# CS2263 F22 Project 9B:

# Space Invaders Solution

Koda Koziol

Other than the features that I didn’t get around to implementing, I don’t have very much that I would change if I were to do this again. Overall, I think that what I built is fairly scalable.

There is one major problem that I know could be improved: how it runs on different displays. My laptop is 4k, so that’s what I’ve been testing it on. Just now I was able to test it on an HD display (this is probably the most common resolution, I’m just making an educated guess that this is what you use), and unfortunately everything was twice as large as it needed to be and moved twice as fast. I cut every value affected in half (except for the enemy ship’s speed, which was already set to one as an integer), so now it’s playable on HD, but I wish that I had considered that and built the game to scale with the window as it’s running from the very beginning. Right now, every size and movement value is in pixels, which is the root of my problem. If I were to do this project again, or continue it, I could have a float unitScale variable that I initialize on form load, or maybe use some built in property that I don’t know about currently, and multiply all necessary values by that. This probably still wouldn’t help me with very small values though, like the enemy ship’s speed I mentioned earlier, so I’m stumped with that. Maybe I could do an intermediate layer separating game logic from display logic. Game logic would be handled with float values in isolation before they are *read* and cast to an integer to draw to the display.

I tried to keep drawing logic (the visuals/sprites) separate from the game logic (movement), and I think that I did okay with that, but I could’ve done better. I put some methods in some of my sprite classes that I’d rather have in an object manager class that references a sprite object.

I did get to learn a few new things to make this project. I became more familiar with the PictureBox tool, which is how the game itself is displayed. I learned how to use async tasks to run animations without interrupting the main GameSession loop. I also became a little more familiar with events, creating a GameOver and Winner event.

Overall, I’m happy with how this game turned out. I have a whole new respect for the attention to detail that game engine developers have to build a game that runs consistently on any device. After all, if it didn’t, it wouldn’t be fair.