CS/CoE 447 – Project 1 – Spring 2016 – Rubric (Draft v0.03)

Requirement	Contribution to Score
Logic/Gameplay:	
The program uses an appropriate data structure (e.g., queue of events)	10
Game takes ~100ms time steps	5
The game itself starts when the player presses the center (b) key	2
Bugs are placed on the game board at random *x* coordinates in the top row	2
Only 3-4 aliens are placed at a time. More bugs are added as the game progresses to increase the level of difficulty. By the end of two minutes, >= 64 bugs should have been added.	2
Multiple bugs are allowed in a column (e.g., there can be aliens at both (30; 50) and (30; 0))	2
A bug moves from its initial position (*x*,0) to (*x*,62). A bug moves down one LED (e.g., (*x*,10) to (*x*,11) every time step	5
Each game element (e.g., bugs, bug-buster, etc.) has the correct color	2
The bug-buster moves right and left appropriately	2
The bug-buster is fired by pressing the up key	2
Pulse moves properly every time step	5
The pulse causes the bug to disappear	10
Each bug-hit produces a burst	7
Burst cascades	7
Burst is limited to 10 pixel radius	2
No bugs in the bug-buster's row (63)	2
Game ends in two minutes	2
Scoring system works correctly	7
The game may be ended early by pressing the down key.	2
When game completes, the program prints the correct message: "The game score is <bug-hits> : <phaser-firings>."	2
Submission: Student submitted a compressed file (.zip) containing the following:	
Submission windfall (Student submitted something)	5
Student submitted README.txt [i.e., file exists]	5
Known problems/issues documented in README.txt. Student used README.txt to explain the algorithm the student implemented. The explanation is detailed enough that the teaching assistant can understand the student's approach without reading the student's source code.	5
Documentation and Format: Student's assembly language program is properly documented and formatted. Student used enough comments to explain the student's algorithm, implementation decisions and anything else necessary to make the student's code easily understandable.	5
Total:	100