

Compte Rendu CPP

Auteurs : Huiling BAO

Heyang LI

- Description de l'application développée.

Nous avons réalisé un jeu de la taupe en utilisant la bibliothèque SDL pour la visualisation. Le rôle est assez simple, tout ce que vous serez faire est de cliquer sur la taupe qui a sauté de la fosse. De plus, nous avons également mis en place des bombes. La cliquer en mode différent conduit au résultat différent.

Ce qui rend notre projet spécial, c'est que notre jeu a trois modes:

1, Le mode normal. Dans ce mode, le jeu est limité pendant 60 secondes et la note est calculée. La note dépend du nombre de fois que vous avez cliqué sur la taupe.

Une taupe de niveau 1 coûte 1 point, celle de niveau 2 coûte 2 points. Mais attention! 2 points seront déduits si vous cliquez accidentellement sur la bombe. Le jeu se termine lorsque le temps est écoulé.

2, Le mode révolutionnaire, où les joueurs atteignent une certaine note et le jeu passera automatiquement au niveau suivant. Il y a aussi une limite sur le temps et la méthode de notation reste la même que le mode normal ci-dessus.

3, Le mode infini, sans limite de temps, mais si le joueur clique accidentellement sur la bombe, le jeu se termine au lieu de la perte des notes.

- Mettre en valeur l'utilisation des contraintes.

- 8 classes :

Window, Rôle, Bombe, Taupe, Game, ModeSimple, ModePromu, ModeInfinie

- 3 niveaux de hiérarchie : Game <- ModeSimple <- ModePromu <- ModeInfinie

- 2 fonctions virtuelles DrawToRenderer(SDL_Renderer*, SDL_Texture*) dans la classe Rôle pour afficher les images (la bombe et la taupe)

Move(SDL_Texture*, SDL_Texture*, int) dans la classe Game pour générer et contrôler le mouvement des taupes et des bombes

- 2 surcharges d'opérateurs ;

- surcharges d'opérateurs <+> dans la classe Taupe pour accumuler la note
- surcharges d'opérateurs <+> dans la classe Bombe pour accumuler la note

- deux conteneurs de la STL ;

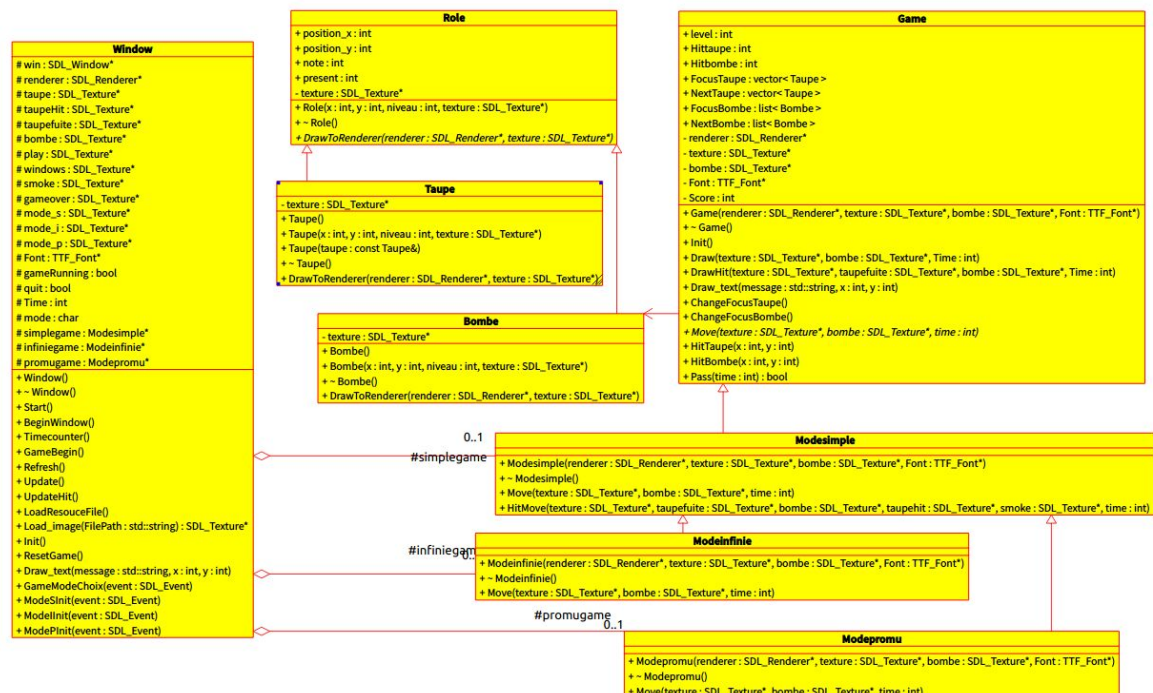
vector<Taupe> FocusTaupe;

vector<Taupe> NextTaupe;

list<Bombe> FocusBombe;

list<Bombe> NextBombe;

- Diagramme UML de l'application.



- Procédure d'installation (bibliothèques ...) et d'exécution du code.

Nous avons appliqué la bibliothèque SDL pour l'affichage graphique de l'interface de jeu, il faut donc d'abord installer la bibliothèque SDL sur l'ordinateur (Les machine de l'école déjà installés). Donc il faut simplement utiliser le makefile pour l'exécution.

- Les parties de l'implémentation dont vous êtes les plus fières.

La réalisation du jeu n'est pas très difficile mais c'est l'utilisation de la bibliothèque SDL qui fait une bonne expérience utilisateur.

Donc les deux classes: <Game> <Rôle> constituent la structure principale de notre code de programme. Il y a trois classes dérivées (3 niveaux de hiérarchie) de la classe <Game> qui représentent les trois mode que nous avons réalisé.

Le chargement de l'image d'arrière-plan du jeu et le rendu de l'interface de jeu sont implémentés dans la classe <Window>.C'est pour rendre le programme plus lisible et concis que nous avons écrit cette classe.Tous les matériaux que nous avons besoin (la fenêtre, le renderer,les texture pour les différents images) sont chargés dans cette partie.

Et il y a deux classes dérivées de la classe <Rôle> qui représentent la taupe et la bombe.