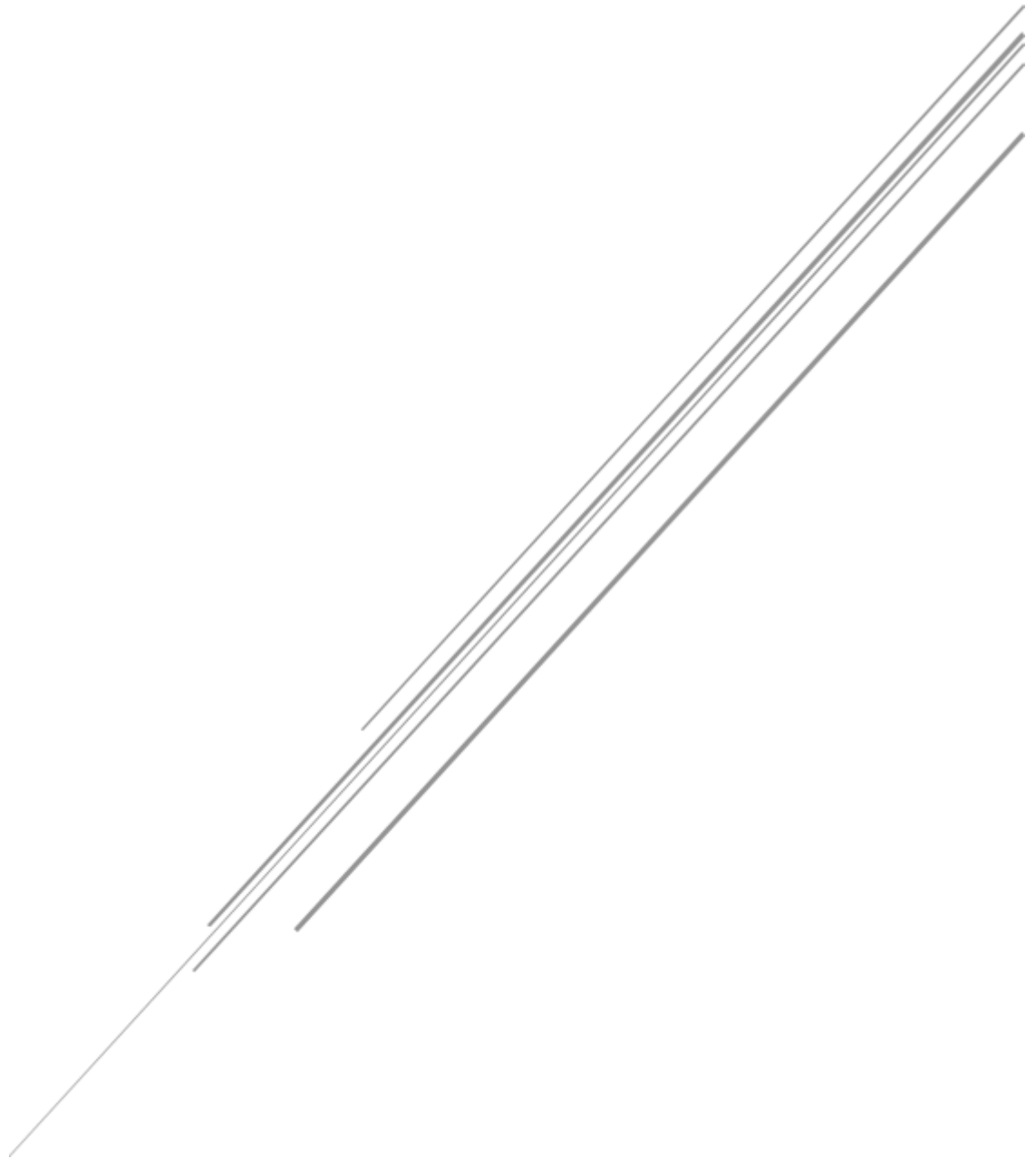


PROJET TUTEURE S2 :

Programmation dans Minecraft



FICHE DESCRIPTIVE DU PROJET TUTEURE DE S2

N° DU SUJET : 227

TITRE : Programmation Minecraft

TUTEUR : Amélie CORDIER

MEMBRES DU GROUPE : - Kévin GUTIERREZ
- Lucas HERAUT
- Youenn STEVANT
- Loic CHEREL
- Flavien LEPINE

Résumé du projet en quelques lignes : Notre objectif pour ce projet sur Minecraft était de déterminer par quelles méthodes on pouvait faire découvrir la programmation à des jeunes jouant à Minecraft.

Technologies envisagées pour la réalisation du projet : Maîtriser les outils de développement web (HTML, CSS, PHP, ...) pour créer un site de tutoriels ainsi qu'un forum correspondant. Le LUA, un langage de programmation, est aussi utilisé pour les tutoriels car nous les écrivons à propos d'un mod nommé ComputerCraft qui permet d'utiliser des Scripts en LUA sur Minecraft.

Méthode de gestion de projet envisagée : Nous avons pour l'instant réalisé un diagramme de Gantt (voir Annexes) détaillant les tâches qu'il fallait effectuer et leur durée approximative de manière à obtenir un planning de réalisation.

Quelle est, selon vous, la probabilité que vous poursuiviez ce projet au prochain semestre ? Le sujet est très intéressant mais pour répondre à notre problématique nous avons préféré partir sur l'optique du « cours » en ligne. La réalisation de ce projet est donc plutôt littéraire. C'est pour cela que la probabilité que l'on poursuive ce projet est entre 50-60%.

Résumé des difficultés rencontrées pour la réalisation de ce projet : Le majeur problème que nous avons rencontré est celui des disponibilités de chacun. A 5 il était difficile de trouver un créneau horaire sur lequel tout le monde pouvait travailler donc pour cela nous avons travaillé plus ou moins de notre côté avec des « réunions » pour se répartir les tâches, donner nos avis et surtout parler du thème général du projet et le développer.

Sommaire

I - Présentation du projet

- a) Objectifs
- b) Problématique
- c) Solution choisie

II - Outils à maîtriser

- a) Langage LUA
- b) Minecraft (+ComputerCraft pour la rédaction des tutoriels)
- c) Programmation Web
- d) Gestion des versions (GIT)

III - Veille technologique

- a) Wiki
- b) ComputerCraft

IV) Contenu du Wiki

- a) Tutoriels
- b) Forum

V) Evolution du Wiki

- a) Directions à prendre pour l'avenir du site Internet
- b)

Diversifications

VI) Annexes

I) Présentation du projet

a. Objectifs

L'objectif de ce projet est de réaliser un outil permettant de faire découvrir la programmation à des enfants à l'heure où l'accès à la technologie est de plus en plus facilité. Mais leur utilisation se limite plutôt au surf sur Internet, jeux sur consoles et tablettes, vidéos...

Au sein des écoles, les divers cours dispensés concernant l'informatique ne se limitent qu'à une simple introduction sur l'utilisation d'un ordinateur ou bien à l'utilisation des outils de bureautique sans avoir une compréhension réelle de la logique de l'informatique et des technologies qu'ils utilisent quotidiennement.

Mais cette tendance est en passe de changer avec notamment l'apparition dans les lycées de l'option ISN (Informatique et sciences du numérique) ainsi qu'une volonté du gouvernement d'introduire les élèves à la programmation dès le collège, même dès le primaire.

Avec cette volonté, il faut donc que les élèves bénéficient d'outils leur permettant de comprendre de manière concrète les principes de la programmation.

b. Problématique

Il existe déjà des outils permettant d'introduire les enfants au monde de la programmation comme : Scratch, Lego Mindstorm, Arduino, Raspberry Pi...



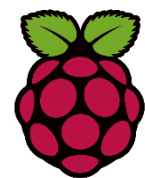
Scratch



Lego Mindstorm



Arduino



Raspberry Pi

Minecraft est un jeu de type “bac à sable”, le joueur se retrouve dans un environnement libre où il peut faire ce qu’il souhaite sans objectif de jeu. Son côté créatif permet aux enfants de laisser exprimer leur imagination, en reproduisant par exemple des éléments de la vie réelle dans le jeu. C’est ce côté ludique qui peut faire de Minecraft un bon outil pour apprendre la programmation, où l’aspect créatif est aussi important.

Problématique : Comment introduire des enfants à la programmation à l’aide de Minecraft ?

Grâce à Minecraft, nous avons plusieurs axes de solutions pour créer un outil d’apprentissage de la programmation :

- Logique booléenne et portes logiques grâce à la redstone présente dans le jeu de base sans ajout externe
- Développement en Java et création d’un mod à l’aide des API de modding telles que Forge
- Développement dans le jeu en LUA à l’aide du mod ComputerCraft

c. Solution choisie

Nous avons choisi d’opter pour la dernière solution, et cela pour plusieurs raisons :

- La première solution offre l’avantage de comprendre la logique de l’informatique mais ne correspond pas réellement à une introduction à la programmation
- La deuxième solution nous semble trop complexe et devrait être réalisée par des personnes ayant déjà des connaissances avancées en programmation
- Le mod ComputerCraft permet de développer directement à l’intérieur du jeu, ce qui permet d’utiliser les éléments déjà présents dans le jeu et ainsi poser des images concrètes sur des concepts de programmation, ce qui est très bien pour une introduction.

Résumé du projet

Réaliser un site internet (type Wiki) comportant une série de tutoriels et d’applications afin de maîtriser les bases de la programmation (plus précisément du langage LUA) à l’aide ComputerCraft et de réaliser un objectif préalablement défini (réalisation d’un mini-jeu par exemple).

II) Outils à maîtriser

a. Langage LUA

Le langage LUA est un langage de programmation apparu en 1993. C'est un langage orienté objet à prototypes ainsi que procédural. Il est de faible typage, ce qui le rend un peu plus difficile d'accès pour des personnes ne pratiquant pas l'informatique par rapport au Python. Il sera donc nécessaire dans les cours d'expliquer avec précision les caractéristiques de ce langage. L'approche orientée objet ne sera pas dans ce tutoriel, au moins au début, car ceci est assez compliqué.

Ce langage de programmation est assez intéressant car est assez utilisé dans les jeux vidéo. Les joueurs de *Minecraft* étant intéressés par les jeux vidéo, les applications de ce langage dans le monde du jeu vidéo peut être enrichissant. Par exemple, on trouve de ce langage dans l'interface du jeu *World of Warcraft*, on en trouve aussi dans le moteur CryEngine.

Le LUA est écrit en C ANSI strict, ceci lui donne donc un grand nombre de systèmes sur lesquels ce langage peut être compilé.

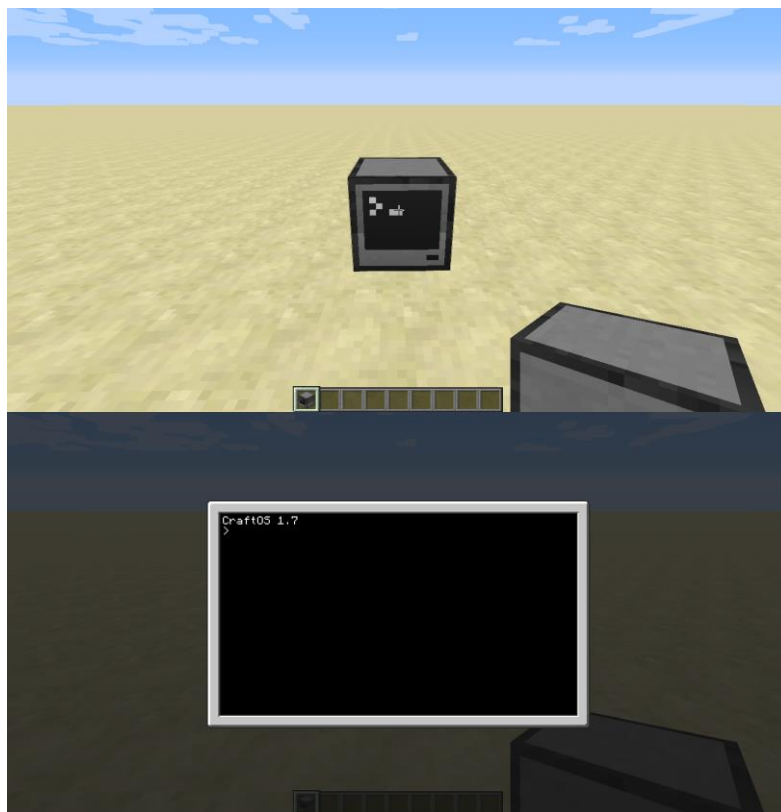
Ce langage n'est pas dépendant du mod *ComputerCraft* (dont on parlera plus tard). Ce tutoriel pourra donc être suivi par une personne souhaitant uniquement apprendre ce langage. Les exercices que nous écrirons seront par contre uniquement disponibles sur *ComputerCraft* pour donner des exemples plus concrets d'application de la programmation pour des personnes souhaitant apprendre les bases de l'informatique.



b. ComputerCraft

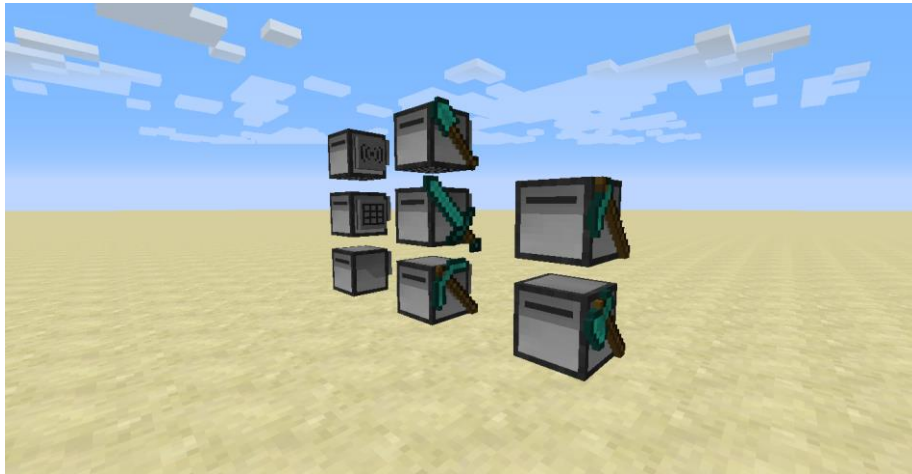
ComputerCraft est un mod pour *Minecraft* permettant de pouvoir programmer en LUA dans le jeu en ajoutant plusieurs blocs.

Le premier bloc est l'ordinateur (Computer). Il permet d'accéder au *CraftOS*, un système d'exploitation spécifique au mod. Sur cet OS, il est possible de rentrer des commandes comme dans une interface de commande d'un OS classique. On y retrouve les commandes `cd`, `mkdir`, ou encore `edit`. Grâce à ces dernières, il est possible de créer des fichiers texte pour pouvoir écrire nos codes. C'est ici que tous les cours appris en LUA pourront être appliqués dans le jeu. En plus des fonctions habituelles du LUA, on trouvera de nouvelles fonctions pour *Minecraft*, comme par exemple activer un courant de redstone.



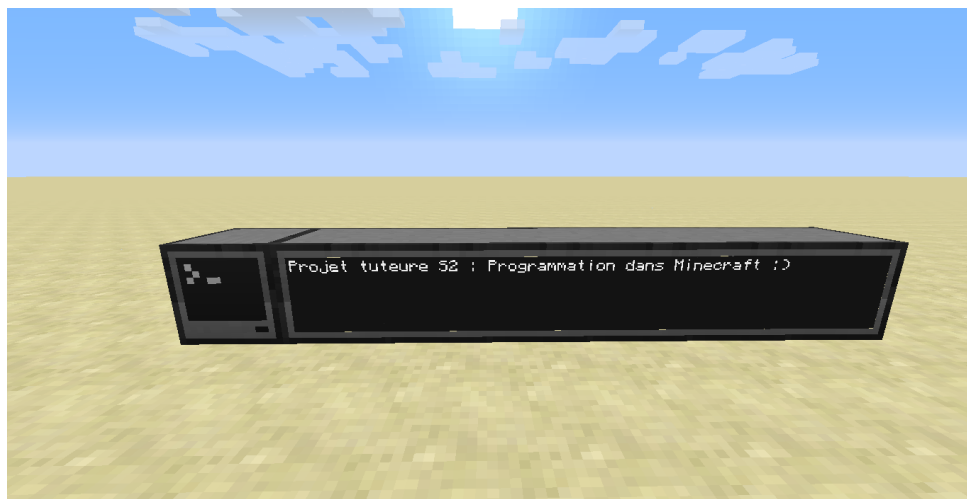
Computer et CraftOS

Ensuite, le deuxième bloc est la Turtle. Il fonctionne comme un robot : il faut programmer son IA dans un code et il réalisera les actions voulues seul. On peut écrire directement le code dans le bloc. On peut aussi lui rajouter des objets du jeu pour lui ajouter des fonctions, comme par exemple une hache pour couper du bois.



Différents types de Turtle

Enfin, le mod rajoute d'autres blocs et objets plus secondaire. Le disque (Disk) et le lecteur de disque (Disk Drive) qui fonctionnent respectivement comme une clé USB et un lecteur USB, permettant de transporter et lire des fichiers entre plusieurs blocs. Les Modems et Cables permettent de communiquer directement entre plusieurs blocs. L'écran (Monitor) permet d'afficher en grand la sortie texte d'un programme et peut avoir une taille de 6x8 blocs.



Utilisation des écrans couplés à un ordinateur

c. Programmation web

Pour ce projet, il faudra des personnes maîtrisant les outils de développement web car le Wiki sera mis en ligne. Il faudra donc maîtriser le HTML et CSS pour tout ce qui est de la partie rédaction des tutoriels et mise en page du site. Il faudra une bonne ergonomie d'ensemble pour que la lecture ne soit pas trop difficile. Ces deux langages vont principalement servir pour la mise en forme du Wiki et de son contenu.

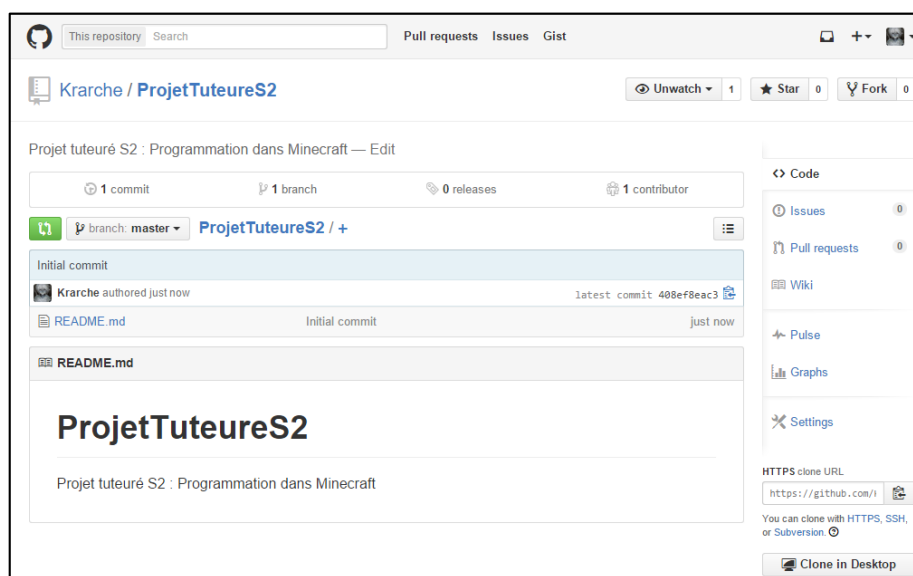
Pour rendre le visuel meilleur, des connaissances en JavaScript seront les bienvenues dans le but de créer un aspect plus dynamique et donc plus ergonomique pour l'utilisateur. Ces connaissances ne sont pas nécessaires.

Enfin, un des langages les plus importants sera le PHP. Ce dernier permettra aux utilisateurs de créer un compte pour le Wiki. Ce compte enregistrera leur progression dans le cours. Il sera utilisé pour accéder aux forum. C'est sur ce dernier que l'on retrouvera des sujets créés par les utilisateurs pour que ceux-ci puissent collaborer et s'entraider. Le forum est une part très importante de ce projet car c'est lui qui permettra au Wiki d'évoluer et de développer une communauté de joueurs voulant apprendre, ou connaissant déjà, les bases de l'informatique.

d. Gestion des versions (GIT)

La gestion des versions sera très importante dans notre projet, non seulement pour faciliter le travail en groupe mais aussi car la majorité des tutoriels seront composés de scripts LUA dont le maintien des codes sources devront être effectués notamment pour faire évoluer le Wiki mais aussi pour s'adapter aux différentes mises à jour de Minecraft mais aussi de ComputerCraft.

Un repository devra donc être mis en place sur Github, et devra être régulièrement utilisé afin d'avoir un réel suivi de l'avancement du projet.



Repository Github pour la gestion des versions

III) Veille technologique

a. Wiki

Pour la mise en forme du site Internet, nous prendrons exemple sur des sites d'enseignement informatique et des wikis déjà existants. Il y a par exemple le site OpenClassrooms, le site francophone le plus connu pour apprendre à coder, ou bien le wiki ComputerCraft, épuré et efficace.

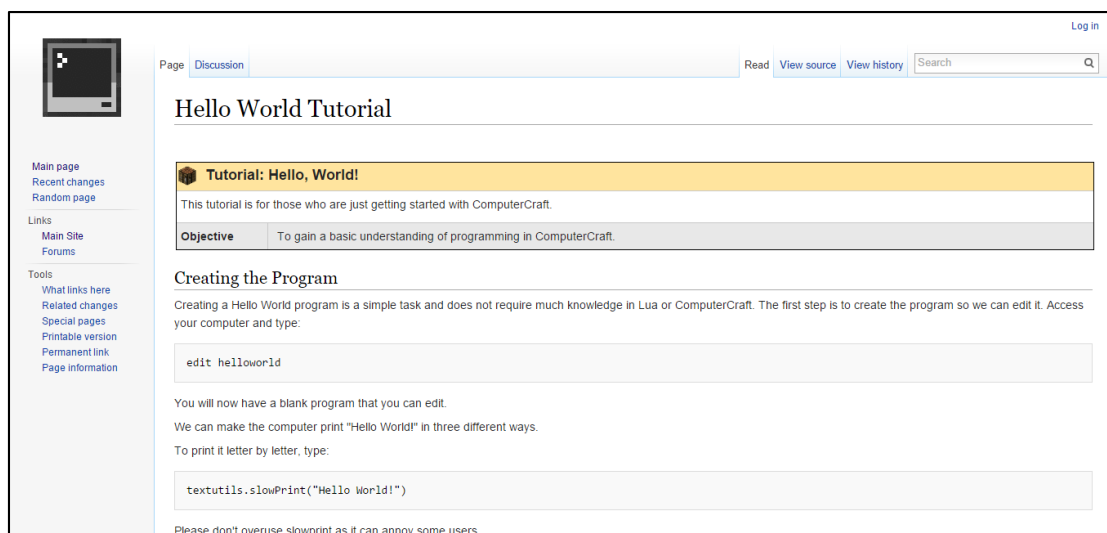
Il faudra aussi se tenir au courant des dernières mises à jours sur les outils de développement Web (HTML, CSS, PHP).

b. ComputerCraft

ComputerCraft et Minecraft reçoivent constamment des mises à jour pour ajouter, améliorer ou déprécier des fonctions ou des éléments de jeu. Il convient donc de s'informer sur tous ces changements qui peuvent affecter nos tutoriels, et la nécessité de changer le contenu de notre site.

Il faut aussi faire attention à la compatibilité des versions entre Minecraft et ComputerCraft. En effet, ComputerCraft n'utilise pas forcément la dernière mise à jour de Minecraft. Par exemple, en ce moment, ComputerCraft est à la version 1.73 et requiert la version 1.7.10 de Minecraft. Or, la dernière version de Minecraft est la version 1.8.7. Les derniers ajouts de Minecraft ne pourront donc pas être utilisés.

Il faut aussi savoir que ComputerCraft est compatible avec une grande partie d'autres mods de Minecraft. Ce mod est par exemple dans le "modpack" *Feed the Beast* comprenant un grand nombre d'autres mods, rajoutant des objets ou un nouveau type de gameplay. Les utilisateurs pourront donc continuer de jouer à Minecraft et installer d'autres mods sans avoir désinstaller ComputerCraft à chaque utilisation.



Exemple de page de tutoriel du Wiki de ComputerCraft

IV) Contenu du Wiki

a. Tutoriels

Le Wiki sera composé de plusieurs tutoriels pour guider les élèves dans lesquelles ils apprendront à programmer dans Minecraft.

Le premier tutoriel sera une aide à l'installation de ComputerCraft ainsi que comment mettre la bonne mise à jour de Minecraft.

Après, il y aura un tutoriel rapide sur les bases en programmation, en LUA, et sur ComputerCraft assez théorique.

Ensuite, nous aurons des tutoriels sous la forme : un apprentissage théorique en LUA pour la programmation, une application sur ComputerCraft, et un exercice pour vérifier si la notion est comprise.

On aura donc dans l'ordre un tutoriel pour l'affichage classique du 'Hello World', puis un sur les variables, un sur les conditions et boucles (If, While), un sur les fonctions, et un sur les tableaux.

Enfin, l'utilisateur pourra tester toutes ses connaissances dans un projet final qui nécessite d'avoir tout compris. Nous ne savons pas encore exactement en quoi il consistera, mais il faudra sûrement faire un programme pour déplacer une Turtle et lui dire de faire une action comme couper du bois et le stocker sans interruption.

Grâce à tous ces tutoriels, l'élève aura normalement la capacité d'écrire ses propres programmes. S'il s'avère que les utilisateurs manquent de certaines autres compétences pour être autonome, nous pourrions alors ajouter des nouveaux tutoriels.

b. Forum

Dans notre wiki, il y aura une partie forum. Les utilisateurs pourront s'inscrire grâce au langage PHP. Dans ce forum, il y aura plusieurs catégories.

Dans chacune de ses catégories, les inscrits seront libres de créer des articles et pourront répondre ou commenter les articles des autres.

Ce forum contiendra une catégorie "Besoin d'aide" où les apprentis codeurs pourront demander aux plus expérimentés des solutions à leurs problèmes.

Ensuite, il y aura une partie "Boîte à idée" où les inscrits pourront proposer de nouvelles piste pour faire évoluer le wiki. Dans cette catégorie, nous pourrions potentiellement ajouter un système de vote comme le site *Reddit* si le nombre d'articles devient trop important.

Il y aura aussi une catégorie "Exercices de la communauté" où tout le monde pourra proposer des exercices pour mieux aider les nouveaux codeurs.

Enfin, on trouvera une catégorie "Projets communautaire" où des personnes pourront se rencontrer pour se lancer dans un projet comme par exemple un mod. Un inscrit pourra créer un article dans lequel il présentera son projet, et demandera aux autres s'ils veulent le rejoindre dans son projet pour un travail collaboratif.

V) Evolution du Wiki

a. Directions à prendre pour l'avenir du site

Le site pourra évoluer de façon collaborative. Pour cela, on retrouvera une boîte à idées sur le forum dans laquelle chaque membre pourra donner ses envies sur un prochain tutoriel. Si l'on voit qu'une idée ressort souvent et que la réalisation s'adapte aux objectifs du site étant de faire découvrir la programmation à des novices en l'appliquant de façon ludique.

Nous espérons que ce site servira à un grand nombre de personnes souhaitant découvrir l'informatique car nos objectifs sont d'appliquer l'informatique aux jeux et d'en faire des tutoriels pour que les novices en programmation comprennent plus facilement.

Un de nos objectifs pour le futur est aussi la possibilité de décerner une certification aux personnes ayant terminé le suivi du cours avec succès. Pour cela, il faudrait rajouter une façon d'évaluer les réponses aux exercices donnés. L'une des options les plus intéressantes est l'évaluation par d'autres personnes suivant elles aussi le Wiki. Il faudra que la communauté grandisse avant de permettre ce genre d'évaluation, mais ce serait quelque chose de plus motivant pour les novices en informatique que d'obtenir une certification.

b. Diversifications

On pourra aussi proposer des tutoriels sur des logiciels de création pour que les inscrits au Wiki puissent par exemple développer leurs mods. Ces mods pourront tout d'abord être développés sur Minecraft, donc dans ce cas-là, nous écrirons un tutoriel sur le Java et l'utilisation d'API pour ensuite développer son mod personnel. Ce tutoriel ne pourra être lancé qu'à la suite de la réalisation du tutoriel sur LUA, plus facile d'accès pour un novice.

D'autres outils pourraient être intéressants à expliquer à des novices comme par exemple la réalisation d'un tutoriel sur l'utilisation de TESCS (The Elder Scrolls Construction Set), un logiciel gratuit de développement de mods sur les jeux The Elder Scrolls (Morrowind, Oblivion ou encore Skyrim). Cet aspect serait différent de la programmation pure et dure mais permettrait quand même à un début appréciant ces jeux de faire ce que bon lui semble comme par exemple rajouter une quête, se créer un équipement personnalisé, créer une maison à un endroit de la carte. Ce tutoriel serait indépendant des tutoriels sur Minecraft, mais réservés à des personnes ayant quelques notions en anglais car ce logiciel ne possède pas de version française et est assez complexe.

Projet Tuteuré S2

31 mai 2015

Diagramme de

3

