

Rapport TPA phase 1

Groupe 3 Bleu : Sokoban

Goron Nathan, De La Rosa Louis-David, Basset Emilien, Demé Quentin



Table des matières

1	Introduction	2
2	Cahier de charges	3
2.1	Déroulé fonctionnel , objectif	3
2.1.1	Phase 1 :	3
2.1.2	Phase 2 :	3
2.2	Contexte de réalisation	3
2.3	Découpage	3
2.4	Pré-requis d'utilisation	3
2.5	Règles du jeu	4
2.6	Fonctionnalité additionnelles	4
2.7	Calendrier , dates fixées	4
2.8	Internationalité	4
2.9	Priorités , importances relatives des fonctionnalités	4
3	détails du fonctionnement	5
3.1	Moteur de jeu	5
3.2	Astar	5
4	Perspectives d'améliorations en phase 2	6
5	Session de tests	7
6	Résumé , conclusion de la phase 1	8

1 Introduction

Parmi les projets proposés , nous avons fais le choix de ne pas retenir les projets n'étant pas des jeux de peur qu'ils limitent notre créativité et notre motivation quant a son développement. Le jeu de Sokoban nous paraissait être un projet ambitieux et intéressant de par la difficulté dans l'implémentation de son IA. Il s'agissait donc de développer un Jeu de Sokoban : Un jeu de puzzle dans lequel un personnage déplace un certains nombres de caisse sur des emplacements dans un niveau fermé ,le joueur gagne la partie si il parvient a boucher toutes les emplacements avec des caisses.

2 Cahier de charges

2.1 D  roul   fonctionnel , objectif

Le programme sera   crit en Python a l'aide de la librairie Pygame et r  pondra au crit  res suivants d'ici la version finale de son d  veloppement :

2.1.1 Phase 1 :

-Le jeu sera jouable par humain via un prototype non-d  finitif(changements possibles en phase2) , le joueur pourra se d  placer dans un niveau import   au format .slc dans lesquels figureront 5 types d'  l  ments diff  rents : l'  l  ment personnage que le joueur contr  le pour d  placer les caisses , les cases murs , les cases socles ou doivent   tre d  plac  es les caisses afin de terminer le niveau , et les cases vides sur lesquelles le personnage ou les caisses peuvent   tre d  plac  es

2.1.2 Phase 2 :

Le jeu se verra ajouter une fonctionnalit   de r  solution automatique gr  ce a l'algorithme A* et aux heuristiques, cet algorithme proposera un chemin optimal(dans la plupart des cas en fonction de la difficult  )pour compl  ter le niveau en le moins de coup possibles.Cet algorithme devra   tre anytime , c  ad capable de compl  ter un niveau quelque soit son point de d  part. Les fonctionnalit  s additionnelles facultatives sont list  es dans la cat  gorie "Fonctionnalit  s additionnelles"

2.2 Contexte de r  alisation

Ce projet est r  alis   dans le cadre de l'  valuation de la valeur "TPA" en L2 info 2016-2017 de l'universit   de Caen

2.3 D  coupage

Le programme sera compos   des modules suivants : — Le module sokoban qui assurera le fonctionnement g  n  ral du programme (moteur de jeu, conditions de victoire,interface graphique, importations des niveaux) — Le module classes qui comprendra toutes les classes et m  thodes n  cessaire au fonctionnement du jeu — Le module constantes stockant les importations d'images pour les personnages et les cases ainsi que les variables n  cessaire pour les variations de tailles de niveaux a l'  cran. Ce d  coupage pourra   tre modifi   en fonction des choix de fonctionnalit  s additionnelles

2.4 Pr  -requis d'utilisation

Le jeu fonctionnera sous n'importe quel syst  me muni de python(2.7 a 3.5) et de la librairie Pygame. Le jeu sera intuitif d'utilisation et ne n  cessitera aucune connaissance informatique avanc  e particuli  re .

2.5 Règles du jeu

le joueur déplace un personnage dans un niveau constitué de murs , de cases vides , de caisses et de socle pour les caisses. Il faudra déplacer les caisses dans le niveau afin de couvrir tous les socles pour terminer le niveau. Le joueur peut que pousser (pas tirer) , il ne peut pousser qu'une seule caisse a la fois Le joueur peut se retrouver bloqué

2.6 Fonctionnalité additionnelles

6 Fonctionnalité additionnelles : Les fonctionnalités suivantes seront susceptible d'être implémentée d'ici la version finale du programme : — Menu permettant le choix du niveau et la modifications d'éventuelles options (sons , design de la fenêtre...) — Bouton "undo" permettant d'annuler le dernier mouvement du joueur — Système de sauvegarde pour conserver son avancée dans un niveau

2.7 Calendrier , dates fixées

Le développement du jeu est découpé en deux phase : -phase 1/1er semestre : réalisation d'un jeu jouable par un humain et recherches sur l'ia (A* , heuristiques) , importation des niveaux -phase 2/2nd semestre : programmation de la résolution automatique anytime

2.8 Internationalité

8 Internationalité : Le jeu ne sera pas traduit et sera utilisable uniquement en Français.

2.9 Priorités , importances relatives des fonctionnalités

Le fonctionnement du jeu sans IA et la possibilité d'importation des niveaux sont prioritaires, la résolution automatique ne sera programmée qu'ensuite .Les fonctionnalités additionnelles ne seront développées qu'une fois les objectifs des eux phases remplies.

3 détails du fonctionnement

3.1 Moteur de jeu

3.2 Astar

4 Perspectives d'améliorations en phase 2

5 Session de tests

6 Résumé , conclusion de la phase 1