## Snap Séance 2

## 1 ) Les listes

##### A. Notion de liste

|  |  |
| --- | --- |
| Au cours de la séance précédente, nous avons découvert la notion de variables. Dans les variables que nous avons créées, nous avons stocké une valeur unique (éventuellement modifiée par la suite).Il existe un autre type de variable, appelé liste. Une liste peut être comparée à un tiroir qui aurait plusieurs compartiments. | tir |

##### *Attention*, en langage de programmation Python (comme dans tous les autres langages) le premier élément de la liste a se notera a[0]. En informatique, on commence en effet à compter à partir de 0, ce qui peut être déroutant. Le langage visuel Snap! a lui choisi de compter à partir de 1.

##### B. Premier exemple

Créez le bloc :

Puis observez ce que vaut la valeur ,  puis

##### C. Jeu de questions / réponses

Créez un script faisant ceci : (indications au verso)

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

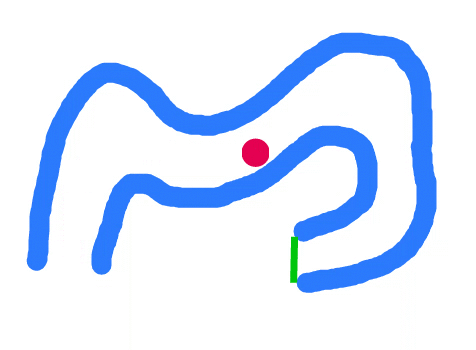
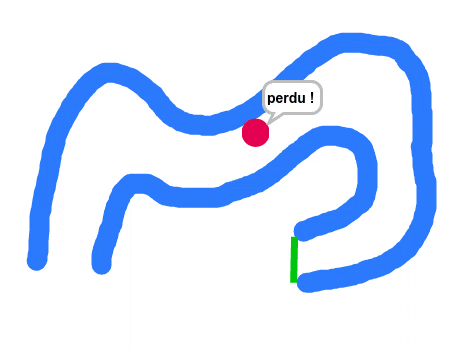
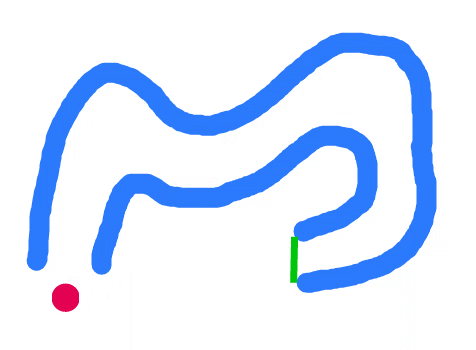
* Pour afficher le personnage, cliquez sur : *Fichier* → *Costumes*
* Il faut créer deux listes : une des pays, une des capitales correspondantes.
* La question posée doit être aléatoire, donc il faut créer une variable qui prendra une valeur aléatoire.
* Le bloc permettant de renvoyer une valeur aléatoire se situe dans la catégorie verte (Opérateurs)
* Pour le test de la bonne ou mauvaise réponse, penser à la structure

Variantes : Éviter d’avoir plusieurs fois la même question, faire un système de compteur de points etc..

(Un exercice pour la fin de séance pour les plus rapides : Alonzo dit les 100 phrases ; le carré de 1 est 1, le carré de 2 est 4, …le carré de 100 est 10000, en créant au passage la liste des 100 carrés)

## 2 ) Gestion d'évènements

##### Exemple 1 : le circuit de course Objectif du jeu : piloter la balle au clavier jusqu’à la fin du parcours (trait vert) sans toucher les bords



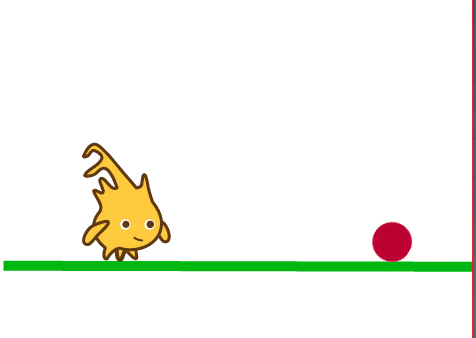
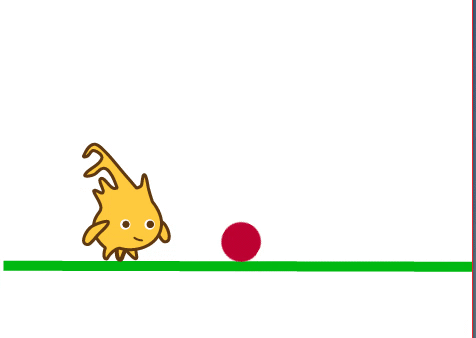
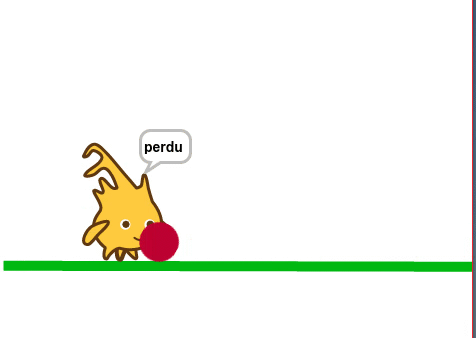
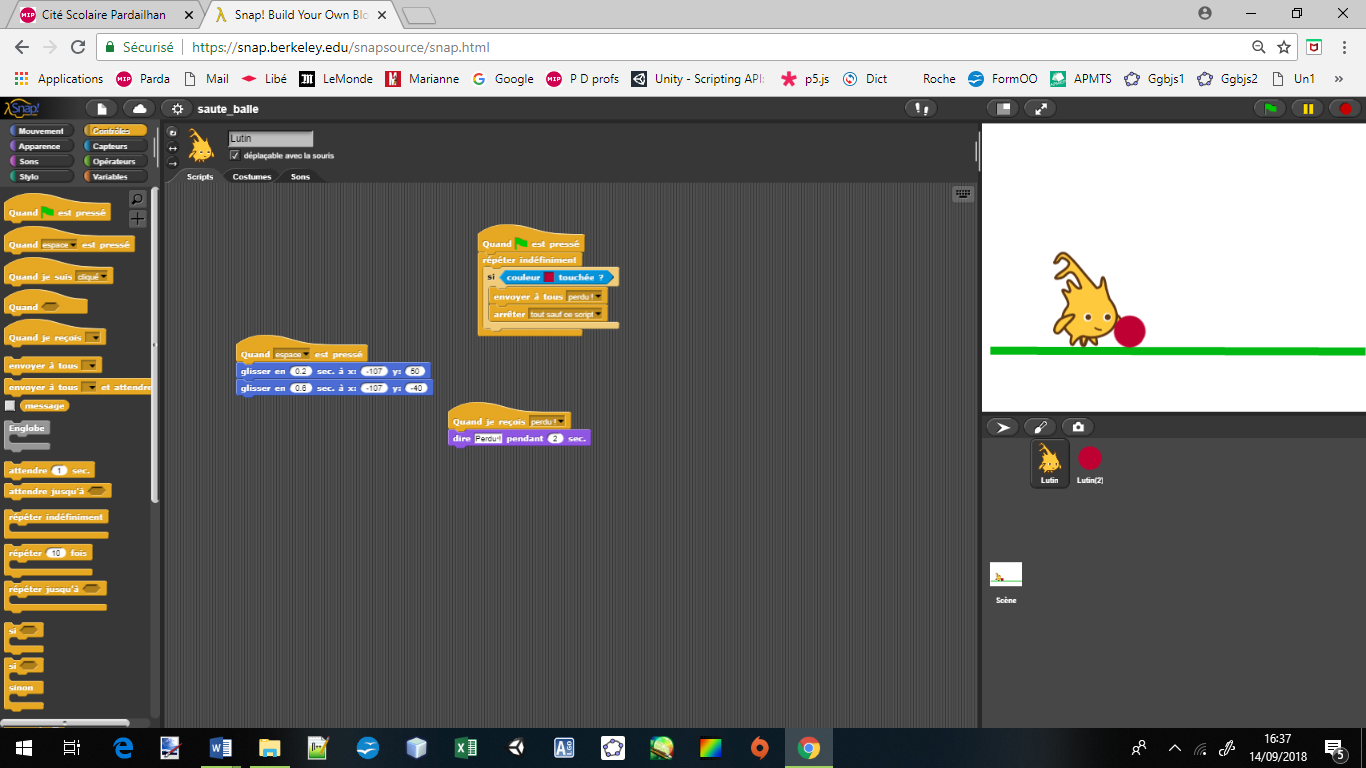
|  |  |
| --- | --- |
| **Étapes :**  1. Modifier le lutin (sprite) en cliquant sur « costumes » puis le pinceau. Dessiner une balle de couleur.  2. En cliquant sur la *scène / stage* (en bas à droite) puis sur *Background*, dessiner sur la scène un circuit (avec une barre de couleur différente à la fin) |  |

3. Faire en sorte de pouvoir piloter la balle au clavier, grâce par exemple aux blocs suivants (ou similaires)

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

##### 

##### Exemple 2 : Alonzo saute par- dessus une balle

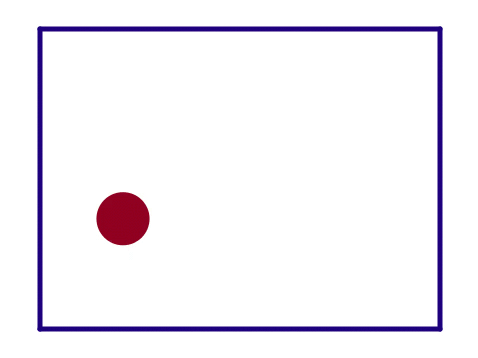
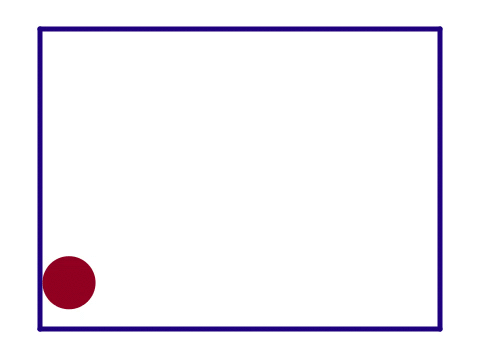
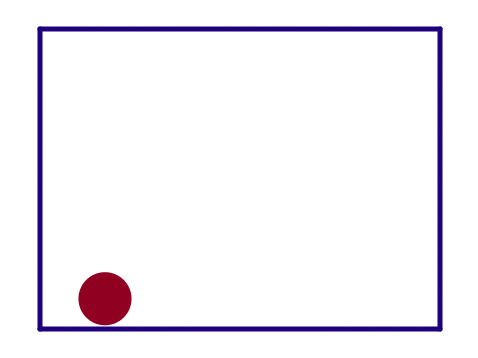


***Objectif du jeu :*** *Faire sauter le personnage avec la touche espace du clavier pour éviter la balle.*

*La commande glisser est utile aussi pour le script de la balle.*

##### 

##### Exemple 3 Mouvement d’une balle qui rebondit



**Indications :**

* Modifier votre scène pour y tracer un rectangle bleu.
* Donner à votre sprite une forme de boule
* Créer deux variables dx et dy qui correspondent au mouvement effectué par la boule toutes les 0,1 secondes (par exemple) pour donner l’effet de mouvement.
* Gérer la collision avec les bords par des tests sur l’abscisse et l’ordonnée de la boule.

|  |  |
| --- | --- |
|  | En français : dx (petite différence pour x) prend la valeur 10.  Pour que la boule rebondisse il faut remplacer dx par (-1)\*(dx), ou bien dy… quand c’est nécessaire… |

## 3 ) Communiquer entre sprites

*Il est possible dans Snap! de créer plusieurs Sprites. Chaque Sprite possède une fenêtre de scripts indépendante. Pour les faire communiquer, on utilise les blocs Envoyer/Recevoir un message*

##### Alonzo et le chien

1. Créez un deuxième Sprite qui aura le costume d’un chien.
2. Dans votre premier Sprite, créez ce script :



1. Dans le sprite du chien, créez un script le faisant apparaître lorsque le message « viens le chien-chien » est reçu. Noter que le chien a deux costumes différents.

