

CS 319

**Object-Oriented Software Engineering  
Fall 2014**

**USER GUIDE**

**GROUP G**

**Cansu Tüzmen**

**Doğancan Demirtaş**

**Günçe Kalyoncu**

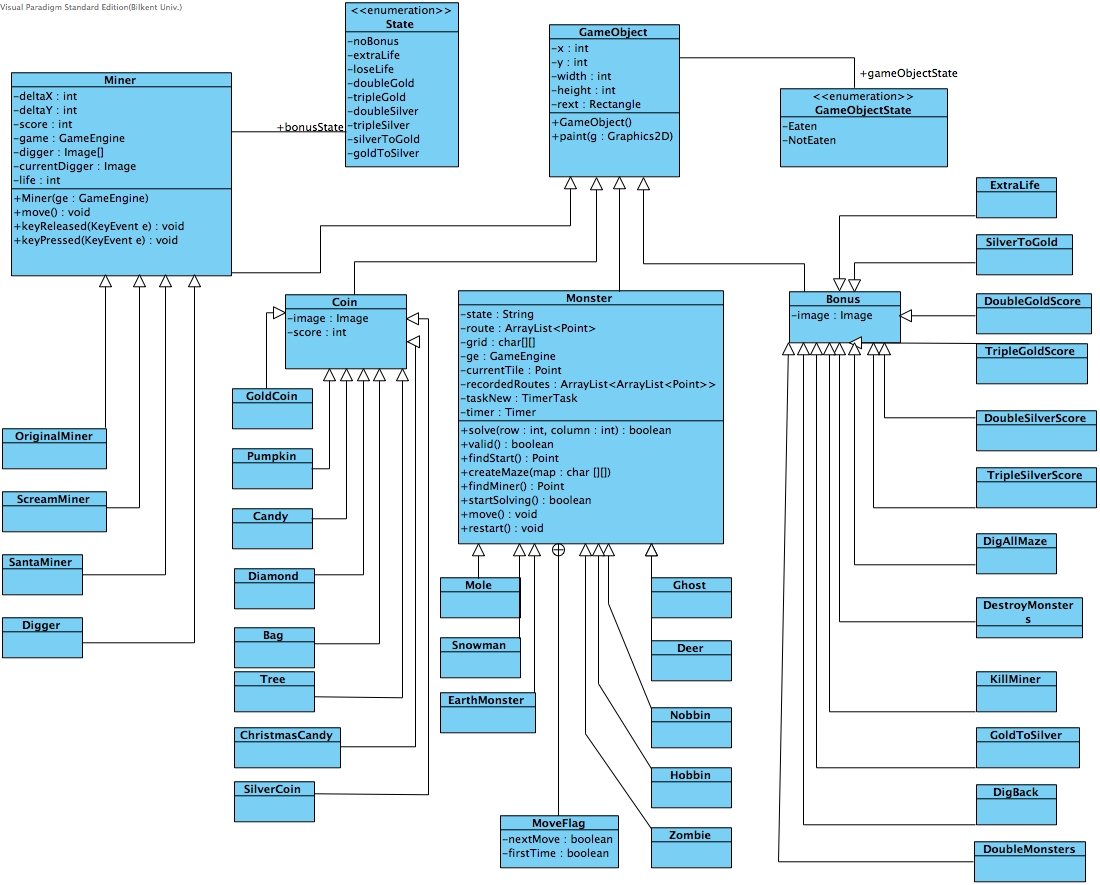
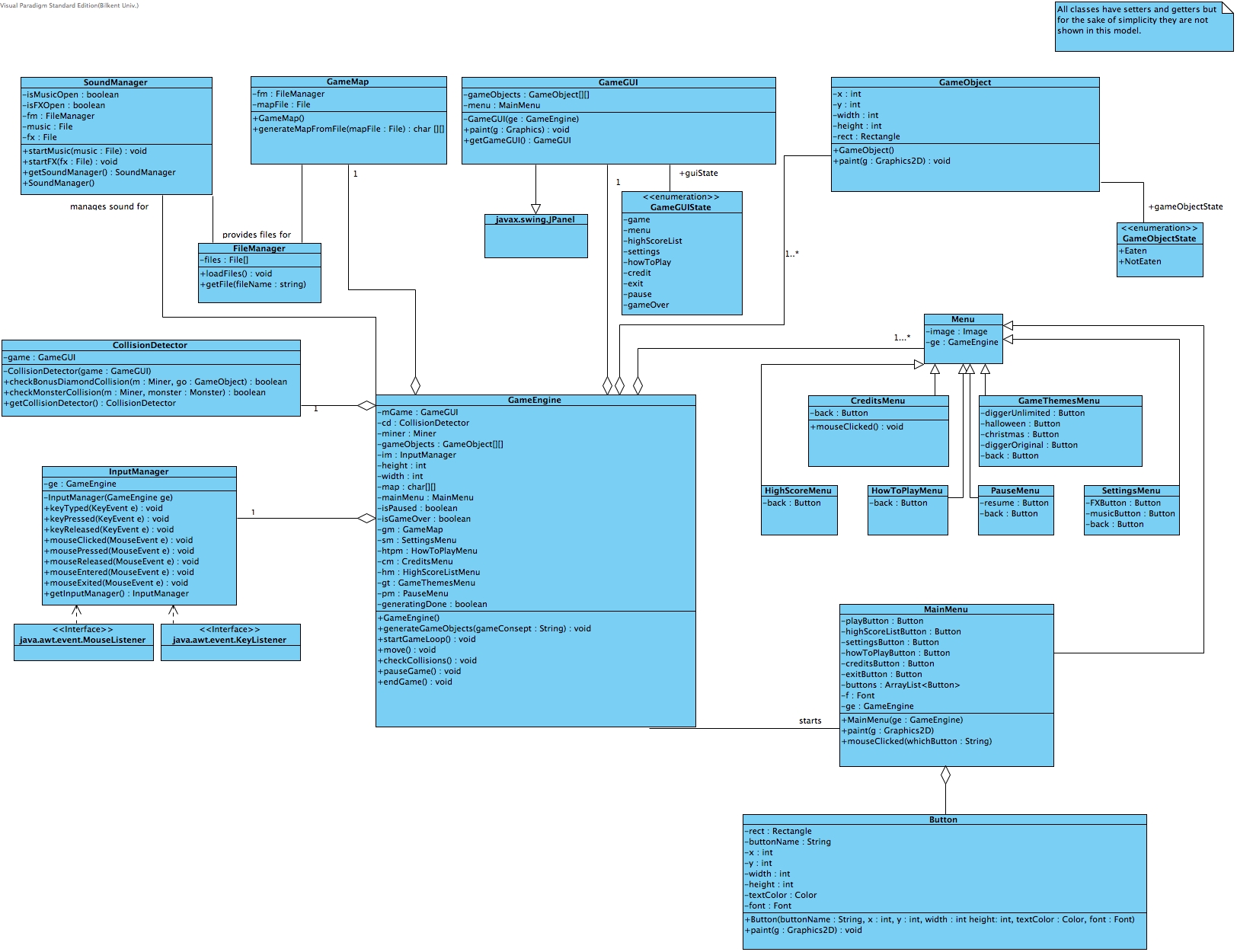
**Erkut Alakuş**

During the implementation process, we rapidly prototyped the game and see the challenges that we would face. Since we were familiar with Java AWT and Swing libraries, graphical developments were easy to implement. However, merging game logic with graphics is one of the hardest sides of this project. We used a TimerTask and a Timer to slow down monster’s movement by scheduling its moves with 1-second delay. We used this approach since game logic is relatively fast when compared with graphics.

Also writing the AI part, making monster to follow miner, was also hard to implement, since there are various exceptions during miner’s movement. We used a recursive algorithm and traced the miner in a matrix, which shows the game’s current components (bonuses, gold, monster and miner).

Additionally, we had some difficulties when restarting the game, and that lead a major bug in level, high score and theme structure. Although we tried various workarounds regarding these bugs, because of the time limit we were not able to fix them and release a stable version.

We were royal to the Detailed Object Design document but we had to make some minor changes in this design. Below, you can see the final object design and compare it with the initial one.

In this project, we get benefits of object-oriented design and its principles. All the game map components are derived from GameObject class and they were painted with a single shot, with different images.

We used both threads and loops in project’s structure. To schedule future events Java Threads were very useful and for the game loop, a simple while loop is used.

Since we used many matrix operations during the game play, memory usage was higher than we expected, however we managed to reduce it by disposing unused matrices, images and sound files.