**Project Design Document**

Team !=LOSERS:

Nour Ali/ 900160807

Sherry Guirguis/900161820

Michael Dmitry/

* Requirement analysis:

The game contains a player, enemies, waves, levels and pick-ups.

The player fires at the enemy, dies if hit by the chicken or rocks or eggs and randomly can collect a pick-up from a dead chicken. After each wave the player faces a beast (a big chicken) in order to move into the next level. If the player fails to defeat the chicken, he restarts the level.

The wave is typically the chickens and the rocks that will show up in the screen. It ends if all the chickens or rocks go down the screen.

A menu which will contain a background picture , music, high scores, settings and exit. all these options will show up on the menu.

* Algorithms:
* Menu will take the size of the window and will show as soon as the game starts. it has the options: play, exit, high scores, settings.
* If player presses on play, the game will initiate at a certain wave with a number of enemies to defeat (these enemies include chickens and rocks). If the player succeeds in killing all the chickens and avoiding the rocks and the eggs, he faces the big chicken. If he defeats it, he moves to the second wave. Each level contains a specific number of waves.
* If the player presses on exit, the game should close.
* If the player presses on high scores, a certain file should open containing the high scores listed in order.
* If the player presses on settings, there will be the option of new game, sounds...
* High score calculation according to the number of chicken dead and collected pick-ups. Example: each dead chicken is worth 2 points, each pick-up is worth 5 points… The highest scores are saved in a separate file that records every new high score and lists them in decreasing order. Each time there’s a higher score, it is appended to the file and the other scores are shifted one place.
* When the player gains a pick-up, it is either an extra bullet, a new life, or extra immunity time (randomly selected). The bullet will also change to a bigger one at the end of the wave, when the player is faced with the big chicken.
* The health of the player: 3 lives maximum. If hit by a chicken, an egg or a rock, he loses one life.
* The music changes when the game starts, when the player shoots his bullets, when he is hit, when he dies...
* List of classes:

**Class menu:** background picture, music, options

**Class button:** set button size, position, string font (color and size), check if the button is clicked.

**Class player:** set picture, move, fire, display, is dead, is alive, collect pickups, set health, set immunity.

**Class screen:** set background, class buttons, music, etc.

**Class enemy** (chicken and rocks): set picture, position, type of movement, display, is dead, is alive, drop eggs, pick-up.

**Class bullet:** set bullet size, color, if hit, if miss, if pick-up.

**Class level:** set enemy, number of enemies, game time, set enemy (the big bird), set bullet, chicken dead, number of pick-ups.

**Class pick-ups:** extra health, extra fire or immunity time (10 seconds).

**Class wave:** random number of chickens, movement of chickens, number of rocks, descent of chickens, descent of rocks.

**Class highscore:** will support inputting from a file, updating the scores, and then outputting the scores to the file at the end.

NOTE: some of the classes will be used in other classes (class within another).