

## Rapport : Projet de Programmation Avancée « Arkanoid »

### Intruduction :

Le but de ce projet est de concevoir une version qui respectera les règles de la version originale de (Arkanoid) en utilisant le maximum de paradigmes de programmation vus en cours, pour mettre en évidence l'élégance du code, le design, la modularité et l'optimisation.

### Utilisation :

Grace à l'outil cmake, la compilation est automatisée, donc il suffit d'exécuter sur le terminal « **cmake .** » pour avoir le fichier Makefile qui est à exécuter à son tour en tapant **make** et puis aller dans le repertoire */bin* et lancer l'exécutable « *./Arkaoid* »

### Implementation :

le projet a été devisé en plusieurs sous partie afin de bien faciliter les tâches et se localiser rapidement en cas de modification ou ajouts d'une composante et de retracer rapidement les erreurs de compilation. Notre implémentation est modulaire et permet de la modifier et l'étendre facilement.

seule la copie de sprite est utilisée à l'aide de `SDL_BlitSurface`.

Les niveaux sont stockés dans des fichiers textes.

### Paradigmes de Programmation :

l'héritage et le design pattern sont le plus utilisé dans notre implémentation, par exemple les médiateurs qui étaient plus au moins facile à intégrer selon l'utilisation.

### Travail réalisé :

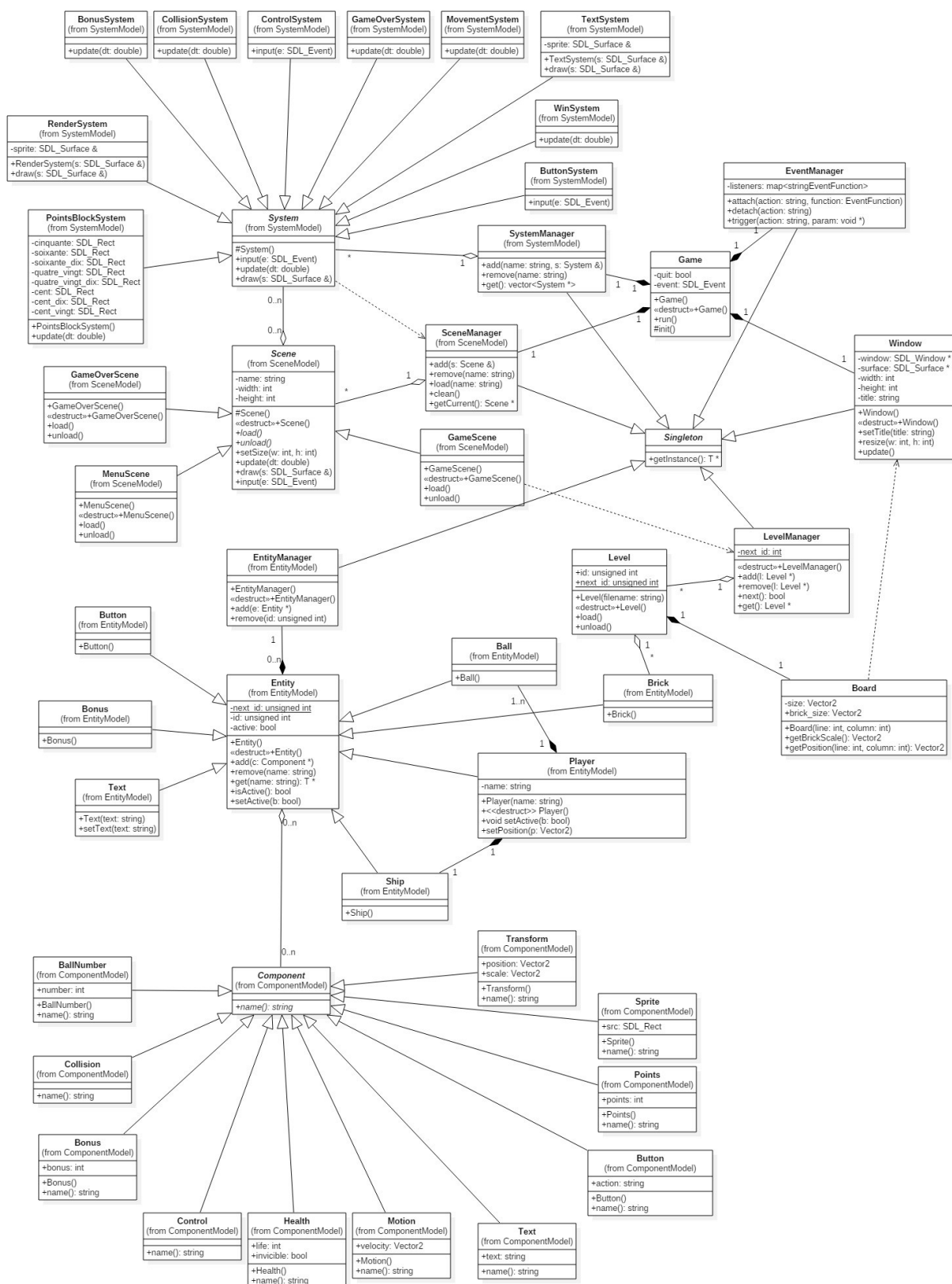
participation à l'ensemble de projet et à sa contribution, la principale tâche est de créer et générer les niveaux sur la console et leur gestion, pour cela, chaque fichier niveau est stocké dans un fichier texte qui contient respectivement : position(line, column) pour la taille variable de niveau, le point de la brick, la position(x, y), le nombre de coup et le point bonus.

les données sont récupérées en fonction du besoin et permettent de générer les niveaux suivant l'état de jeux pour passer d'un niveau à un autre.

Gestion de niveaux tailles variables est traité dans la suite.

## UML :

Nous avons conceptualisé le projet en utilisant un diagramme de classe permettant de représenter toutes les classes et leurs liens. Figure ci-dessous, qui est également disponible sur le dépôt git.



## **Tâche Générale Réalisées :**

- ✓ taille du plateau de jeu variable (original 13x25 briques)
- ✓ plateau “cylindrique” (pas de bord gauche/droite)
- ✓ gestion score
- ✓ plusieurs niveaux (fichier)
- ✓ niveaux tailles variables
- ✓ l’héritage
- ✓ polymorphisme
- ✓ design-pattern
- ✓ containers STL
- ✓ template
- ✓ et autres amélioration ...

## **Travail collectif :**

étant un groupe de deux personnes dans deux groupe différents ,l'utilisation de moyen de partage à distance « git » nous a bien aider à travailler en collaboration et à repartir les tâches et d'échangé en temps réel nos données,cela nous a permet de se debugger et détecter les éventuelles erreurs afin d'aboutir à une amélioration du code.

## **Problème rencontré :**

étant donné mon premier projet en c++,donc je suis parti de mon propre code pour débiter ce projet en me basant sur le code d'exemple et les ressources disponibles sur [http://igg.unistra.fr/people/thery/Prog\\_Avancee](http://igg.unistra.fr/people/thery/Prog_Avancee) pour avoir une base solide pour la suite de projet. En suite, le fait d'utiliser les Entity-Component-System (ECS) qu'est l'idée de mon binôme, c'est quelque chose de nouveau pour moi, que j'ai eu des difficulté à assimiler et comprendre ce principe et au final c'était un bon choix car il nous permet de concevoir des entités de notre choix sans modifier l'ensemble de code. On a aussi eu quelques problèmes de conceptualisation d'où les erreurs de segmentation et les erreurs qui surviennent à la compilation, gdb il était indispensable dans ces cas ,il fallait constamment débbugger le programme afin de les résoudre.

## **Conclusion :**

pour conclure, je vous remercié pour le sujet proposé mais étant très intéressant et passionnant amenant l'étudiant à faire d'énorme recherches et apprendre par conséquent.un projet de tel ampleur nécessite beaucoup de temps pour avoir une conception finale optimisée et avec toutes les options demandées.