1. Intro: 15 minutes
   1. But et description de l’atelier
   2. Bases de scratch
   * L’environnement Scratch est divisé en 3 sections. La section du milieu est ou se trouve les blocs qui sont le code qu’on va utiliser
   * Le principe est comme des legos, les blocs se connectent l’un à l’autre. S’il n’y a pas de trou en haut, c’est un bloc de départ et s’il n’y a pas de languette en bas, c’est un bloc de fin.
   * Pour ajouter un bloc à notre code, il faut le glisser et le laisser tomber dans notre environnement
   1. Bases de programmation (while loop, for loop, if/else:si/alors)
2. Programmer: 45 minutes
3. Conclusion, feuilles de commentaires et ranger les ordis: 15 minutes

**Intro**

Bonjour tout le monde! Merci de nous avoir accueillis aujourd'hui! Mon nom est \_\_\_, c'est \_\_\_ et ceci est\_\_\_\_ et nous faisons parti de l’organisme Esprit Science. Notre but aujourd’hui et comme organisme est de vous présenter les domaines de sciences, technologie, ingénierie et mathématiques que vous ne voyez pas nécessairement dans les cours de sciences.

Aujourd'hui, vous allez programmer un jeu avec Scratch. Bien que notre jeu ne sera pas un comme Call Of Duty ou NHL 2018, mais il sera tout de même assez impressionnant pour un programmeur débutant! Qui ici à déjà entendu parler de Scratch ou l’a déjà utilisé?

(ils lèvent ou pas leur mains)

Bon, avant de commencer, on va vous expliquer les bases de Scratch et de la programmation pour vous permettre de mieux comprendre comment programmer votre jeu.

**Théorie**

Le principe de Scratch ressemble beaucoup à celui des légos, c’est-à-dire qu’on utilise des blocs qui s'emboîtent les uns dans les autres pour créer le code. S’il n’y a pas de trou en haut, c’est un bloc de départ et s’il n’y a pas de languette en bas, c’est un bloc de fin. Les blocs de départ indiquent comment le code sera commencer. L’exemple le plus courant est: lorsque le drapeau est cliqué. Pour construire le code, il suffit de glisser et déposer le bloc dans l’environnement à gauche.

L’élément le plus important à savoir pour votre code aujourd’hui et un élément extrêmement utile pour un programmeur sont les événements. Nous allons faire un exemple avec vous pour vous montrer ce que sont les événements.

**If/else loops**

Vous le savez peut-être déjà, mais en anglais les mots if et else veulent dire si et sinon. Avec Scratch, nous pouvons utiliser le block ‘’if \_\_\_, then \_\_\_’’ pour décider des actions qui vont se si quelque chose arrive.

Assignons une certaine valeur à x et à y. Disons qu’ils valent tous les deux 100. Avez-vous des suggestions sur comment peut-on faire dire au chat “X est égal à Y’’, lorsque les deux valeurs sont égales(comme dans notre situation)?

Nous allons d’abord utiliser l’opérateur égal, et y insérer les variables x et y. Maintenant, nous ouvrons **Control**, et nous allons utiliser le block **if, \_\_\_ then**. On insère le block vert x=y dans le petit espace réservé à cela. Finalement, nous ajoutons le block mauve qui fait parler le chat à l’intérieur du **if, then \_\_\_**, comme cela.

C’est l’heure de tester! i on choisi 100 et 100, et l’on appuis sur le bouton vert, voici ce que ça donne.

Maintenant, changeons la valeur de y, et regardons ce que ça fait.

Ça marche! Mais qu’est-ce qu’on fait si on veut que le chat dit ‘’X est plus grand que Y’’ lorsque c’est le cas? Nous utilisons le bloc if/else, qui va nous permettre de performer une autre action lorsque x n’est pas égal à y. Donc l’action que nous allons performer est de dire ‘’X est plus grand que Y”

Qu’est-ce que vous pensez que ca va faire? Verifions.

Malheureusement, ce n’est pas ce que l’on veut! En effet, nous n’avions pas vérifié si x aurait pu être plus petit que y; en fait, nous savons seulement si x est égal à y ou pas. Donc essayons quelque chose d’autre.

Ce qu’il faut faire c’est de rajouter une condition à notre code. Si x=y, dire “X est égale à Y”, si X<Y dire “X st plus petit que Y” et si ce X n’est ni plus petit, ni égal à Y ça veut dire qu’il est plus grand que Y. Pour faire ça, il faut inclure un autre **if** bloc. Vérifions si ça marche.

Comme on a prévu, ça marche!

Cela conclut les événements if/else, maintenant faisons les boucles for et while (pendant).

**While loops**

Une boucle ‘’for’’ est utilisé lorsque nous nous avons besoin de faire fonctionner un morceau de code exactement un certain nombre de fois. Prenons par exemple le chat. On veut le faire tourner sur soi-même une fois. Voici le code que l’on pourrait essayer

Mais lorsque l’on teste le chat, rien n’arrive! C’est parce que nous avons seulement changer sa direction de 360 degrés, donc ça va retourner à sa place initiale. Essayons, à la place, de le faire tourner de 180 degrés deux fois.

C’est déjà mieux. Et si l’on le fait tourner 4 fois de 90 degrés? 8 fois de 45 degré? Dans le fond, ça va être de mieux en mieux. Mais voici ce que le code va ressembler.

Comment pourrions nous rendre ça plus “beau’’ et plus simple? Avez-vous des suggestions?

On pourrait utiliser des boucles for! Dans scratch, vous devez sélectionner, dans la section **Control**, et le block **repeat \_\_**, et tu choisis le nombre de fois que tu veux répéter le code. Disons que l’on veut le faire répéter 8 fois, et à chaque fois on fait tourner le chat de 45 degré. Voici à quoi ça ressemble.

C’est beaucoup mieux! Et si, à la place de faire tourner le chat exactement 10 fois, on le fait tourner jusqu’à ce qu’il se rend à sa place initiale. Comment pensez vous qu’on pourrait le faire?

Essayons ça.

Ça marche! Dans le fond, une boucle while est utilisé lorsque nous avons besoin de faire fonctionner un morceau de code plusieurs fois, jusqu’à-ce que la condition donnée est atteinte. Dans notre cas, la boucle while est représenté par le bloc repeat until \_\_\_, et la condition est que le chat s’arrête de tourner lorsqu’il fait face à la droite.

**Atelier**

Bon, maintenant qu’on a fini avec la théorie, il est temps de rentrer dans les choses sérieuses. L’atelier que vous allez faire aujourd’hui consiste à programmer une variation du jeu Flappy Bird.

Qui connait le jeu ici?

Pour commencer vous pouvez ouvrir le document d’instructions. Si vous avez des questions à n’importe quel stade du jeu, n’hésitez pas de lever la main, on est là pour ça!

**Conclusion**

Juste avant de vous laisser partir pour la période Élite, nous avons quelques petits commentaires à faire. Tout d’abord, merci d’avoir participer à notre atelier, nous espérons que vous avez aimé.

Ce serait très apprécié d’avoir vos commentaires, qu’ils soient bons ou mauvais. Nous tentons toujours de s’améliorer et votre avis est super important. Vous pourrez les écrire sur cette feuille ou le dire à votre enseignant qui pourrait nous relayer le message.

Aussi, si cette activité vous a plu sachez que le monde de la programmation est immense et en essor. Les possibilités d’emplois sont énorme! J’étudie moi-même en génie informatique, qui est le mélange entre l’électronique et la programmation, et les gars sont en informatique pure. Par contre, il est important de savoir que si vous voulez devenir un programmeur de jeu vidéo pour Ubisoft ou autre compagnie, ou travaillé en informatique, vous devez suivre le parcours de sciences et mathématiques au secondaire. Vous devez donc réfléchir à tout cela lorsque vient le temps de faire vos choix de cours l’année prochaine.

Finalement, si jamais vous avez d’autres questions ou vous voulez plus d’information, n’hésitez pas de venir nous voir ou de visiter notre site web ou notre page Facebook.