Space Shooter Game Design Document

# Description:

Space Shooter is a turn-based, tactical level, simple simulation of ship to ship space combat. It is played on an 8x8 grid that represents one quadrant and there are 8x8 quadrants in the galaxy. (See SpaceShooterUI.psd in the PhotoShop Documents folder.) To win, the player must destroy all enemy ships within a set amount of time (turns)—which is randomly set at the beginning of each game. If the player’s ship is destroyed or runs out of energy, or runs out of time before all enemy ships are destroyed, the player loses.

Each quadrant might contain enemy ships, a space station, and/or stars. The number of stars is fixed for each quadrant at the start of the game and never changes. The enemy ships and space stations can be destroyed and, if destroyed, are not replaced. Enemy ships and the space station never move to a new quadrant. The player docks with space stations to refuel and speed up repairs.

The player controls shields, movement, long range scans, and firing weapons. The player’s ship, in addition to taking damage (in the form of losing energy) can also suffer damage to critical systems, such as shields, hyper drive, and weapons. Damage to critical systems can result in the temporary loss of functionality of those systems. Damage is repaired over time. Below is a list of actions and the results of loss of a critical system:

1. **Shields**. The player can increase or decrease the amount of energy allocated to the shields. The shields offer protection against critical system damage in proportion to the amount of energy in the shield. If the shields fail, the ship—if hit—is destroyed.
2. **Movement**. The player has both hyper drive (faster than light) and sub light propulsion systems. Movement between quadrants requires the use of hyper drive—with the max distance being 8 quadrants. Movement within a quadrant requires the use of sub light engines. If the hyper drive fails, the player can still move within the quadrant using the sub light engines. If the sub light engines fail, then the player can move between quadrants, but cannot maneuver within a quadrant. If both engines fail, the player cannot move until one of the propulsion systems is repaired.
3. **Weapons**. There are two types of weapons systems: energy weapons (Blasters) and projectile weapons (missiles). Blasters target all enemy ships in a quadrant with a single shot, but divide the total energy by the number of targets. Damage also decreases with distance to the target. Missiles only damage one target and the damage is not affected by distance (although missiles cannot pass between quadrants). Missiles can be affected by objects between the target and the firing ship—such as other enemy ships, stars, or the space station. In which case, they will not reach the target. The probability of a hit decreases with distance. Failure due to critical systems damage to weapons results in the loss of those weapons until they are repaired.
4. **Long Range Scan**. The long range scan shows how many enemy ships, space stations, and stars there are in each of the adjacent quadrants. Failure of the long range scan results in not getting this information.
5. **Galaxy Map**. The galaxy map gives the same information as the long range scan, but for the entire galaxy. If a quadrant has not been visited or scanned, the data in the galaxy map will be blank. The galaxy map reflects that last known state of each quadrant. The galaxy map is not affected by critical systems damage.

# Definitions:

*Blasters*: Energy weapons found on both player and enemy ships, although damage from enemy ships is calculated slightly differently from damage from the player’s ship.

*Docking*: Since only one unit is allowed in each sector, the player’s ship docks with a space station by stopping in one of the eight adjacent squares around the space station.

*Enemy Ship*: This is a computer controlled unit. Destroying all enemy ships in a specified number of turns is the game’s single victory condition.

*Galaxy*: The entire play area composed of 64 quadrants in an 8x8 grid.

*Missile*: Projectile Weapons found only on the player’s ship.

*Player’s Ship*: This is the only unit the player controls.

*Quadrants*: An area of the galaxy composed of 64 sectors in an 8x8 grid.

*Sectors*: The smallest unit of area in the game. A sector can hold only one of the following: an enemy ship, a space station, a star, or the player’s ship. A sector can also be empty.

*Shields*: Energy based armor for the player’s ship. Shields are also found on the space station.

*Space Station*: Refueling and repair base. Docking at a space station instantly refuels the player’s ship. Repairs proceed at a faster rate while docked. The player’s ship is fully protected by the space station’s shields while docked.

*Star*: Stars act as obstacles to both movement and missile firing.

# Walk Through:

When the game starts, it determines the number of enemy ships, space stations, and stars for each quadrant –just the number, not the positions within the quadrant. It also generates the number of turns the player has to destroy all enemy ships and it picks a starting quadrant for the player’s ship. The player’s ship, enemy ships, space station, and stars are then placed into the quadrant—in that order.

The player is given an option to read the rules.

After the player has either read the rules or opted not to read the rules, the game waits for the player to give one of the following orders: movement, long range scan, fire Blaster, fire missile or raise shields.

**Giving Orders**: *(The following information refers to the UI design document, “SpaceShooterUI.psd”, which is subject to change.)*

1. **Movement**:
   1. **Hyper Drive**: If the hyper drive has failed, the hyper drive button will be “greyed” out and will not work. Otherwise, the player left-clicks on hyper drive button and the galaxy map is displayed. The player left-clicks on a quadrant. The target quadrant is created with the appropriate number of units (enemy ships, space station, stars). Each time a quadrant is created, placement is random, so if the player has already visited a quadrant, units will not be in the same locations when he returns. The new quadrant is displayed and the game waits for further orders from the player. It takes the enemy ships at least one turn to react to the arrival of the player’s ship.
   2. **Sub Light Engines**: If the sub light engines have failed, the sub light engine indicator will indicate “damaged” and no movement within the quadrant is possible. If a player left-clicks on a sector with a space station in it, the player moves to the sector next to the space station and docks with the space station. If the player left-clicks on a star, the player is given a warning of an illegal action and nothing else happens. Left-clicking on an enemy ship results in a missile attack on that ship. Otherwise, if the player left-clicks on an empty target sector, the ship will move to that sector after the computer calculates the route to the target sector and determines:
      1. **Enough Fuel**: Does the player’s ship have enough fuel to make the move? If yes, check for obstacles. If no obstacles, move to target sector. If there is not enough fuel, warn the player with a “low fuel” alarm and borrow energy from the shields. If still not enough fuel, ignore the command.
      2. **Obstacles**: Did the player run into a star, enemy ship, or space station? If so, then:
         1. **Star**: Player’s ship takes minor damage and stops in the sector before the star.
         2. **Enemy Ship**: Player’s ship is attacked and damage is assessed. Player may not actually be blocked (random condition). If blocked the player’s ship stops in the sector before the enemy ship, otherwise, the player’s ship continues toward the target sector.
         3. **Space Station**: The player docks with the space station (stops in the sector before the space station’s sector) if the player’s ship has less than 10% of its maximum energy capacity or has one or more failed critical systems. Otherwise, the player’s ship continues toward the target sector.
2. **Weapons:** 
   1. **Blasters:** the player may optionally adjust the energy level to the blasters prior to firing. If the energy level to the blasters is zero or the blasters have failed, the blaster fire button is “greyed” out and the fire button will not work. Otherwise, to fire, the player left-clicks the blaster fire button. The computer then calculates the damage to the enemy ships in the quadrant.
   2. **Missiles:** If the number of remaining missiles is zero or the missile launcher has failed, the missile launcher is “greyed” out and missiles cannot be fired. Otherwise, to fire, the player left-clicks on the targeted enemy ship. The computer calculates the route to the target ship and determines:
      1. **Obstacles:** Does the missile run into a star, enemy ship, or space station? If so then:
         1. **Star:** There is a 90% chance the missile is destroyed, otherwise the missile continues toward the target.
         2. **Space Station:** There is a 10% chance the missile and space station are destroyed, otherwise the missile continues toward the target. (Note: if a player destroys all his space stations, the game is over and the player loses.)
         3. **Enemy Ship:** There is a 50% chance the missile and enemy ship are destroyed, otherwise the missile continues toward the target.
      2. **Hit**: If the missile makes it to the target, the computer calculates whether or not the enemy ship is destroyed. (see table in “Algorithms”)
3. **Shields:**
   1. **Set Energy Level**: If the shields have failed due to critical systems damage, the shields “+” and “-“ buttons are “greyed” out and the shields are inoperable. Otherwise, the player left-clicks “+” button on shields display to increase energy to the shields or left-clicks “-“ button on shield display to decrease energy to the shields. The energy display for the ship is offset by the corresponding amount. If the shield’s energy is increased by 100, the ship’s energy is decreased by 100.
4. **Long Range Scan:**
   1. If the long range scanner has failed due to critical systems damage, the long range scan area will be “greyed” out and no data will be displayed. The long range scan will not respond to left-clicking. Otherwise, left-clicking anywhere in the long range scan area will refresh the data.
5. **Galaxy Map:**
   1. *The present UI image does not have a galaxy map button, but one must be added.* The galaxy map is not subject to critical systems damage. Left-clicking the galaxy map button will result in quadrant map being replaced by a galaxy map. Left-clicking again toggles back to the quadrant map. All buttons are disabled when viewing the galaxy map, but left-clicking anywhere on the screen will toggle back to quadrant view and will re-enable the buttons.
6. **Quit**:
   1. *The present UI image does not have a quit button, but one must be added.* The player left-clicks the quit button to end the game. This will require a second left-click in a dialog box to confirm the player really wants to quit and didn’t click on the button by mistake.

# Algorithms:

1. **Initialization**:
   1. **Game Values:**
      1. **Game Turns**: 25 – 35. [random(0 to 10) + 25]
   2. **Board Setup Values:**
      1. **Order of Unit Placement**: Player ship, enemy ships, space station, stars
      2. **Stacking Rule**: Regardless of type, only one unit is allowed per sector.
      3. **Probability of Units in a Quadrant**:
         1. **Enemy Ships**: Calculate in the following order
            1. 2% chance quadrant will have 3 enemy ships
            2. 5% chance quadrant will have 2 enemy ships
            3. 20% chance quadrant will have 1 enemy ship
            4. otherwise, the quadrant contains no enemy ships
         2. **Space Station:** 
            1. 4% chance of 1 space station
            2. Otherwise, the quadrant contains no space station
         3. **Stars**: Random(0 to 7)
      4. **Special Placement Rule**: If no space stations are generated after all quadrants have been created, put a space station in the same quadrant as the friendly ship and increase the number of enemy ships by one—unless there are already 3 enemy ships in the quadrant, in which case, leave the number of enemy ships at 3. Randomly select a new quadrant and sector for the friendly ship.
   3. **Player Ship Values:**
      1. **Initial Position**: Randomly select quadrant and sector
      2. **Initial Ship Energy/Maximum Ship Energy**: 3000
      3. **Initial Shield Energy**: 0
      4. **Maximum Missiles**: 10
   4. **Enemy Ship Values**:
      1. **Initial Ship Energy**: 200
2. **Movement**:
   1. **Player Ship**:
      1. **Cost**: 10 units of energy plus 1 unit for every sector travelled. Energy will be automatically drained from shields if required. The game is over and the player loses when the total ship energy and shield energy is zero.
      2. **Stacking Rule**: A sector can only contain one unit.
      3. **Obstacles**: If a player’s ship passes through a sector containing a unit, calculate the following.
         1. **Enemy Ship**: In separate calculations, each ship has a 10% chance of 10-50 units of damage. [random(0 to 40) + 10]
         2. **Space Station**: If any of the following conditions exist, player’s ship does not enter the sector with the space station, but instead docks next to it.
            1. **Remaining Ship Energy**: The player’s ship has 10% or less of maximum energy. [300 = 3000 \* .1]
            2. **Critical Systems Damage**: The player’s ship has damage to any critical system.
         3. **Star**: 30% chance of 10 – 100 units of damage. [random(0 to 90) + 10]
   2. **Enemy Ship**: Enemy ships have a “Combat Mode” attribute which can be either true or false. All enemy ships when first placed in a quadrant set Combat Mode to false. There is a 75% chance that the Combat Mode will change to true each time the player moves. This is calculated separately for each enemy ship. All enemy ships set their Combat Mode to true once the player has fired. There is no energy cost for enemy movement.
      1. **Combat Mode is false**: Enemy ship doesn’t move.
      2. **Combat Mode is true**: Enemy ship randomly selects an empty sector and moves there.
   3. **Missiles**:
      1. **Cost**: 2 units of energy for every missile fired.
      2. **Obstacles**: If a missile passes through a sector containing a unit, calculate the following
         1. **Enemy Ship**: 50% chance of destroying this enemy ship instead of the original target.
         2. **Space Station**: 10% chance of destroying the space station instead of the original target.
         3. **Star**: 90% chance missile is destroyed.
3. **Combat Results**:
   1. **Player Ship**: When both the ship and shield energy are reduced to zero or less, the ship is destroyed.
      1. **Blaster Attack**: Evenly divide the blaster energy between all enemy ships in the quadrant. Divide this value by the distance between the player’s ship and the target times random(2 to 3). Distance is calculated by taking using the largest of either the number of rows or number of columns between ships. If the resulting number (hit value) is less than 15% of the remaining enemy power, no damage is done, otherwise reduce the enemy power by the hit value.
      2. **Missile Attack**: Check for obstacle effects as documented in “Movement” section. If missile has not been destroyed by obstacle effects the enemy ship will either be destroyed or missed using the following: (.5 + (1/distance)) chance of enemy ship being destroyed. Distance is calculated as described in blaster attack.
      3. **Critical Systems Damage**: 5% chance of critical systems damage to be calculated for each critical system when the enemy attack results in a hit energy greater than 20% of the shield energy.
         1. **Effects of a critical hit**:
            1. **Long Range Scanner**: Long range scan doesn’t work until it is repaired. Repair time is 3 to 10 turns.
            2. **Blasters**: Cannot fire blasters until they are repaired. Repair time is 1 to 10 turns.
            3. **Missile Launcher**: Cannot fire missiles until launcher is repaired. Repair time is 1 to 5 turns.
            4. **Shields**: Shield energy drops to zero, but 50% of the energy is returned to ships energy. Energy cannot be moved back into shields until they are repaired. Repair time is 1 to 3 turns.
            5. **Hyper Drive**: Cannot move out of the quadrant until hyper drive is restored. Repair time is 4 to 10 turns.
            6. **Sub Light Engines**: Cannot move within a sector until the sub light engines are repaired. Repair time is 2 to 4 turns.
   2. **Enemy Ship**: When the enemy ship’s energy is reduced to zero or less, the enemy ship is destroyed.
      1. **Attack**: Hit energy is enemy ship energy divided by the product of the distance and a random number between 2 to 3. Enemy ship energy is reduced by 1/random (3 to 4).
      2. **Damage**: Player ship shield energy is reduced by the amount of the hit energy. If shield energy drops below zero, player ship is destroyed and game is lost.
4. **Space Station**:
   1. **Restocking**:
      1. **Energy**: Ship energy is reset to 3000.
      2. **Shields**: Shields must be lowered to dock, so shield energy is set to zero.
      3. **Missiles**: Missiles are reset to the maximum capacity of 10.
   2. **Repairs**: Repair times are two to three times faster when docked. Shields are immediately repaired.
5. **Ship Battle Condition Status:**
   1. **Red**: Quadrant has one or more enemy ships.
   2. **Yellow**: No enemy ships in quadrant, but ship energy is 10% or less of maximum energy.
   3. **Green**: No enemy ships in quadrant and energy is above 10%.
6. **Victory Conditions**:
   1. **Win**: All enemy ships are destroyed before time runs out.
   2. **Lose**:
      1. Enemy ships remain after time runs out.
      2. Player’s ship is destroyed.
      3. Player’s ship runs out of energy.
      4. All space stations are destroyed.