In building the code, we wanted to start off with small things, test them, add another small thing and test that. To keep weirdness and bugs at a minimum, we believe this is the right way to do it.

For the same reasons, we also tried to make the code as simple as we could make it.

Firstly, we started off with the most basic functionality – having a player and moving it. That was mostly straightforward since it was building some basic blocks like recording keyboard events and then just applying velocity every frame in a direction dependent on the key pressed.

After that, work started on level collision. That was a complete nightmare to figure out and then implement properly. In the end, after a month of fiddling around, we got it to some acceptable level and decided to stop spending time on it.

We continued with the enemies. First we made their class. Initially it was a copy of the player class, so we could manually move the enemies to debug them. After that worked, we had to figure out how to make them follow the player. When we eventually got that going, we wanted to mess around with the parameters to delay their rotation for a more “goofy” effect. With that out of the way, we needed to have something which actually spawned the enemies. Initially they did that at specified points outside the screen and the code would pick a random one for spawning. Afterwards, lead programming implemented some great code to randomize their spawn locations still outside of the screen, so the players wouldn't expect them.

Then we came back at collision. This time we made it much simpler because we just needed to detect whether an enemy is hitting the player. The code just checks if the player is within a certain amount of pixels of an enemy and then executes the game over code. The cool thing is that we have all of the enemies in an array, so with a simple for loop we can check if any enemy is hitting the player.

After this we created the whole code for the abilities. It checks if space is pressed, and if it is, upon the player clicking on the screen, an ability is spawned with a destination towards the current mouse position. Since there can be multiple abilities on the screen at once, they are stored in an array of abilities. A for loop goes through all of them to apply their respective velocity every frame and to check if they collide with any enemy in a double for loop.

After this simple elements needed to be implemented. The most major one is the rune bar on the bottom of the screen. It also implements an array of runes, which is easily manipulable when removing runes, clearing runes and resetting the game.

Score was added next. It just adds one to the score counter every time an enemy dies. The counter is displayed through a text box which is updated every frame.

The timer was next. Very simply, it just adds one to itself every frame, and is then divided by 60, which is our framerate to get the number of seconds the player has been alive.

To top it off, we implemented some simple sounds for shooting and killing and a nice 8bit song from the internet to fit the tempo of the game.