Explanation of the state invariants of the system

The Asteroid game is made up of four machines.

1. Space – For definitions of the space and its boundaries.
2. Asteroid – For definitions of the asteroids and its positions. Asteroid SEES Space.
3. Spaceship – For definitions, states and operations of the spaceship. Spaceship SEES Space and Asteroid.
4. Game – Top level machine that integrates above three to make up the game and the game rules. Game EXTENDS Spaceship.

These machines consist of 2 enumerated SETS,

```
ALERT = { Exceeding_Boundry, Astroid_Strike, No_Power,
Moved_Up, Moved_Right, Moved_Down, Moved_Left, Warped, Ready }
```

to define the alert messages that are sent at the end of each operation, and

```
GAME_STATUS = { READY, PLAYING, WON, LOST }
```

to define the status of the game as an enquiry operation.

The state invariant explanations are as follows,

Space

The space doesn't have any invariants. Its initialized with the boundary values.

Asteroids

```
asteroids : xRegion <-> yRegion & card(asteroids) = 11
```
The 11 asteroids can only be within the boundary of the space.

Spaceship

```
shipXPosition : xRegion & shipYPosition : yRegion &
(shipXPosition |-> shipYPosition) /: asteroids
```
The spaceship can only be within the limits of the space and cannot be the asteroid location.

```
currentPower : NATURAL & collisionsCount : NATURAL
```
Power remaining in the ship can never be below zero. The number of collisions has to be a non-zero full number.

```
visitedSpaceRegions : seq(space)
```
The visited space path can only have values within the boundary of the space.

Thiloshon Nagarajah | 2015298