Final Report

By Hang Liang, Jaspal Singh, Avery Berchek

**Project Description**

Using freeGLUT technology, we made a simple 2-D game. The concept of the game is reverse-whack-a-mole, where the players will be punish by the moles if the players touches the moles. To complete a level and move to the next, player must traverse through obstacles, such as walls that block traversal, as fast as possible. The game uses a time-mechanic where the game over is defined when the time the player has reaches 0. There are coin objects, disperse throughout the map, that the player can collect to increase the time before the game over.

**Members – Contributions**

We split the work in three ways. One person works on the logic of the game, another one to help others, and one to work on animations.

Avery worked on logic of the game and some other shit.

Jaspal worked on compilation, collision, displaying the text and little bit on animations. He was the helper for both Avery and Hang.

Hang worked on collideable and animations.

**Implementation**

**Results**

**Lessons/Conclusions**

**Appendix**

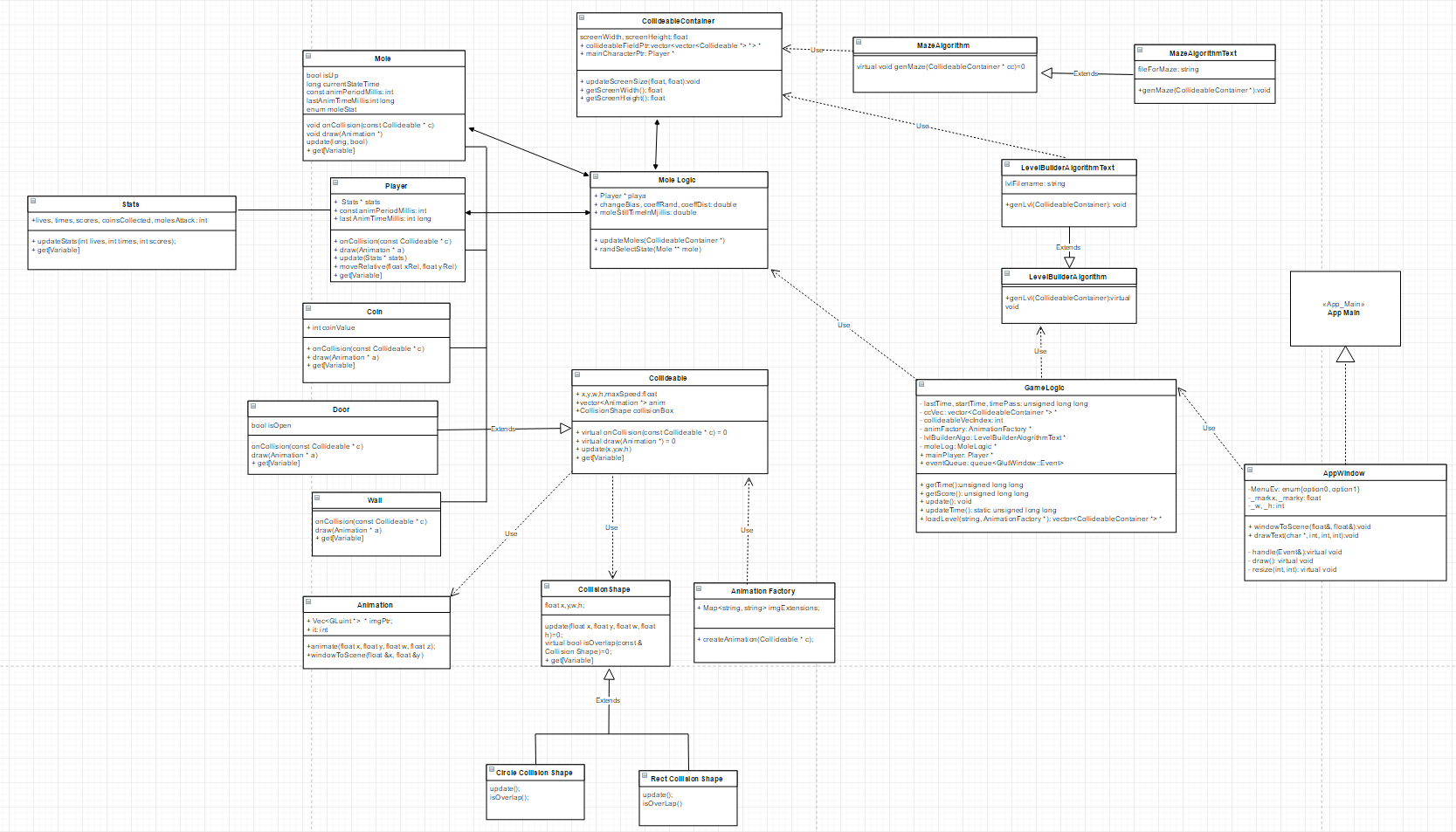


Figure 0 Full UML

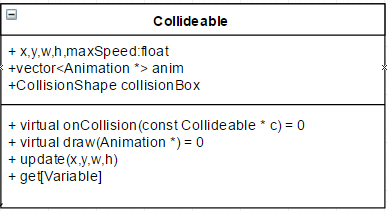


Figure 1 Parent Class of the objects

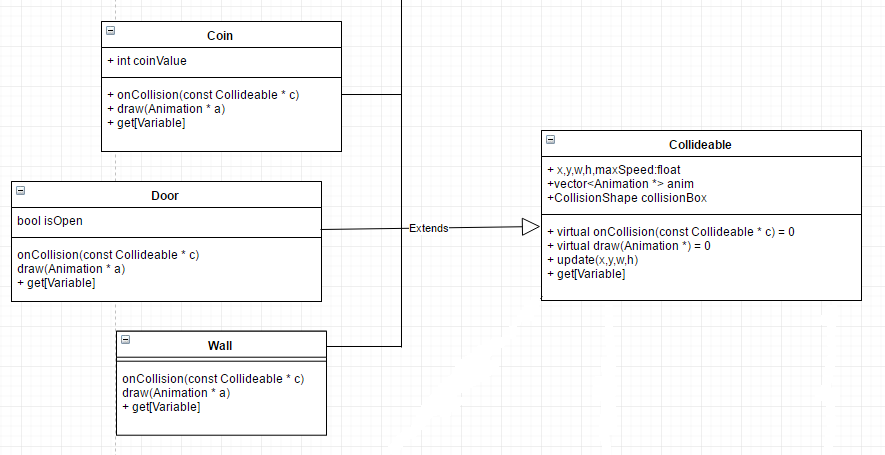


Figure 2 Coin, Wall, Door extending from Collideable

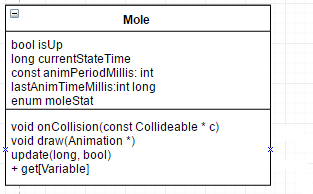


Figure 3 Mole extending from Collideable

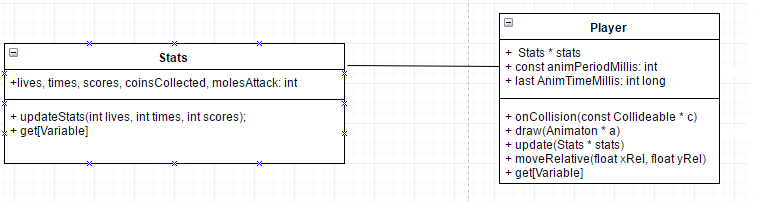


Figure 4 Player Objects extending from Collideable and associates with Stats Class

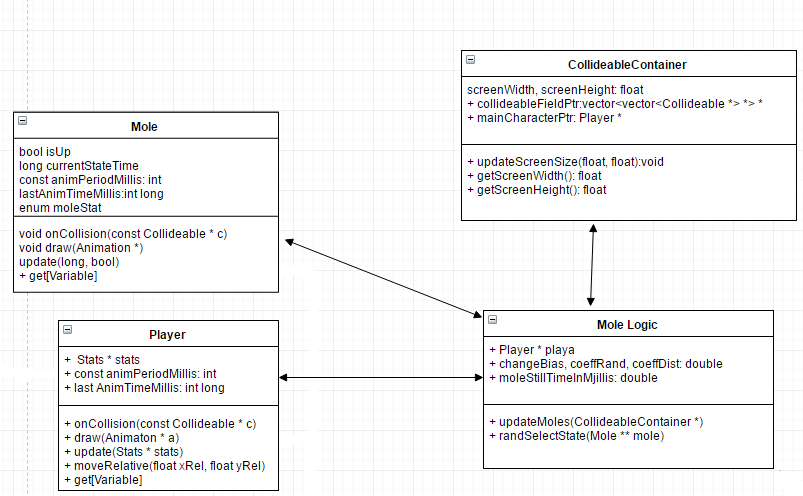
****

Figure 5 Mole Logic and its association

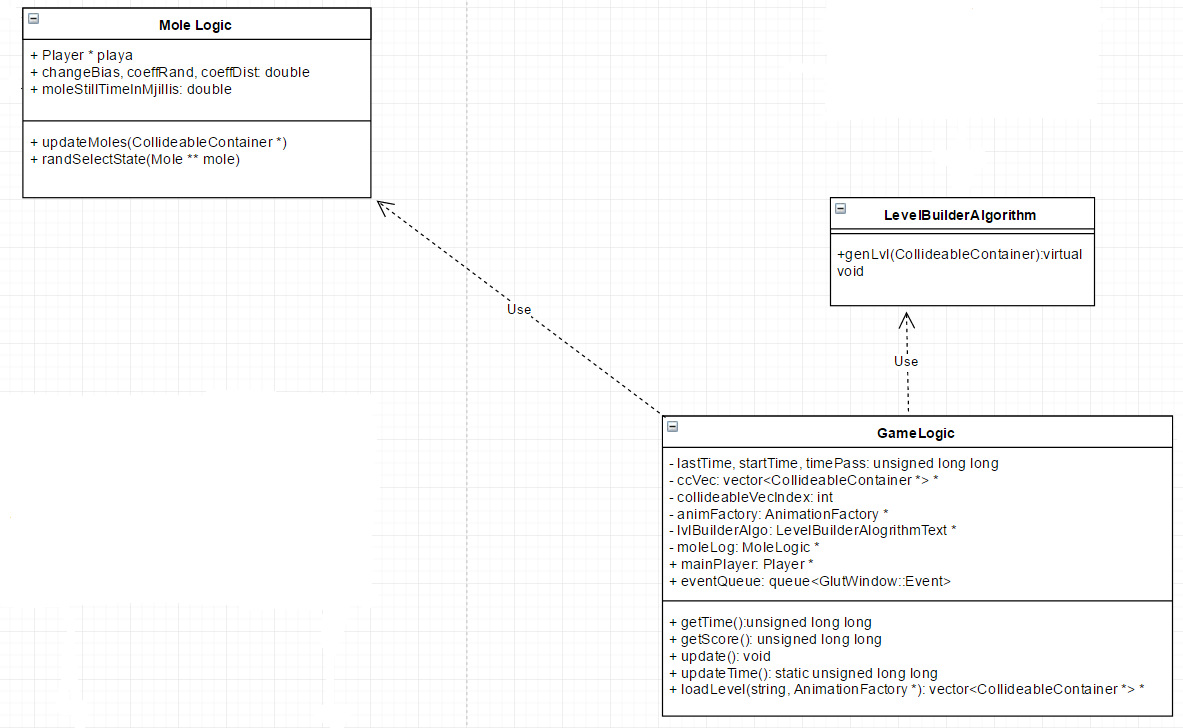


Figure 6 Game Logic and its dependency

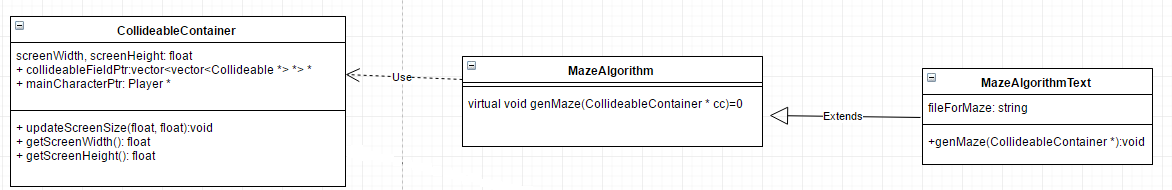


Figure 7 Maze Algorithms hierarchy and its dependency

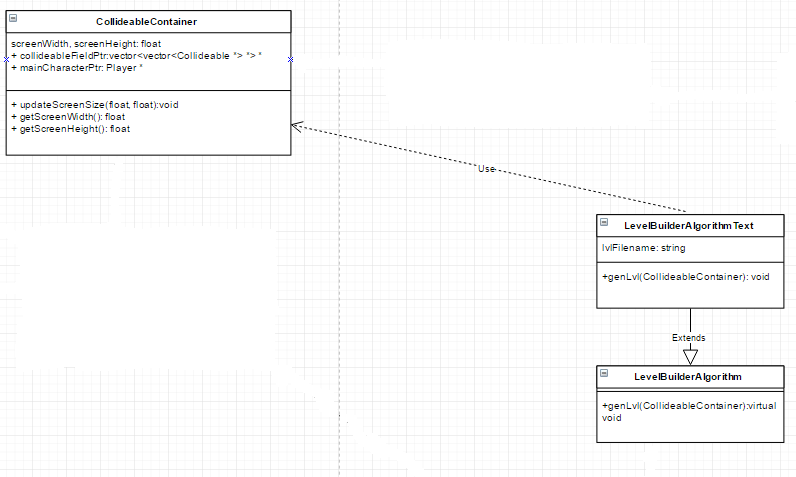


Figure 8 Level Builder Algorithm hierarchy and its dependency

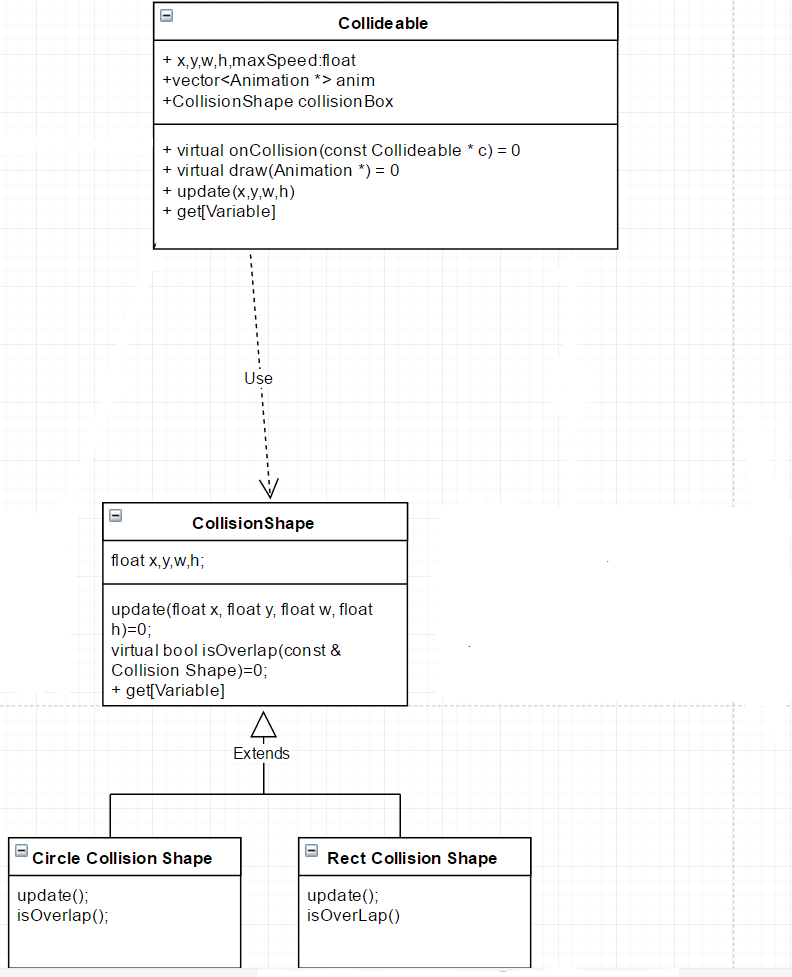
****

Figure 9 CollisionShape hierarchy and its dependencies.

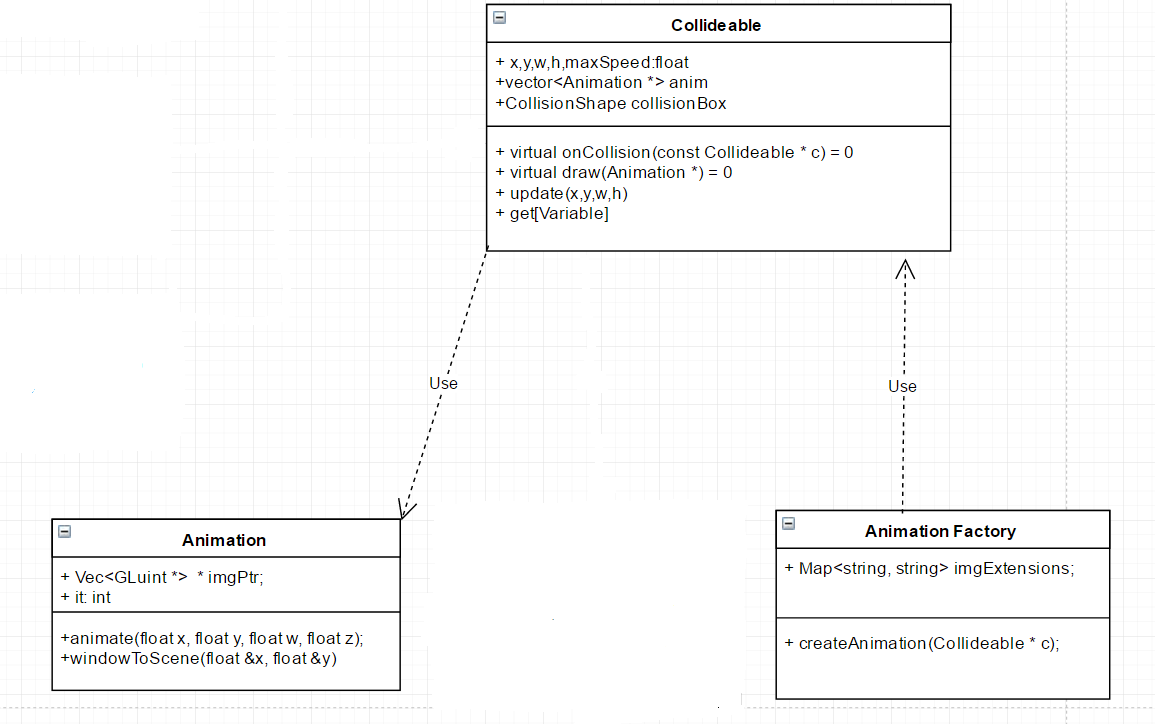


Figure 10 Animation and AnimationFactory

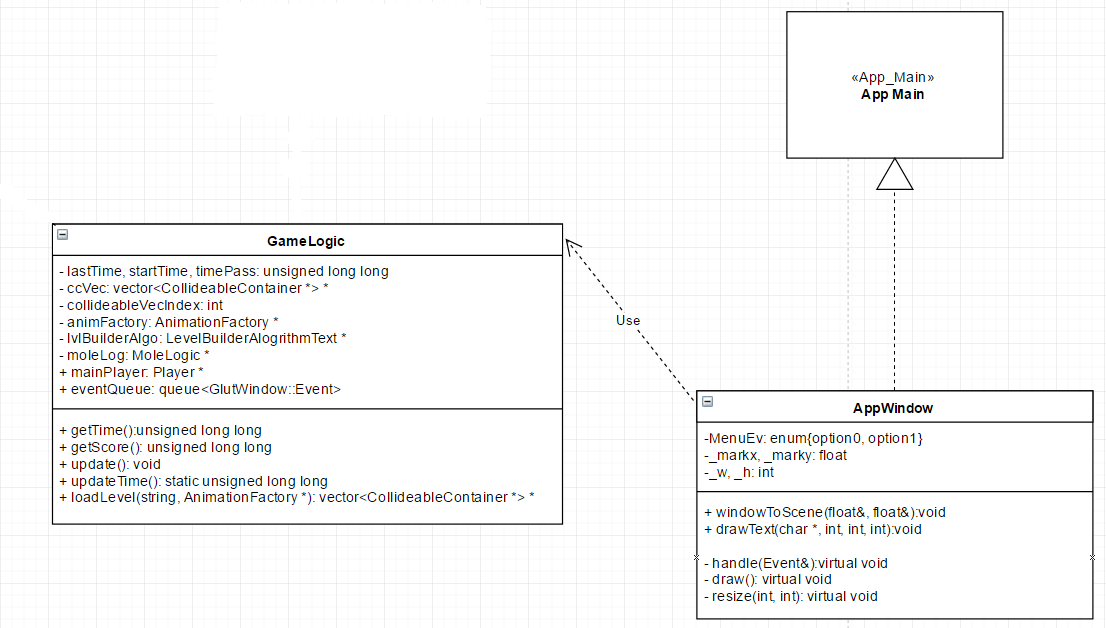


Figure 11 App Main Interface