INFORMATICS INSTITUTE OF TECHNOLOGY

UNIVERSITY OF WESTMINSTER

INFORMATICS INSTITUTE OF TECHNOLOGY

in collaboration with

UNIVERSITY OF WESTMINSTER

Module: 6SENG003C

Reasoning about Programs Coursework

Student Name : Mohamed Boosary Fathima Farheen

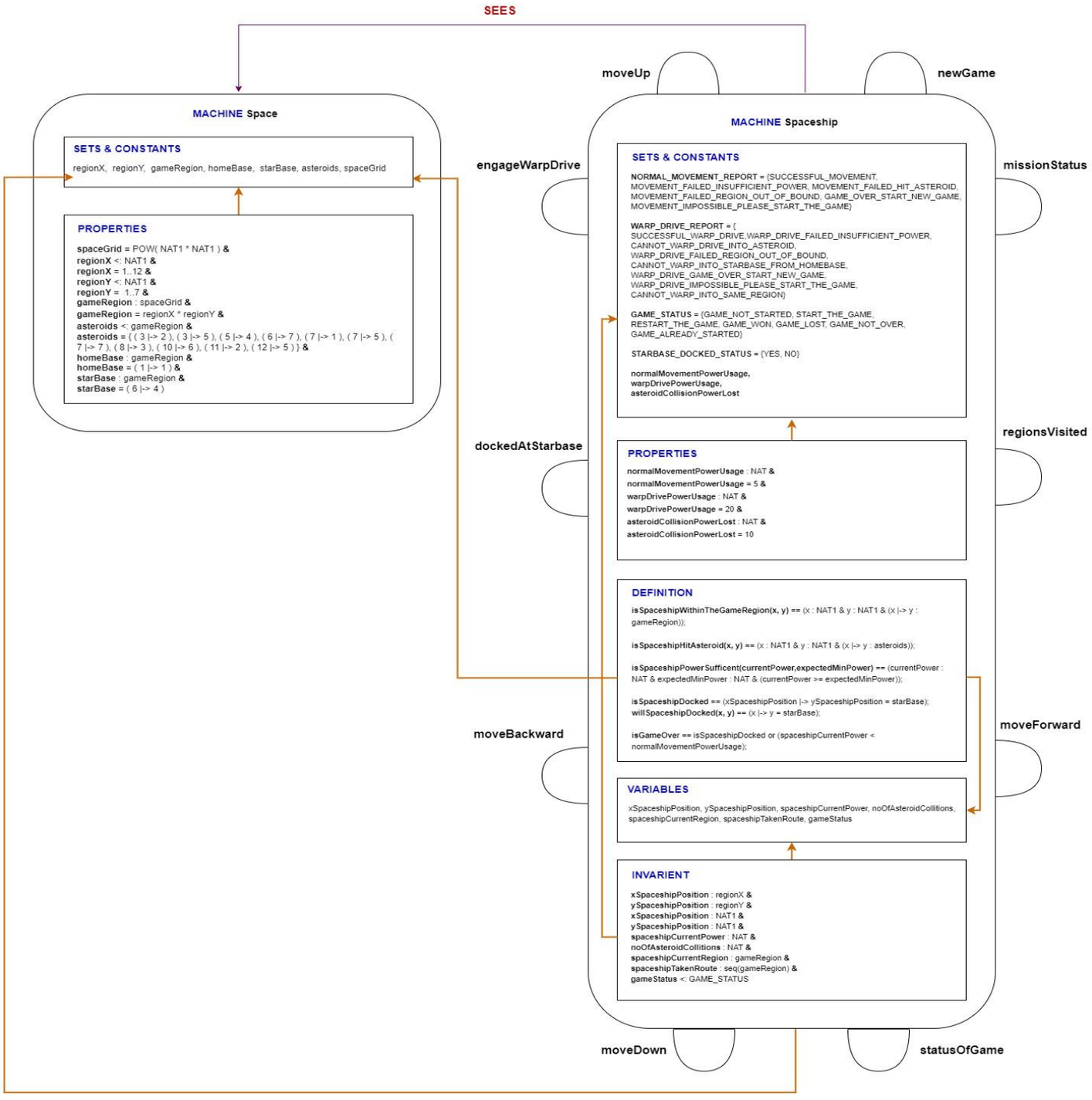
Student Id : 2018323

UOW Id : w1742063

Degree : BEng (Hons) in Software Engineering

Date of Submission : 3rd December 2021

1) Structure diagram of the Spacehsip & Asteriods game system



2) Assumptions taken, Invariant & Sets explanation (in plain english)

a) Assumptions taken in to count in the spaceship & asteriod system

- Spaceship cannot peform warp drive movement **directly** from Homebase to Starbase it should at least do one normal movement to be able to perform direct warp drive movement to Starbase.
- The initial power of the spaceship is taken as zero power and its is required to **perform newGame operation** in order to start and initialize the spaceship power.
- The newGame operation will be consider as **both game start & restart operation** rather only considering the operation as a reset/restart game operation.
- If the spaceship docked in the starbase and WON the game then the spaceship cannot do any normal or warp drive jump movement its need to restart the game from the beginning as the game is over at that point.
- Warp drive jump is not allowed to perform to the same position that spaceship is current located, it will only allow to perform Warp drive jump to a new position.
- The **newGame(power)** operation will allow to assign power for the spaceship starting from 25 onwards to any power value, as it will be the minimum power required to win the game.

b) State invariant Description

- xSpaceshipPosition : regionX

  The x value/position of the spaceship is a member of the game space X region, which means that x movement value of the spaceship should be a value in the game space X region.

- ySpaceshipPosition : regionY

  The y value/position of the spaceships is a member of the game space Y region, which means that y movement value of the spaceship should be a value in the game space Y region.

- xSpaceshipPosition : NAT1

  The x dimension of the spaceship is a member of natural number starting from 1.

- ySpaceshipPosition : NAT1

  The y dimension of the spaceship is a member of natural number starting from 1.

- spaceshipCurrentPower : NAT

  The spaceship will contain a power in order to move through the space. So the current power of the spaceship will take a value of natural number set that start from 0.

- noOfAsteroidCollitions : NAT

  The number of asteroid collitions of the spaceship is a member of natural number set starting from 0, as there will be possibilities of spaceship colliding with asteroids as well not colliding with any asteroids at all.

- spaceshipCurrentRegion : gameRegion

  The current position (x,y) of the spaceship is a member of the game space region

- spaceshipTakenRoute : seq(gameRegion)

  All the regions covered by spaceship is member of the sequence of game space regions.

- gameStatus <: GAME_STATUS

  The current game status is a subset of the GAME_STATUS set.

c) Set Description

NORMAL_MOVEMENT_REPORT = {
SUCCESSFUL_MOVEMENT, MOVEMENT_FAILED_INSUFFICIENT_POWER,
MOVEMENT_FAILED_HIT_ASTEROID,
MOVEMENT_FAILED_REGION_OUT_OF_BOUND,
GAME_OVER_START_NEW_GAME,
MOVEMENT_IMPOSSIBLE_PLEASE_START_THE_GAME
} ;

Contains all the possible results that occur in an attempted normal movement of the spaceship such as successful, failure dues to collition of asteroids, out of boundary etc.

WARP_DRIVE_REPORT = {
SUCCESSFUL_WARP_DRIVE,
WARP_DRIVE_FAILED_INSUFFICIENT_POWER,
CANNOT_WARP_DRIVE_INTO_ASTEROID,
WARP_DRIVE_FAILED_REGION_OUT_OF_BOUND,
CANNOT_WARP_INTO_STARBASE_FROM_HOMEBASE,
WARP_DRIVE_GAME_OVER_START_NEW_GAME,
WARP_DRIVE_IMPOSSIBLE_PLEASE_START_THE_GAME,
CANNOT_WARP_INTO_SAME_REGION
};

Contains all the possible results that occur in an attempted warp drive jump movement of the spaceship such as successful, failure dues to collition of asteroids, out of boundary etc

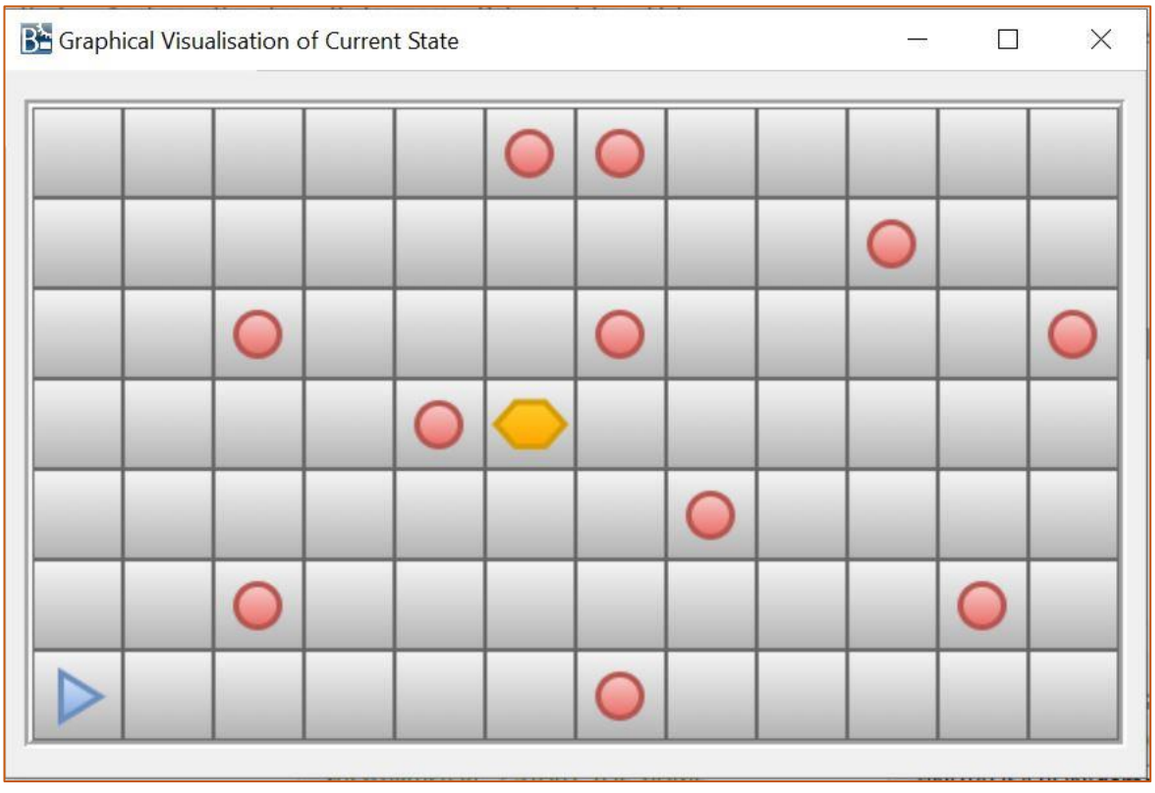
GAME_STATUS = {
GAME_NOT_STARTED,
START_THE_GAME,
RESTART_THE_GAME, GAME_WON,
GAME_LOST, GAME_NOT_OVER,
GAME_ALREADY_STARTED
};

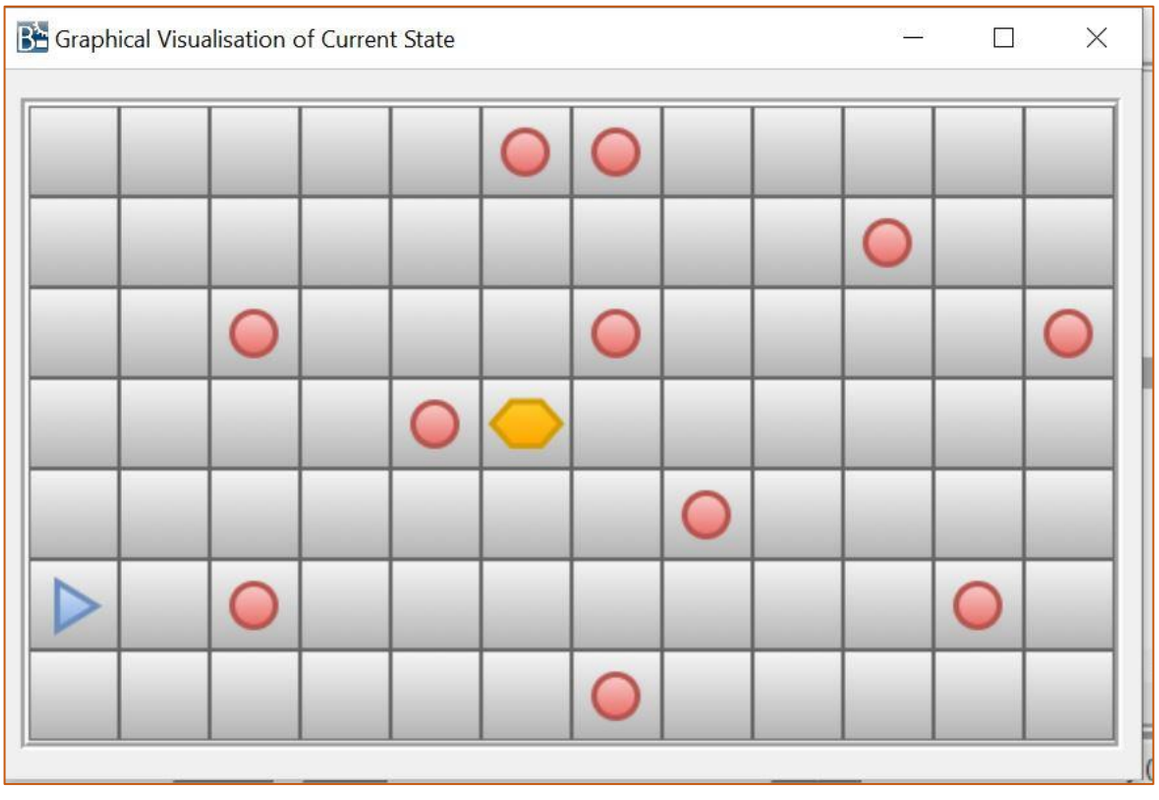
Contains a list of game status related to the spaceship in the game such as WON, LOST, GAME NOT OVER etc.

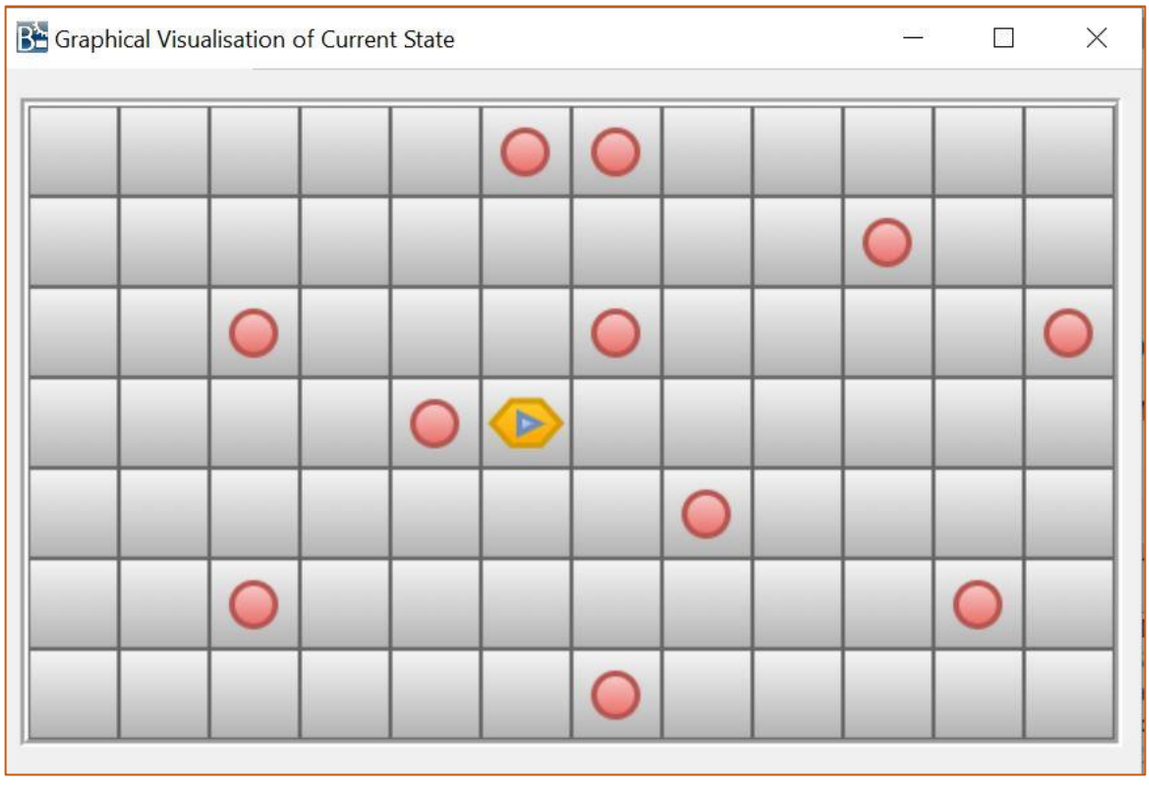
STARBASE_DOCKED_STATUS = {YES, NO}

A List (SET) that contain the docking status of the spaceship in the Starbase, such as YES or NO.

3) Screenshots of the Graphical Visualization of the specified system






Graphical Visualisation of Current State