

Projet commun POO – COO : Puru Puru Digger

Date de remise du TM impérative : vendredi 4 avril 2014 par binôme

Vous trouverez dans ~/Bibliotheque/M2013-Projet/ un exécutable du jeu Puru Puru Digger. Cet exécutable contient toutes les fonctionnalités minimales demandées.

0 – Le document

Le TM est à réaliser en binôme.

Un dossier papier vous est demandé impérativement pour le vendredi 4 avril 2014.

Sur chaque page, vous devez indiquer vos noms et vos prénoms, le nom de votre groupe. Les pages doivent être numérotées.

Le document ne doit pas être manuscrit vous pouvez utiliser un AGL.

Le document doit être agréable à lire (font 11 ou 12), chaque diagramme doit avoir un titre, être commenté ...

1 – Principe

Vous pouvez trouver un exemple du jeu original sur le site de Big Fish Games <http://www.bigfishgames.fr/jeux-en-ligne/8638/puru-puru-digger/index.html>

Dans ce jeu, il s'agit de déplacer un mineur dans une matrice afin d'atteindre un score tout en évitant certains objets (bombes).

Au départ d'une partie, la matrice contient 4 types de cases :

1. des cases numérotées de 1 à 6 (la plupart des cases sont de ce type)
2. des cases numérotées de 1 à 6 visiblement de type bonus (ces cases sont de couleur différentes (peu de cases sont dans ce cas))
3. des cases "bombe"
4. une case contenant le mineur

Le joueur peut déplacer le mineur dans les 8 directions autour de lui (gauche/droite/haut/bas/bas-gauche/bas-droit/haut-gauche et haut-droit). Pour cela, il suffit de cliquer dans la case voisine correspondante. Quelle que soit la direction choisie, le mineur se déplacera de n cases dans cette direction (n étant la valeur inscrite dans la case cliquée).

Attention, le joueur ne peut pas se déplacer dans une direction si la première case dans cette direction n'a pas de numéro.

A chaque fois que le mineur passe pour la première fois sur une case ; la case est découverte et la valeur inscrite (multipliée par 10) dans la case est ajoutée au score courant du niveau.

Le gain d'une case est multiplié par deux si la case est de type bonus.

Pour chaque niveau, lorsque le joueur atteint ou dépasse un score donné (l'objectif), le niveau est gagné et on passe au niveau suivant.

Initialement le joueur a 3 vies. Le joueur perd une vie si :

1. le mineur passe sur une bombe,
2. le mineur repasse sur une case déjà découverte, ou
3. le mineur "sort" de la matrice.

2 - Objectifs

Trois niveaux de fonctionnalités sont proposés pour ce projet. Attention, avant de passer aux fonctionnalités suivantes toutes les autres fonctionnalités doivent avoir été implémentées.

Fonctionnalités minimales :

1. Écran d'introduction du jeu
2. Écran de menu permettant via des boutons :
 1. de démarrer le jeu et de le quitter
3. Écran de jeu permettant d'afficher :
 1. un arrière-plan
 2. le mineur
 3. les bombes
 4. les cases numérotées
 5. le score du niveau courant et le score à atteindre
 6. nombre de vies restantes
 7. Le nombre de points total (sur la partie)
 8. Des écrans de transition lorsque l'on perd une vie, lorsque l'on termine un niveau et lorsque la partie est terminée
9. Pendant le déroulement du jeu :
 - Déplacement du mineur après un clic sur une case numérotée et valide (voisine)
 - 3 niveaux jouables

IMPORTANT : si toutes ces fonctionnalités n'ont pas été implémentées, cela ne signifie pas nécessairement que vous n'aurez pas la moyenne. D'autre part, toute forme de triche (plagiat) sera **TRÈS** sévèrement sanctionnée !

Fonctionnalités supplémentaires à réaliser dans l'ordre indiqué :

1. Gestion du temps : l'objectif du niveau doit être atteint en un temps donné. Un bonus de points peut être accordé en fonction du temps restant
2. Animation des déplacements
3. Ajout de bonus cachés dans les cases numérotées permettant de gagner du temps, des points ou des vies
4. Gestion des meilleurs scores avec affichage d'un écran des meilleurs scores depuis le menu
5. Ajout de son (fond sonore, déplacement, case découverte,...)

Fonctionnalités bonus (non ordonnées) :

- Éditeur de niveau
- Générateur aléatoire de niveau
- Vérificateur de faisabilité de niveau
- Nombre indéfini de niveaux
- Changement de charte graphique (modifiable depuis l'écran d'options)
- Support multi-langues (modifiable depuis l'écran d'options)

Cette liste n'est pas exhaustive, vous pouvez choisir d'implémenter d'autres fonctionnalités bonus.

Questions :

1. Réaliser le diagramme de contexte statique
2. Réaliser le diagramme de classes
3. Etablir la liste des événements externes
4. Etablir la liste des événements résultats
5. expliquer les règles que vous implémenterez pour la gestion du temps et ces bonus cachés.