

Degree Program: BEng. (Hons) in Software Engineering

Lecturer Name: Dr.Thilini Piyatilake



**MODULE: 6SENG005C.1 Formal Methods**

Student Name: H.K.J.N.Gunaweera

IIT ID: 20200003

UoW ID: w1810567

**Table of Contents**

[1.0 Structure Diagram – B specification 4](#_Toc148595871)

[2.0 Rationale and Elaboration 5](#_Toc148595894)

[2.1 SETS 6](#_Toc148595895)

[2.2 CONSTANTS & PROPERTIES 8](#_Toc148595896)

[2.3 VARIABLES & INVARIANTS 10](#_Toc148595897)

[2.4 INITIALISATION 15](#_Toc148595898)

[3.0 Implementation Screenshots 16](#_Toc148595894)

# List of tables

[Table 1- SETS 6](#_Toc148559176)

[Table 2- CONSTANTS & PROPERTIES 9](#_Toc148559177)

[Table 3- VARIABLES & INVARIANTS 11](#_Toc148559178)

# List of Figures

[Figure 1 Structure Diagram 4](#_Toc148596458)

[Figure 2 SETS 6](#_Toc148596459)

[Figure 3 CONSTANTS & PROPERTIES 8](#_Toc148596458)

[Figure 4 VARIABLES & INVARIANTS 10](#_Toc148596459)

[Figure 5 INITIALISATION 15](#_Toc148596458)

# 1.0 Structure Diagram – B specification

A close-up of a computer screen

Description automatically generated

Figure 1 Structure Diagram

Visit <https://drive.google.com/file/d/1HS33AT6JUt8qlj2LLbcvWzhdlK1eMQzE/view?usp=sharing> to see the diagram more clearly

Model Type: Finite State Machine (FSM)

Model Overview:

* The model employs a finite state machine (FSM) to represent the spaceship's behavior within the game environment.
* The FSM is comprised of a finite set of states, interconnected by transitions triggered by specific events.
* The model's initial state is denoted as "Game\_not\_over."
* The model incorporates tables for sets, constants, properties, and variables, providing further context for its operation.

States:

* Game\_not\_over: The initial state, from which transitions to the following states can occur:
  + MoveUp
  + MoveDown
  + MoveForward
  + MoveReverse
  + EngageWarpDrive
* MoveUp: Represents the spaceship's upward movement. Transitions to:
  + Moved Successfully (upon successful movement)
  + Move Failed Unable to Visit Unknown Space (upon attempting to access an unknown area)
* MoveDown: Represents the spaceship's downward movement. Transitions to the same states as MoveUp.
* MoveForward: Represents the spaceship's forward movement. Transitions to:
  + Freespace (upon successful movement)
  + Asteroid Collision (upon collision with an asteroid)
* MoveReverse: Represents the spaceship's reverse movement. Transitions to the same states as MoveForward.
* EngageWarpDrive: Represents the spaceship's engagement of its warp drive. Transitions to:
  + Freespace (upon successful warp)
  + Warp Failed states (upon unsuccessful warp)
* Terminal States:
  + Moved Successfully
  + Freespace
  + DockedAtStarBase
  + These states signify the completion of a particular action or sequence, with no further transitions possible.

# 2.0 Rationale and Elaboration

This project entails the creation of a B specification for a simplified iteration of the Spaceship & Asteroids arcade game, utilizing B tools. A detailed exposition of the assigned and employed SETS, CONSTANTS, PROPERTIES, VARIABLES, and INVARIANTS is presented below.

## 2.1 SETS

The project involves two main machines: SpaceRegion and Spaceship. Each machine comprises several sets crucial for their functionality. The SpaceRegion machine sets define the spatial characteristics of the game, while the Spaceship machine sets are focused on the spaceship's movements, messages, and overall game status.

A screenshot of a computer

Description automatically generated

Figure 2 SETS

|  |  |  |  |
| --- | --- | --- | --- |
| **Machine** | **Set** | **Elements** | **Description** |
| SpaceRegion | **MAX\_X\_AXIS** | 1..12 | Maximum value for the X-axis in the space region. |
| **MAX\_Y\_AXIS** | 1..7 | Maximum value for the Y-axis in the space region. |
| **SPACE\_REGION\_COORDINATES** | All pairs (x, y) where x is in MAX\_X\_AXIS and y is in MAX\_Y\_AXIS | Set of all coordinates in the space region. |
| **ASTEROID\_COORDINATES** | {(3,2), (3,5), (5,4), (6,7), (7,1), (7,5), (7,7), (8,3), (10,6), (11,2), (12,5)} | Set of coordinates occupied by asteroids. |
| **EMPTY\_SPACE\_COORDINATES** | All coordinates in SPACE\_REGION\_COORDINATES not in ASTEROID\_COORDINATES | Set of coordinates available for spaceship movement (excluding asteroid coordinates). |
| **HOME\_BASE\_COORDINATE** | (1,1) | Coordinate of the home base. |
| **STAR\_BASE\_COORDINATE** | (6,4) | Coordinate of the star base. |
| Spaceship | **MOVEMENT\_TYPE** | {MOVE\_UP, MOVE\_DOWN, MOVE\_FORWARD, MOVE\_BACKWARD, WARP\_DRIVE} | Movement types of spaceships |
| **REPORT\_MESSAGE** | {SUCCESSFULLY\_MOVED\_UP,  SUCCESSFULLY\_MOVED\_DOWN,  SUCCESSFULLY\_MOVED\_FORWARD,  SUCCESSFULLY\_MOVED\_BACKWARD,  SUCCESSFULLY\_WARPED,  CANNOT\_MOVE\_UP\_DUE\_TO\_SPACE\_BOUNDRY,  CANNOT\_MOVE\_DOWN\_DUE\_TO\_SPACE\_BOUNDRY,  CANNOT\_MOVE\_FORWARD\_DUE\_TO\_SPACE\_BOUNDRY,  CANNOT\_MOVE\_BACKWARD\_DUE\_TO\_SPACE\_BOUNDRY,  CANNOT\_WARP\_DUE\_TO\_SPACE\_BOUNDRY,  ASTEROID\_COLLISION,  INSUFFICIENT\_POWER,  CANNOT\_WARP\_INTO\_ASTEROID,  CANNOT\_WARP\_TO\_THE\_SAME\_POSITION,  CANNOT\_IMMEDIATELY\_WARP\_INTO\_STARBASE} | Appropriate report messages for various actions and conditions. |
| **DOCKED\_STATUS** | {DOCKED\_AT\_STARBASE, NOT\_DOCKED\_AT\_STARBASE} | Spaceship docked status. |
| **GAME\_STATUS** | {GAME\_WON, GAME\_LOST, GAME\_NOT\_OVER} | Game status messages. |
| **InitialPower** | 100 | Initial power for the spaceship. |
| **PowerForNormalMove** | 5 | Power usage for a normal move. |
| **PowerForWarpDrive** | 20 | Power usage for a warp drive. |
| **PowerLossForAsteroidCollision** | 10 | Power loss for asteroid collision. |

Table 1- SETS

## 2.2 CONSTANTS & PROPERTIES

CONSTANTS and PROPERTIES play a crucial role in defining the characteristics and behavior of the SpaceRegion and Spaceship machines.

A screenshot of a computer

Description automatically generated

Figure3 CONSTANTS

A screenshot of a computer

Description automatically generated

*Figure4 PROPERTIES*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Machine** | **CONSTANT** | **CONSTANT explanation** | **PROPERTY** | **PROPERTY explanation** |
| SpaceRegion | **MAX\_X\_AXIS, MAX\_Y\_AXIS** | Maximum values for the X and Y axes in the space region. | **MAX\_X\_AXIS <: NATURAL1 & MAX\_X\_AXIS = 1..12** | Ensures that MAX\_X\_AXIS is a natural number between 1 and 12. |
| **SPACE\_REGION\_COORDINATES** | Set of all coordinates in the space region based on MAX\_X\_AXIS and MAX\_Y\_AXIS. | **MAX\_Y\_AXIS <: NATURAL1 & MAX\_Y\_AXIS = 1..7** | Ensures that MAX\_Y\_AXIS is a natural number between 1 and 7. |
| **ASTEROID\_COORDINATES** | Set of coordinates occupied by asteroids in the space region. | **SPACE\_REGION\_COORDINATES : MAX\_X\_AXIS <-> MAX\_Y\_AXIS** | Establishes a relationship between SPACE\_REGION\_COORDINATES and the X and Y axes. |
| **EMPTY\_SPACE\_COORDINATES** | Set of coordinates available for spaceship movement, excluding those occupied by asteroids. | **ASTEROID\_COORDINATES <: SPACE\_REGION\_COORDINATES** | Specifies that ASTEROID\_COORDINATES is a subset of SPACE\_REGION\_COORDINATES. |
| **HOME\_BASE\_COORDINATE, STAR\_BASE\_COORDINATE** | Coordinates of the home base and star base within EMPTY\_SPACE\_COORDINATES. | **EMPTY\_SPACE\_COORDINATES = SPACE\_REGION\_COORDINATES - ASTEROID\_COORDINATES** | Defines EMPTY\_SPACE\_COORDINATES as the set difference between SPACE\_REGION\_COORDINATES and ASTEROID\_COORDINATES. |
|  |  | **HOME\_BASE\_COORDINATE : EMPTY\_SPACE\_COORDINATES & HOME\_BASE\_COORDINATE = (1** | Specifies that HOME\_BASE\_COORDINATE is within EMPTY\_SPACE\_COORDINATES and defines its value. |
|  |  | **STAR\_BASE\_COORDINATE : EMPTY\_SPACE\_COORDINATES & STAR\_BASE\_COORDINATE = (6** | Specifies that STAR\_BASE\_COORDINATE is within EMPTY\_SPACE\_COORDINATES and defines its value. |
| Spaceship | **InitialPower, PowerForNormalMove, PowerForWarpDrive, PowerLossForAsteroidCollision** | Constants for initial power, power usage for normal move, power usage for warp drive, and power loss for asteroid collision. | **InitialPower = 100** | Sets the initial power for the spaceship to 100. |
|  |  | **PowerForNormalMove = 5** | Sets the power usage for a normal move to 5. |
|  |  | **PowerForWarpDrive = 20** | Sets the power usage for a warp drive to 20. |
|  |  | **PowerLossForAsteroidCollision = 10** | Sets the power loss for an asteroid collision to 10. |

Table 2- CONSTANTS & PROPERTIES

## 2.3 VARIABLES & INVARIANTS

These VARIABLES and INVARIANTS are crucial for maintaining the consistency and correctness of the system's state and behavior during the execution of operations. They help define constraints on the allowed values and relationships between different components of the system.

**Variables**

Variables represent the dynamic state of a system. They are used to model values that can change during the execution of the system.

**A screenshot of a computer

Description automatically generated**

*Figure4 VARIABLES*

*Figure A screenshot of a computer

Description automatically generatedVARIABLE Coverage*

A screenshot of a computer

Description automatically generated *Figure VARIABLE Read/Write matrix*

**Purpose of variables**

Variables help track the changing aspects of a system over time. In the context of the provided machines (SpaceRegion and Spaceship), variables include elements such as the current position of a spaceship (xPosition and yPosition), the remaining power of the spaceship (power), the movements made by the spaceship (movements), the route taken by the spaceship (missionRoute), and the number of collisions the spaceship has encountered (numberOfCollisions).

**Usage of variables**

They are manipulated by operations in the machine. Operations can modify the values of variables based on certain conditions and constraints.

**Invariant**

Invariants are logical conditions that must hold true throughout the execution of the system. They provide constraints on the possible values of variables.

**A screenshot of a computer

Description automatically generated**

*Figure3 INVARIANTS*

*A screenshot of a computer

Description automatically generated*

*Figure Specialized INVARIANTS*

***A screenshot of a computer

Description automatically generated***

*Figure INVARIANT Analysys*

**Purpose of invariants**

Invariants ensure that certain properties remain unchanged during the execution of the system. They help in specifying correctness conditions that should be preserved at all times.

**Usage of Invariants**

They are conditions that must be satisfied before and after the execution of any operation. In the provided machines, invariants include conditions such as the current position being within the bounds of the space region, the power being an integer, the movements being a sequence of predefined types, the mission route consisting of coordinates in empty space, and the number of collisions being an integer.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Machine** | **VARIABLE** | **VARIABLE**  **description** | **INVARIANT** | **INVARIANT**  **description** |
| SpaceRegion | **(none)** | No specific variables defined in the SpaceRegion machine | **xPosition : MAX\_X\_AXIS** | The xPosition variable, representing the current x-coordinate of the spaceship, is within the bounds specified by the MAX\_X\_AXIS constant.  This ensures that the x-coordinate of the spaceship is a valid value within the defined space region along the X-axis. |
|  |  | **yPosition : MAX\_Y\_AXIS** | The yPosition variable, representing the current y-coordinate of the spaceship, is within the bounds specified by the MAX\_Y\_AXIS constant.  Similar to the previous invariant, this ensures that the y-coordinate of the spaceship is a valid value within the defined space region along the Y-axis. |
|  |  | **xPosition |-> yPosition : EMPTY\_SPACE\_COORDINATES** | The pair (xPosition, yPosition) represents a coordinate in the empty space as defined by EMPTY\_SPACE\_COORDINATES.  This ensures that the current position of the spaceship is within the set of coordinates available for movement, excluding positions occupied by asteroids. |
|  |  | **power : INTEGER** | The power variable, representing the current power level of the spaceship, is an integer.  This ensures that the power level is a whole number, conforming to the mathematical definition of integers. |
|  |  | **movements : seq(MOVEMENT\_TYPE)** | The movements variable, representing the sequence of movements made by the spaceship, is a sequence of elements from the MOVEMENT\_TYPE set.  This guarantees that the recorded movements are a valid sequence of predefined types, ensuring consistency in the recorded actions. |
|  |  | **missionRoute : seq(EMPTY\_SPACE\_COORDINATES)** | The missionRoute variable, representing the sequence of coordinates in the spaceship's route, is a sequence of coordinates in empty space.  This ensures that the recorded route consists only of coordinates in the empty space, excluding positions occupied by asteroids. |
|  |  | **numberOfCollisions : INTEGER** | The numberOfCollisions variable, representing the count of collisions encountered by the spaceship, is an integer.  This ensures that the count of collisions is a whole number, conforming to the mathematical definition of integers. |
| Spaceship | **xPosition** | This variable represents the current x-coordinate of the spaceship in the space region. | **xPosition : MAX\_X\_AXIS** | This invariant ensures that the xPosition variable remains within the allowed bounds specified by MAX\_X\_AXIS. |
| **yPosition** | This variable represents the current y-coordinate of the spaceship in the space region. | **yPosition : MAX\_Y\_AXIS** | This invariant ensures that the yPosition variable remains within the allowed bounds specified by MAX\_Y\_AXIS. |
| **power** | This variable represents the current power level of the spaceship. It is used to model the available energy or resources that the spaceship has for its operations. | **xPosition |-> yPosition : EMPTY\_SPACE\_COORDINATES** | This invariant ensures that the current position of the spaceship is within the set of empty space coordinates, meaning it is a valid position for the spaceship to occupy. |
| **movements** | This variable is a sequence that stores all the movements performed by the spaceship. Each element in the sequence corresponds to a specific type of movement (e.g., MOVE\_UP, MOVE\_DOWN). | **power : INTEGER** | This invariant ensures that the power variable takes on integer values. |
| **missionRoute** | This variable is a sequence that stores the coordinates representing the route taken by the spaceship. The coordinates are expected to be in empty space, as specified by the invariant. | **movements : seq(MOVEMENT\_TYPE)** | This invariant ensures that the movements variable is a sequence composed of elements from the MOVEMENT\_TYPE set, representing valid spaceship movements. |
| **numberOfCollisions** | This variable keeps track of the total number of collisions encountered by the spaceship. | **missionRoute : seq(EMPTY\_SPACE\_COORDINATES)** | This invariant ensures that the missionRoute variable is a sequence of coordinates that are all within the set of empty space coordinates. |
|  |  | **numberOfCollisions : INTEGER** | This invariant ensures that the numberOfCollisions variable takes on integer values. |

*Table 3- VARIABLES & INVARIANTS*

## 2.4 INITIALISATION

The initializations provided for both the SpaceRegion and Spaceship machines are a set of instructions that define the starting state of the machines when a new instance of the game is initialized. These initializations ensure that the game starts with the spaceship positioned at the **home base, full power, no previous movements or collisions**, and with the **necessary data structures initialized** for tracking the spaceship's journey.

A screenshot of a computer

Description automatically generated

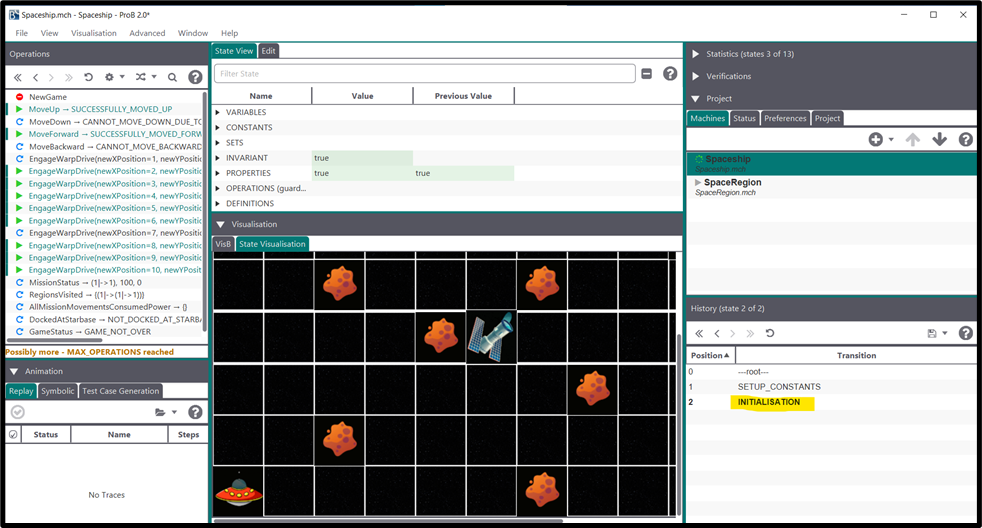
*Figure 4 INITIALISATION*

# 3.0 Implementation Screenshots

1. A screenshot of a computer

   Description automatically generatedInitialising the Game

1. Start of the Game



1. Successful Normal Movement -MOVE\_UP

A screenshot of a computer

Description automatically generated

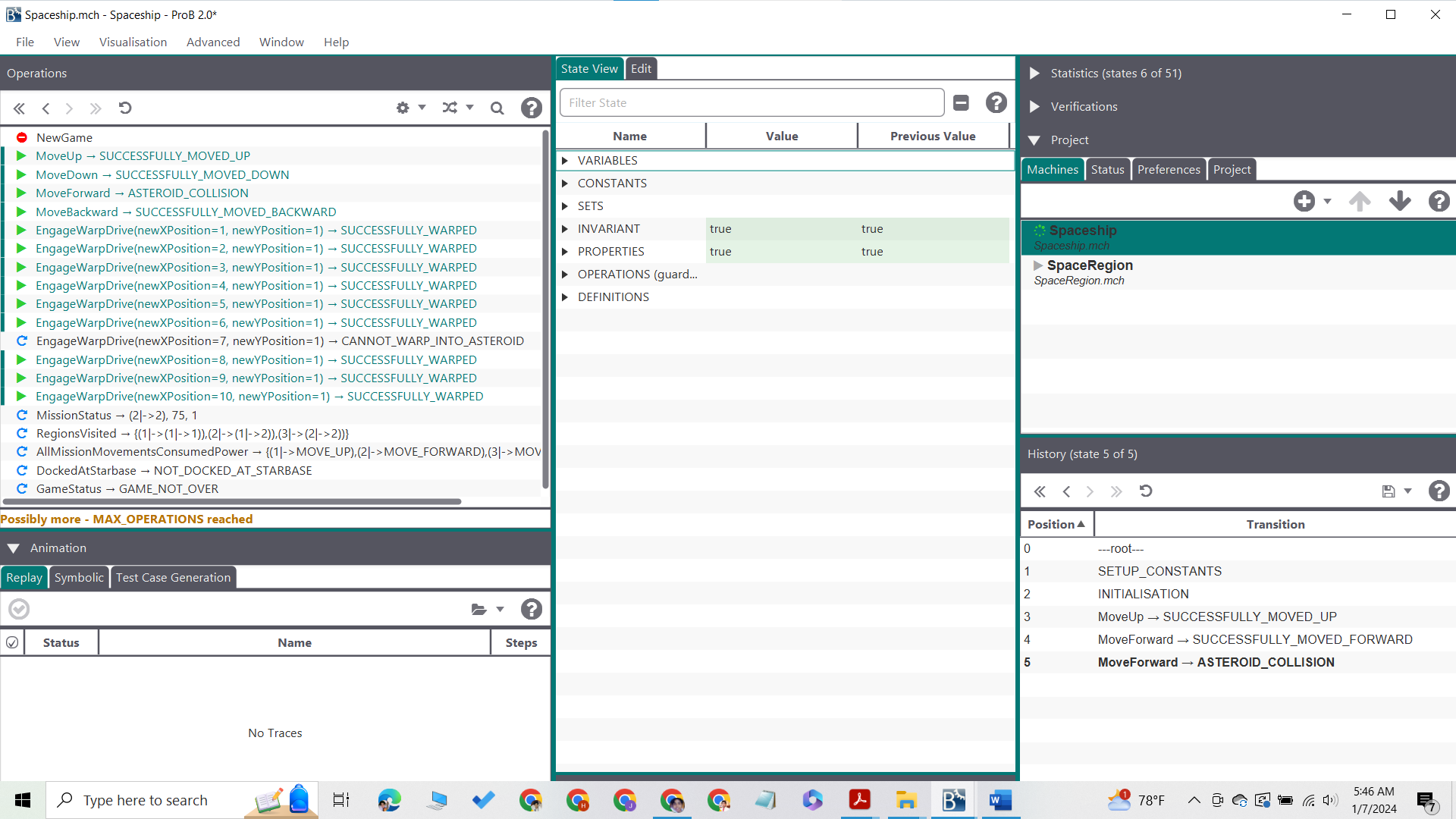
A screenshot of a game

Description automatically generated

A screenshot of a video game

Description automatically generated

1. Failed movement due to unknown space and asteroid collision



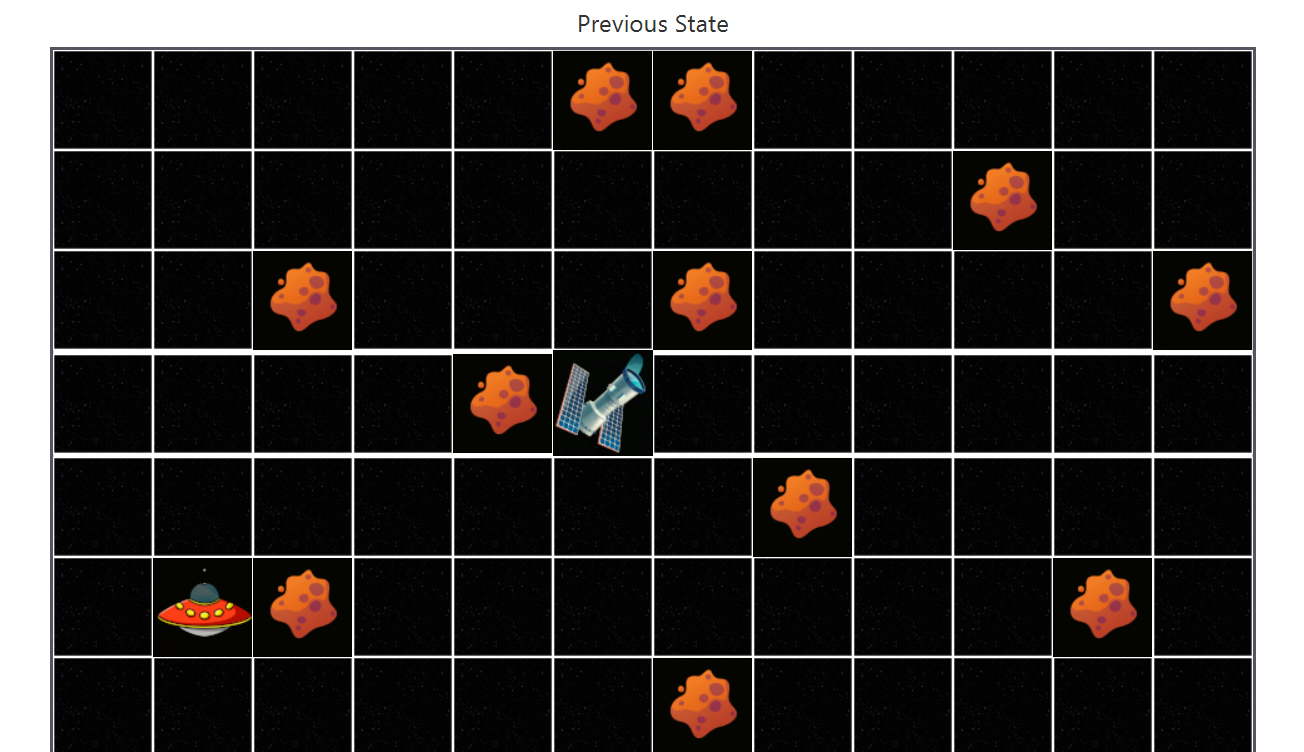
A screenshot of a game

Description automatically generated

1. Successful warp jump

A screenshot of a computer

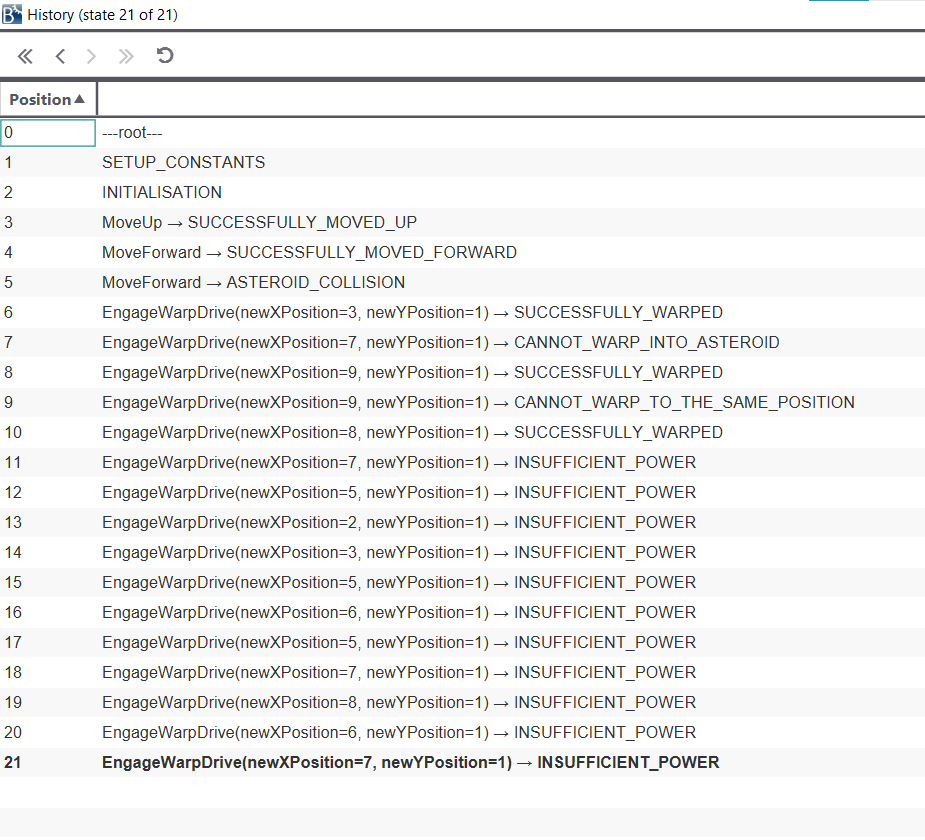
Description automatically generated



A screenshot of a video game

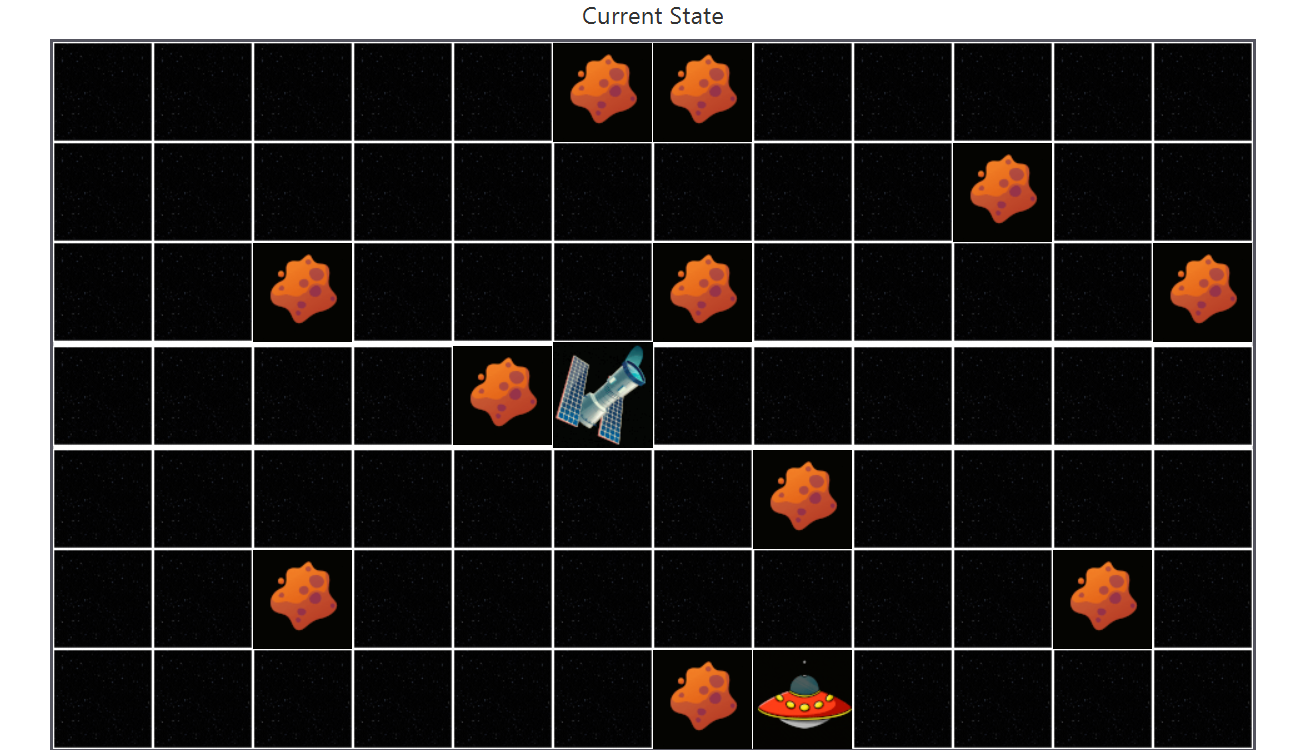
Description automatically generated

1. A screenshot of a computer

   Description automatically generatedFailed warp jump due to same position, asteroid collision and unknown space errors
2. Failed warp jump due to insufficient power

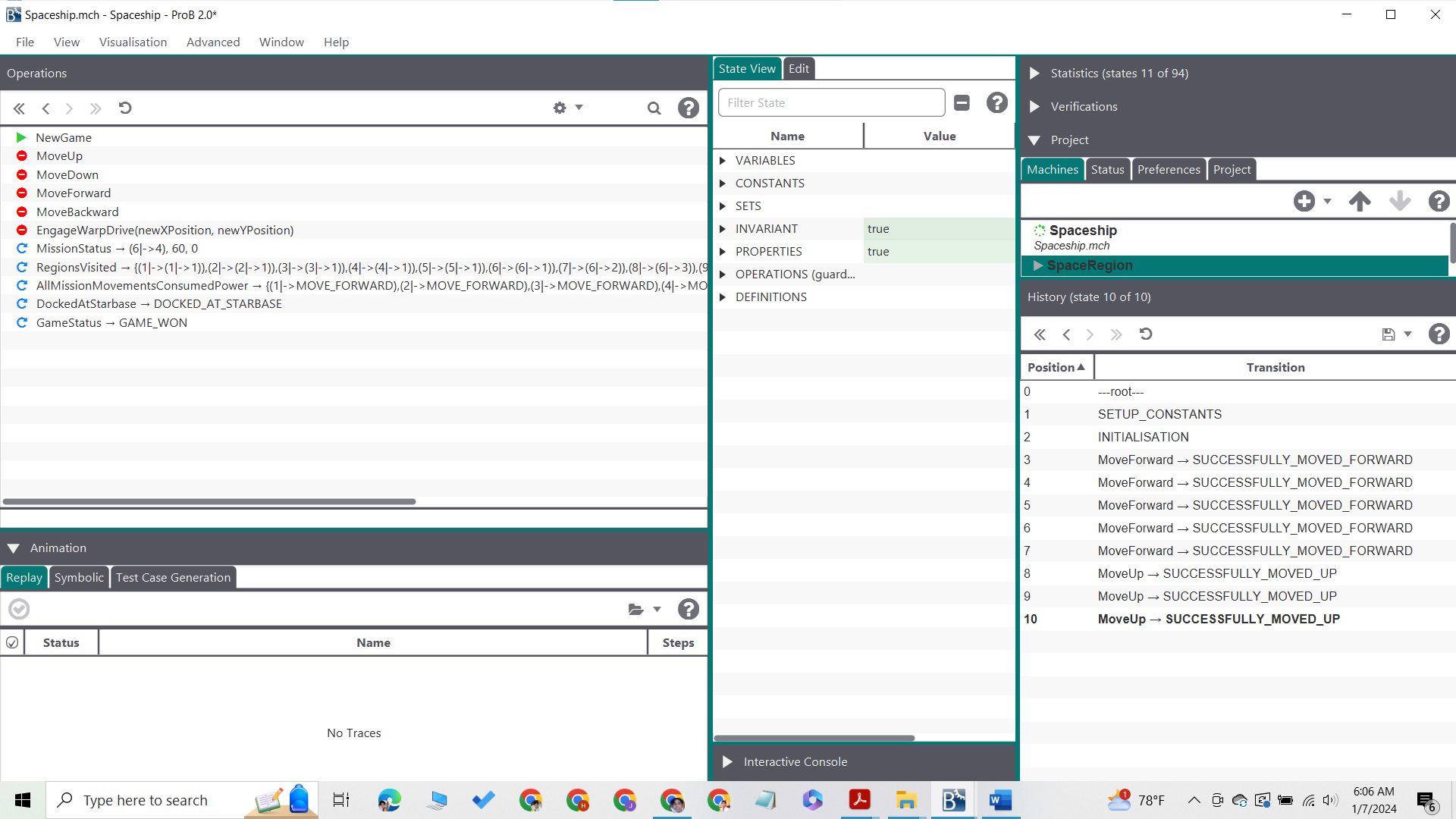
A screenshot of a video game

Description automatically generated



1. A screenshot of a computer

   Description automatically generatedGame Lost
2. Game Won



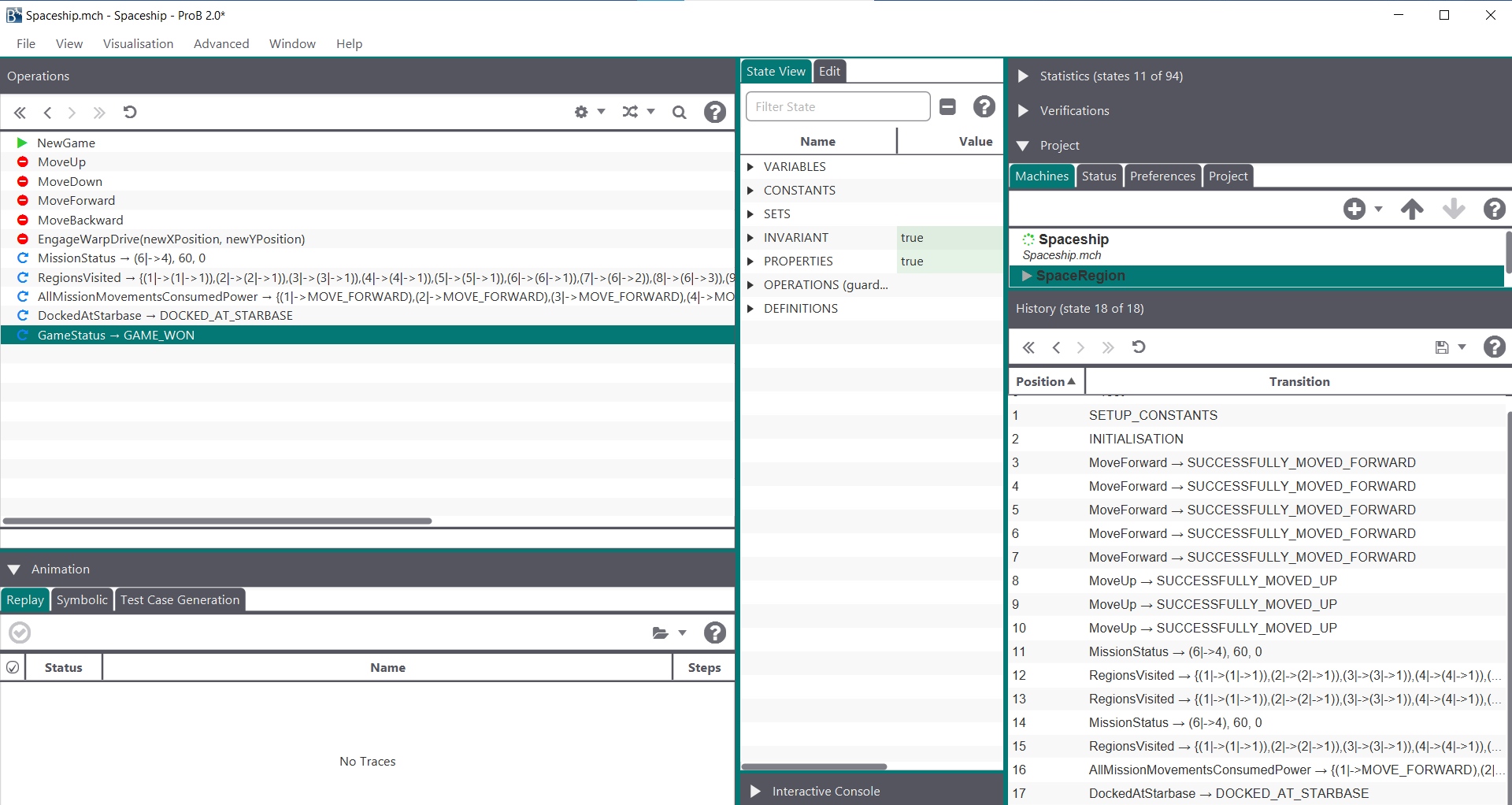
A screenshot of a video game

Description automatically generated

A screenshot of a game

Description automatically generated

1. Status of the mission, Spaceship visited regions, Spaceship docked at starbase or not



11. Screenshot of the Atelier B before Type Check

A screenshot of a computer

Description automatically generated

12.Screenshot of the Atelier B after Type Check

A screenshot of a computer

Description automatically generated