GROUPE 8

BILAN UNITY

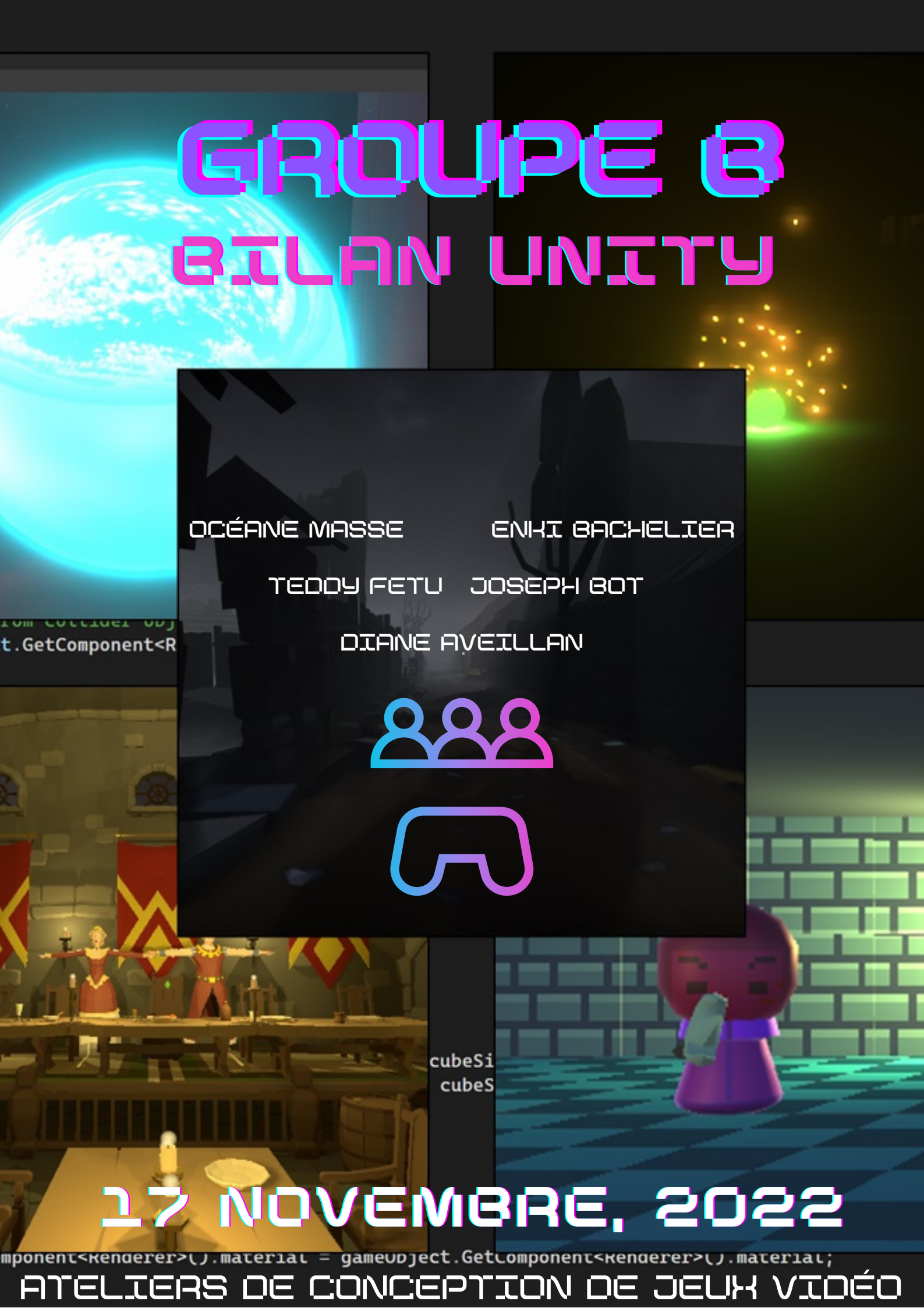
OCÉANE MASSE

ENKI BACHELIER

TEDDY FETU

JOSEPH BOT

DIANE AVEILLAN



17 NOVEMBRE, 2022

ATELIERS DE CONCEPTION DE JEUX VIDÉO

– Bilans Personnels –

Enki – A Blink Of Light

A Blink Of Light est un 'plateformer/horror' où le lancer d'objet avait pour but d'éclairer le chemin pour le joueur à l'aide de lucioles (boids). Le lancer d'objet était relativement simple à intégrer, ce n'est que de l'équilibrage pour trouver le bon feeling. L'IA, basée sur l'écoute sonore, était également simple à coder et à rendre « stressante ». Le visuel est un point plus compliqué. Autant l'ambiance visuelle (via les volumes d'URP) était simple et bien réussie, autant le joueur a du mal à se repérer dans la pénombre (malgré quelques techniques). En plus de cela, je considère que l'ambiance sonore n'est pas assez immersive même si certains l'ont trouvé suffisante. Pour m'améliorer, je dois creuser encore plus le Level Design et comment suggérer implicitement des mécaniques.

Diane – Sphere Gatherer

Sphere Gatherer est un prototype d'un jeu de type 'platformer', où j'ai principalement mis un point d'honneur à essayer de comprendre les cours et l'implémentation de ceux-là dans mon projet plutôt que de rendre quelque chose de joli (par manque de temps et à cause de mes propres difficultés). Pour ma part, faire le contrôleur, la caméra, le son et les animations était plutôt simple par rapport au reste (l'IA, les boids, et le level design), bien que j'ai fait l'erreur de mettre ma caméra au mauvais endroit. J'ai mal géré mon stress et je ne maîtrise pas Unity, j'ai donc, malheureusement, dû passer outre certaines parties des cours (que je vais revoir sur mon temps libre). Je pense faire d'autres projets pour continuer d'approfondir les notions déjà vues et les optimiser à l'avenir.

Teddy – At the top of Silver-Ground

At the top of Silver-Ground est un jeu d'infiltration ou l'on doit apprendre les patrouilles des gardes afin de les esquiver. L'IA a été le plus intéressant à faire et le plus simple. Cependant, j'ai eu beaucoup de mal à manier le Navigation Mesh : les IAs ne voulaient pas monter et considéraient leur patrouille sur le toit comme complétée. La création du terrain était simple car je le maitrisais déjà auparavant. Malheureusement, cette fonctionnalité a pris un temps assez conséquent, j'ai donc dû réduire les efforts dessus, chose que j'ai reportée sur le contrôleur où j'y ai passé un bon temps. J'ai appris beaucoup de choses sur ce projet, par contre je pense que je dois améliorer mon organisation sur le projet et reconsidérer le temps que prend chaque fonctionnalité.

– Bilans Personnels –

Joseph – Hello Patrick

Hello Patrick, référence au jeu Hello Neighbor dont je me suis principalement inspiré, est un jeu d'infiltration où il vous faudra trouver 3 clés dissimulées dans l'environnement afin de découvrir les secrets que renferme Patrick. Pour ce projet, j'ai décidé d'utiliser le HDRP d'Unity afin de pouvoir vraiment donner l'ambiance visuelle que j'imaginais, chose plutôt réussie grâce aux effets post-processing très avancés de ce pipeline de rendu. Toute la partie "brute", c'est-à-dire le contrôleur, l'animation ainsi que la state machine de l'IA, a été particulièrement simple à intégrer, étant donné mes précédentes expériences avec Unity 3D. Cependant la partie plus avancée telle que les différentes patrouilles de l'IA et ses comportements plus poussés n'ont pas été une mince affaire, sans compter le shader de surlignage des objets ramassables. Par manque de temps, beaucoup de mécaniques n'ont pas pu être pleinement implémentées et la partie audio a été brève, c'est par conséquent ce que j'améliorerai / ajouterai à mon projet.

Océane – Escape Dungeon

Dans un objectif de perfectionnement de mes compétences, j'ai réalisé un jeu d'évasion : **ESCAPE DUNGEON**. Mon objectif était de découvrir les principaux aspects expliqués par le cours avec une approche personnelle, afin d'avoir une meilleure conscience de la démarche et du travail à fournir dans la réalisation d'un jeu. Le premier point était pour moi le contrôleur que je souhaitais créer entièrement, car il influence énormément l'expérience de jeu. Bien que les retours soient vraiment positifs, l'orientation de la caméra reste peu fluide par rapport au déplacement du joueur. Je trouve qu'un bon contrôleur doit contribuer à l'expérience de jeu, ce qui est encore un niveau au-dessus à atteindre. Je suis en revanche très satisfaite de mon level design, qui porte à lui seul quasiment l'intégralité de mon gameplay, accompagné d'IAs aux comportements scriptés. Comme les retours l'ont mentionné, elles restent encore assez simples mais le déroulement du cœur du gameplay (jet d'objet pour distraire les gardes et recherche de clés pour s'échapper), ainsi qu'une réelle interaction avec l'environnement permettent d'entrevoir une bonne évolution du jeu. La démarche pour progresser en devient assez simple : se renseigner et multiplier les essais !

- Partage de pratiques Unity 3D -

Aborder le projet

Au sein du groupe, deux manières d'aborder le projet se sont opposées. La première, académique, consistait à suivre le fil du cours en réalisant les exercices demandés, bien que ces derniers ne soient pas toujours liés. La seconde consistait à utiliser le cours comme une annexe et de se concentrer principalement sur la vision personnelle du projet.

En comparant ces deux méthodes, nous pouvons identifier le fait qu'il est important de suivre le cours, qui peut apporter de nouvelles manières de faire ou de nouvelles connaissances. Cependant, il est nécessaire, voir essentiel, de garder en tête l'ébauche finale du projet afin de se rendre compte de l'importance qu'il faut donner à chaque module.

Nos niveaux et notre progression

Nous avons au sein de notre groupe des niveaux clairement distinctifs, car nous avons tous un background différent (études, projets personnels...). Beaucoup ont déjà programmé avant ou touché à Unity - ce qui a pu les aider lors de la réalisation du projet. Certain ont eu besoin de plus de temps afin de mieux comprendre les cours et d'implémenter ce qu'ils viennent de voir, tandis que d'autres avancent vite et ont la possibilité d'enrichir leurs projets, de les optimiser, voir d'apprendre de nouvelles choses. Nous commençons à prendre conscience des différents aspects dans la conception d'un jeu vidéo et du temps que cela peut prendre, nous avons donc acquis de nouvelles méthodes organisationnelles - ce qui fait que nous serons plus efficace et autonome à l'avenir.

Au niveau de notre progression personnelle à chacun, nous avons tous appris des choses nouvelles ou approfondis des connaissances que nous avions déjà avant. Par exemple nous avons tous acquis une meilleure utilisation d'Unity de manière générale, nous n'avions aussi quasiment jamais étudié l'animation, réalisé des IA, ni même l'audio - ce qui ajoute un vrai plus désormais dans nos portfolios et pour les projets futurs. Le fait de pouvoir mettre tous ces acquis en communs afin d'en discuter et de voir les projets des autres à la fin est aussi très enrichissant.

- Partage de pratiques Unity 3D -

Bonnes pratiques d'Unity

D'une part, les bonnes pratiques que nous avons identifiées au sein de notre groupe sont majoritairement d'ordre organisationnel : il faut ranger ses fichiers dans des dossiers et sous-dossier (scripts, audio, sprites...), mais également organiser la hiérarchie des objets disposés dans la scène afin de mieux s'y retrouver. Cela implique qu'il faut au préalable clarifier ce que l'on souhaite faire comme projet Unity. Par ailleurs, il faut donc bien estimer le temps d'une implémentation dans Unity ; et l'une des manières pour économiser ce temps est la création de prefab. Enfin, pour Unity et de manière plus générale, il ne faut pas hésiter à commenter son code et à demander des conseils / avis.

D'autre part, nous avons identifié quelques bonnes pratiques techniques :

- utiliser des headers (voir Unity Engine - HeaderAttribute) qui permet d'ajouter un texte au-dessus des champs dans l'inspecteur, et d'organiser visuellement les variables dans le code.
- l'utilisation variée de gizmos personnalisés pour le confort visuel dans l'éditeur : waypoints (reliés, numérotés) ou le gizmo en mesh pour la vue.
- mettre des rigidbody sur des objets demande beaucoup de ressouces, alors n'en mettez que lorsque nécessaire. Par exemple un objet à ramasser (non static) mais qui ne bouge pas n'a pas besoin de rigidbody.

Mauvaises pratiques d'Unity

Nous pouvons distinguer deux catégories : organisationnelle et technique. Au niveau de l'organisation, il est nécessaire d'éviter à tout prix de partir dans des projets trop ambitieux tant que ce qui est demandé initialement n'est pas respecté (par souci de deadline et pour mieux gérer son stress). Au niveau technique, un code peu ou pas commenté n'est pas souhaitable dans son apprentissage (par souci de compréhension). Il est également indésirable de tout coder dans un unique script. Pour pallier à ce problème, il est nécessaire d'appliquer la programmation objet mais d'une façon réfléchie et logique (ne pas abuser de l'héritage). Un autre problème se trouvait au niveau de la fonction Update. En effet, c'est une mauvaise idée de réaliser des boucles additionnelles dans cette fonction, qui est déjà une boucle à elle-même. Pour finir, un problème récurrent concernait les animations. Il faut bien penser à séparer l'animation du mouvement. L'animation se fait sur le modèle qui est enfant d'un gameObject vide, s'occupant lui du mouvement. Si cela n'est pas fait, nous allons observer des téléportations dues à l'animation.

– Bilan du TP Unity 3D –

Aspect général

Tout d'abord, l'accès au cours se fait de manière rapide et efficace grâce à une bonne organisation par thématique, en plus de fournir les principaux mots clés de chacune d'entre elles ainsi qu'un synopsis pour circuler rapidement au travers du cours vers les notions voulues.

L'ordre induit par la présentation est plutôt logique et n'empêche pas la bonne progression dans le cours. Cependant, les deux grands aspects code / visuel s'entremêlent et on se retrouve à devoir au moins lire en diagonale l'intégralité des cours avant pour mieux prendre conscience de l'ordre des notions et du développement d'un jeu. Cela nous amène à penser qu'il aurait fallu regrouper les cours par deux approches principales : celle du moteur en lui-même et de ses outils (l'environnement de travail, comment s'y déplacer, créer un objet/environnement simple, animation non procédurale…) et celle de la programmation en interaction avec le moteur (contrôleur, script, audio, animation procédurale, IA…).

Enfin, pour chaque cours il aurait été très utile de fournir le projet complet (scripts, scènes, assets…) à télécharger afin de non seulement comprendre le cours de manière interactive, mais aussi pour comparer avec notre essai, surtout si nous n'arrivons pas à reproduire les exemples du cours. Nous avons, en effet, remarqué plusieurs erreurs de frappe dans des exemples de code, ce qui gênait à la compréhension et sème le doute en copier/collant dans notre projet.

Retours sur les explications

Les cours ont des explications plutôt disparates : certaines sont trop courtes alors que nous souhaitons plus, d'autres sont très complètes et vont loin, bien plus loin. En revanche, la progression nous semble trop brutale : si la première page et la deuxième sont plutôt bien vulgarisées en amenant la nouvelle notion, le niveau des pages suivantes augmente bien trop rapidement. Du C# un peu trop poussé / pas assez expliqué (les notions de POO ne sont pas encore autant acquises), des exemples très intéressants mais très approfondis, … Cela nous soulève la question, à quel point faut-il comprendre et maitriser le contenu expliqué ? Est-il attendu de comprendre sur la semaine du TP ou est-il attendu que nous revenions sur les cours au fur et à mesure de notre apprentissage et jusqu'au rendu du JabJob ?

– Bilan du TP Unity 3D –

Les explications de la relation du cours avec les critères de notations, et des modules entres eux n'étaient pas très clairs. En effet, certains étaient perdus car chaque thématique a des exemples qui étaient à réaliser indépendamment (plage, panda, ie NavMesh...) tandis que le projet TP / JabJob se passait dans une seule scène et avait certaines contraintes (vue à la première personne, ...). D'autres se sont donc servis du cours comme appui mais sans forcément pousser les mêmes aspects, car c'était dans la démarche du TP fil rouge et par extension le JabJob.

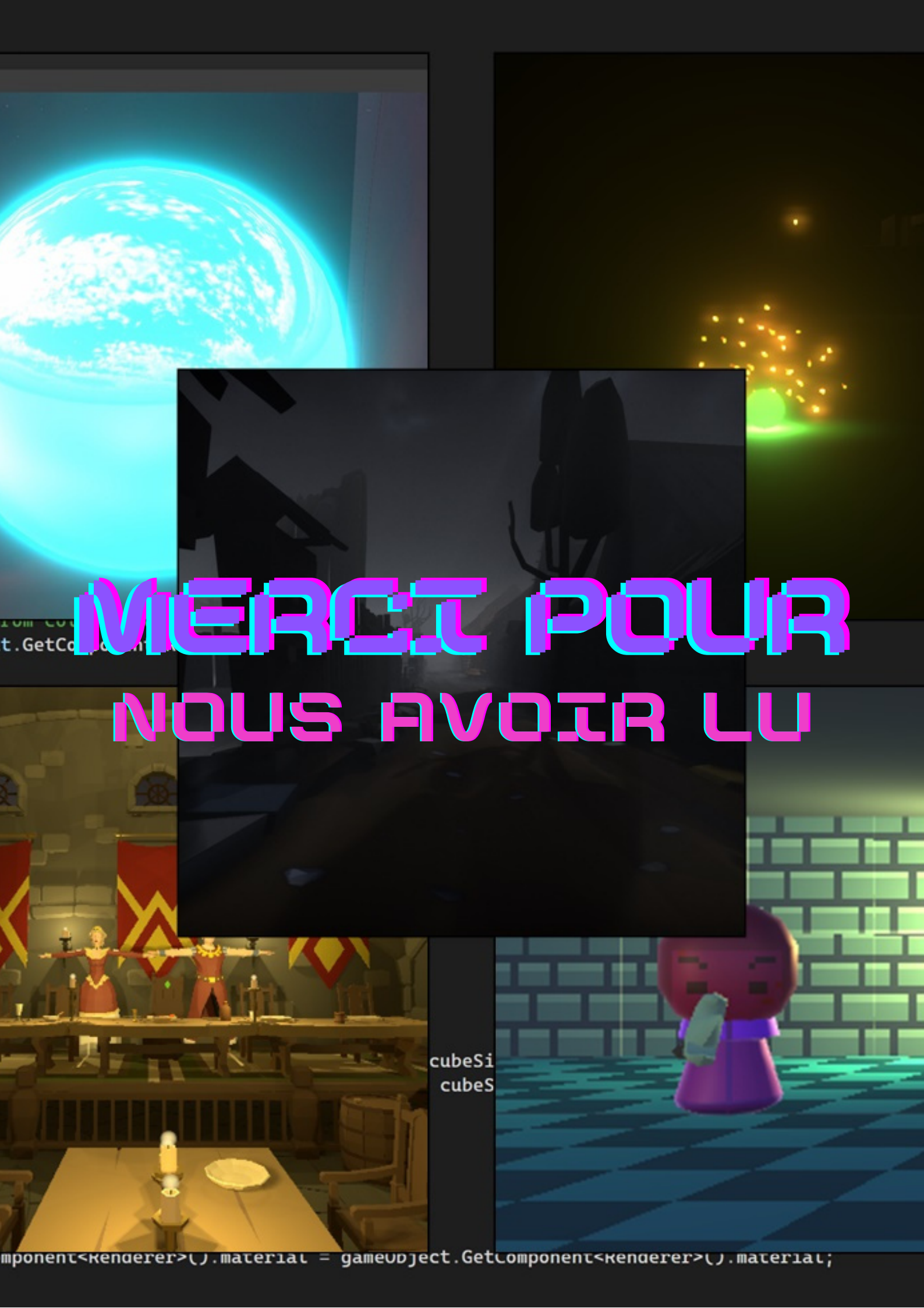
Suggestion d'amélioration

Séparer ce qu'il faut comprendre, savoir refaire et approfondir serait un grand plus afin de bien diriger notre projet et notre apprentissage. Mettre les projets de chaque module accessibles à tous et faire un lien logique entre chaque module (par exemple, mettre des exemples cohérents entre ces modules qu'on pourrait éventuellement utiliser et insérer dans nos projets) aiderait grandement à la compréhension mais également à l'attention des étudiants.

De plus, il serait intéressant d'avoir davantage de documentation sur les codes et également d'avoir différentes approches sur le même sujet (à l'aide d'autres sources par exemples).

Pistes à creuser

Nous aurons aimé passer plus de temps sur les animations, approfondir les contrôleurs, savoir comment faire des cinématiques et également un module sur URP/HDRP qui se révèle très utiles !


MERCI POUR
NOUS AVOIR LU