

Bureau d'études : Rapport Semaine 2

Groupe : K-A-C-A-Team

Membres :

- Kellian BALLENTINE*
- Alexandre LOUVEAU*
- Corentin MADRE*
- Anthony BORDEAU*

I) Rapport d'activité

Premièrement, nous avons essayé de reprendre notre installation du jeu, de NetBeans et de Pogamut afin que tout fonctionne. La solution est d'installer le jeu en anglais. En effet, si le jeu est en français il y a des fonctionnalités qui ne fonctionnent pas (par exemple, le raycasting).

Nous avons aussi réglé le problème provenant de la Javadoc : il faut installer JDK 1.7 au lieu du 1.8. En faisant cela, la Javadoc fonctionne à peu près : il reste cependant un warning. Nous décrirons la procédure complète dans la partie suivante (voir partie II).

Nous avons donc pu faire des tests sur le raycasting, ce que l'on avait pas pu faire la semaine d'avant car cela ne fonctionnait pas.

Nous avons tous installé GitBash pour Windows. Nous allons tous l'utiliser sous peu, même si pour l'instant on utilise plutôt les fonctionnalités de base disponibles sur GitHub.

Plusieurs membres du groupe ont aussi lu la documentation de Pogamut.

Dans un autre temps, nous avons émis des hypothèses sur ce que l'on allait faire comme bot. Ces hypothèses constituent, en gros, la spécification pour la conception du bot de notre équipe. Cependant, si lors du codage nous nous rendons compte de problèmes non décelés, nous pourrions modifier cette spécification (nous détaillerons l'idée dans la partie 3).

Nous n'avons pas encore réparti les tâches car nous réfléchissons encore à comment nous allons paramétrer le profil du joueur que notre bot doit incarner.

II) Procédure d'installation

Nous avons commencé par regarder le tutoriel du site de Pogamut. Cependant, en suivant les étapes décrites dans le tutoriel, Pogamut ne marchait pas. En effet, le site de Pogamut ne mentionnait pas certaines informations, qu'il nous a fallu chercher, afin de faire en sorte que cela fonctionne.

Le site disait qu'il fallait une version de JDK 1.7 ou plus alors qu'il ne fallait que la 1.7 car, avec la 1.8, la Javadoc de Maven ne fonctionne pas bien. La semaine précédente nous travaillions

avec la version 1.8 de JDK et pour que la Javadoc ne nous gêne pas il nous fallait modifier le 'pom.xml' ou rajouter l'option '-Dmaven.javadoc.skip=true verify' avant de clean et build notre projet. Cependant nous ne devons pas mettre l'option avant le chargement d'un nouvel archétype car cela empêchait son chargement.

Le raycasting ne fonctionnant pas, nous avons donc cherché pendant plusieurs jours pourquoi. Finalement, nous nous sommes rendu compte que cela venait du fait que notre jeu était en français et qu'il le fallait en anglais.

Nous avons donc désinstaller et réinstaller l'ensemble de l'environnement de programmation un bon nombre de fois avant de trouver d'où venaient les problèmes. Cela nous a pris un temps considérable, mais nous avons aujourd'hui une version fonctionnelle de Pogamut et même si il reste quelques warnings à la compilation au niveau de la Javadoc, mais ils n'empêchent en rien le bon fonctionnement du bot.

Voici les étapes que l'on a fait dans l'ordre :

- Mettre les paramètres de contrôle utilisateur (Panneau de configuration → Compte d'utilisateur) à ne pas m'avertir ;
- Installer UT2004 en anglais et le patcher si il n'est pas à la bonne version ;
- Installer Maven (dézippé et collé le dossier dans le dossier programme) ;
- Mettre le dossier bin de Maven dans la variable d'environnement Path (Panneau de configuration → Systèmes → Paramètres systèmes avancées → Paramètres systèmes avancées → Variables d'environnement) ;
- Installer la version 1.7 de JDK jumelée à Netbeans 8.0.2 (ou JDK 1.7 puis Netbeans) en tant qu'administrateur ;
- Créer la variable d'environnement JAVA_HOME où l'on renseigne le dossier où est installer JDK ;
- Ajouter le dossier bin du JDK dans la variable d'environnement Path ;
- Lancer NetBeans une fois en tant qu'administrateur ;
- Installer Pogamut en tant qu'administrateur (c'est à dire en lançant le jar par une fenêtre cmd lancer en tant qu'administrateur) ;
- Lancer Netbeans en tant qu'administrateur ;
- Faire un nouveau projet avec un archétype Maven ou ouvrir un existant, normalement cela fonctionne à part un warning non gênant lorsqu'on build.

III) Spécification de notre bot

Nous souhaitons définir un profil de joueur paramétré pour notre bot. Ce profil permettra de définir les paramètres de notre bot par rapport aux caractéristiques du joueur réel qu'il doit incarner (assurance du joueur, skill du joueur, etc).

Les paramètres, que l'on mettra dans ce profil, nous permettront de faire prendre des décisions à notre bot (engager un combat, se ravitailler, fuir vers de la vie, de l'armure ou pour aller se cacher le temps d'être en mesure de combattre, etc) en cohésion avec le type de joueur qu'il incarne.

Nous pourrons aussi permettre au profil de s'améliorer tout au long d'une partie en mettant des paramètres évolutifs et en faisant en sorte qu'il puisse analyser les capacités et les habitudes de ses adversaires afin d'adapter sa manière de jouer. Nous essaierons donc de coder un bot qui a la capacité de s'adapter en apprenant.

IV) Remarques

Nous avons eu des disponibilités limitées durant la semaine donc nous n'avons pas pu travailler tous les jours. Cependant, nous avons maintenant notre environnement de travail (logiciels) qui est fonctionnel et nous avons aussi une idée de ce que l'on compte faire.

Pour ce qui est du rapport précédent, nous avons appris que vous ne l'aviez pas reçu en début de semaine dernière. Nous vous l'avons renvoyé en utilisant une adresse de l'UPS car avec GMAIL il arrive que cela ne fonctionne pas. Nous espérons que vous l'avez bien reçu ce coup ci. Si vous le souhaitez nous avons la preuve des deux envois de notre premier rapport.