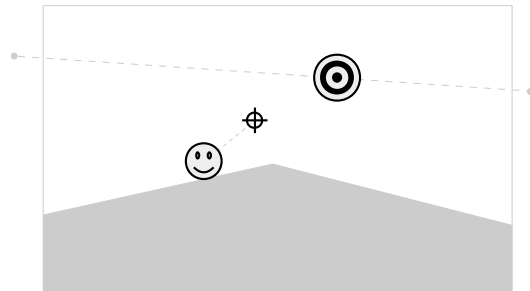


Mathématiques appliquées

Jeu de balistique

I - Règles :

Dans un jeu de balistique en « vue de côté », un personnage doit lancer une balle sur une cible mouvante.

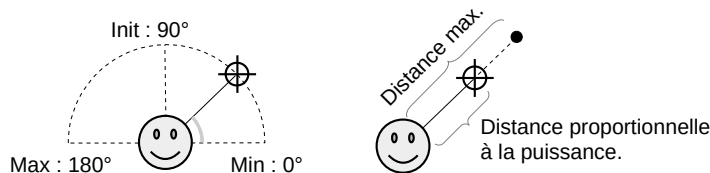


Le personnage ne bouge pas ! Seule la visée est contrôlée par le joueur. Par défaut, la visée est vers le haut (90°). Le joueur contrôle la visée avant le tir avec les touches directionnelles :

- Gauche : La visée pivote autour du personnage dans le sens anti-horaire (vers la gauche) avec un maximum de 180° .
- Droite : la visée pivote autour du personnage dans le sens horaire (vers la droite) avec un minimum de 0° .
- Haut : la puissance du tir augmente progressivement avec un maximum de 1.
- Bas : la puissance du tir diminue progressivement avec un minimum de 0.

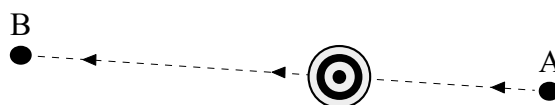
La visée est représentée par un réticule placé autour du personnage.

La puissance du tir est représentée par la distance entre le personnage et ce réticule.



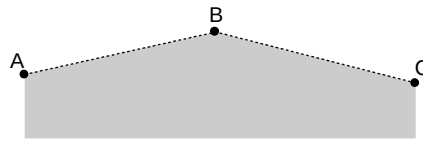
Une balle est tirée lorsque le joueur appuie sur la barre espace. La balle tirée se déplace dans la direction du tir et sa vitesse dépend de la puissance du tir.

La cible traverse l'écran de droite à gauche de manière cyclique en allant d'un point A à un point B hors écran. La vitesse de la cible est fixe et réglable en éditeur.



Lorsque le centre de la balle touche la cible, la cible est touchée et la balle est détruite.

Le sol est en forme de colline représentée par 3 points formant un sommet comme montré sur le schéma ci-dessous. La balle est détruite lorsqu'elle touche le sol.



La vitesse de la balle est affectée par deux accélérations constantes :

- La gravité (accélération verticale)
- Le vent (accélération horizontale)

II - Consignes :

Sous Unity, réaliser les règles ci-dessus en s'appuyant sur les consignes suivantes. Aucune solution ne doit utiliser de RigidBody ni de Colliders.

1- Implémenter les règles de contrôle de la visée et de la puissance. Les variations d'angle et de puissances se feront à chaque Update en fonction des inputs et avec des vitesses constantes (vitesse angulaire, vitesse de variation de puissance). Afficher la cible au bon emplacement. La cible ne doit pas être enfant du personnage dans la hiérarchie de la scène.

Utiliser « Input.GetKey(...) » avec les valeurs : « KeyCode.UpArrow », « KeyCode.DownArrow », « KeyCode.LeftArrow » et « KeyCode.RightArrow ».

Conseils :

- *Pensez aux coordonnées polaires !*
- *Les changements de valeurs se font en fonction du deltaTime*
- *N'oubliez pas de borner les valeurs.*

2- Implémenter le tir de la balle avec une vitesse constante en fonction de l'angle de visée et de la puissance de tir. Pour cette étape, ne pas prendre en compte les accélérations de gravité et de vent. Utiliser « Input.GetKeyDown(KeyCode.Space) » et GameObject.Instantiate(...).

Conseils :

- *Pensez à faire un prefab contenant le script de balle dessus.*
- *L'instanciation et l'initialisation de la balle se font depuis le script de visée réalisé en « 1 ».*
- *Pour simplifier les échanges entre le script de visée et le script de balle, ne pas hésiter à passer certaines variables publiques.*

3 – Implémenter l'animation cyclique de la cible. La vitesse de déplacement doit être réglable en éditeur. Pour rappel, la durée totale du déplacement, donc la durée de l'animation, peut être déterminée à partir de la vitesse et de la longueur à parcourir.

4 – Implémenter la collision de la balle avec la cible. Lorsque la balle touche la cible, un message est affiché et la balle est détruite.

Conseils :

- *Transmettez la référence de la cible au script de balle lors de l'instanciation de la balle.*
- *Mettre le rayon de la cible dans une variable réglable en éditeur.*
- *Afficher la cible avec OnDrawGizmos et Gizmos.DrawSphere ou Handles.DrawWireDisc.*

5 – Dans un script dédié au sol, implémenter la collision de la balle avec le sol. La balle est détruite lorsqu'elle touche le sol.

Conseils :

- *Tracer le sol avec au choix `Debug.DrawLine` ou `Gizmos.DrawLine`.*
- *Attention à ce que les trois points définissant le sol forment bien un sommet.*
- *Le sol est l'intersection de deux demi-plans.*
- *Attention à la direction des vecteurs normaux utilisée !*
- *Transmettez la référence du sol au script de balle lors de l'instanciation de la balle.*

6- Compléter les règles de balistique en appliquant la gravité et le vent. Ces deux accélérations sont constantes et appliquées à chaque Update pour modifier la vitesse de la balle.

IMPORTANT :

Affichez des informations de debug à chaque exercice pour visualiser le résultat. Pour afficher ces informations de debug, utiliser au choix :

- **Des sprites de couleur,**
- **Les méthodes `Debug.Draw[...]` dans la méthode `Update`,**
- **Les méthodes `Gizmos.Draw[...]` et `Handles.Draw[...]` dans la méthode `OnDrawGizmos`,**
- **etc.**

N'implémentez rien de plus que ce qui est demandé. Si vous avez un doute sur un choix de programmation gameplay, allez toujours au plus simple !

Les supports de cours sont autorisés. N'hésitez pas à me demander pour toute incompréhension dans l'énoncé ou dans le cours.