**Space Invaders Post-Mortem**

**Brandon Wilson**

April 25, 2018

The first feature I wish I could have implemented is a death animation. All the other transistions in the game are handled with message boxes (Game over, level advancement, etc.), but the death is handled simply by pausing the game for three seconds. This helps to draw attention to the fact that the user has been hit, and is thus that much closer to losing the game. However, it still feels clunky and a blinking screen or player cannon would have felt more satisfying.

The second feature I wish I could have implemented is a persistent High Score feature. As implemented there is a High Score feature, but it only remembers the highest score from the present session. Once the game is exited, this information is lost. This feature would make the game much more compelling. The user could be in continual competition with themself and others to get further into the game than before. Better still would be to implement a full High Score board, to remember the previous 10 highest scores and associate a name or initials with them.

The original class design of the game survived intact, but in hindsight, it still has a number of flaws. Most of those problems stem from having more classes than was really necessary.

The Projectile class is referenced in exactly one line of code, when removing all projectiles from the screen at the beginning of a new level. This serves only as classification and has no distinct code. It could easily have been replaced with a second "or" statement.

The biggest violation of good object-oriented design is the decision to make each screen entity from a list of "parts" (for example, the player cannon was made of two rectangles, one representing the base, and one the "turret"). The primary difficulty was that since each "part" contained its own drawing method, the "x" and "y" values had to be changed at each update step, along with the "x" and "y" values of the overarching object. Thus the properties had to be made public, when protected would have been preferable. This "parts" feature also does not generalize well to when the rectangle and ellipse graphics will be replaced by a gif in later versions.