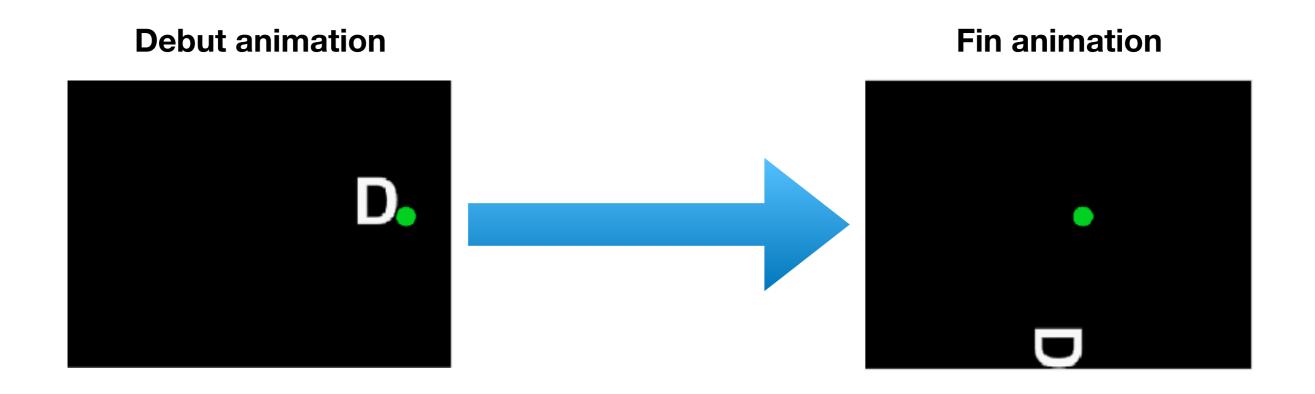


Scratch fun

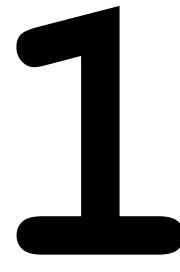
VF v1

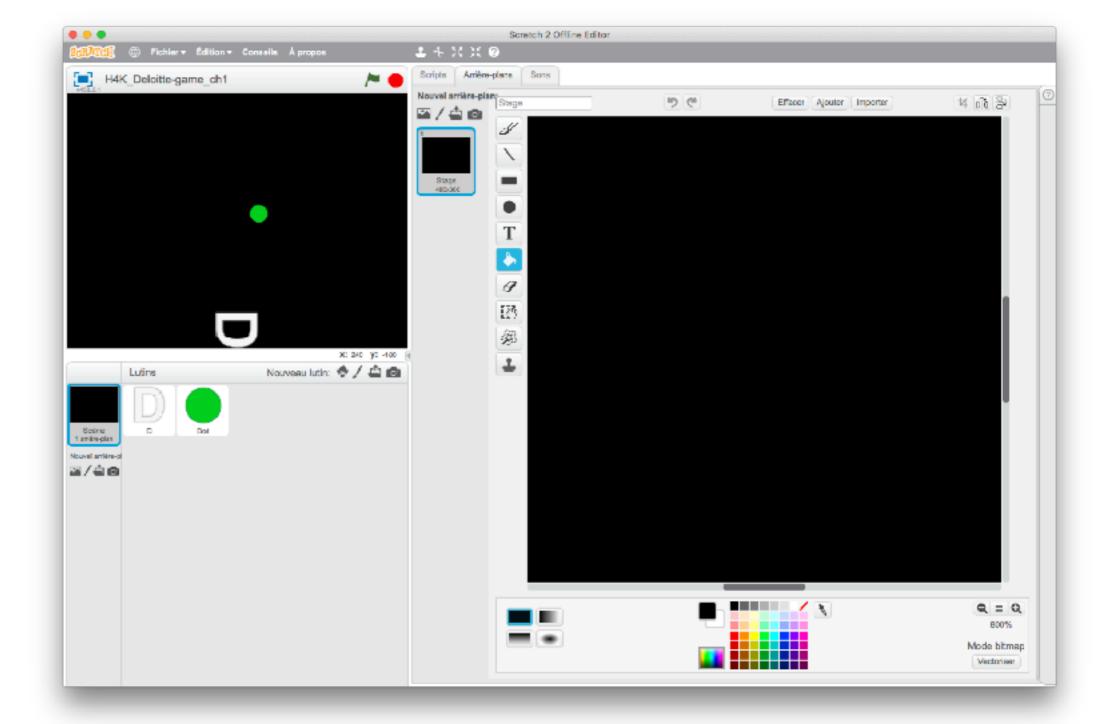
Partie 1 - Preparation et animation de base du jeu



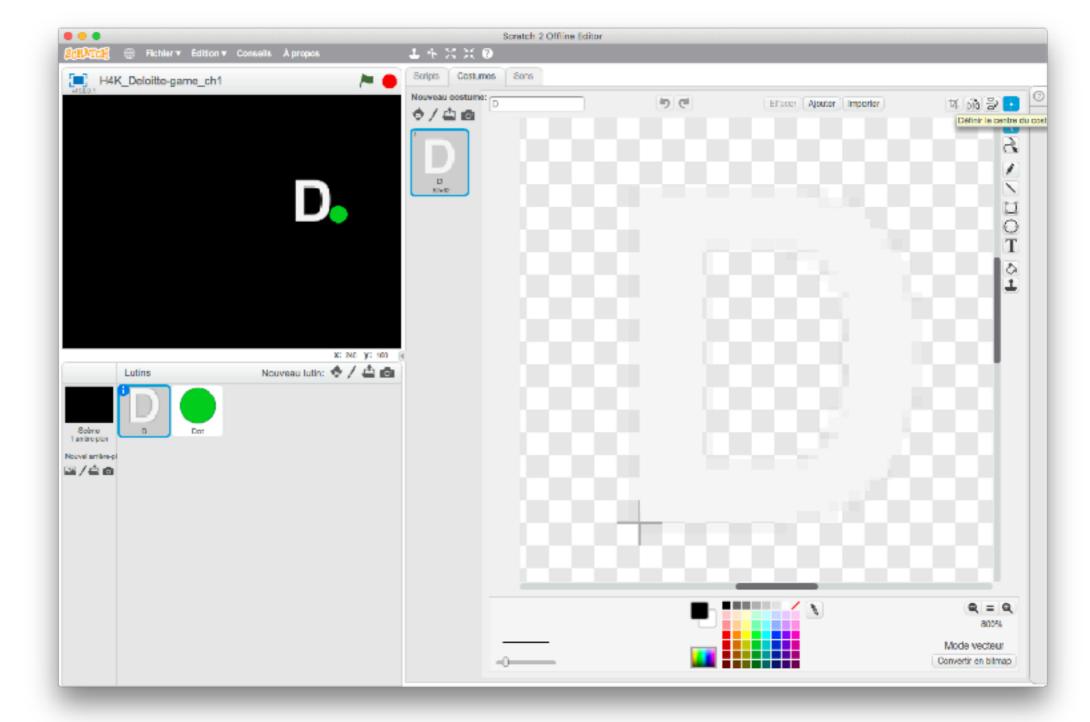


Partie Pa





L'arrière plan doit être peint en noir. Sélectionner la scène, puis le pot de peinture. Sélectionner la couleur noir, et dans l'onglet arrière plan, cliquer sur l'image. Partie 1

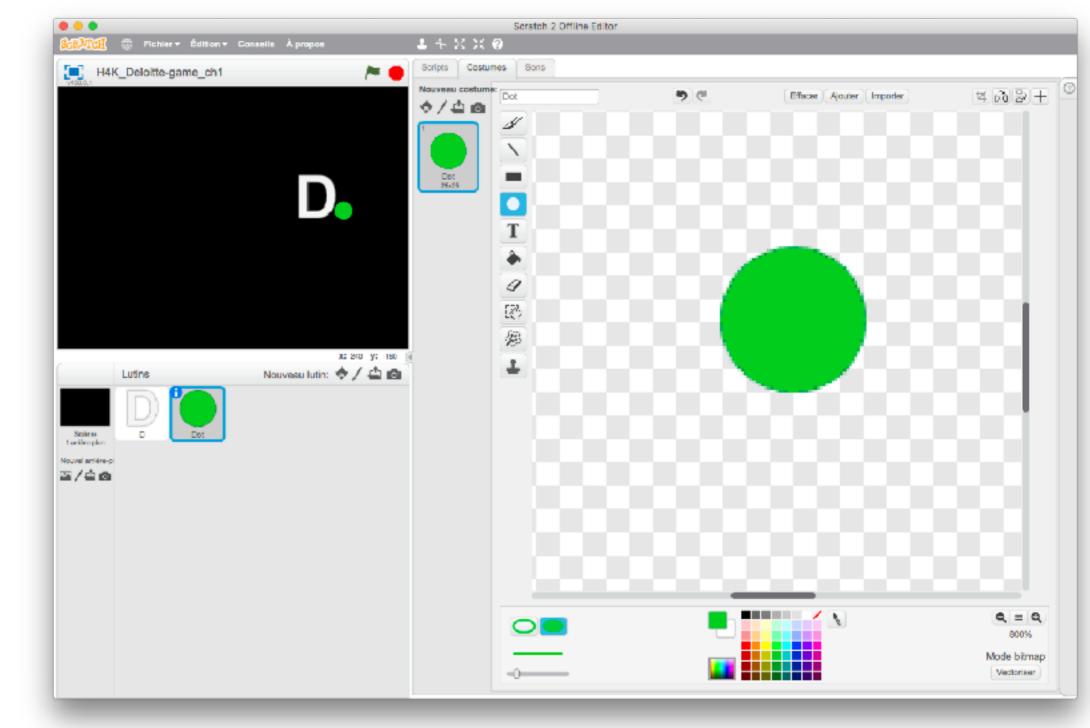


Créer un premier lutin pour la plateforme. Mettre une lettre D majuscule de couleur blanche. Définir le centre du « costume » en bas à gauche de la lettre. Attention: La dimension de la lettre devrait être de 60 pixels de hauteur.

Conseil: Le « i » sur les lutins permet de les renommer

Partie Partie Pot





Créer un deuxième lutin pour la balle (outil ellipse en appuyant sur la touche majuscule pour faire un cercle) de couleur verte. Définir le centre du « costume » au milieu de la balle. Attention: La dimension de la balle devrait être de 25 pixels de hauteur.

The control of the co

```
quand est cliqué

aller au premier plan

s'orienter à 90 v

aller à x: 170 y: 0

glisser en 1 secondes à x: -30 y: 0

glisser en 1 secondes à x: -30 y: -130

répéter 30 fois

tourner ( de 3 degrés

envoyer à tous Anim End v
```

Code de la lettre. Animation basique. On commence au premier plan, à droite de l'écran, puis la lettre se déplace au centre, et tourne sur elle meme afin de présenter une raquette (30 fois 3 degrés, soit un angle droit).

Conseil: Le message (envoyer à tous) a été créé via « événements »

Partie Partie Pot

```
Solution Effects for the Amore Solution Solution
```

```
quand est cliqué

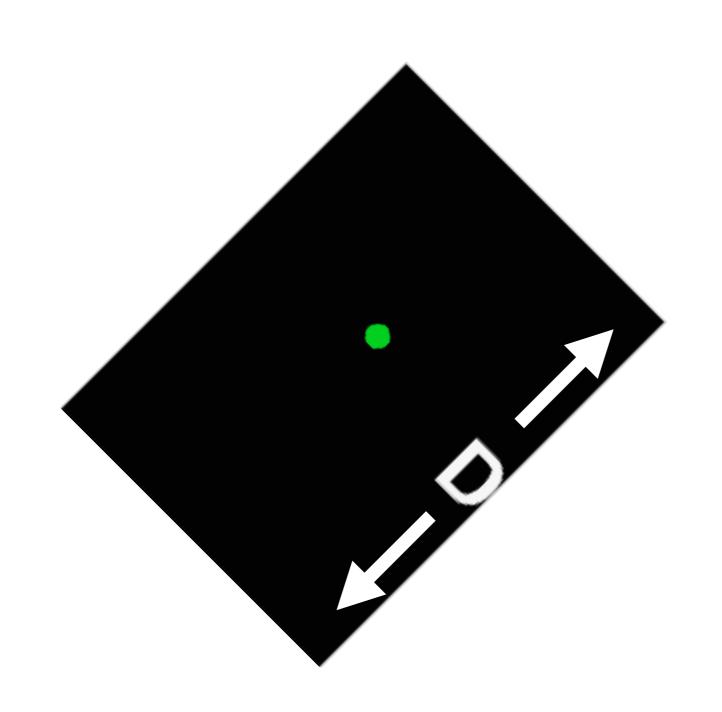
aller à x: 230 y: 10

glisser en 1 secondes à x: 30 y: 10
```

Code de la balle. Encore plus simple. Elle apparait à droite et se déplace au centre.

Conseil: Clique sur le drapeau vert pour tester ton animation

Partie 2 - Donner de la vie au jeu





```
Partie 2
Travailsur
```

```
South Contract Armon Arm
```

```
quand je reçois Anim End ▼
répéter indéfiniment

si touche flèche droite ▼ pressée? alors

ajouter 7 à x

si touche flèche gauche ▼ pressée? alors

ajouter -7 à x
```

Animation de la raquette: Elle bouge horizontalement le long du sol $(Axe\ X)$.



```
Contact Assess Arrays

Arrays
```



```
quand je reçois Anim End s'orienter à nombre aléatoire entre -30 et 30 répéter indéfiniment avancer de 10 rebondir si le bord est atteint
```

Animation de la balle. Elle va tout droit, et rebondit contre les murs.

```
Partie 2

Travaile Doi

Travaile Doi
```



```
quand je reçois Anim End 
s'orienter à nombre aléatoire entre -30 et 30

répéter indéfiniment

avancer de 10

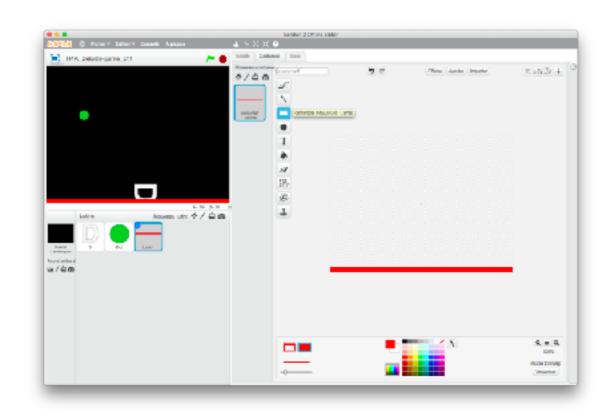
rebondir si le bord est atteint

si D ▼ touché? alors

s'orienter à nombre aléatoire entre -30 et 30
```

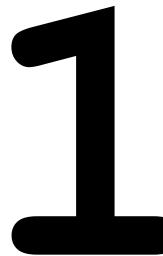
Interaction entre la raquette et la balle: si la balle touche la raquette (Nom du lutin de la raquette), elle rebondit.

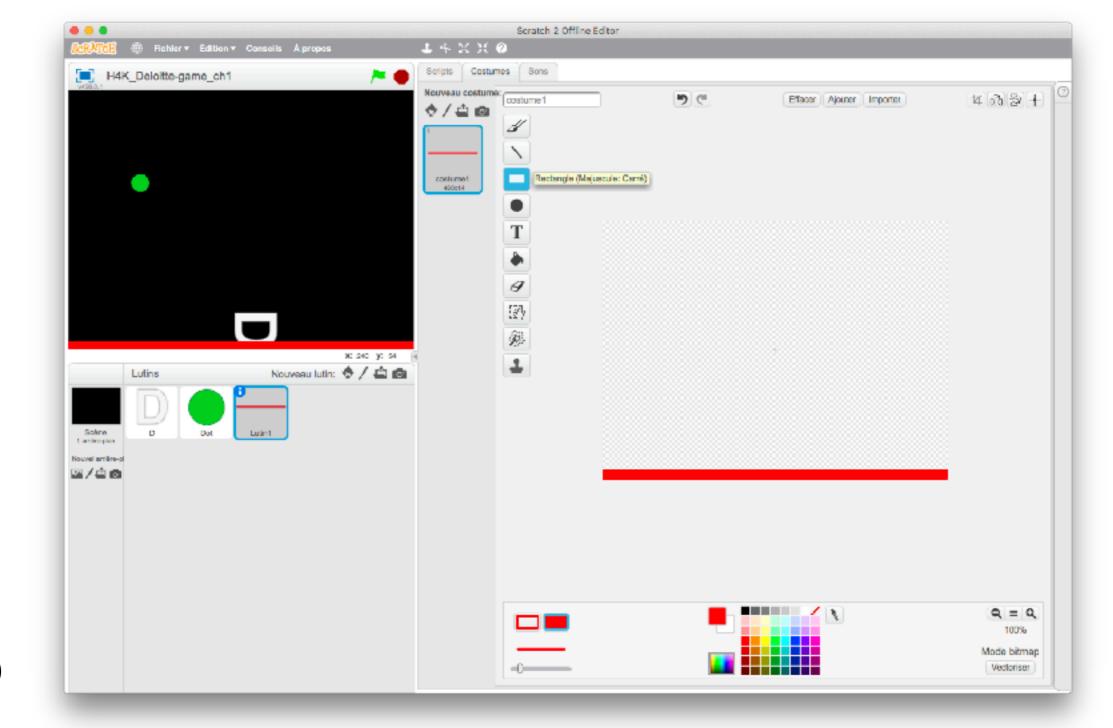
Partie 3 - Rendre le jeu plus intéressant





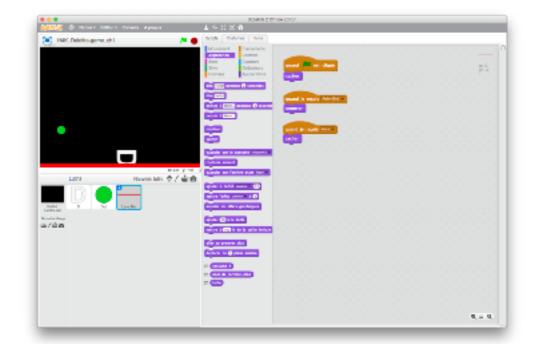
Parties
Parties
Parties
Parties
Parties
Parties





Création de la lave, un piège mortel. Dessiner un rectangle rouge sur toute la largeur du dessin de costume. Le positionner en bas via la fenêtre de jeu (cliquer et glisser).









Code pour la lave. Elle est cachée pendant l'animation de début. Elle apparait pendant le jeu. Puis elle disparait de nouveau.

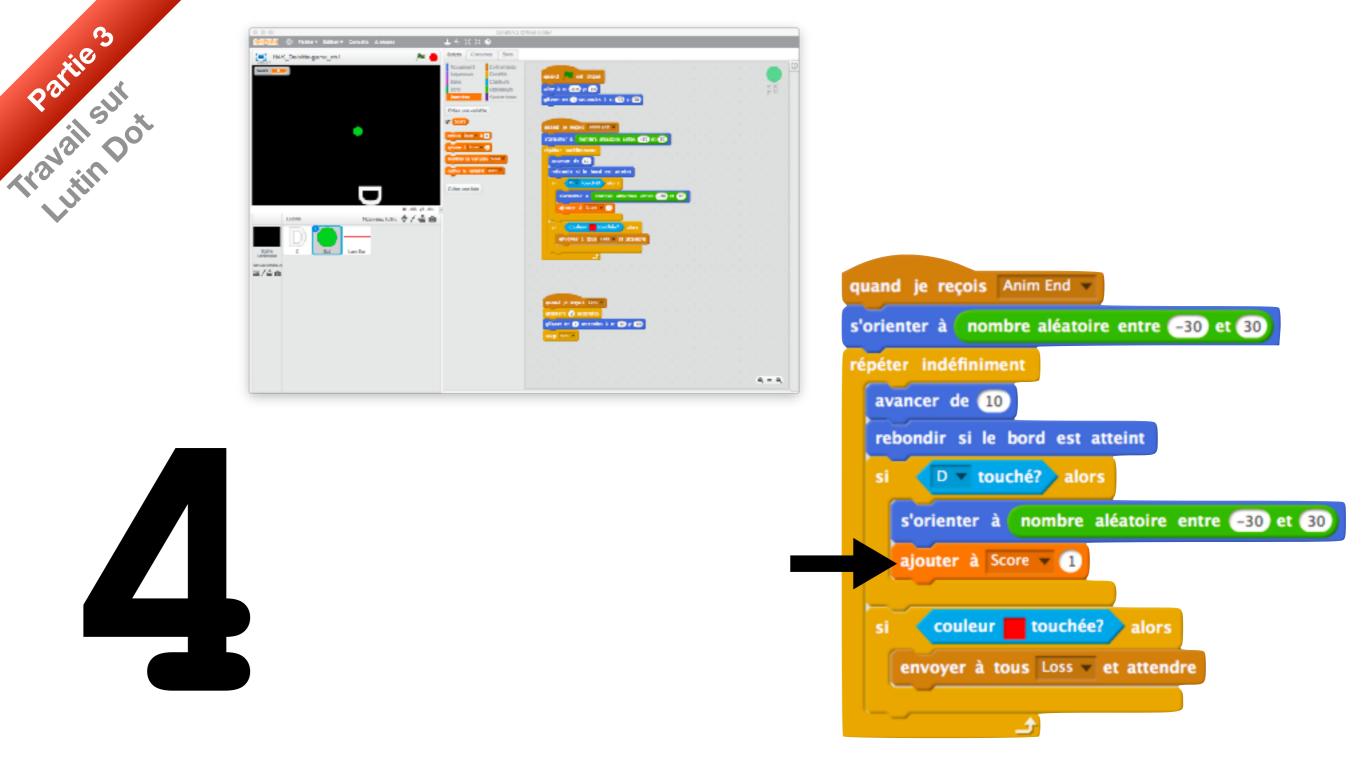
Conseil: On créé un nouveau message « Loss » qui nous sera vite utile

```
Partie 3
Travail sur Dot
```

```
Accorded Collect Edition 1 Section 2 Section 2
```

```
quand je reçois Anim End
s'orienter à nombre aléatoire entre -30 et 30
répéter indéfiniment
  avancer de 10
  rebondir si le bord est atteint
        D v touché? alors
     s'orienter à nombre aléatoire entre -30 et 30
        couleur touchée? alors
     envoyer à tous Loss et attendre
quand je reçois Loss
attendre 2 secondes
 glisser en 1 secondes à x: 30 y: 10
stop tout ▼
```

Interaction entre la balle et la lave. Si la balle touche la couleur de la lave, le programme envoie un message « Loss », et la balle après 2 secondes revient au centre.



Notre premiere variable: « Score ». A chaque fois que la balle touche la raquette, on marque un point.

Conseil: Pour créer une variable, il faut aller dans « Données », « Créer une variable »

quand est cliqué mettre Score ▼ à 0 aller au premier plan s'orienter à 90▼ aller à x: 170 y: 0 glisser en 1) secondes à x: (-30) y: 0) glisser en 1 secondes à x: -30 y: -130 répéter (30) fois tourner (4 de 3 degrés envoyer à tous Anim End 🔻 quand je reçois loss 🔻 Top Score < Score alors mettre Top Score a à Score glisser en 1 secondes à x: -30 y: 0 répéter 30 fois tourner 🔼 de ᢃ degrés

Ne pas oublier de mettre la variable à 0 au debut du jeu. On créé une deuxième variable pour tenir un high score. Enfin à la fin du jeu, le lutin « D » revient à sa place et se remet vertical (on tourne 30 fois 3 degrés dans l'autre sens).

Conseil: Le code vérifie si le « high score » est plus petit que le « score ». Si c'est le cas, un nouveau « high score » est affiché.



```
So MA Decide general formation in the second state of the second s
```

```
répéter indéfiniment

ajouter à Time 

attendre 1 secondes

si Time 

o alors

envoyer à tous Loss
```

quand est cliqué mettre Score v à 0 mettre Time v à 60 aller au premier plan s'orienter à 90▼ aller à x: 170 y: 0 glisser en 1 secondes à x: -30 y: 0 glisser en 1 secondes à x: -30 y: -130 répéter 30 fois tourner (4 de 3 degrés envoyer à tous Anim End v

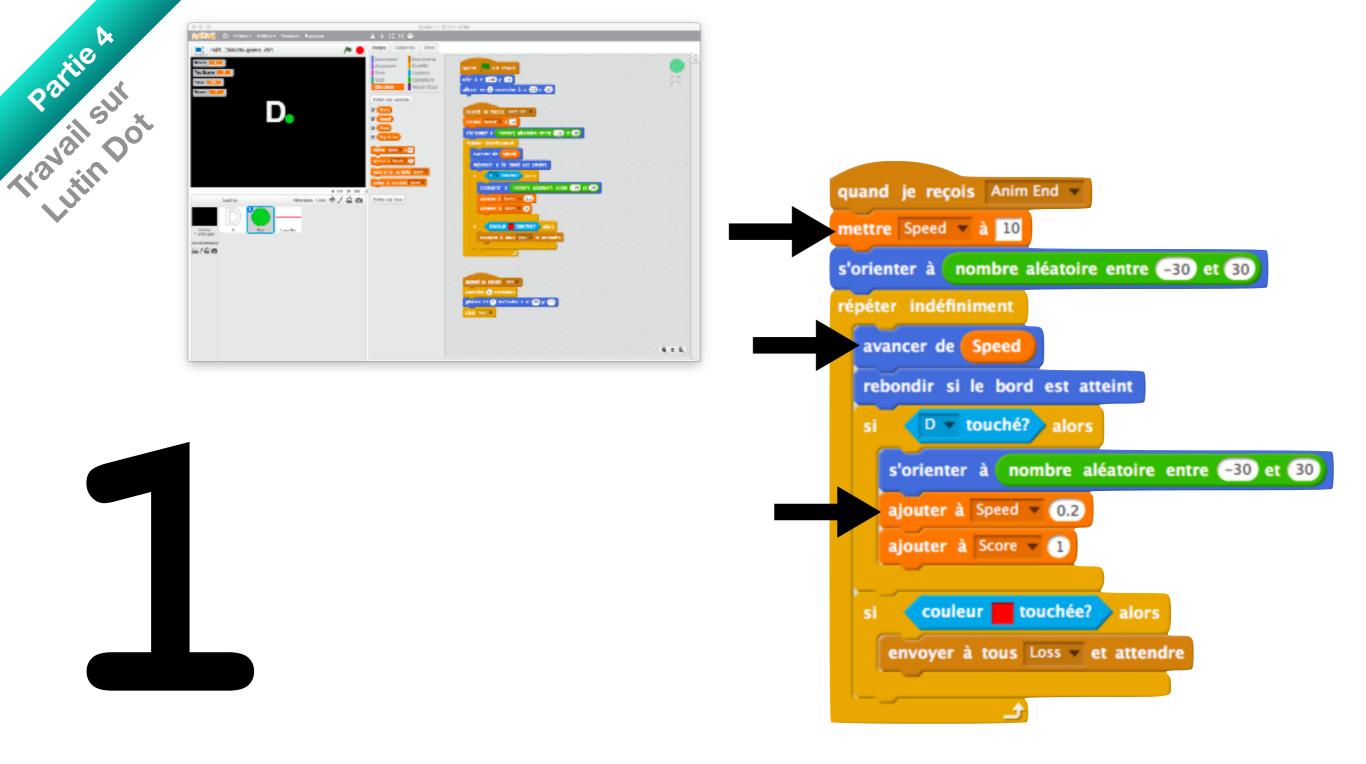
Un temps limité? Une nouvelle variable, « Time », est créée. Elle va avoir une valeur de 60, et va baisser de 1 toutes les secondes. A la fin du temps (donc si la valeur de la variable « Time » est plus petit que 0), le jeu est fini.

Partie 4 - Pour les experts...

Le jeu va se compliquer...

- 1 A chaque fois que la balle va toucher la raquette, elle va accélérer
- 2 L'endroit où la balle va toucher la raquette va impacter la direction du rebond





On va faire varier la vitesse de la balle. Au lieu d'être fixe, la vitesse de la balle devient une nouvelle variable « speed ». A chaque contact avec la raquette, elle va augmenter de 0.2 unités.

Conseil: Attention. Pour indiquer 0.2, utilise un point, et pas une virgule. Si tu ne veux pas afficher une variable, décoches-la.



```
Enter the Collection of the Co
```

```
répéter indéfiniment

mettre X-D v à abscisse x

si touche flèche droite v pressée? alors

ajouter 7 à x

si touche flèche gauche v pressée? alors

ajouter -7 à x
```

Plus compliqué... Maintenant, si la balle tape à droite de la raquette, elle rebondit vers la droite. Idem à gauche. Pour le faire, on sait que notre raquette fait 60 pixels de large. Mais il va nous falloir une nouvelle variable (« X-D »), pour savoir où est la raquette.

```
TO STATE OF THE PARTY OF THE PA
```



```
quand je reçois Anim End
mettre Speed a 10
s'orienter à nombre aléatoire entre -30 et 30
répéter indéfiniment
  avancer de Speed
  rebondir si le bord est atteint
        D v touché? et abscisse x - X-D < 30 alors
    s'orienter à nombre aléatoire entre (-30) et (1)
    ajouter à Speed v 0.2
    ajouter à Score v 1
        D v touché? et abscisse x - X-D > 30 alors
    s'orienter à nombre aléatoire entre 0 et 30
     ajouter à Speed ▼ 0.2
    ajouter à Score - 1
      couleur touchée? alors
    envoyer à tous Loss v et attendre
```

```
D ▼ touché? et abscisse x - X-D < 30
```

On sait où est la raquette grâce à la variable « X-D ». Il suffit de comparer l'endroit où la balle va taper la raquette (à droite ou à gauche) par rapport à ce point de référence.

Conseil: Courage! :-)