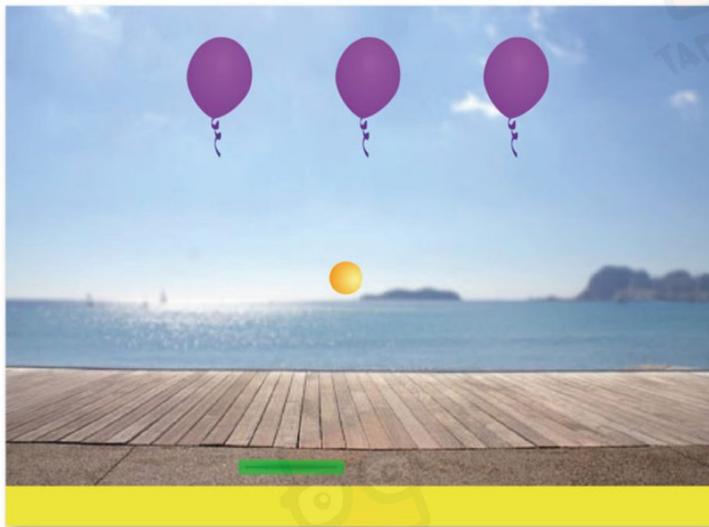


## 2 Programmer un jeu d'action

# Éclate-ballons

Projet  
ceinture verte

- instructions conditionnelles
- boucles
- capteurs
- variables
- opérateurs



**But du jeu :** Le joueur doit faire rebondir la balle sur la raquette verte pour faire éclater les trois ballons. Il perd si la balle touche la bande jaune située sous la raquette.

### Étape 1 L'arrière-plan et les lutins *Balloon*



Un rectangle jaune doit apparaître en bas de la scène. Les trois ballons violets doivent être symétriques par rapport à l'axe des ordonnées.

- À quoi sert le rectangle jaune ? Pourquoi a-t-on choisi cette couleur ?

.....

.....

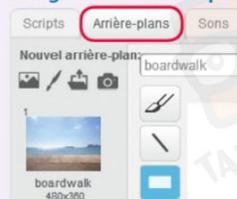
.....

.....

.....

#### Coup de pouce

Pour dessiner la bande jaune, tu peux utiliser la palette graphique dans l'onglet des arrière-plans.



- Que peut-on dire des coordonnées du centre des ballons de gauche et de droite ?

### Étape 2 Les lutins *Paddle* et *Ball*



Le lutin *Paddle* doit être dirigé horizontalement à l'aide de la souris, en bas de la scène, au-dessus du rectangle jaune. Le lutin *Ball* doit être orienté de façon aléatoire au début du jeu, avancer constamment et rebondir contre les bords de la scène.

- Quelle est la coordonnée de *Paddle* qui ne doit pas varier au cours de l'exécution du programme ?

.....

- Quel type de boucle permet d'assurer le déplacement de *Paddle* tout au long du jeu ?

.....

- Le lutin *Ball* devrait-il éviter certaines orientations au départ ? Pourquoi ?

.....



في دارك... اتمنون علي قرابتك واصفارك



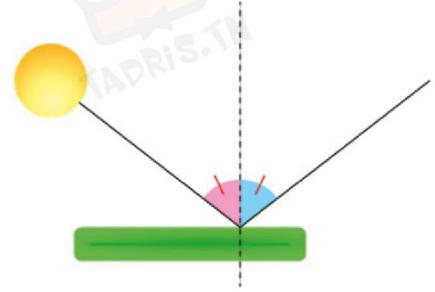
### Étape 3 Interactions entre lutins



La balle doit rebondir de façon symétrique lorsqu'elle touche la raquette. Le ballon doit disparaître quand il est touché par la balle. Pour suggérer que le ballon éclate, on doit entendre un son.

#### Coup de pouce

Tu peux utiliser la variable **direction** qui contient, à chaque instant, l'orientation du lutin.



- Comment faut-il orienter la balle lorsqu'elle rebondit sur la raquette ?

### Étape 4 Arrêt du jeu



Le jeu doit s'arrêter si la balle arrive dans la zone jaune ou si le joueur a éclaté les trois ballons.

- Quelle condition faut-il ajouter dans le script de *Ball* pour faire arrêter le jeu ?

.....

.....

.....

- Comment peut-on faire compter au programme le nombre de ballons éclatés ?

.....

.....

.....

#### Pour aller plus loin :



- Augmenter la vitesse de la balle après chaque ballon éclaté.
- Mettre en mouvement les ballons pour qu'ils se déplacent lentement sur la scène.
- Montrer un ballon seulement au début du jeu. Chaque ballon éclaté est remplacé par deux nouveaux ballons. Le joueur gagne lorsqu'il a éclaté dix ballons.



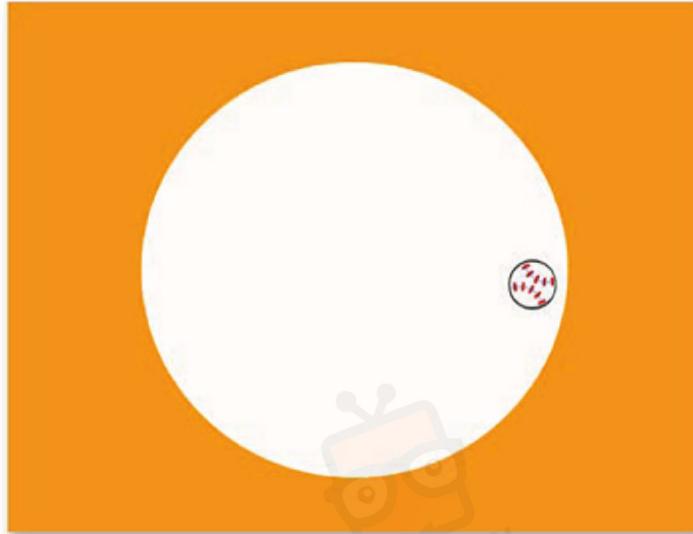
في دارك... اتمنون على قرابتك واصفارك



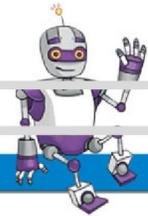
## Dans le mille !

Projet  
ceinture verte

- variables
- boucles
- instructions conditionnelles
- capteurs
- opérateurs



**But du jeu :** La balle apparait aléatoirement sur la scène à chaque essai déclenché lorsque l'utilisateur presse la touche *espace*. On souhaite savoir combien de fois la balle apparait dans la cible blanche par rapport au nombre total d'essais.



### étape 1 Programmer les essais et le comptage



La scène doit faire apparaitre le lutin *Baseball* réduit à 50 % de sa taille et l'arrière-plan *light 494x372*.

L'utilisateur doit pouvoir faire autant d'essais qu'il le souhaite en appuyant sur la touche *espace*. À chaque fois, la balle doit se placer aléatoirement sur la scène. Des variables permettent de mémoriser les résultats.

#### Coup de pouce

La scène est un rectangle de 480 pixels de large et de 360 pixels de haut.

- Quelles sont les valeurs minimales et maximales des coordonnées  $x$  et  $y$  de la balle ?

.....

- Combien de variables sont nécessaires ? Que représentent-elles ?

.....

.....

.....

.....

- Lorsque la balle apparait en dehors de la cible blanche sans la toucher, comment les valeurs des variables changent-elles ?

.....

.....

- La somme des variables qui représentent le nombre de fois où la balle apparait à l'intérieur et à l'extérieur de la cible est-elle égale au nombre d'essais ? Pourquoi ?

.....



في دارك... اتمنى على قرابتك واصفارك

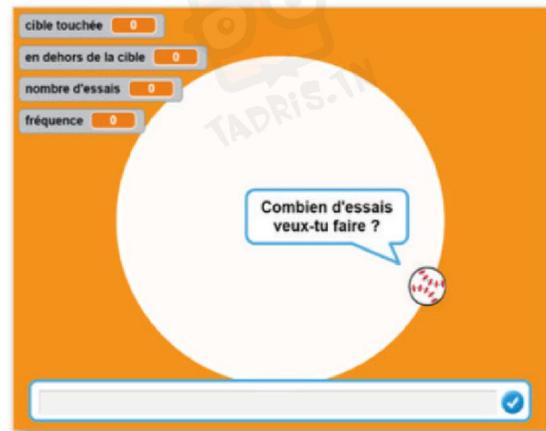
## Étape 2 Faire un grand nombre d'essais

- ↓ Pour que l'expérience soit plus rapide, on veut que l'utilisateur donne le nombre d'essais qu'il souhaite effectuer pour que le programme effectue automatiquement ces essais. Il doit ensuite afficher la fréquence de l'évènement « La balle a atteint la cible ».

### Coup de pouce

Calcul de la fréquence  $f$  en pourcentage :

$$f = \frac{\text{nombre de fois où la cible a été touchée}}{\text{nombre d'essais}} \times 100$$



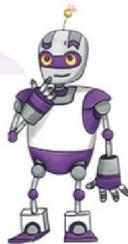
- Compléter le tableau de valeurs.

Nombre d'essais	50	100	500	1000
Nombre de fois où la cible a été touchée				

- Quelle est la fréquence de l'évènement « la balle a atteint la cible » en pourcentage pour 2 000 essais ?  
.....
- Mettre la balle à 1 % de sa taille et refaire l'expérience avec un grand nombre d'essais. Que se passe-t-il ?  
.....
- Y-a-t-il un lien entre le pourcentage de l'aire de la scène occupée par la cible blanche et la fréquence calculée par le programme ? Si oui, lequel ?  
.....  
.....  
.....

### Coup de pouce

Tu peux augmenter la vitesse d'exécution du programme en activant le « Mode Turbo » dans le menu « Édition ».



### Pour aller plus loin :

- ↓ Modifier l'arrière-plan et refaire la même expérience afin de connaître les chances que le lutin atteigne une zone précise de la scène.
- ↓ Dessiner un arrière-plan composé d'un triangle équilatéral coloré puis utiliser cette expérience pour déterminer une valeur approchée de son aire.