Document de Conception

Projet Générique «Action-RPG»

Présenté à Jean-Christophe Demers

Par Renaud Bigras

20 Février 2012

# Présentation générale

Le projet «Action-RPG» est un jeu vidéo pour ordinateur mélangeant des éléments des jeux d’action, jeux d’aventure et des jeux de rôle. C’est un jeu temps réel vu de haut (top-down), dans lequel le joueur incarne un aventurier à la recherche de reliques dans des donjons diverses peuplés d’ennemis. Il pourra changer son équipement et atteindre de nouveaux niveaux de compétence au fur et à mesure que le jeu progresse. Un élément de labyrinthe est présent sous la forme des donjons, d’où l’élément d’exploration.

# Présentation détaillée

L'utilisation du langage Java et des librairies externes

Le projet «Action-RPG» est réalisé à l'aide de la plateforme Java et de la librairie interne Graphics2D. Le projet de base vise à utiliser à ses pleines capacités les fonctionnalités de Graphics2D pour avoir un jeu 2d incorporant des « tilesets » et des « sprites » en tant que la majorité de ses graphiques. Le projet de base n’intégrera ni son ni support pour des périphériques de contrôle autres que le clavier. Le développement tiendra compte de la possibilité d’intégrer des libraires audio tels que FMod pour le son et JInput pour les contrôles.

L’écran principal

L’écran principal englobe tout ce qui est présenté lors du démarrage du jeu. Le joueur sera présenté avec un écran titre et des choix de jeu et d’options. Le joueur pourra débuter une nouvelle partie, charger une partie précédente ou accéder aux options du jeu. Débuter une nouvelle partie affichera le menu de sélection de niveau, tous encore incomplets, et sans niveaux de compétences ou d’objets acquis au préalable. Charger une partie lira un fichier qui pourra être créé entre les niveaux contenant simplement l’inventaire du joueur, son niveau de compétence et la liste des niveaux qu’il a complété. Les options seront tout simplement le son et la musique, mais possiblement la résolution graphique également.

La scène du jeu et ses éléments

La scène sera d’une largeur et d’une hauteur prédéfinie; elle ne fera pas le niveau au complet, évidemment. Un objet « caméra » s’occupera de tenir compte d’où dans le niveau la scène doit être. La largeur et la hauteur d’affichage seront les mêmes peu importe la résolution du jeu; ceux-ci équivaudront à un certain nombre de cases sur la grille du niveau (voir Les niveaux et leur construction). La hauteur d’affichage équivaudra soit à la hauteur complète du niveau ou à une fraction de celle-ci.

Un « affichage tête-haute » sera présent sur la scène. Les éléments de ceux-ci seront principalement des informations pertinentes au joueur : sa vie, son niveau de compétence, le nombre de clefs qu’il possède, les objets qu’il a d’équipé, etc. Les éléments seront placés à des endroits stratégiques pour ne pas obstruer la vue, tels les coins de l’écran.

Un système d’animation sera présent, dans lequel si un objet / acteur entreprend une animation, il est possible d’afficher celle-ci comme importante, c'est-à-dire que le jeu ne progressera pas tant que l’animation ne sera pas terminée.

Les niveaux et leur construction

Les niveaux auront une taille prédéfinie, possiblement à partir de certains choix de grosseur équivalant à un nombre prédéterminé de cases horizontales et verticales sur un système de grille. Chaque niveau, ou plutôt donjon, consistera d’une série de salles, ou plutôt écrans, liés les uns aux autres. La différence de taille pourra s’appliquer entre les salles et non pas être prédéterminée pour le donjon en entier. Les donjons pourront aussi être composés de plusieurs épaisseurs de salles reliés ensemble, sois des étages.

Les salles eux-mêmes seront disposées en forme de grille, avec des possibilités pour des cases spéciales. Du plancher, des trous, des murs, des pièges, etc... Le contour de la grille sera toujours des murs automatiquement, et les portes devront se trouver adjacentes à des murs.

Il sera aussi ajustable, dans les paramètres d’une salle, des options pour empêcher la sortie de la salle si, par exemple, le joueur n’a pas activer un objet, ramasser tous les objets ou détruit tous les ennemis.

Le système de fichier pour les niveaux

Les niveaux seront sauvegardés dans des fichiers encodés en tant que séries de matrices. Une matrice par couche, une erreur étant lancée si la taille de la matrice n’est pas équivalente à la définition de taille du niveau;

Une entrée donjon possèdera plusieurs entrées salles, qui eux possèderont des informations tel l’étage, le fond, les tuiles à utiliser, la grandeur de la salle et un lien vers une autre salle pour le nord, sud, est et/ou ouest.

L’éditeur de niveau

Un éditeur de niveau sera potentiellement inclus, pour des raisons de développement. Celui-ci servira à rendre la réalisation des niveaux plus simple. Sa structure de base sera la même que celle du jeu, mais les éléments du jeu comme tel ne seront pas présent et seront remplacer par des éléments d’édition. L’éditeur lira et écrira les définitions du niveau, et sauvegardera les matrices du niveau dans le fichier approprié. L’éditeur aura accès aux données graphiques du jeu pour représenter correctement les tuiles du niveau et les acteurs du jeu.

Les acteurs

Les acteurs sont le joueur, les objets, les ennemis, les portes, les obstacles et autres qui ne font pas partis du décor (fond, sol, avant, arrière, etc.). Un acteur pourra avoir d’autres acteurs qui si rattache, comme par exemple le joueur et son arme. Chaque acteur aura des attributs génériques, tel sa grandeur, ca largeur et son « hitbox », sa zone de contact. Par exemple, la zone de contact du joueur définit ou le joueur doit être touché pour être blessé, de même pour les ennemis. La zone de contact des armes servira à savoir si le joueur touche à un ennemi, etc.

Les acteurs sont placés sur la grille mais bougent librement.

Le joueur et les ennemis

Le joueur lui-même possède plusieurs caractéristiques, tel son inventaire, son niveau, sa vie, sa vie maximale, sa magie, sa défense et sa vitesse. Un système d’expérience sera aussi présent, laquelle augmentera en combattant des ennemis ou en complétant d’autres objectifs. Le joueur pourra décider par lui-même, par l’entremise d’un menu d’augmentation de niveau, quelle caractéristique il veut améliorer. Les ennemis aussi auront des caractéristiques, statiques cette fois-ci. Ils n’auront pas d’inventaire, de niveau ou d’expérience, mais tous les autres attributs seront présents. Ceux-ci auront soit de vagues mouvements prédéfinis et ignoreront le joueur ou changeront de mouvement lorsque le joueur s’approche. Il y aura aussi des ennemis plus fort, de type « boss » présents dans la plupart des niveaux.

Cas d’usages UML



Diagramme de classes UML

