Space Potaters

***Final Report***

Team Ramrod*:*  
Tan, Joshua  
Anderson, Travis  
Birrenkott, Chris  
Wako, Shambel

CSCI 161  
Section 1

December 9, 2011

Instructions to run program:

To run the program, simply run the executable jar, using :

java –jar “Space Potaters.jar”

In addition, if your Operating System is correctly associated with executable jar files, you may double click the jar icon.

Features Implemented:

* Moveable and fireable space shooter.
* Enemy space potaters that fire randomly and move according to the original Space Invaders pattern.
* Potato and Gollum themed graphics for the sprites.
* Damageable protective barriers (with French Fry container theme).
* Space potater animations.
* Lives system for Gollum.
* Score and record system with persistent data for leaderboard information.
* Randomly created bonus space potaters.
* Different difficulty levels for game. Difficulty increases as space potaters get progressively closer to the bottom of the screen.
* Menu system with in-game instructions.
* Ability to pause or quit to main menu from game.

Features Not Implemented:

* Implementation of WindowListener methods for SpacePotater class (i.e. pause when minimize window, unpause when maximize window, etc.)
* Sound effects for game.
* Animation for Gollum death.
* Increasing difficulty in-between levels (each time player beats a particular level).

What you have learned from working as a team? (Positive things /Negative things):

Working as a team gave the added benefit of constant feedback for design/coding ideas. This helped to modulate some ideas from members and reaffirm others, which was beneficial overall. Also, we could split the workload up among members.

A negative aspect of working as a team was trying to find a way to split up the project into components that were independent of one another. Also, whenever a particular team member failed to produce results, other members had to pick up his responsibility on top of whatever was already assigned to them.

Comments:

The approach taken to various aspects of the game could probably be improved – in particular, the brute force approach to checking for collisions or checking for certain component conditions seems particularly wasteful of resources. Because of this, the game might not run very well on a slower computer (slow CPU, limited RAM). Also, it might have been possible to improve the efficiency of the game using other specialized data structures (instead of just using ArrayList for most game structures).