Ce mémo est le récapitulatif de ce que les participants ont appris durant le stage de développement de jeu vidéo avec Guillaume, l’intervenant.

Unity est le logiciel qui a été utilisé pendant le stage. Vous pouvez le télécharger gratuitement en suivant ce lien : <https://unity3d.com/fr/get-unity/download>

Unity est également un logiciel très riche qui possède une documentation en ligne. Cette dernière permet de trouver rapidement des exemples pertinents d’utilisation de lignes de code (en anglais) : <https://docs.unity3d.com/Manual/index.html>

Vous pouvez également retrouver des vidéos pour approfondir ou revoir les notions de programmation que nous avons vu ensemble : <https://www.youtube.com/watch?v=RAZjcibFE1A&list=PLUWxWDlz8PYLKlr6F_fwCs02DH1g2hrgS>

1. Questions

1] Quel type de jeu avez-vous développé cette semaine ? Pouvez-vous citer un jeu de ce genre ?

Le jeu était un Plateformer comme Mario.

2) Quels sont les deux axes d’un jeu en 2D ?  
Ce sont les axes x et y. L’axe x est horizontal et l’axe y est vertical. En 3D, on y rajoute l’axe z, pour la profondeur.

3) Qu’est qu’un GameObject sur Unity ?  
Un **gameObject** est l’élément de base utilisé par Unity. Comme son nom l’indique, on peut le considérer comme un objet. Il possède un **transform** qui nous indique sa position, sa rotation et sa taille. On peut aussi lui donner d’autres comportements.

4) Qu’est qu’un Component sur Unity ?  
Un component est un composant que l’on peut attribuer à un **gameObject** pour lui donner un nouveau comportement ou de nouvelles propriétés. Nous avons vu par exemple le **rigidbody2D** qui permet d’ajouter un poids fictif à un **gameObject** pour le faire constamment tomber.

5) Qu’est qu’un script ?

Un script est un texte contenant du code. On peut associer du script à un **gameObject** pour lui donner le comportement qu’on aura codé.

6) Qu’est qu’une variable en programmation ?

Une variable est une information qu’on pourra utiliser dans un code. Une variable peut être un chiffre, un texte ou encore un component (exemple : **rigidbody2D**).

7) Quelle est la différence entre la fenêtre scène et game ?

La fenêtre game représente ce que le joueur verra en jeu. La scène est l’endroit où on placera nos éléments.

8) Qu’est qu’un collider et que signifie le mettre en trigger ?

Le **collider** et le **collider2D** sont des components qui permettent d’obtenir les « bords » d’un **gameObject**. De base, ces bords sont solides, mais on peut les rendre « traversables » en les mettant en trigger.

9) Qu’est qu’un canvas ?

Un **canvas** est **gameObject** particulier contenant l’interface de notre jeu. Avec Unity, le **canvas** peut s’adapter en fonction de l’écran utilisé.

1. Quelques exemples importants

1] Exemples d’une fonction que l’on peut appeler avec un Rigidbody2D :

Une fois que vous avez mis un **Rigidbody2D** dans une variable (Rappelle : public **Rigidbody2D** monRigidbody ;), vous pouvez appeler certaines fonctions comme :  
monRigidbody.AddForce(Vector3.up \* vitesseSaut) ; Ici vitesseSaut est une variable de type « float ». Cette fonction permet d’appliquer une force dans la direction du Vector3.up (à savoir une force vers le haut, sur l’axe y, up veut dire haut en angler), nous nous en sommes servie pour sauter.

2] Exemples de fonctions que l’on peut appeler avec un Transform :

A l’inverse des autres composants qui existent, comme le **Rigidbody2D** ou **le Sprite Renderer**, le composant « **Transform** » (Rappelle : **Transform** est le composant qui ne peut pas s’enlever sur un **gameObject** et qui contient les positions, les rotations, et le scale du **gameObject** sur lequel il se trouve) est directement accessible dans les scripts que l’on crée. On peut y accéder avec le mot clef « **transform** » :  
**transform**.position -> Donne la position du **gameObject** sous forme d’un vecteur (X,Y,Z || Par exemple, la position de mon avatar est de X = 22.65, Y = 0, Z = 0).

**transform**.localScale -> Donne le scale (la taille) du **gameObject** sous forme d’un vecteur (X,Y,Z || Par exemple, le scale de mon avatar est de X = 2, Y = 1, Z = 1)  
**transform**.eulerAngles -> Donne la rotation du **gameObject** sous forme d’un vecteur (X,Y,Z || Par exemple, la rotation de mon avatar est de X = 90, Y = 0, Z = 0)  
Pour attribuer une nouvelle valeur à l’une des valeurs contenu dans le **transform** (position, rotation, scale), il suffit de faire :  
transform.eulerAngles = new Vector3(0,0,0) ; Dans cette exemple, on change la rotation du **gameObject** en X = 0, Y = 0, Z = 0 ;