DIFFICULTIES

To a player, the game would seem very basic. But even while create a game which looks very simple from the outside, we had to face a lot of difficulties. One of the major difficulties we faced was to maintain the continuity of the game. The game menu screen must be displayed first, then the main game screen and finally the game over screen. The difficulty was that after the game over screen, the game must return to the game menu screen at the click of a spacebar. To ensure this we had to create nested while loops along with several check variables. Once, the problem of the hierarchy of screens was sorted, many more problems emerged in the main game. Adding a new feature often led to errors which majorly consisted of wrong scope (local and global) of variables, function call at wrong time, infinite while loop, incorrect blitting sequence etc. Thus, debugging was a bigger problem than thinking of the logic for the feature. Implementing the feature correctly often took many hours. While debugging, one thing that we found useful was to print arbitrary strings to the terminal at the points we thought that the error could occur and this helped us to judge whether the loop had reached that point in the code or not. While setting the bars (obstacle) at the start of the game and while adding a new bar, we had to ensure that the random integer returned for different bars didn’t lead to the collision of the bars. Thus, we had to create a list of bars and check the coordinates of the new bar against all the old bars. While creating the dropping bars feature, we had to ensure that the, coordinates of the bar being dropped didn’t match with any of the already present static bars. Also, while playing the music, we had to ensure that the background music paused when the ball was being eaten by the snake so that the eating sound could be played. After, that we had to play the background music again. Also, we had problems while setting text on the backgrounds. To ensure that the text was in the right position, there was no other option to check other than running the game over and over again and checking.

FLEXIBILITY

The game we have made is very flexible. We have assigned some value to the variables corresponding to no. of lives, no. of bars, screen width etc. in the beginning of the code and we have written the entire program in terms of these variables. So the initially assigned values to the variables can be changed by user and the code still functions properly. For example- the screen height and width are global variables which can be changed and the text position gets adjusted according to it, the no. of bars to be displayed at the beginning according to the level can be changed, the no. of lives provided at beginning can be changed according to level and whatever number of lives is they will come always comein centre in screen.

CONTRIBUTION:

Raghav:

Movement of the snake, Loop structure to ensure continuity of game menu, main game screen etc., sound effects, function to ensure that the bars being displayed don’t collide, checking for the collision of snake with itself, background images and text.

Nishikant:

Setting obstacles (bars), adding new bar randomly, dropping bar feature and checking none of them collide by making a list of all bars, lives and its working, displaying life ball randomly and adding extra life if eaten, displaying score ball randomly and increasing score if eaten, writing and reading from file.