de la fenêtre

Idée !

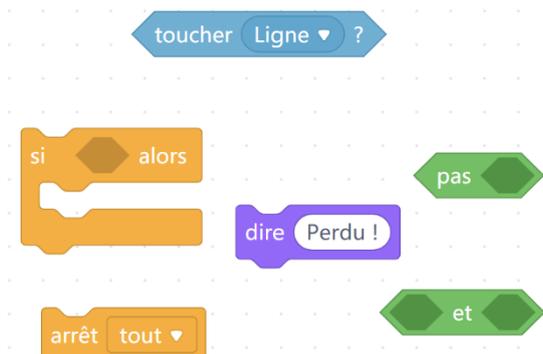
Tu vas créer un lutin qui sera une simple ligne en bas de l'écran :



Lutin en forme de **ligne**

Si tu touches la ligne **ET** la raquette, **alors** le jeu continue, la balle repart et rebondit

Si tu touches la ligne **ET** que tu ne touches **pas** la raquette, tu affiches « Perdu ! » et tu arrêtes le jeu. Tu as donc besoin des blocs suivants :



4- Compter les points

Si tu touches la balle avec la raquette, tu marques un point ; si tu la rates, le jeu s'arrête...

Tu auras besoin de créer ici une **variable** « Points ». Une « variable » est une grandeur qui varie (si !) : le temps évolue d'1s seconde toutes les secondes (si !), et tu marques un point à chaque fois que tu renvoies la balle (promis !)

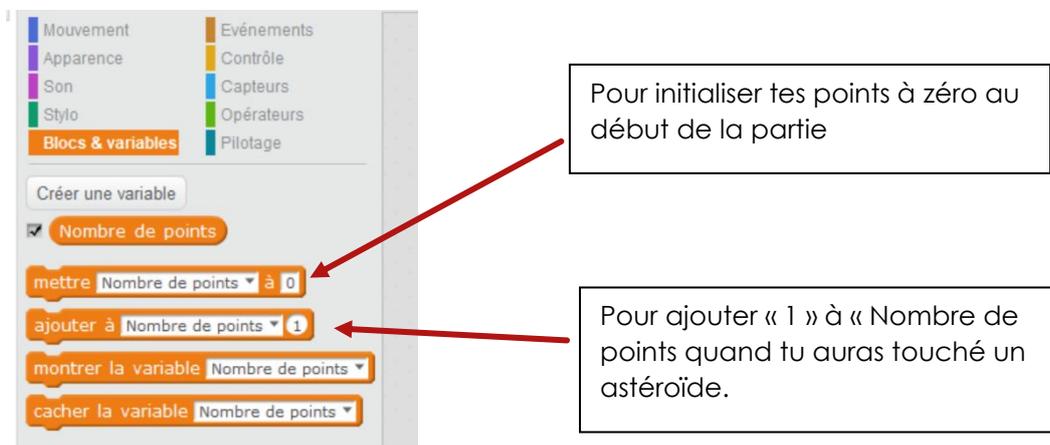
La grandeurs « Nombre de points » n'existant pas dans Scratch, c'est à toi de la **créer**, de l'**initialiser** et de la modifier quand il le faudra.

Clique sur la palette « **Blocs et variables** » puis sur « **Créer une variable** » et appelle ta première variable « **Nombre de points** » :



Tu verras alors apparaître de nouvelles commandes associées à ta nouvelle variable :

Tu pourras insérer ces nouvelles commandes dans tes scripts.



On peut créer un score qui augmente de 1 quand la balle rebondi sur la raquette

1. Dans le script du lutin « balle » Crée une variable « Nombre de points » et mets-là à 0 au tout début de ton script.
2. Ajoute 1 à ta variable dans la condition « si la raquette est touchée alors... ».

Question-bonus : comment faire varier progressivement la vitesse au fur et à mesure que tu marques des points ? (...pense à utiliser ta variable « Nombre de points » par exemple)