

Osto Vargas
Professor Matthews
CSC 211
16 April, 2015

Final Program - Space Invaders

Testing Plan

I am not 100% sure how I would get everything to come together for the final project, as a matter of fact a lot might change. I am planning on using **BufferedImage** objects to hold individual ship images to load up per ship object. The *Player* ship obviously has a distinct sprite.

I plan on testing if I can utilize the **render()** method in the **Game** to call upon the render methods in the ship objects to process their **BufferedImage** information in one go. I separated rendering and updating to account for mutator methods on the ships locations and rendering to process their images new location.

The tightest problem for me would have to be **Missile** to **Entity** collision. I would have to find a way to repeatedly check if a fired missile is within bounds of the **Entity** instance it's suppose to affect. I might have to do all that within the **Game** class at his has a broader scope. All this will be done in the **tick()** method.

It's possible I might introduce pixel array processed **BufferedImage**'s by storing their *RGB* values. This is a slightly more complex method but it allows me to utilize a **Sprite** class to process all the game assets from. Otherwise I would just have to individually load up a **BufferedImage** to each **Ship** and **Missile** class using a external resources.

I also decided to implement a separate **InputHandler** class to add the **KeyListener** implementation into the game class and keeping the modularity. I would have to find a way to process the key events into movements for the player. The great thing is that it can be implements in any **Game** made by anyone! I borrowed it from open-source code Markus "Notch" Persson released on the internet!

There's no reference to game rule logic, I decided that missiles colliding with alien ships will just omit them from the array, and I will have a sort method to shift all alien ships down to prevent null holes in the array. The game ends when the array reached a max capacity and has no alien ships or the player dies.