

Manuel du projet irover

Lundi 9 Novembre 2015

Manuel du project irover.

Table des matières

1 Introduction

Le projet a pour objectif la création d'une application graphique où un petit personnage (robot) réaliserait une tâche particulière. Pour cela nous avons imaginé un scénario de mini jeu semblable au jeux en deux dimensions où notre robot prendrait la forme d'un héros sans peur et sans reproche. Le manuel suivant se composera de plusieurs parties : La première vous aidera à comprendre le but du jeu ses règles et l'histoire de notre petit monde. La seconde décrira comment installer l'application et les différentes options d'installation. La troisième partie constituera la documentation utilisateur. La quatrième quand elle sera dédiée à la description détaillée de l'infrastructure logicielle sur la forme d'une documentation dédiée aux développeurs ou utilisateurs confirmés et enfin nous vous proposerons un tutoriel retrassant une simulation pour illustrer le rendement de notre jeu.

2 iRover, l'histoire d'un héros

Dans la vallée perdue du Vertou, là où nul aventurier n'a plus mis les pieds depuis des années. On raconte qu'il y a plusieurs siècles, un grand pirate du nom de Stevy J. y aurait caché un trésor : "le chamalo magique". Notre héros, jeune et intrépide aventurier du nom de Rover est à la recherche de cet artefact et se voit donc lancer dans cette grande quête afin de retrouver cette sucrerie antique. Cependant, la cette mystérieuse vallée est habitée par d'étranges créatures, des chou-kêtes. Cette horde de créature est farouchement attachée à leur territoire et n'hésiteront pas à attaquer quiconque y pénétrera.

But du jeu :

L'objectif principal du jeu (ou plutôt de notre jeune ami) est de ramasser tous les trésors présents sur le terrain afin d'atteindre gloire et richesse et surtout un jour trouver le chamalo magique.

Ces coffres seront disséminés sur la carte, souvent derrière des obstacles que le héros devra contourner. Des ennemis (les chou-kêtes) pourront également attaquer notre héros et donc un combat sans merci s'enclanchera entre héros et monstre.

Le jeu prendra fin une fois tous les coffres ramassés ou si notre héros ne peut plus continuer son aventure.

2.1 Gestion de notre héros

Rover, notre héros, est contrôlé par une intelligence artificielle d'où son petit surnom de robot. Il pourra se déplacer sur la carte, case par case et rencontrera des obstacles. Il pourra interagir avec d'autres éléments du monde.

N'ayant pas appris à nager et du à des crises de vertiges récurrentes, il ne pourra ni se déplacer sur l'eau ni grimper sur les rochers, les arbres, les murs et même les buissons !

Il pourra cependant ramasser des objets au sol (des clefs), se battre contre les ennemis et ouvrir les coffres.

2.2 Gestion des clés

Un certain nombre de clés seront présents sur la carte ou peuvent être obtenus en tuant des ennemis. Pour certains coffres, il peut être nécessaire d'avoir une clé. Une fois le coffre ouvert, la clé est consommée et le robot devra en récupérer d'autres pour ouvrir d'autres coffres.

Pour récupérer une clé, il faut que le robot se trouve sur la même case que celle-ci, elle disparaît alors de la carte et vient s'ajouter à son inventaire de clés.

2.3 Gestion des ennemis

Certains ennemis seront présents sur le terrain et mettront le robot en difficulté. Le robot devra alors combattre ces ennemis s'il les rencontre afin de rester en vie.

Lorsque le robot se trouve à côté d'un ennemi, il est obligé de le combattre. Cela se traduit par un algorithme de combat qui fera gagner le duel au robot ou à l'ennemi.

Une fois l'ennemi vaincu, celui-ci disparaît du terrain mais dans le cas contraire, notre héros ne sera plus en mesure de continuer et notre jeu prendra fin.

3 Manuel d'installation

3.1 makefile

3.1.1 gestion des systèmes d'exploitation

3.2 .config ? ? ? ? ? ? ?

4 Documentation utilisateur

4.1 Le robot

Le robot se déplace uniquement horizontalement et verticalement, il ne peut donc pas se déplacer en diagonale. Il ne pourra se déplacer que d'une case à la fois. Le robot disposera d'un algorithme de déplacement qu'on appellera de son petit nom : le path finding. Il lui arrivera de temps en temps d'avoir plusieurs choix de déplacement, et cet algorithme lui permettra de choisir où aller pour atteindre son objectif.

Il peut également avoir conscience de son environnement.

4.2 L'environnement

La carte sur laquelle le robot se déplace est construite à l'aide de Tiled, il s'agit d'un logiciel permettant de créer des cartes case par case. Pour les cartes modélisées, on distingue deux types de cases :

1. les cases où le robot peut se déplacer
2. les cases où le robot ne peut pas se déplacer

La carte est sous forme d'un fichier XML que le programme interprète pour renvoyer une carte exploitable par le robot.

L'environnement servira de base également pour la disposition des autres entités présente, coffre, clef ennemie. Ceux-ci sont disposés aléatoirement sur la carte à des endroits accessibles par le héros mais ils ne peuvent pas se déplacer. La rencontre avec un de ces éléments entraîne la gestion des événements.

4.3 La gestion des événements

Le robot, notre ami rover, devra faire face à beaucoup d'événements, ouvrir un coffre, se battre contre un ennemi, ramasser une clef, se cogner contre un mur etc.. Tout ceci est géré par le comportement de chaque objet mais aussi pour sa hiérarchie sur la carte.

4.3.1 Rencontre avec un ennemi

Un ennemi et notre héros devront avoir la même importance physique : ils ne peuvent pas se superposer. Un ennemi et notre héros n'auront besoin que d'un regard pour enclancher le combat ! 1 case de différence.

4.3.2 Ouverture d'un coffre

Un coffre et notre héros n'ont pas la même importance physique : le héros devra pouvoir "marcher" sur le coffre. Un coffre et notre héros devront impérativement être sur la même case pour que le coffre puisse s'ouvrir et ce dernier, le héros, devra avoir une clé pour que cela soit possible.

4.3.3 Ramasser une clé

Une clé et notre héros n'ont pas la même importance physique : le héros devra pouvoir "marcher" sur la clé.

5 Documentation logicielle

5.1 Le robot

5.2 L'environnement

6 tutoriel