

Examen médian de RV01

Novembre 2020

Durée : 45 Minutes

Documents autorisés : Tous documents autorisés, accès à internet autorisé, corrigés des TDs autorisés, photocopiés de TD de SY02 et MB11 autorisés.

Le sujet est sur 20 points hors bonus. Une attention particulière sera accordée à la qualité et la clarté du code rendu, de l'arbrescence de projet et des objets dans la scène, qui seront noté sur 3 points. Le rendu doit être fait sur moodle sous forme de package Unity. Tout manquement sera sanctionné par la perte des 3 points de clarté.

On souhaite organiser un jeu révolutionnaire. Une sphère devra se déplacer dans un espace clos jusqu'à trouver son objectif. Plus elle sera proche du but, plus sa couleur évoluera vers le vert, plus elle sera loin, plus sa couleur évoluera vers le rouge. Si vous avez besoin d'une ressource, elle devrait être dans le dossier remis avec le médian.

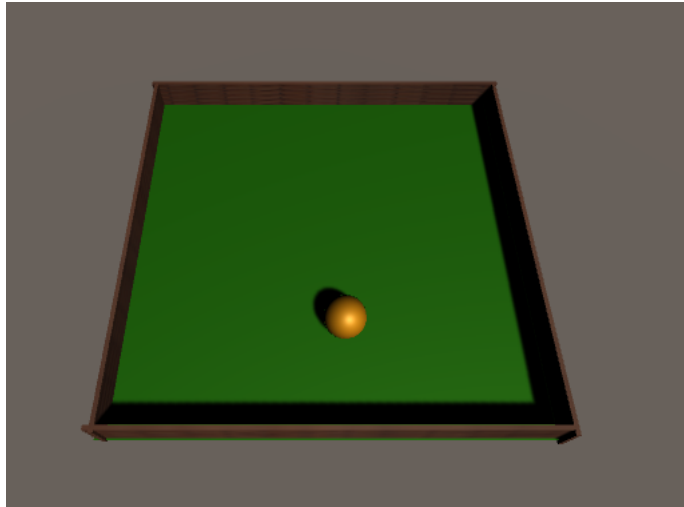


FIGURE 1 – La scène que l'on souhaite reproduire

1 Préparation des préfabriqués

1. (0.5 points) Créez le préfabriqué mur en utilisant l'objet approprié.
2. (0.5 points) Créez le préfabriqué sol en utilisant l'objet approprié.

- (0.5 points) Créez le préfabriqué boule en utilisant l'objet approprié.
- (0.5 points) Créez le préfabriqué objectif en utilisant l'objet approprié. L'objet doit être invisible et il devra déclencher des événements quand on s'approche de lui et quand on rentre dedans.

2 Ajout de la physique

- (1 point) Donnez à l'objet boule une masse de 0.1
- (1.5 points) Appliquez une force dépendant d'un paramètre force lorsque l'utilisateur appuie sur une touche (e pour haut, s pour gauche, d pour bas, et f pour droite, car oui, notre projet est vraiment révolutionnaire, il change les contrôles). Vous veillerez à ce que les développeurs suivants puissent modifier la valeur de la force appliquée.

3 Ajout des textures

- (0.5 points) Texturez le sol en vert et les murs avec les textures fournies.
- (1 point) Créez trois préfabriqués de texture : un rouge, un orange et un vert, que vous nommerez respectivement *sphere_away*, *sphere_default* et *sphere_close*. Appliquez la texture *sphere_default* à votre préfabriqué de sphère.

4 Instanciation

- (1 point) Implémentez la scène décrite en figure 1 à partir de tous les préfabriqués des questions précédentes.

5 Script

Dans cette partie, nous allons implémenter le changement de couleur. On veut avoir une texture rouge quand la sphère est loin de l'objectif, et une texture verte quand la sphère est proche de l'objectif. Le material d'un objet est accessible par `GetComponent<Renderer>().material`

- (2 points) Implémentez le fait que, quand l'objet est trop loin, la texture devient rouge. *On utilisera pour cela une fonction distance.*
- (2 points) Implémentez le changement de la texture au vert quand l'objet est trop proche. Votre code comprendra une fonction *onXEnter()*,
- (1 point) Implémentez l'arrivée à l'objectif. Quand l'utilisateur atteint l'objectif, on mettra dans un premier temps simplement un message de votre choix dans la console. Votre code comprendra une fonction *onXEnter()*.
Vérifiez que tous les cas limite sont traités par votre script.

6 GUI

On souhaite maintenant que notre jeu, en plus d'être passionnant, soit addictif. Pour cela, il faut féliciter le joueur quand il réussit, et recommencer immédiatement une partie. On le félicitera donc en affichant un message au milieu de l'écran avant de déplacer l'objectif du jeu à un endroit aléatoire. Enfin, on souhaite mettre en place un score qui augmente de 10 000 à chaque fois que l'utilisateur atteint l'objectif.

- (1 point) Désormais, quand le joueur atteint l'objectif, téléportez l'objectif à un autre endroit
- (0.5 points) Ajoutez un texte pour le score.

15. (1.5 points) Créez un script qui gèrera les actions importantes du score de ce jeu.
16. (0.5 points) Désormais, quand le joueur atteint l'objectif, ajoutez 10 000 à son score.
17. (1.5 points) Désormais, quand le joueur atteint l'objectif, il doit être félicité par un message apparaissant au centre de l'écran en vert. Votre message sera choisi aléatoirement parmi 4 messages prédéfinis.

7 Défaite du joueur (Bonus)

Quand le joueur perd, on souhaite afficher un écran de fin de partie. Cet écran de fin de partie contiendra un titre, une image, un unique bouton rejouer, et relancera automatiquement le jeu après un certain temps.

18. (1 point) Ajoutez une image, un bouton rejouer, et un texte mentionnant que le joueur a perdu.
19. (1 point) Ajoutez un script qui chargera la scène quand on clique sur le bouton
20. (0.5 points) Lancez ce script automatiquement après 10 secondes d'attente, on considèrera pour cet exercice qu'attendre 24 images équivaut à attendre une seconde
21. (1 point) Affichez le score du joueur sur l'écran de fin de jeu.