Orbitron (Strategy Game) Documentation – Stanton Parham (stparham@ncsu.edu

First off, I want to apologize for the state of the majority of my code. I have so many areas where my code is extremely redundant and inefficient, but I was in a rush to finish a lot of the features of the game so I implemented them in the fastest way that I could instead of taking the time to find a better way. I'm also really sorry about how large some of these classes are.

- Packages – I created several packages to group my classes.
  - main – holds the classes directly responsible for running the program itself and the game
  - supportingClasses – holds several classes that I couldn't describe any other way than "supporting"; they are vital to the execution of the game
  - staticStorage – holds several classes that basically serve as databases for large amounts of information needed to run the game
  - dynamicStorage – holds two classes that can have their values changed and serialized into files independent of Java
- Terminology – While making the game, I coined several terms that have been used innumerable times throughout my source code.
  - Cell – just short for a GameCell; Cells are the basis on which all gameplay depends. These can be thought of as city blocks or buildings.
  - Cell Type – determines all of the characteristics of a cell; The different cell types are basically just different buildings that you would find in a city: Apartments, Skyscraper, Bar, Church, Mall, Grocery Store… See: CTInfo.java, CTVS.java, GameCell.java
  - Cell States – A cell has several "states" that it can be in. A cell can be scouted, captured, destroyed, hostile, or neutral. A cell must be scouted if it is captured or destroyed. See: GameCell.java
  - Cell Potential States – A cell has several "potential states" that could affect the cell's actual "states" if the player acts upon these "potential states". A cell can be scoutable, capturable, or destroyable. See: GameCell.java
  - Cell Compatibilities – A cell has compatibilities with certain structures. All this means is that certain buildings can be built on the cell and certain buildings cannot. See: CTInfo.java and CTVS.java
  - Cell Buildings/Structures/Conversions – When a cell is built on, it is converted into a different cell type. A

cell can be built on and converted into two of the following structures: Housing, Robotics Factory, Storage, Steel Mill, Energy Grid, Solar Farm. See: CTInfo.java, CTVS.java

  - Step – a form of measurement that I use in the game to determine whether a cell can be scouted, captured, destroyed, etc. There is a very detailed explanation of steps in the GameLogic source code. See: GameLogic.java
  - Enemy Spread – an event where enemies spread from any cells that they inhabit to neighboring cells; These occur on a regular basis that depends on the difficulty of the game. See: GameLogic.java
  - **There are probably more terms that I used throughout the project that I may have forgotten about or overlooked. Please ask me if you have any questions.

- Gameplay – There are just a few basic points to know when playing the game.
  - The player starts at the Space Port.
  - The player's main goal is to capture all 3 Objectives around the map.
  - Scouting - The player can scout the cells around the Space Port.
  - Capturing - After cells have been scouted, the player can capture these cells by using soldiers. Each soldier can defeat 2 enemies.
  - Building - The player can build structures on captured cells if he/she has enough resources to do so, but the player cannot build on the Space Port or any captured Objectives. These structures benefit the player by either increasing the maximum amount of resources that he/she can have or convert two types of resources into the third type of resource.
  - Defending - The player must be wary of enemies while moving to find and capture the Objectives. The Objectives are extremely hostile and produce enemies. To protect the cells that he/she has already captured, the player can station soldiers in a captured cell. These soldiers will defend the cell against all enemy attacks.
  - Enemies - When capturing hostile cells, the player can weaken the enemies in that cell first by bombarding the cell. The player uses energy cores to bombard a cell. Each energy core used will defeat one enemy. The player can also destroy a cell by bombarding a cell with one more energy core than there are enemies. When a cell is destroyed, neither enemies nor the player can move into or do anything with the cell.

    - Ending the Game – the game can end in 1 of 3 ways: the player loses the game by allowing his/her Space Port to be captured by enemies, the player forfeits the game by choosing the “Forfeit” option on his/her Space Port, or the player wins the game by capturing all 3 Objectives
- Resources and Assets – For my fonts, images, and icons, I used online resources.
    - I downloaded a free icon set from which I used a few.
    - I created a lot of the cell pics on logomakr.com.
    - I used Google’s Orbitron font. This is actually where I got the idea for the name of the game from. I just thought that the font was so cool that I named my game after it.