ÉPREUVE FINALE

Pondération : 50 %

Date de remise : Jeudi 12 décembre 2024 (*GitHub*)

Lien GitHibClassroom : https://classroom.github.com/a/vNWQJJ-J

Travail à effectuer : Créez un jeu qui respecte les contraintes énoncées ci-dessous. Vous pouvez utiliser ou non le thème suggéré. Si vous choisissez un autre thème, vous devez en discuter avec l'enseignant. Remettez avec votre jeu un bref document décrivant les mécaniques attendues de votre jeu.

ÉLÉMENTS DU JEU

Votre jeu doit respecter les exigences suivantes :

1. Durer entre 90 et 120 secondes;
2. Gérer le déplacement d'objets;
3. Gérer des collisions (entre des objets et dans une zone);
4. Gérer les contrôles de l'utilisateur;
5. Avoir des éléments de UI (dont 2 contrôles de saisie);
6. Proposer au moins une animation créée par vous (incluant deux clips et un contrôleur au minimum);
7. Intégrer une musique et un effet sonore;
8. Fournir des indications expliquant comment jouer au jeu (qui demeurent accessibles tout au long du jeu);
9. Un document d'analyse qui présente le jeu;
10. Une gestion de l'éclairage adéquat pour votre type de jeu.

Également, votre code doit inclure :

1. La gestion de deux événements (*event* ou *UnityEvent*);
2. L'utilisation d'une coroutine (la coroutine qui contient seulement une instruction de *Wait*… n'est pas acceptable);
3. L'intégration d'un ou l'autre de ces concepts :
    a. Une machine à état composée d'au moins 3 états;
    b. L'utilisation de 2 ScriptableObjects avec chacun au moins 2 instances.

*Intégrer la machine à états **et** les Scriptable objects ne vous donne pas de points bonis, mais contrairement à mon habitude, je vais vous donner les points pour la **meilleure des deux intégrations**.*

EMPRUNT D'ASSETS ET DE CODE

Si vous empruntez des éléments, vous devriez organiser votre projet de la façon suivante :

- Assets (créé par Unity)
    - emprunte
        - Animations
        - Prefabs
        - Scripts
        - … (autres éléments standards)
    - cree
        - Animations
        - Prefabs
        - Scripts
        - … (autres éléments standards)

Les éléments dans « emprunté » ne sont pas corrigés, mais peuvent être utilisés pour générer l'expérience de jeu. Par exemple, un script dans « emprunté » ne donne pas de point. Un son dans le dossier « emprunté » qui est appelé dans un AudioSource ajouté sur un G*ameObject* réalisé par vous est tout à fait adéquat, car l'élément évalué est l'intégration des sons et non leur création.

Tout élément dans « cree » qui n'est pas de votre production pourra être considéré comme du plagiat et entraîner les sanctions prévues au règlement.

THÈMES SUGGÉRÉS

JEU SEUL – JEU DE DÉFENSE DE TOUR

Créer un jeu où une série d'ennemis procède vers un objectif. Si trop d'entre eux atteignent le bout du trajet, la partie est perdue. Les actions de la personne joueuse sont de construire des tours et de les améliorer. La destruction des ennemis par les tours génère des ressources.

JEU EN ÉQUIPE DE 2 – MINI RPG

En équipe de 2, vous pouvez faire un mini-RPG composé de deux scènes : un village où vous recevez une ou plusieurs quêtes et un donjon où vous réalisez ses quêtes. **ATTENTION : comme il s'agit d'une épreuve finale individuelle, chaque élément demandé doit être réalisé indépendamment par chaque membre de l'équipe. Dans votre structure de projet, vous avez trois dossiers : emprunté, fait par 1 et fait par 2.**

SIMULATION DE MÉTÉORES

Simulez une pluie de météorites sur une ville et l'effet de destruction engendré. Vos modèles n'ont pas à être nécessairement tous détruits (à la façon du pot de fleurs). Prévoyez des paramètres à votre pluie de météores pour augmenter la rejouabilité. Il faudrait pouvoir simuler de 3 à 5 pluies de météores.

AUTRES THÈMES

Vous pouvez utiliser un autre thème libre qui doit être vérifié auprès de l'enseignant.

GRILLE D'ÉVALUATION

Critère	Indicateur	Échelle descriptive			
		Excellent	**Bien**	**En progression**	**Insuffisant**
Description **juste** et **pertinente** du système	Rédaction d'un document d'analyse **précis** Présence d'indications de jeu **claires**	Le document d'analyse est **complet** et son contenu est **précis** et **pertinent**. Des indications **claires** expliquant **tous** les contrôles sont intégrées au jeu. Les indications sont **accessibles** tout au long du jeu. Le code est **systématiquement** documenté de façon **adéquate**.	Le document d'analyse est **complet** et son contenu est **pertinent** et **généralement précis**. Des indications **claires** expliquant **tous** les contrôles sont intégrées au jeu. Le code est **systématiquement** documenté de façon **pertinente**.	Le document d'analyse est **complet** et son contenu est **pertinent**. Des indications **généralement claires** expliquant **tous** les contrôles sont intégrées au jeu. Le code est **systématiquement** documenté.	Le document d'analyse est **incomplet** et son contenu manque de pertinence. Des indications pour **certains** ou **aucun** des contrôles sont intégrées au jeu. Le code est **rarement** documenté.
Pondération : 10 points		**9-10**	**8**	**6-7**	**0-5**
Utilisation **adéquate** d'un moteur de jeu 3D	Utilisation **adéquate** des composantes de gestion de la physique	Les simulations physiques ont un aspect **réaliste** [1]. Elles sont accomplies en utilisant les composants de façon **adéquate**. Les scripts qui utilisent ces composants le font de façon **judicieuse** et **correcte**.	Les simulations physiques ont un aspect **généralement réaliste** [1]. Elles sont accomplies en utilisant **la majorité** des composants de façon **adéquate**. Les scripts qui utilisent ces composants le font de façon **correcte**.	Les simulations physiques ont un aspect **généralement réaliste** [1]. Elles sont accomplies en utilisant **certains** composants de façon **adéquate**. Les scripts qui utilisent ces composants le font de façon **correcte**.	Les simulations physiques ont un aspect **manque de réalisme** [1]. Elles sont accomplies en utilisant les composants de façon **inadéquate**. Les scripts qui utilisent ces composantes le font de façon **erronée**.
	Utilisation **adéquate** des fonctionnalités de la boucle de jeu	Les traitements des données sont effectués à l'étape **adéquate** de la boucle de jeu.	Les traitements des données sont **presque toujours** effectués à l'étape **adéquate** de la boucle de jeu.	Les traitements des données sont **souvent** effectués à l'étape **adéquate** de la boucle de jeu.	Les traitements des données sont **rarement** effectués à l'étape **adéquate** de la boucle de jeu.
	Interaction **adéquate** avec la personne joueuse	Le choix des contrôles est **pertinent.** Ils sont en **lien** avec l'effet produit ou ils sont **standards**. Le jeu réagit **correctement** aux contrôles.	Le choix des contrôles est **généralement pertinent**. Ils sont **souvent** en **lien** avec l'effet produit ou ils sont **standards**. Le jeu réagit **correctement** aux contrôles.	Le choix des contrôles manque de **pertinence**. Le jeu réagit correctement **à la plupart des** contrôles.	Le choix des contrôles rend le jeu **difficile d'utilisation**. Le jeu réagit **rarement** ou de façon **inattendue** aux contrôles [4].
	Utilisation **adéquate** de l'algèbre vectorielle pour déplacer des objets	Les déplacements ont un aspect **réaliste** [1]. Les fonctionnalités du moteur de jeu sont utilisées **adéquatement**.	Les fonctionnalités du moteur de jeu sont **généralement** utilisées **adéquatement**.	Les déplacements ont un aspect **généralement réaliste** [1]. Les fonctionnalités du moteur de jeu sont utilisées **correctement**.	Les déplacements ont un aspect **peu réaliste** [1]. Les fonctionnalités du moteur de jeu sont **peu** utilisées ou le sont de façon **incorrecte**.
	Configuration **correcte** des éclairages	Les éclairages sont configurés **adéquatement** avec un aspect **réaliste** [1].		Les éclairages sont configurés **correctement** avec un aspect **parfois réaliste** [1].	Les éclairages ne sont pas configurés **correctement**.
Pondération : 30 points		**27-30**	**22-26**	**18-21**	**0-17**

Intégration **correcte** d'éléments visuels et sonores	Développement **correct** d'une interface utilisateur	L'interface utilisateur affiche les informations **pertinentes**. Elle réagit **en temps réel** aux changements dans le jeu.	L'interface utilisateur affiche **la plupart** des informations **sur le contexte du jeu**. Elle réagit **souvent en temps réel** aux changements dans le jeu.	L'interface utilisateur affiche des informations **sur le contexte du jeu**. Elle réagit **souvent en temps réel** aux changements dans le jeu.	L'interface utilisateur **peu** ou **aucune** information. Elle réagit **rarement en temps réel** aux changements dans le jeu.
	Intégration **correcte** d'animations	Les animations ont un aspect **réaliste** [1]. Elle est **correctement** synchronisée avec le jeu. Les contrôleurs d'animation sont **correctement** utilisés et configurés.	Les animations ont un aspect **réaliste** [1]. L'animation est **approximativement** synchronisée avec le jeu. Les contrôleurs d'animation sont **correctement** utilisés.	Les animations ont un aspect **généralement réaliste** [1]. Elle est **approximativement** synchronisée avec le jeu. Les contrôleurs d'animation sont **correctement** utilisés.	Les animations **manquent** de réalisme [1]. L'animation **n'est pas** synchronisée avec le jeu. Les contrôleurs d'animation sont **incorrectement** utilisés.
	Intégration **correcte** d'éléments sonores	L'équilibrage des sons est réalisé **adéquatement**. La projection des sons est **correctement** synchronisée avec le jeu.		L'équilibrage des sons est réalisé **correctement**. La projection des sons est **approximativement** synchronisée avec le jeu.	L'équilibrage des sons est réalisé **incorrectement**. La projection des sons **n'est pas** synchronisée avec le jeu.
Pondération : 25 points		**23-25**	**18-22**	**15-17**	**0-14**
Application **adéquate** des bonnes pratiques en matière de développement de jeu	Respect **systématique** des bonnes pratiques orientées objet	Les bonnes pratiques de programmation orientée objet en C# sont **systématiquement** appliquées [2].	Les bonnes pratiques de programmation orientée objet en C# sont **presque toujours** appliquées [2].	Les bonnes pratiques de programmation orientée objet en C# sont **généralement** appliquées [2].	Les bonnes pratiques de programmation orientée objet en C# sont **parfois** appliquées [2].
	Organisation **correcte** du projet	Le projet est organisé **conformément** aux pratiques standard [3].		Le projet est organisé **en grande partie** en respectant les pratiques standard [3].	Le projet est organisé en respectant **certaines** des pratiques standard [3].
	Utilisation **judicieuse** d'un concept avancé en programmation de jeu (machine à états ou *ScriptableObjects*)	Une machine à états **pertinente** est **correctement** intégrée au jeu.		Une machine à états est **correctement** intégrée au jeu.	Une machine à états est **incorrectement** intégrée au jeu ou celle-ci comporte un nombre d'états **insuffisants**.
		Plusieurs données sont gérées de façon **pertinente** par des *ScriptableObject* **correctement** intégrés au jeu.		**Certaines** données sont gérées par des *ScriptableObject* **correctement** intégrés au jeu.	Les données sont gérées par des *ScriptableObject* **incorrectement** intégrés au jeu ou aucune donnée n'est générée par des *ScriptableObjets*.
Pondération : 35 points		**31-35**	**26-30**	**21-25**	**0-20**

Tout élément non réalisé ou absent du jeu se voit affecter la cote « insuffisant »

Critères à correction négative	**Critères d'évaluation**		
Non respect à la durée du jeu	**Moins de 90 secondes**	**Moins de 60 secondes**	**Moins de 30 secondes**
	-2	**- 5**	**- 20**
Qualité de la langue	Présence de **quelques** erreurs dans les documents d'analyse ou les commentaires.	Présence de **nombreuses** erreurs dans les documents d'analyse ou les commentaires. Présence d'erreurs dans les **interfaces du jeu**.	
	de -1 à -4	**de -5 à -10**	

Définition de termes :

Adéquat : l'élément choisi est pertinent dans son contexte et rempli sans fonctionnalité. De plus, il est intégré de façon correcte.

Correct : intégrer de façon à permettre un fonctionnement sans erreur.

Judicieux : l'élément est sélectionné de façon à tirer profit de façon efficace des fonctionnalités du moteur de jeu ou de la situation.

Notes :

[1] : un aspect est réaliste, s'il est suffisamment proche de la réalité pour que l'on comprenne de quoi il s'agit sans explication. Il est possible d'approximer plusieurs phénomènes. Un test que vous pouvez faire pour vous assurer de respecter ce critère est de faire jouer quelques collègues au jeu et regarder s'ils comprennent vos représentations.

[2] : inclut de façon non limitative : la division du code en méthodes, le respect de la longueur des méthodes, les modificateurs de visibilité, les standards de nomenclature. Bref, tout ce que vous faites habituellement en C#. Au besoin, référez-vous à la documentation officielle d'Unity ou de C#.

[3] : se référer aux consignes et au projet en démonstration.

[4] : un élément important de réagir de « façon attendue » est la reproductibilité du résultat. Utiliser le même contrôle dans les mêmes circonstances produira le même résultat.