

2015-2016

2ème année du cycle ingénieur

Semestre 8

Projet

Jurassic Island

KOZISEK Arnaud
POLAT Stéphane
VINCONNEAU Mathieu

Sous la direction Mr RICHARD Paul



REMERCIEMENTS

Nous souhaitons tout d'abord remercier tout le corps enseignant de l'ISTIA pour les connaissances qu'ils nous ont transmises, et plus particulièrement monsieur Paul RICHARD, notre tuteur responsable du projet, pour les conseils, sa disponibilité et son aide. Nous souhaitons également remercier madame Eulalie VERHULST pour nous avoir suivis et assistés tout au long du projet. Nous souhaitons également remercier monsieur Guillaume CHARBONNIER pour ses conseils sur le management de projet et la méthode agile. Enfin, nous souhaitons remercier madame Mireille LOIRAT, madame Nadège VERRIER et madame Melany PAYOUX pour avoir écrit le scénario et apporté l'aspect psychologique.



| | |
|----------------------------------|----|
| Introduction | 4 |
| 1 Objectifs | 5 |
| 2 Présentation du projet | 6 |
| 3 Organisation du projet | 12 |
| 4 Développement technique | 14 |
| 5 Conclusion | 15 |
| Présentation du projet final | 16 |
| 1 Présentation du projet final | 16 |
| 2 Organisation du projet | 18 |
| 3 Développement technique | 20 |
| 4 Conclusion | 20 |
| Points d'améliorations | 21 |
| 1 Animation d'un dinosaure | 21 |
| 2 Suivi de la pente du dinosaure | 22 |
| 3 Autres améliorations | 22 |
| Conclusion | 23 |
| Annexe | 24 |
| Résumé | 29 |
| Abstract | 29 |

Introduction

Au cours de la deuxième année du cycle ingénieur de l'ISTIA, nous nous sommes intéressés à un projet de réalité virtuelle dans un but médical. En effet, de plus en plus de médecins s'intéressent à la réalité virtuelle afin de faire de nouvelles thérapies. La réalité virtuelle possède de nombreux avantages comme la possibilité de centraliser toutes les thérapies, améliorer la sécurité des patients et économiser du temps et de l'argent. Malgré un coût d'achat élevé, les nombreuses applications des thérapies permettent d'amortir l'investissement rapidement.

Ce projet s'inscrit dans le contexte du projet ENJEUX[X], financé par l'Université d'Angers et la Région Pays de la Loire afin de procéder à l'évaluation et la rééducation de troubles moteurs, cognitifs et émotionnels à l'aide d'un serious game. Jurassic Island permet de tester plus particulièrement les troubles de l'attention et de mémorisation de l'enfant.

Pour notre projet, nous avions un projet existant. Dans un premier temps, nous devions l'améliorer afin d'ajouter plus d'interactions avec l'utilisateur et le plonger encore plus dans un univers jurassique. Cependant, nous avons reçu un changement du projet à la fin du mois de février. L'objectif étant totalement différent, nous avons décidé d'établir une nouvelle organisation en essayant d'adapter un maximum d'éléments créés pour le projet précédent.

Présentation du projet initial

1 Objectifs

L'objectif du projet initial est de développer un environnement virtuel dans la période du jurassique. Pour cela, nous avons à notre disposition une île élémentaire avec un mode de jeu simple que nous allons décrire dans le chapitre suivant.

Le premier objectif était de passer le projet d'Unity 4 à Unity 5.3. Ce passage a été particulièrement délicat car quasiment tous les scripts et les interactions sont à refaire. Cependant, cette application est nécessaire afin de pouvoir utiliser les dernières fonctionnalités et d'ajouter plus facilement la Kinect et l'Oculus Rift.

Ensuite, l'élément le plus important du projet est d'ajouter des dinosaures avec des IA différentes. En effet, l'objectif est d'ajouter des dinosaures avec des comportements différents. Certains doivent être curieux, d'autres agressifs ou encore craintifs. Cette tâche se décompose en deux étapes. Dans un premier temps, nous devons trouver des modèles de dinosaure et les animer. Tous les dinosaures doivent posséder quatre boucles d'animation: une pour la marche, une pour la course, une pour l'attaque et une pour l'arrêt. Ensuite, nous devons créer les différents comportements et leurs déplacements. Ces déplacements sont aléatoires mais doivent respecter des contraintes afin d'améliorer le réalisme. Par exemple, un dinosaure ne peut pas marcher au fond de l'eau s'il est submergé ou encore grimper des pentes trop importantes.

Pour réussir à plonger encore plus dans l'univers le joueur, nous devons ajouter l'Oculus Rift et la Kinect pour gérer les déplacements. Le passage à la version Unity 5.3 permet l'utilisation des lunettes de réalité virtuelle très facilement. De la même manière, il existe des programmes permettant de récupérer le squelette d'un enfant à l'aide de la Kinect de Microsoft. Cependant, il est nécessaire d'associer une action à un mouvement que nous aurons préalablement enregistré.

Enfin, nous devons améliorer l'environnement en modifiant les objets à ramasser ou en ajoutant des nouveaux éléments de décors comme un volcan. Cette partie est une partie qui sera réalisée en fin de projet et permettra d'améliorer la richesse et la diversité de notre île.

2 Présentation du projet

Le projet fourni consistait en un parcours délimité par des cailloux blancs, sur lequel divers objets étaient à ramasser, avec quelques dinosaures faisant du bruit à l'approche du joueur et diverses autres créatures. Nous allons vous présenter brièvement le fonctionnement du jeu.

Le jeu commence par le menu suivant.

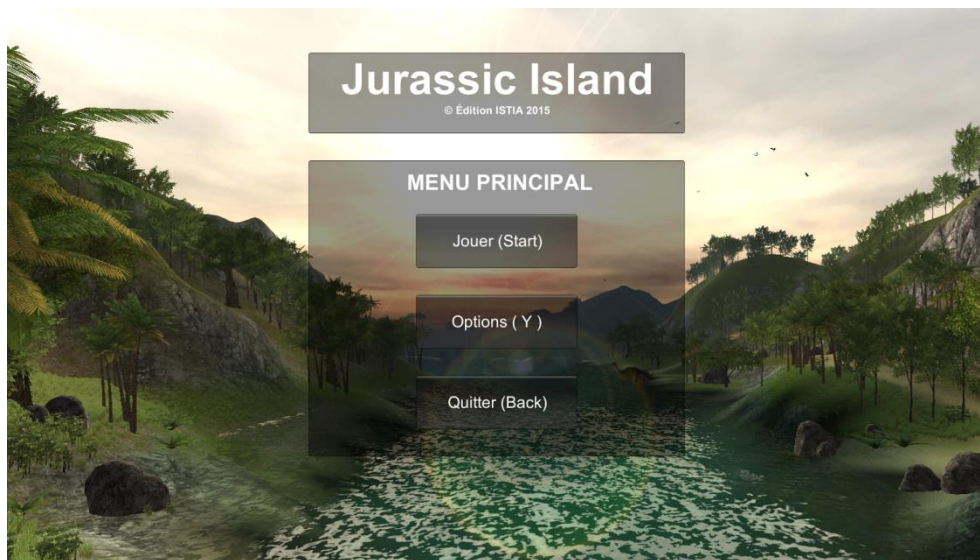


Figure 1: Menu de démarrage

Il faut tout d'abord aller dans les options (clic sur le bouton avec la souris ou touche Y de la manette Xbox) pour régler quels objets à ramasser seront présents dans la scène.



Figure 2: Choix des objets

Dans le menu précédant on peut choisir de ramasser le coffret, l'eyeball, la splitmetalball, la wheelball et le coffret_trésor, Le choix des objets est faisable uniquement à la souris et pas à la manette. Après avoir sélectionné les objets souhaités, on retourne au menu avec le bouton X de la manette. De retour au menu, on choisit ensuite Jouer.

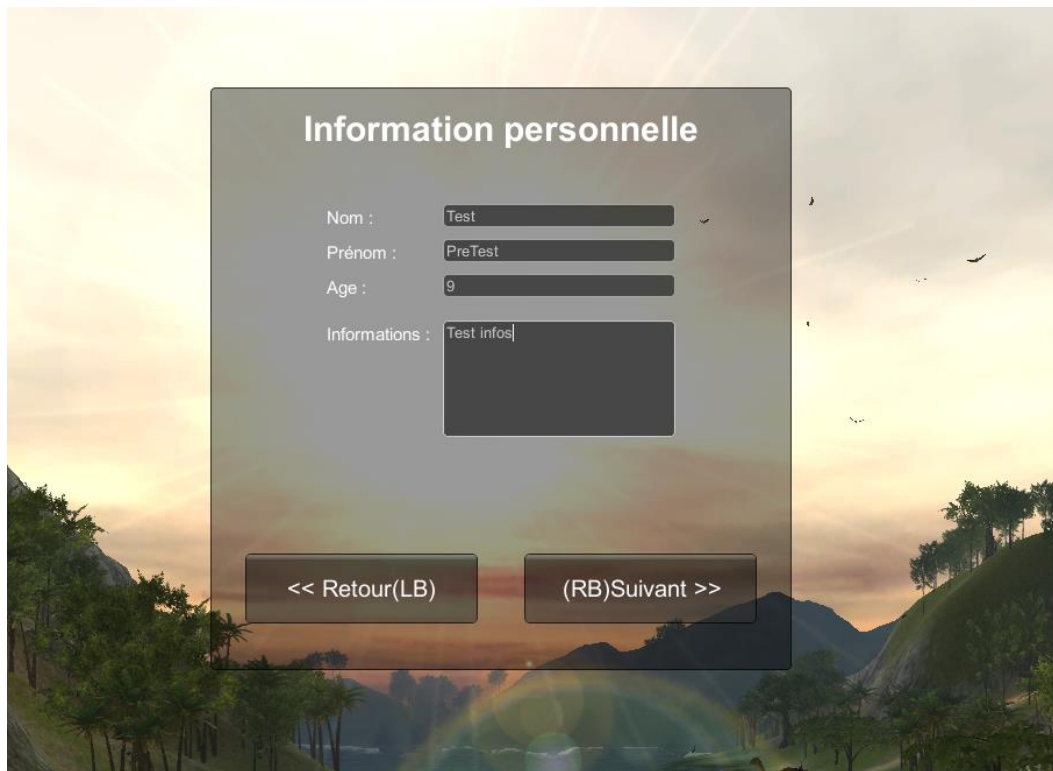


Figure 3: Menu: Information personnelle

Il faut ici simplement saisir les informations relatives à l'enfant ainsi que des informations supplémentaires qui pourraient être utiles au thérapeute. On clique ensuite sur suivant (ou bouton RB de la manette) pour lancer le jeu.



Figure 4: Ecran d'arrivée dans le jeu

En bas à gauche, on voit un perroquet, qui est sensé donner des instructions, mais il est inactif. En haut à droite, on peut lire le nom et prénom de l'enfant, le temps passé dans le jeu et le nombre d'objets ramassés. Notons cependant que le compteur de secondes défile trop rapidement. On peut voir également des boutons pour retourner au menu, pour afficher les résultats (nous aborderons ce point plus bas) ainsi qu'un bouton d'accès à l'aide qui est malheureusement non fonctionnel.

Lorsqu'on ramasse un objet, une petite animation est jouée et le compteur en haut à droite est incrémenté.



Figure 5: Animation ramassage



Figure 6: les objets choisis sont présents dans la scène

Le jeu ne possède pas de fin à proprement parler. Une fois qu'on a effectué ce que l'on souhaitait, on affiche donc les résultats.



Figure 7: Vue résultat

La page de résultat affiche la carte de l'île vue de dessus, et nous donne uniquement le trajet parcouru (trait rose). Aucune autre information n'est disponible sur cette page.

Le projet comportait ensuite quelques problèmes. Il y a tout d'abord le problème des commandes. Comme on l'a vu plus haut, la navigation dans les menus ne peut se faire entièrement à la manette. Les contrôles du personnage posent aussi quelques problèmes; Il est impossible de le contrôler totalement uniquement à la manette ou au clavier/souris. En effet, la rotation de la caméra est impossible au clavier/souris, tandis qu'il est impossible de courir en utilisant la manette (il faut appuyer sur la touche shift du clavier).

Il y a de plus certains problèmes de textures lorsqu'on entre puis ressort de l'eau. Un script permet de simuler un effet de flou sous l'eau.



Figure 8: Vue sous l'eau

Mais lorsqu'on on retourne à la surface, plusieurs textures (notamment d'arbres) deviennent phosphorescentes.

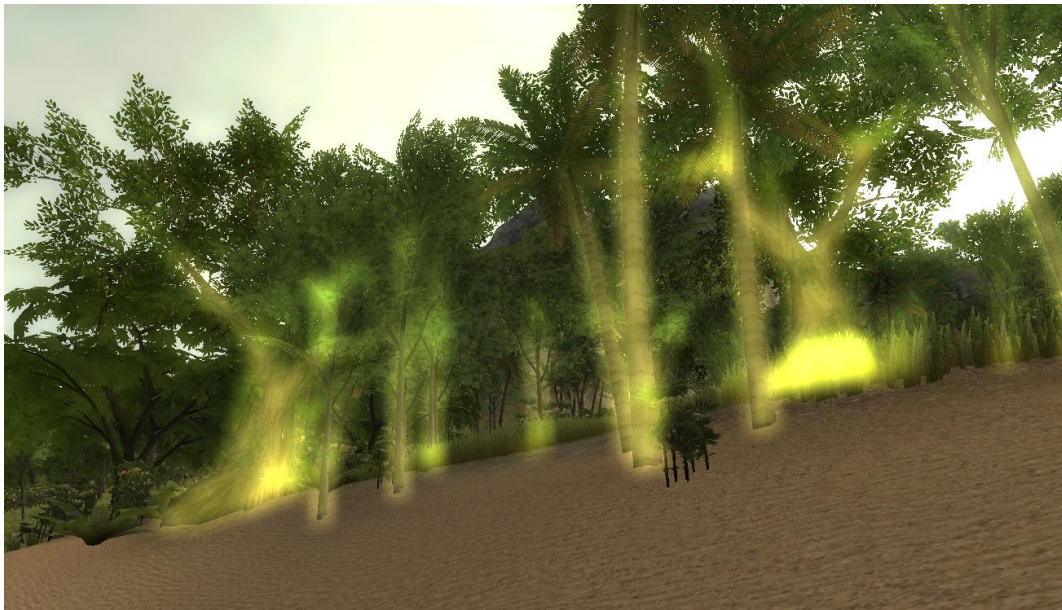


Figure 9: Bug d'affichage des arbres

Enfin, toutes les captures d'écran présentées jusque là étaient effectuées sur l'exé déjà fourni avec la projet Unity, car il s'avère que plusieurs textures (notamment d'eau) étaient manquantes sur le projet fourni, ce qui donnait un rendu de couleur rose.

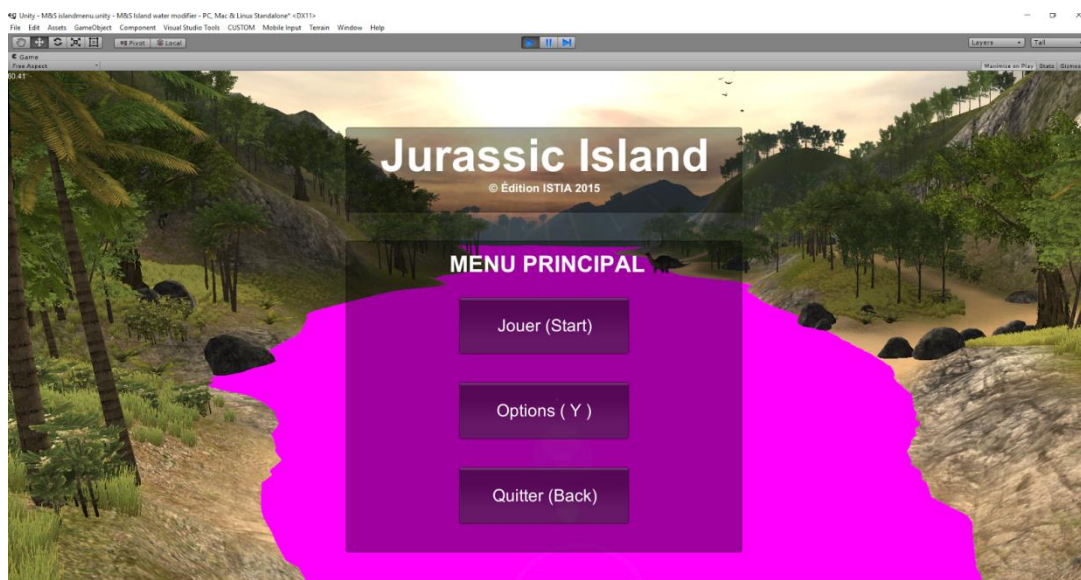


Figure 10: Ecran d'accueil

Considérant plusieurs autres facteurs tels que la complexité de plusieurs scripts du projet et son organisation, le nombre incalculable d'assets inutiles et hors sujet (des carcasses d'avion, des caisses d'arme, des immeubles, des voitures...) présentes dans le projet et le fait que plusieurs éléments potentiellement intéressants de l'asset store n'étaient pas disponibles pour Unity 4, nous avons décidé de récupérer uniquement la structure de l'île et de repartir avec un nouveau projet créé sous Unity 5.

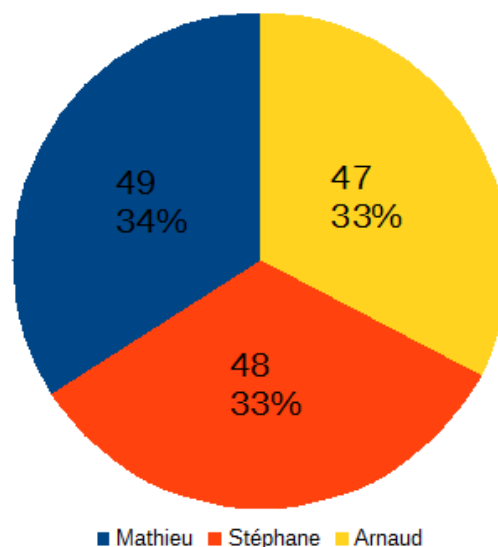
3 Organisation du projet

Pour la gestion de notre projet, et du fait des risques de demandes d'ajout de nouvelles fonctionnalités par exemple, nous choisissons d'utiliser la méthode agile.

Après découverte du projet existant et la décision de repartir quasiment à zéro, nous décidons après avoir déterminé les objectifs, d'une répartition des tâches correspondant aux compétences de chacun.

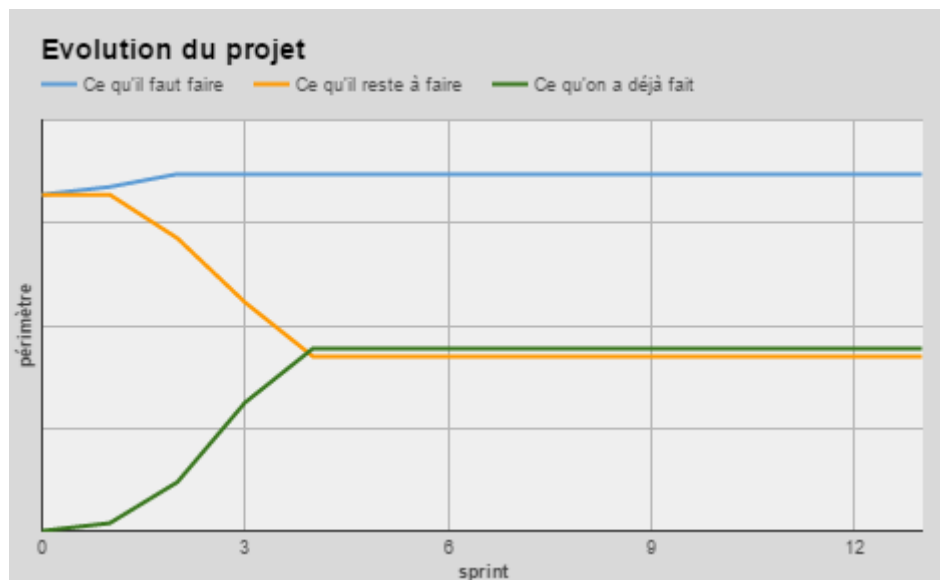
| N° tâche | Pers. | Catégorie | Priorité | Difficulté | Description |
|----------|----------------|---------------|----------|------------|---|
| T1 | Mathieu | Graphisme | P1 | 2 | Réalisation de la carte |
| T2 | Arnaud | Graphisme | P3 | 3 | Ajout dinosaures |
| T3 | Stéphane | Programmation | P1 | 13 | Ajout Kinect |
| T4 | Stéphane | Configuration | P3 | 1 | Coniguration des contrôles (clavier-souris/manette) |
| T5 | Stéphane | Programmation | P3 | 21 | IA dinosaure agressif |
| T6 | Mathieu | Programmation | P3 | 21 | IA dinosaure craintif |
| T7 | Stéphane | Programmation | P2 | 13 | Animation dinosaures |
| T8 | Arnaud/Mathieu | Configuration | P2 | 13 | Recherche d'objets à ajouter sur l'Asset Store |
| T9 | Arnaud | Graphisme | P2 | 13 | Ajout objets à ramasser (défini/procédural ?) |
| T10 | Arnaud | Programmation | P2 | 5 | Script ramasser objet |
| T11 | Arnaud | Programmation | P3 | 8 | Menu configuration objets |
| T12 | Mathieu | Programmation | P3 | 13 | Exportation des données |
| T13 | Arnaud | Programmation | P3 | 5 | Déplacement dinosaures |

Cela donne une charge de travail par membre à peu près équilibrée



Malgré la répartition initiale, il arrivera cependant fréquemment que chacun s'entraide ou effectue une partie du travail d'un autre, tout en restant cependant concentré principalement sur la tâche qui lui est assignée. Bien que l'ajout de la Kinect fut indiqué en priorité, le manque d'accès au matériel (utilisé par d'autres groupes de projet) et aux conditions d'utilisation idéales (grand écran, salle avec de l'espace) feront que l'on décidera finalement de réserver cette tâche pour la fin.

A l'issue des 4 premiers sprints, on remarque que la moitié du projet est déjà terminée, sans compter d'autres tâches déjà avancées mais pas encore complètement finies.



Nous pensons donc à ce stade pouvoir finir la majorité des parties restantes dans les 2 ou 3 prochains sprints, pour pouvoir ensuite peaufiner les parties qui seraient améliorables.

4 Développement technique

Le projet ayant de nombreux problèmes de programmation, nous avons donc décidé de recommencer le projet en récupérant uniquement le relief de la carte. La première tâche était donc de faire les textures de ses reliefs ainsi que de rajouter des éléments de décor afin d'améliorer l'immersion dans un univers jurassique.

Une fois la carte réalisée, nous avons rajouté de l'eau. Le défi majeur de cette tâche est de créer l'eau la plus réaliste possible. En effet, pour éviter de gâcher l'expérience de jeu, il faut que l'eau reflète le plus parfaitement les éléments terrestres. De plus, nous avons créé un phénomène de léger courant tout en diminuant les vagues imposées par le modèle de base d'Unity.

L'environnement terminé, nous avons intégré des dinosaures à la carte, auxquels nous avons appliqué plusieurs animations. Pour chaque créature, nous avons ajouté les quatre animations nécessaires: une pour la marche, une pour la course, une pour l'arrêt, une où le dinosaure se défend si on approche trop près de lui. De plus, nous avons rajouté une animation pour la mort du personnage même si nous ne nous en servons pas.

Ensuite, nous nous sommes attardés sur l'IA des dinosaures. Pour cela, nous avons créé différents scripts pour les comportements des dinosaures. Tout d'abord, nous avons utilisé le navMeshAgent d'Unity 5 pour pouvoir déplacer le dinosaure d'un point A à un point B en évitant les obstacles. Cependant, le dinosaure marchait au fond de l'eau et montait des pentes impossible à gravir. Pour cela, nous avons créé une carte de contrainte interdisant l'accès aux zones où le dinosaure peut se noyer et en ajoutant des coefficients de difficulté en fonction de la pente. Ces coefficients permettent au dinosaure d'éviter des zones trop pentues même s'il peut toujours y accéder s'il y est contraint. Ensuite, nous avons créé le comportement du dinosaure en fonction de celui du joueur. En effet, comme le montre le script du dinosaure agressif en annexe 1, nous avons changé les déplacements du dinosaure en fonction de la distance le séparant du personnage contrôlé. Par exemple, si le joueur est trop proche, le dinosaure va courir dans sa direction et même l'attaquer s'il le peut. Ensuite, une fois que celui-ci est en dehors de la zone, on choisit une nouvelle destination aléatoire. Pour la destination aléatoire, nous avons créé un tableau avec une dizaine de destinations possible. Une fois que le dinosaure a atteint la destination, une valeur aléatoire permet de donner une nouvelle destination qui sera atteinte par le meilleur parcours grâce aux calculs du navMeshAgent.

5 Conclusion

Lors de ce projet, tous les aspects techniques étaient fini. Même s'il restait encore deux taches techniques à faire qui sont l'implémentation de la Kinect et de l'Oculus Rift. De plus, nous étions en avance sur nos prévisions et pensions pouvoir commencer la partie d'amélioration de l'environnement avec l'intégration d'un volcan rapidement. Cependant, fin février, nous avons reçu un nouveau scénario par courriel. Ce scénario se décompose en 11 étapes permettant de parcourir toute l'île.

On découvre, à notre grande surprise, que plus aucun autre dinosaure ne doit se promener sur l'île. C'est donc un coup dur pour l'équipe sachant que ce développement avait pris la majeure partie du temps consacré à ce projet. Cependant, nous décidons de consacrer une séance afin de pouvoir adapter un maximum des travaux réalisés au prochain projet. De plus, nous avons réalisé une nouvelle planification en fonction de nos compétences.

Présentation du projet final

1 Présentation du projet final

Le nouveau projet se déroule sur la même île que la précédant. Cependant, la déambulation dans l'île est maintenant scénarisée. Zozor un jeune dinosaure bipède et vert cherche à aider sa maman à retrouver ses œufs. Pour cela, il invite l'enfant à le suivre afin de rapporter les précieux œufs à sa maman situé de l'autre coté de l'île. Lors de ce périple, ils vont rencontrer des animaux, grimper aux arbres, manger des bananes ou encore ramasser des fleurs. A la fin, Zozor retrouve sa maman et lui remet les œufs ramassés pendant le trajet. A ce moment là, la maman du petit dinosaure demande à l'enfant de lui raconter l'aventure vécue. Cette étape permet de vérifier la mémoire de l'enfant et sa concentration. Le scénario est donc décomposé en 11 étapes comme le montre la carte ci-dessous.



Figure 11: Carte prévisionnelle de Jurassic Island

La première étape consiste en un tutoriel. Le dinosaure se présente et explique les bases de l'intrigue à l'enfant. Ensuite, L'enfant apprend à se déplacer dans l'environnement et à effectuer des étapes de base. Premièrement, il suit le dinosaure pour arriver à une pile de caisses. Il doit arriver au sommet afin de ramasser des pièces. Ensuite, il doit redescendre et aller ramasser une fleur. Une fois cette étape finie, il est prêt à partir à l'aventure.

La scène deux est une étape transitoire, l'enfant monte au sommet d'une montagne sur un chemin parsemé de pierre. Un fois arrivé au sommet, l'enfant aperçoit le premier œuf au sommet d'un arbre. Il doit

alors escalader pour l'attraper. Ensuite, il descend la montagne par l'autre sommet. Cependant, des arbres bloquent le passage. Le joueur doit donc sauter par dessus pour continuer l'aventure.

Nous sommes maintenant à la scène cinq. L'enfant longe le bord de l'eau et peut ramasser des noix de coco et des fleurs. Ensuite, il emprunte un chemin pour arriver au sommet d'un second sommet. En route, il croise des singes et un perroquet qui possède un œuf du dinosaure. Pendant la scène 7, le temps se couvre et on entend des éclairs. On peut même apercevoir un arc en ciel. Ils traversent une rivière puis accèdent à une nouvelle plage. Sur cette plage, nos deux aventuriers trouvent un nouvel œuf et empruntent un bateau pour naviguer sur les flots.

Pendant leur traversée en bateau, le joueur aperçoit des tortues, des ptérosaures qui volent au-dessus de leurs têtes et un bateau de pirates. Ils arrivent alors sur un rocher où ils trouvent le dernier œuf. Enfin, Zozor retrouve sa maman sur une plage et le joueur doit raconter son périple.

Ce scénario possède de nombreux points complexes comme la réalisation et l'animation de nombreux animaux différents. De plus, il faut réaliser de nombreux codes pour toutes les interactions demandées. Devant le peu de temps imparti et la difficulté du travail demandé, nous avons décidé d'effectuer l'impasse sur certains points demandés. Sachant que nous ne pouvions pas finir notre projet dans les délais, nous avons essayé de faire un code propre et commenté afin de pouvoir être réutilisé par nos successeurs. En effet, nous apportons un point important à la continuité de projet car nous avons été confronté à ce problème.

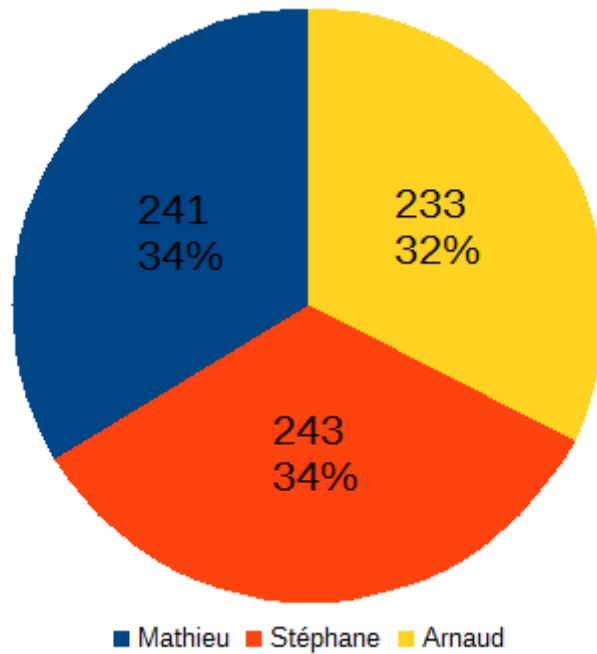
2 Organisation du projet

Malgré des objectifs plus clairement définis que lors des premières semaines du projet initial, nous avons décidé de rester sur une organisation agile.

Ci-dessous, la répartition des tâches pour le "nouveau" projet.

| N° tâche | Pers. | Catégorie | Priorité | Difficulté | Description |
|----------|----------|---------------|----------|------------|--|
| T1 | Arnaud | Configuration | P1 | 8 | Intégration et modification du précédent projet (assets, récolte objets, animations) |
| T2 | Mathieu | Programmation | P1 | 13 | Script dialogues |
| T3 | Stéphane | Configuration | P3 | 5 | Gestion de l'ambiance sonore |
| T4 | Stéphane | Configuration | P2 | 5 | Gestion des voix |
| T5 | Mathieu | Configuration | P1 | 8 | Suivi du scénario |
| T6 | * | Graphisme | P2 | 34 | Recherche et ajout de modèles 3D |
| T7 | Mathieu | Programmation | P1 | 21 | Scène 1 (tutoriel) |
| T8 | Stéphane | Programmation | P1 | 21 | Scène 2 |
| T9 | Arnaud | Programmation | P1 | 21 | Scène 3 |
| T10 | Mathieu | Programmation | P2 | 21 | Scène 4 |
| T11 | Stéphane | Programmation | P2 | 21 | Scène 5 |
| T12 | * | Programmation | P2 | 34 | Scène 6 |
| T13 | Arnaud | Programmation | P2 | 21 | Scène 7 |
| T14 | Mathieu | Programmation | P2 | 21 | Scène 8 |
| T15 | * | Programmation | P2 | 34 | Scène 9 |
| T16 | Stéphane | Programmation | P2 | 21 | Scène 10 |
| T17 | Arnaud | Programmation | P2 | 13 | Scène 11 |
| T18 | Stéphane | Programmation | P2 | 13 | Ajout Kinect |
| T19 | * | Graphisme | P3 | 55 | Animations et "effets spéciaux" |
| T20 | Arnaud | Programmation | P3 | 13 | Gestion de la météo |

Encore une fois, nous arrivons à une répartition globalement équitable de la charge de travail de chacun.



Cependant, nous savions déjà vu le peu de séances restantes (7) et la charge énorme de travail à réaliser, qu'il nous serait impossible de réaliser tout le projet. Nous décidons donc de nous attaquer aux 3 premières scènes ainsi qu'aux scripts nécessaires au bon fonctionnement de celles-ci, pour que nos successeurs puissent se concentrer sur la partie 3D/animation. Malgré un manque évident de motivation sur les 2 premières séances (du fait du changement soudain de direction du projet), nous avons néanmoins passé le reste des séances à travailler sur nos scènes respectives afin d'avoir quelque chose de présentable en tant que démo.

3 Développement technique

Tout d'abord, nous avons utilisé le package `FirstPersonController` fourni par Unity 5 afin que le joueur puisse jouer avec une vue à la première personne. Nous l'avons ensuite appliqué à un objet 3D simple appelé capsule. Cette forme permet de définir le box collider du joueur (zone définissant la collision entre deux objets). Une fois ajouté à la scène, nous avons donc réglé les paramètres de poids, de taille et de vitesse afin qu'il corresponde au mieux à un enfant.

Enfin, nous avons réalisé un script en C# permettant le bon déroulement de l'histoire. En effet, ce script exécute des coroutines successives afin de vérifier si le joueur effectue correctement les actions demandées. Pour réaliser les différentes actions des coroutines nous utilisons un script élémentaire disponible partiellement en annexe 3. A chaque actualisation de la fonction, le programme vérifie qu'aucune coroutine ne soit cours à l'aide de la variable `Firstpass`. Si la condition est vérifiée, il lance alors la coroutine demandée ou vérifie une condition comme nous pouvons le voir en annexe 4 et 5. Comme nous pouvons le voir dans l'annexe 4, nous utilisons une coroutine appelée "wait" avec différents paramètres afin de pouvoir l'utiliser dans toutes les conditions possibles. En effet, nous pouvons afficher un message à l'écran, déplacer le dinosaure et attendre selon les paramètres donnés. Enfin, la fonction "arrive" vérifie si le dinosaure est arrivé à proximité du point d'arrivée afin de poursuivre l'aventure. Le passage de la variable `fin` à vrai permet d'incrémenter l'étape en cours et de réinitialiser les variables. En annexe 5, nous pouvons voir les scripts permettant de vérifier si un objet est toujours présent. Pour cela, nous avons utilisé un tag sur les objets en question afin d'être sûrs de récupérer le bon objet.

Nous avons donc intégré différents objets tels que des œufs et des fleurs rouges et violettes que le joueur doit ramasser en approchant de leur position. Comme nous pouvons le constater avec le script de l'annexe 6. Dès que le joueur s'approche de l'objet, celui-ci disparaît et la personne peut poursuivre le scénario. Nous avons intégré des cailloux et des végétaux faisant partie intégrante des scènes du jeu car le joueur doit pouvoir, dans certaines scènes, monter sur un arbre.

4 Conclusion

Ce projet a donc été une course contre la montre. En effet, le temps réduit ne nous a pas permis de terminer le projet. Cependant, nous avons réussi à développer le squelette du projet. En effet, grâce au script en annexe 3, nous pouvons réaliser quatre-vingt dix pourcent des actions demandées dans le scénario de base.

Cependant, dû à un manque de temps, nous avons réalisé en intégralité seulement les trois premières étapes. C'est pour cela que de nombreux points sont encore à améliorer. Nous allons donc détailler ces améliorations dans la partie suivante.

Points d'améliorations

1 Animation d'un dinosaure

Comme nous l'avons vu précédemment, Zozor le dinosaure doit être un dinosaure vert bipède. Or, nous ne possédons pas de dinosaures correspondant à la demande. De plus l'animation d'un dinosaure bipède serait longue et complexe.

C'est pour cela que nous avons décidé d'utiliser un modèle que nous avons animé dans le premier projet. Nous avons élu le modèle du triceratops du projet précédent. Cependant, le triceratops n'est pas vert. Nous avons donc créé une texture verte s'appliquant au modèle. Cette texture réalisée de toute pièce pendant le projet à l'aide de PhotoFiltre est présente en annexe 2. Le dinosaure est vert mais il n'est pas bipède.

Un point important d'amélioration serait donc de concevoir un dinosaure vert bipède ainsi que ses animations. Le dinosaure devra être enfantin et il faudrait prévoir une version adulte. De plus, il ne doit pas ressembler à l'un des dinosaures du dessin animé Dino train (cf. image ci-dessous) et doit porter un sac à dos.



Figure 12: Personnage dino train

Lors de son périple, Zozor doit effectuer plusieurs actions qu'il faudra donc animer préalablement. Voici une liste non exhaustive: sauter, marcher, courir, ramasser une fleur, tomber, taper dans les mains, se taper les fesses, grimper aux arbres. Enfin, une dernière animation pourrait permettre à Zozor de bouger les lèvres lorsqu'il parle.

2 Suivi de la pente du dinosaure

Naturellement, les objets sur Unity ne s'adaptent pas automatiquement au terrain se trouvant sous leurs pieds (et ce malgré la gravité simulée). Ainsi, il arrive fréquemment de retrouver un personnage sur un terrain en pente mais se tenant parfaitement droit. Cela ne pose qu'un problème mineur lorsqu'il s'agit d'un humanoïde se déplaçant sur ses deux pieds, mais il arrive souvent pour des objets plus long que des parties du mesh rentrent dans le décor, ou une différence bien trop importante entre le niveau du sol et les extrémités de l'objet.

Il faut donc trouver un moyen pour que les objets suivent la forme du relief. Une solution possible serait de mesurer constamment la distance entre deux points situés à chaque extrémité de l'objet (par exemple patte avant et arrière) et le sol et d'effectuer une rotation de l'objet jusqu'à ce que ces deux distances soit nulles (ou les plus courtes possible). Néanmoins cette solution ne permet pas de savoir l'orientation de l'objet par rapport au relief (typiquement, un personnage qui se déplacerait de façon latérale sur un flanc de montage)

3 Autres améliorations

En plus de ces deux points majeurs, de nombreuses autres améliorations mineures peuvent être implémentées au projet. En effet, pour augmenter l'immersion du joueur, on peut ajouter une Kinect de Microsoft. Ce système permettra au personnage de reproduire les mouvements du joueur et améliorer son immersion. Dans une optique d'améliorer les conditions de jeu du joueur, on peut également ajouter un casque de réalité virtuelle. Le casque conseillé est l'Oculus Rift dont le module est intégré à Unity 5.

Ensuite, nous pouvons améliorer l'environnement afin d'être plus proche de la réalité. On peut par exemple ajouter un effet de vision sous l'eau. On peut également ajouter un système de météo variable. Ce système permettra de changer la météo en fonction de la position du joueur. Ainsi, on peut créer un orage ou encore faire apparaître un arc en ciel au moment voulu.

Enfin, l'animation d'autres animaux comme des singes ou encore des oiseaux pourront améliorer l'univers et le rendre plus vivant. Cependant, ces tâches sont longues et complexes par rapport à l'objectif premier du serious game.

Enfin, il faudrait ajouter une voix au dinosaure. Ce travail n'est pas difficile car les différentes pistes vocales peuvent être lancées à l'aide d'une autre variable dans les coroutines. Une voix à Zozor le dinosaure permettrait à l'enfant de comprendre l'histoire sans avoir à lire ou à se faire conter l'histoire par l'encadrant.

Conclusion

Ce projet de création de serious game à but médical et thérapeutique fut très intéressant et enrichissant pour nous. Il nous a permis dans un premier temps de mettre en oeuvre les compétences techniques acquises au cours de notre formation, et également d'en développer des nouvelles (IA, animation, ...) et ainsi de mieux maîtriser Unity.

Nous avons également dû apprendre à faire face à un changement soudain de cahier des charges et adapter le travail déjà réalisé en amont afin de limiter au maximum le temps perdu durant les premiers mois de travail. Cela nous a montré qu'un manque de communication entre la maîtrise d'ouvrage et les développeurs peut amener à des situations imprévues, et qu'un projet n'est jamais totalement figé est peut être amené à bouger au cours des mois, selon les besoins du client. La fait d'avoir choisi la méthode agile pour gérer ce projet nous a cependant peut-être aidé.

Nous avons fait notre possible pour réaliser le maximum de travail sur la nouvelle version, tout en essayant de rendre le projet facilement accessible (par le biais de scripts adaptatifs à chaque scène) pour nos successeurs afin qu'ils puissent se concentrer sur la création des nouvelles scènes et les autres pistes d'amélioration.

Annexe 1

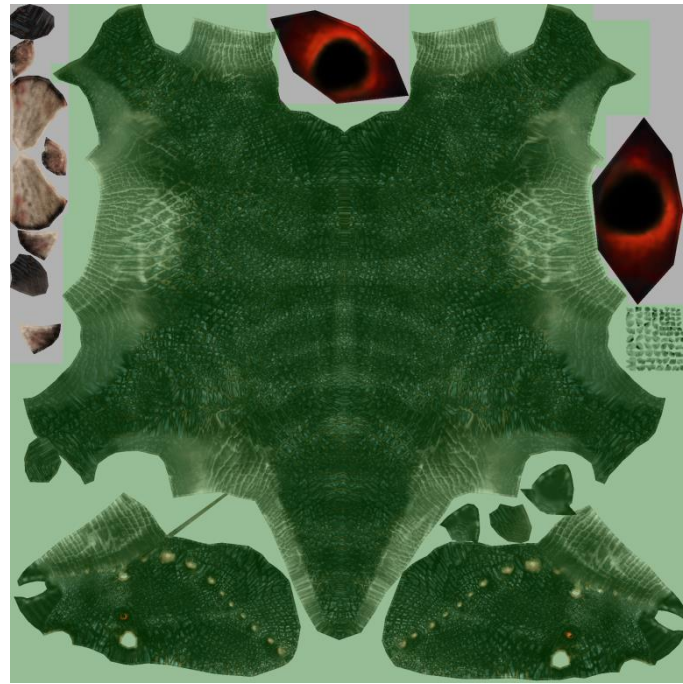
```
public class StegosaurusController : MonoBehaviour {
    Animator anim;
    public GameObject player;
    public GameObject dest;
    public NavMeshAgent agent;
    public float distToAttack = 10f;
    public float distToRun = 15f;
    public float distToWalk = 20f;

    // Use this for initialization
    void Start () {
        anim = GetComponent<Animator>();
    }

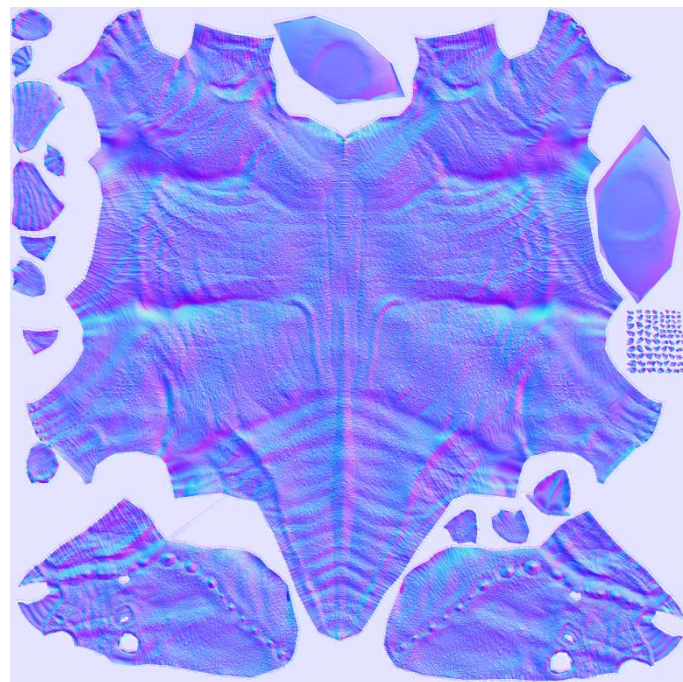
    // Update is called once per frame
    void Update () {
        float dist = Vector3.Distance(transform.position, player.transform.position);
        if (dist < distToAttack)
        {
            anim.SetInteger("AnimState", 3);
            agent.destination = transform.position;
        }
        else if (dist < distToRun)
        {
            anim.SetInteger("AnimState", 2);
            agent.destination = player.transform.position;
        }
        else if (dist < distToWalk)
        {
            anim.SetInteger("AnimState", 1);
            agent.destination = dest.transform.position;
        }
        else
        {
            anim.SetInteger("AnimState", 0);
        }
    }
}
```

Annexe 2

Texture:



Normal map:



Annexe 3

```

void Start()
{
    step = 1;
    fin = false;
    firstPass = true;
    agent.destination = this.transform.position;
    egg_tuto = GameObject.FindGameObjectsWithTag("egg_tuto");
    flower_tuto = GameObject.FindGameObjectsWithTag("flower_tuto");
    text = GameObject.FindGameObjectsWithTag("text");
    egg_tuto[0].transform.localScale = new Vector3(0, 0, 0);
    flower_tuto[0].transform.localScale = new Vector3(0, 0, 0);
}

// Update is called once per frame
void Update()
{
    print(fin);
    if (firstPass) {
        if (step == 1)
        {
            CoroutineWait(5.0f, "Nous sommes sur une île où vivent des dinosaures comme toi, de nombreux autres animaux et aussi quelques pirates!");
            StartCoroutine(Coroutine);
            firstPass = false;
        }
        if (step == 2)
        {
            StopCoroutine(Coroutine);
            StartCoroutine(Wait(5.0f, "Viens m'aider à retrouver les œufs de ma maman. Il retrouver ses 4 œufs, ils sont tous jaunes avec des arcs noirs blancs"));
            firstPass = false;
        }
        if (step == 3)
        {
            /*
            //incrémenter des étapes
            if (fin)
            {
                step += 1;
                fin = false;
                firstPass = true;
            }
            */
        }
    }
}

```

Annexe 4

```
IEnumerator wait(float time, float x, float y, float z, string text)
{
    yield return new WaitForSeconds(time);
    fin = true;
    this.transform.position = new Vector3(x, y, z);
    agent.destination = this.transform.position;
    instruction.text = text;
    yield return true;
}

IEnumerator wait(float time, float x, float y, float z)
{
    yield return new WaitForSeconds(time);
    fin = true;
    this.transform.position = new Vector3(x, y, z);
    agent.destination = this.transform.position;
    yield return true;
}

bool arrive(float x, float z)
{
    if (dino.transform.position.x > x - 1 && dino.transform.position.x < x + 1 && dino.transform.position.z > z - 1 && dino.transform.position.z < z + 1)
    {
        fin = true;
        return true;
    }
    else
    {
        return false;
    }
}

IEnumerator wait(float time)
{
    yield return new WaitForSeconds(time);
    fin = true;
    yield return true;
}

IEnumerator wait(float time, string text)
{
    yield return new WaitForSeconds(time);
    instruction.text = text;
    fin = true;
    yield return true;
}
```

Annexe 5

```
using UnityEngine;
using System.Collections;

public class ramasse : MonoBehaviour
{
    private GameObject player;
    // Use this for initialization
    void Start()
    {
        player = GameObject.FindGameObjectWithTag("Player");
    }

    // Update is called once per frame
    void Update()
    {
    }

    void OnTriggerEnter(Collider other)
    {
        if (other.tag == "Player")
        {
            Destroy(gameObject);
        }
    }
}
```


RÉSUMÉ

Dans le cadre de notre projet tutoré EI4, nous avons dû réaliser un serious game sous Unity, destiné à aider les enfants atteints de troubles de l'attention. Le jeu consiste en une île tropicale sur laquelle l'enfant doit suivre Zozor le dinosaure pour l'aider à retrouver les 4 oeufs de sa maman répartis sur l'île. L'enfant va devoir interagir avec son environnement et devra à la fin de la partie raconter ses aventures à la maman de Zozor.

Une version de base du projet, que nous avons dû convertir sous Unity 5, nous était fournie; seulement le scénario définitif (ne nous ayant été donné que 2 mois après le début du projet) s'avéra très différent, et nous avons dû tout presque tout recommencer. Nous avons ainsi créé les 3 premières scènes ainsi que les scripts nécessaires à leur fonctionnement, afin de faciliter la tâche à nos successeurs. De nombreux axes d'amélioration, tel que l'ajout de la Kinect ou de l'Oculus Rift, sont envisageables.

Ce projet nous a permis de mettre en oeuvre les compétences techniques acquises au cours de notre formation ainsi que d'en développer de nouvelles, notamment en ce qui concerne la gestion de projet et le travail en collaboration avec des non-ingénieurs.

mots-clés : Unity, Serious Game, Réalité Virtuelle, C#

ABSTRACT

In the context of our EI4 tutored project, we had to create a serious game using Unity, designed to help children with attention disorders. The game consists in a tropical island where the child has to follow Zozor the dinosaur to help him find 4 of his mom's eggs spread across the island. The child will have to interact with his environment and will at the end of the game tell his adventures to Zozor's mom.

A base version of the project, that we had to convert to Unity 5, was available; however the definitive scenario (that we were given only 2 months after the beginning of the project) turned out to be quite different, thus we had to start almost all over again. Thereby, we created the first 3 scenes as well as the scripts necessary to run them, in order to facilitate our successors' job. Several areas for improvement, such as adding the Kinect or the Oculus Rift, can be considered.

This project allowed us to use the technical skills acquired throughout our education as well as to develop new ones, especially project management and working in collaboration with non-engineers.

keywords : Unity, Serious Game, Virtual Reality, C#