Errata

Page 97 : Le schéma d'intégration temporelle utilisé dans le logiciel Wolf est un schéma Runge-Kutta d'ordre 3 qui ~~permet~~ entraine un temps de calcul ~~réduit parce que le pas de temps est~~ plus important .

Page 100 : Tableau 16

	SARndbox	Wolf2D
Temps simulé (s)	60	
Temps de calcul (s)	10	~~60~~ 3h22 min
Nombre de pas de temps	21744	~~12277~~ 26549
Pas de temps moyen (s)	0,0027594	~~0,005~~ 0,00226

Tableau 16 : Performances de calcul du programme SARndbox et Wolf2D

Page 104 : Tableau 17

	SARndbox	Wolf2D
Temps simulé (s)	30	17,5s
Temps de calcul	43 s	~~60 s~~ 3h25min
Nombre de pas de temps	80157	~~12094~~ 33019
Pas de temps moyen (s)	0,0074883	~~0,0025~~ 0,00053

Tableau 2 : Performances de calcul du programme SARndbox et Wolf2D

Page 106 : Oubli d'insertion du fichier de conclusion dans la version finale

Conclusion

L'objectif de ce travail était double, dans une première partie, il visait à définir les composants et réaliser un bac à sable à réalité augmentée en utilisant un capteur Kinect de deuxième génération. Dans la deuxième partie du travail l'objectif était d'étudier le schéma d'écoulement utilisé dans le programme informatique et d'établir les performances de ce programme en prenant à titre de comparaison le programme d'écoulement Wolf2D.

L'analyse de l'état de l'art a permis de définir les exigences, sélectionner et mettre en œuvre les différents composants du bac à sable à réalité augmentée : ordinateur, caméra 3D, projecteur, bac à sable et potence.

Une fois le matériel reçu, l'installation d'une version Linux Mate 64 bit en dual boot a été réalisée ainsi que l'installation du logiciel SARndbox Version 2.3. Il s'agit d'un logiciel Open Source contenant

environ 8800 lignes de code. Il utilise aussi les librairies Vrui VR Toolkit en version 4.2-006 (250 000 lignes de code) , Kinect 3D Processing version 3.2 (26000 lignes de code) ainsi que les librairies UNIX standard. Cette version, selon l'auteur, était fonctionnelle avec la caméra 3D Kinect de première génération et contenait de fichiers pour utilisation avec la Kinect de deuxième génération appelée aussi Kinect XBox One. Dans cette phase, des adaptations des procédures d'installation ont dû être réalisées et sont désormais documentées.

Après l'installation, la mise en œuvre des outils de compilations et la génération de code a pu être réalisée. Il est à noter que l'environnement de développement ne repose pas sur un environnement avec débugger intégré de type Code Bloc ou Visual C++ mais se fait exclusivement en ligne de commande à partir de makefile. N'ayant que des connaissances de base de ce type d'environnement et vu la complexité des programmes, nous n'avons pas pu non plus ajouter des fonctions de débogage. La mise au point de toute modification dans ce travail s'est donc faite en ajoutant des messages sur la console ou en utilisant de sortie sur fichiers.

Les premiers essais avec la Kinect seconde génération retournaient un code d'erreur de non détection de la caméra. Des investigations ont été menées d'abord en instrumentant le programme SARndBox puis avec le programme Protonect utilisant une librairie indépendante Libfreenect2. Ce programme a permis de s'interfacer avec la Kinect version 2 et de visualiser les images de la caméra couleur et infrarouge. Il a aussi permis de constater que le choix du port USB est primordial : il faut en effet que la caméra soit connectée sur un port USB 3.0 directement connecté sur la carte mère pour fonctionner. Malgré cette découverte la caméra version 2 ne fonctionnait pas dans le logiciel SARndBox. Diverses modifications ont été tentées dans ce logiciel dont la tentative d'utilisation de la librairie Libfreenect2. Etant donné le temps passé dans ces investigations infructueuses, cette piste d'utilisation de la Kinect Xbox One a donc été mise de côté. Il a été décidé de poursuivre avec une Kinect de première génération.

Le travail s'est poursuivi par la mise en place d'un bac à sable fonctionnel avec l'utilisation du programme SARndbox en passant par la construction du bac à sable et de la potence, l'installation du logiciel et les étapes de calibrations de la Kinect et le projecteur. Cet outil permet de représenter en couleur le comportement de l'eau sur une topographie modelée dans le sable. Le comportement visuel de l'écoulement est mis à jour en temps réel grâce à la représentation de phénomènes hydrologiques. Les fonctions de base du logiciel permettent notamment d'imposer de la pluie sur l'ensemble de la surface en imposant une intensité de précipitations ou de générer une pluie localisée au droit de la main en détectant le passage de celle-ci dans le champ de la caméra. Le logiciel permet aussi de récupérer une topographie réelle (Digital Elevation Model, DEM), de colorier le sable pour reproduire cette topographie et pouvoir ensuite simuler un écoulement.

Pour aborder la seconde partie qui consiste à comparer le modèle du programme avec le programme d'écoulement Wolf2D, le détail des équations et leur implémentation dans le logiciel a été étudiée en détail pour pouvoir ensuite modifier celui-ci.

Afin d'avoir des performances temps réel pour la simulation d'écoulement, il faut savoir que dans le programme SARndbox les équations d'écoulement ne sont pas traitées dans le programme de base sur le CPU principal comme dans un programme classique mais par l'utilisation de GSLS shaders exécutés sur le co-pocesseur GPU de la carte graphique. Très brièvement, les matrices sont transformées en image aussi appelées textures où la composante rouge est égale à l'élévation de la

surface libre, les composantes verte et bleue correspondent au débit spécifique dans deux directions. Les équations d'écoulement sont alors résolues par les paramètres des shaders exécutés sur la carte graphique en même temps que les opérations de rendu visuel. Le résultat peut être ensuite récupéré sur le CPU.

Après cette analyse diverses pistes de modification du programme ont été étudiées et testées, l'une d'elle a abouti à un programme modifié fonctionnel qui permet d'imposer la topographie à tester, de fixer les conditions d'écoulement, et sauver les résultats sous forme de fichier.

L'analyse des premiers résultats a permis de mettre en évidence des inexactitudes dans l'implémentation du schéma numérique dans le code du programme. Deux corrections ont été apportées. La première a été une augmentation de la taille de la texture de bathymétrie pour éviter que, lors de la reconstruction de la bathymétrie au centre de la cellule, des pixels hors du domaine puissent être atteints. En effet, si on désire obtenir l'information d'un pixel hors du domaine, la valeur utilisée est celle du pixel le plus proche à l'intérieur du domaine. Cette opération amenait donc des erreurs sur les valeurs aux bords. La seconde correction a concerné les conditions aux limites en passant des conditions aux limites de type ouverte vers une impossibilité de l'eau à sortir du domaine en imposant une topographie plus importante sur les pixels du périmètre de la texture de bathymétrie. Après ces 2 modifications les allures des résultats sont cohérentes avec ce qui est attendu. Cependant, à l'analyse en détails des valeurs, un très faible apport d'eau est généré sur le modèle dès que l'écoulement touche un des bords du domaine, ceci provoque une augmentation du volume. Cette erreur détectée en fin de ce travail, n'a pu être corrigée à ce jour.

Au niveau des performances le fait d'utiliser la carte graphique pour résoudre les équations d'écoulement permet, sur les écoulements à étudier, de passer d'un temps d'exécution de plus ou moins 1 minutes sur un programme type Wolf2D à 10 secondes pour une topographie plate avec le logiciel SARndbox. Des mesures complémentaires et plus détaillées devraient confirmer ces observations.

Concernant la précision, aucune analyse quantitative n'a été effectuée. Au niveau qualitatif, signalons que le logiciel travaille sur des valeurs réelles en simple précision (32 bits).

Ce travail a été pour moi l'occasion de découvrir et m'investir considérablement tant dans le domaine informatique et dans l'utilisation des shaders, qui n'est pas mon cœur de métier, que dans des aspects pratiques de réalisation, investigations ainsi que dans le détail des équations d'écoulement. Comme probablement dans mon futur travail, aucune étape ne s'est déroulée sans encombre. A chaque fois j'ai envisagé, modifié et testé des solutions alternatives afin d'essayer d'atteindre au mieux les objectifs fixés.

Ce travail est une première étape et pourrait être poursuivi sur plusieurs domaines.

Au niveau des performances, on pourrait envisager de pousser plus loin les comparaisons d'écoulement pour s'assurer de l'exactitude du modèle après corrections des dernières erreurs ainsi que de réaliser une comparaison de précision des calculs dans différents scénarios. Quant aux performances de calcul il serait peut être utile de voir l'influence ou les limites liées à la taille des modèles de topographie.

Concernant le logiciel, bien que peu nécessaire pour un bac à sable, réaliser son adaptation pour une Kinect V2 pourrait être envisagée.

Une autre évolution serait d'utiliser le logiciel que nous avons modifié et lui adjoindre la partie représentation en réalité augmentée, dans le cas d'une topographie imposée. Plus qu'un simple outil didactique, cela permettrait, avant d'analyser en détail les résultats chiffrés, de se faire une idée des effets et de jouer sur les paramètres. On pourrait même envisager sur les études modèle réduit en laboratoire avec plusieurs cameras et projecteurs de vérifier la topographie reproduite et colorier les vrais écoulements.

A travers l'utilisation cet outil, la rapidité des simulations effectuées a été mise en évidence. Cet avantage d'un calcul d'écoulement sur carte graphique avec l'utilisation des GSLS shaders, pourrait s'appliquer à une évolution ou variante du logiciel WOLF par exemple sur des modèles complexes. Il faut noter que dans ce cas, l'application devrait être ré écrite avec les shaders. Ce choix même s'il est dépendant de certaines cartes graphiques puissantes permettra une pérennité et un gain de performance en fonction des améliorations de performance constante de ce type de matériel.

D'autres modifications plus ludiques peuvent être apportées au bac actuellement construit, comme le paramétrage d'une télécommande, la simulation pour l'utilisateur en récupérant des informations de google earth de simuler l'effet de pluies abondantes dans sa région. On peut aussi envisager la création d'une sorte d'imprimante qui permettrait de déposer le sable à partir d'un modèle d'élévation digitale d'un terrain.

Pour conclure à travers ce travail, il apparaît clairement que le bac à sable à réalité augmentée est un outil visuel qui a été privilégié au détriment de l'exactitude des simulations générées. La correction des erreurs amènerait donc à un outil puissant et exploitable dans le domaine scientifique. Cette première approche laisse encore de nombreuses opportunités d'investigations et d'améliorations d'un bac à sable à réalité augmentée tant d'un point de vue universitaire que d'un point de vue plus éducatif.