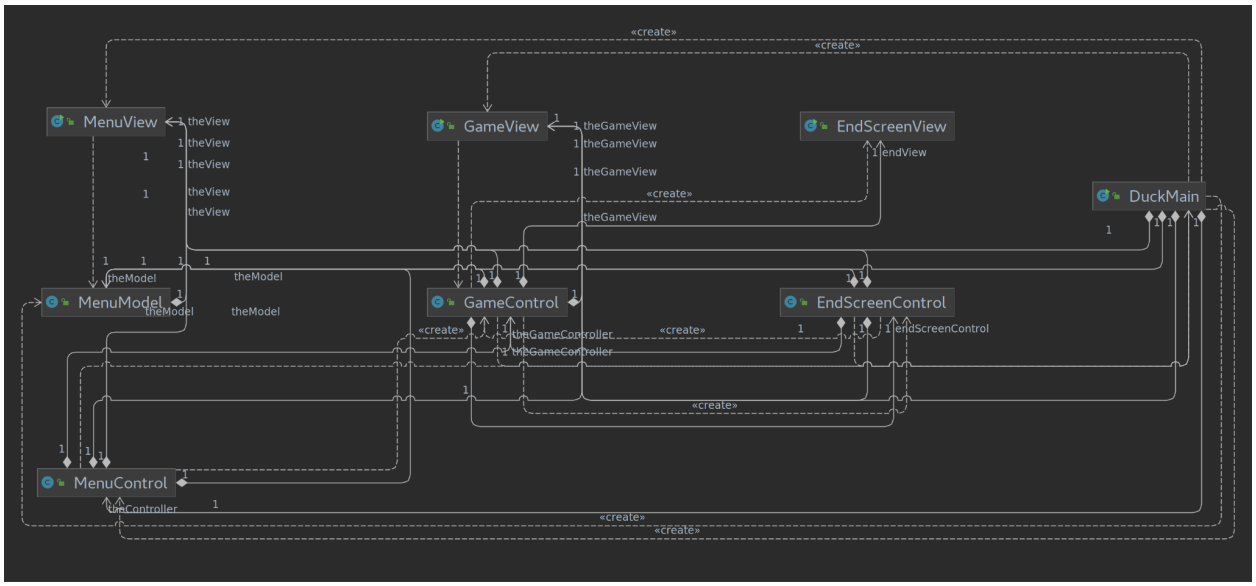
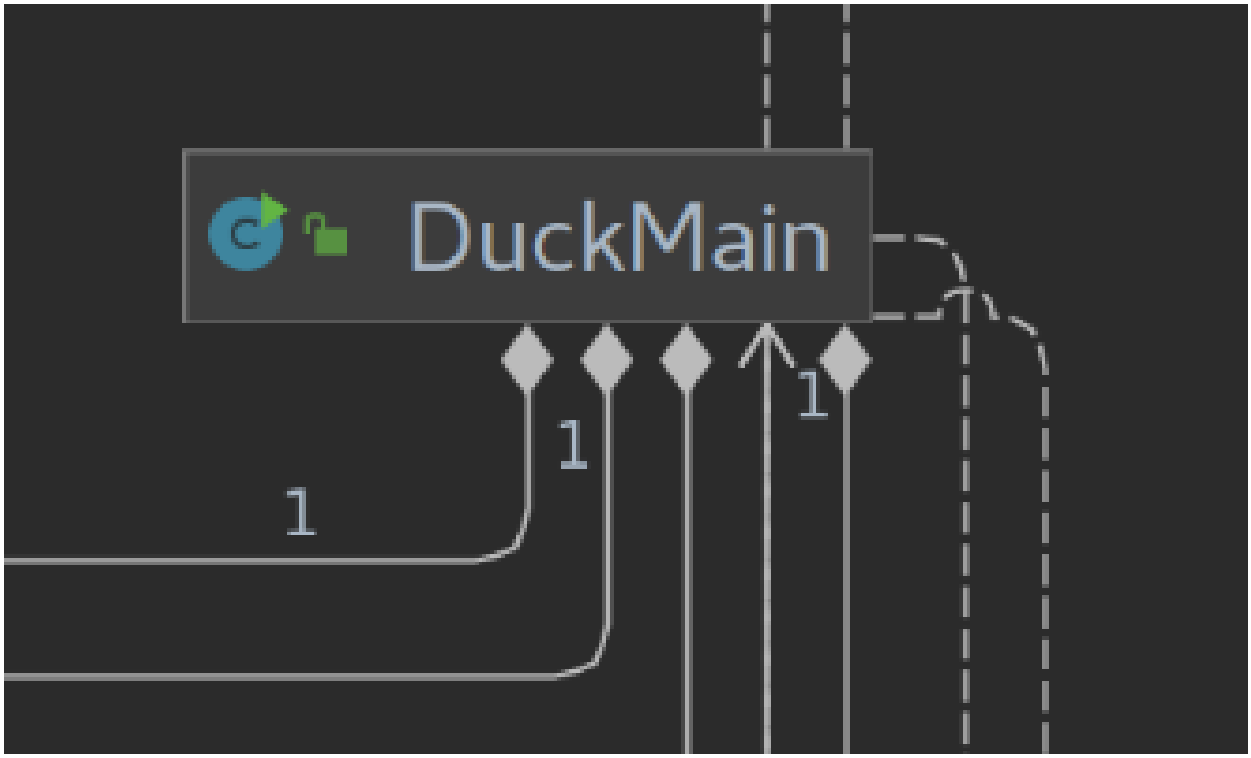
Our team's implementation of Duck Hunt intended to follow the MVC design process for JavaFX, or model, view, control. We split the Duck Hunt application into three parts:

- The menu
- The game
- The end screen

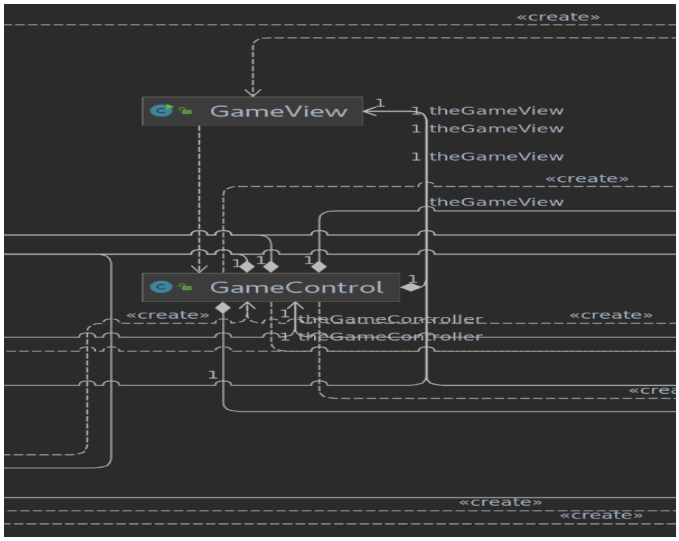
We found it somewhat unnecessary to include model portions for each part but we could not get our code to function properly without a model for the menu. The view portion of each part began with the Duck Hunt background but each also included unique nodes of their own. For instance the menu had two buttons and a label, the end screen only had a label and the game had several view components, too many to list here. Provided below is the IntelliJ generated UML diagram for our project, I will zoom in on a couple of components to explain our design more in depth.



Provided below is our main driver class DuckMain. As you can see, there is a composition relationship between many of the key classes in our project and DuckMain, which is expected as it is the file that instantiates and initializes each component of the application. The numbers that are seen are the number of instances of each composition, which is only one in this case as there is only one MenuView and one MenuControl and so on.

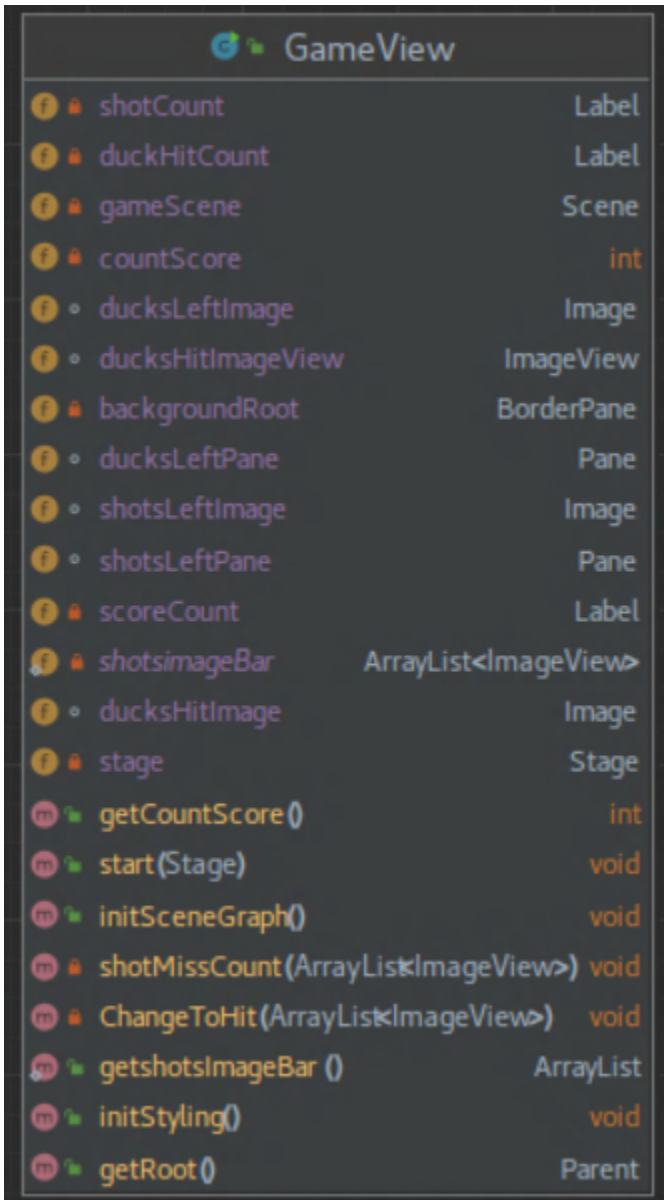


Provided below are the GameView and GameControl for the application which are different from the other MVC portions of the application as they are instantiated and initialized only when the user decides to play the game, so technically, if the user were to only load up the application and quit then there technically would be no instances of GameView or GameControl. This dependency on being instantiated in another class shows with the lines connected to it.



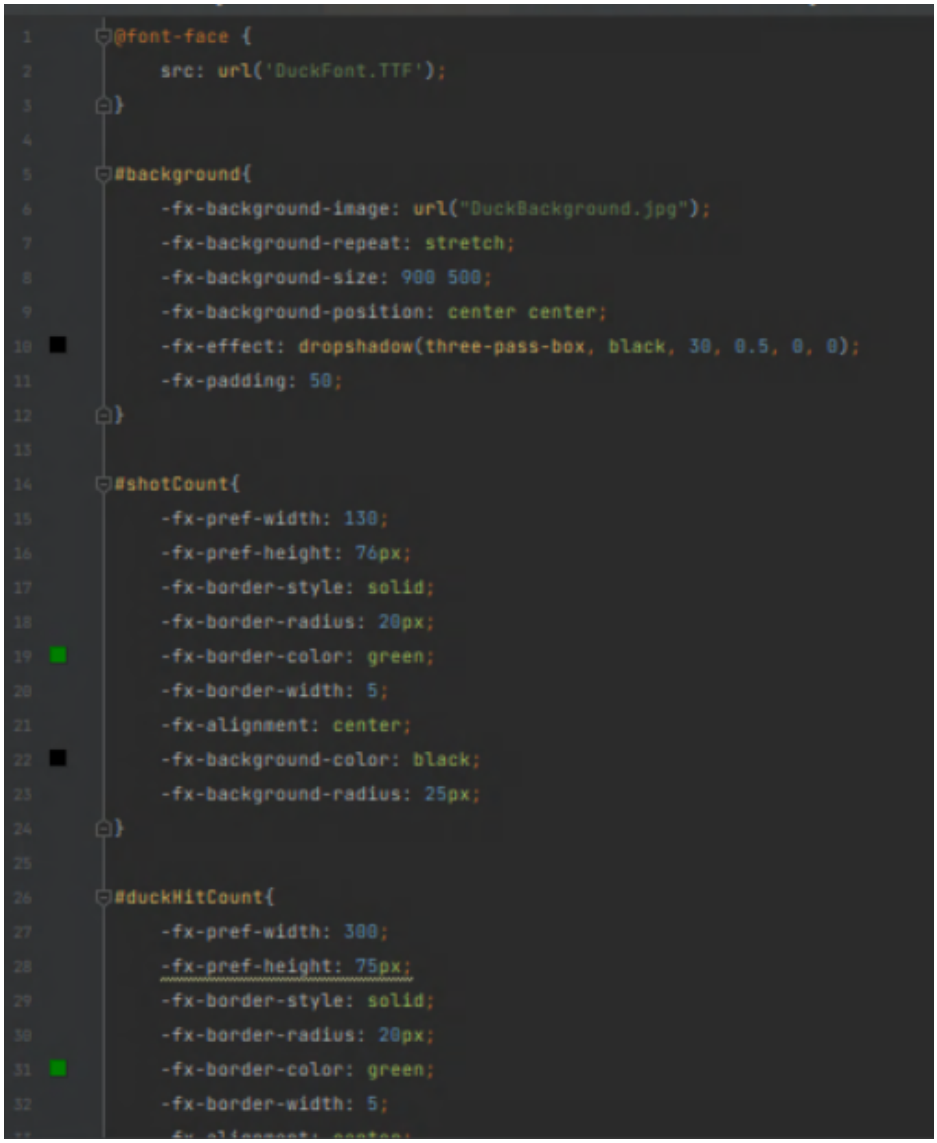
Looking deeper into a class like GameView, we can see the guts of the class. It extends the JavaFX application so we can see that we chose a BorderPane to be our root, and also instantiated a Scene object that will contain all the nodes for the scene graph. The scene graph consists of a lot of objects including Images, Labels, Panes, and more Labels. All of these come together to create what is seen in the dynamic HUD in the actual game. As you can see, there

are two methods that actually should have been moved to some sort of GameModel class, this was an oversight from the developers. Regardless of this fact, the shotMisscount() and changeToHit() methods change the shot count, score count and duck count and are pivotal to the performance/functionality of the HUD.



All of our styling was done in CSS, from the font, to the background to each node that had any styling to it. Below is a screenshot of the CSS file for our GameView class. As you can see, we coded styling to position the HUD properly and ensure it has the correct color, spacing and

padding, which is not as easy as it seems.



```
@font-face {
    src: url('DuckFont.TTF');
}

#background{
    -fx-background-image: url("DuckBackground.jpg");
    -fx-background-repeat: stretch;
    -fx-background-size: 900 500;
    -fx-background-position: center center;
    -fx-effect: dropshadow(three-pass-box, black, 30, 0.5, 0, 0);
    -fx-padding: 50;
}

#shotCount{
    -fx-pref-width: 130;
    -fx-pref-height: 76px;
    -fx-border-style: solid;
    -fx-border-radius: 20px;
    -fx-border-color: green;
    -fx-border-width: 5;
    -fx-alignment: center;
    -fx-background-color: black;
    -fx-background-radius: 25px;
}

#duckHitCount{
    -fx-pref-width: 300;
    -fx-pref-height: 75px;
    -fx-border-style: solid;
    -fx-border-radius: 20px;
    -fx-border-color: green;
    -fx-border-width: 5;
```

```
        -fx-pref-width: 300;
        -fx-pref-height: 75px;
        -fx-border-style: solid;
        -fx-border-radius: 20px;
        -fx-border-color: green;
        -fx-border-width: 5;
        -fx-alignment: center;
        -fx-background-color: black;
        -fx-background-radius: 25px;
        -fx-font-family: "M29_DUCK HOUND";
        -fx-text-fill: white;
    }

    #scoreCount {
        -fx-pref-width: 175;
        -fx-pref-height: 75px;
        -fx-border-style: solid;
        -fx-border-radius: 20px;
        -fx-border-color: green;
        -fx-border-width: 5;
        -fx-alignment: center;
        -fx-background-color: black;
        -fx-background-radius: 25px;
        -fx-font-family: "M29_DUCK HOUND";
        -fx-text-fill: white;

    }
```