1. Introduction

Dans le cadre de la première année de bachelier en Sciences Informatiques à l’Université de

Mons, un projet de programmation est proposé aux étudiants. Le but de celui-ci est de

participer au développement d’un logiciel de taille plus conséquente que ceux proposés le

long de l’année, d’y appliquer des notions d’algorithmique et de programmation orientée objet en se basant sur les cours enseignés.

Le projet de cette année est de créer un jeu de puzzle similaire à Baba Is You [1], qui est lui même inspiré du Sokoban. La particularité de ce jeu se situe dans les règles. Celles-ci sont dynamiques et peuvent être modifiées au cours de la partie.

Afin d’initier les étudiants au travail collaboratif, la charge de travail est répartie par groupe de deux étudiants de BAB1.

2 répartition des taches

Lors de la création du groupe, il a été décidé de séparer les taches en trois. L’un devait se charger de faire un début d’interface graphique et les deux autres le moteur de du jeu.

Mais étant donnée l’abandon de l’un (nom) et le manque de motivation de l’autre (nom ), je me suis retrouve à faire le projet tout seul chose qui n’a pas été facile

1. Choix de la modélisation d’un niveau

Comme tous jeunes programmeur , plusieurs implémentation ont été la considere.

* La premiere était la suivante : un niveau est constitue d’un tableau a de dimension de chaine de caractère où dans lequel chaque entite est represente par son nom(‘baba’, ‘Wall ’, ‘stop’  ….) . cette implementation a été brievement essayee avant d’etre abandonee a cause de sa complexite et des giga condition observee dans le code
* La seconde était la suivante : le niveau est constitue d’un tableau a deux dimention de Words(enumerateur contenant l’ensemle de toutes mes entitees qui a pour constructeur un ‘type’, ‘un schemin relatif vers son image’ et ‘un nom’) . cette approche a été tres bien aborder mais souffrais d’un probleme. En effet je n’avais pas la possiblite de mettre plusieurs entitees sur une meme case
* La troisieme était la suivante :le niveau est constitue d’un tableau a deux dimension de Arraylist d’entitees(**Arraylist<Entities>[][] grid**) . ainsi ma classe entities prends comme constructeur une un Block(TypesofWords), une position(creation de la classe position), une direction(int) et un attribut d’instance imagevieux(ImageVieux) qui nous permet d’avoir l’imageVieux de chaque entitees.

1. non utilisation de l’heritage pour materialiser certains block

mais l’erreur a été remarquée assez tardivement et pour éviter une charge supplémentaire de travail, elle n’a pas été rectifiée.