**7. Notes to future programmers of our project: directions that would make it better**

Name: *Matthew Meade and Abhinav Agrahari*

Since we never finished our release schedule, that would be one thing that a future programmer of our project could do. As we never did our boss level (yet?) we have reduced nLastLevel to 20.

Due to the way that the gravity function works, the falling spikes may or may not go partially “into” the platforms/boxes. The rate of Acceleration accumulates in Velocity, and Velocity is constrained by a Velocity limit. The gravity function adds the velocity to the y position of the object, so the gravity function sometimes adds *16* to the y position, effectively skipping the beginning of the box when falling (causing to move into the box).

To fix this, you could make another variable “fYOrig”, and start using “fYStart” for collision detection instead. Right now, fYstart is used for resetting the moving spikes to their original position when the user dies on that level.

If you press CTRL + G the button does not work, but if you press G + CTRL (in that order) than the button does work. We are not sure why this happens; maybe it is the way Processing stores key inputs.

The collision detection with the platforms and obstacles is with a rectangle (the image) so the crab collides before it seems like it should. His claws extend over his legs, and the floor/spikes, and other hazards, have collision detection with his entire image(claw-to-claw) instead of the specific areas of the image. This can be fixed if there were multiple collision detections for each section of the image. Then various functions could be called depending on which section of the image is returning true in collisions. For example, if the empty space in the image returned true, nothing would happen, but if his claw or leg area returned true then he could hit the spike and die.

The beginning of Level 19 may or may not lag on the school computers. We do not know why this is, maybe the LaserGun shoots too fast?