Software Requirements Specification v1.0b

Stargate: Galaxy

Peter Mitchell
Andrew Krix
Karlos White
Phil Lavender
Kane Stone

7/9/2011

Table of Contents

1   Tables and Figures

2   Revision History

Name	Date	Reason For Changes	Version
General typos and formatting corrections.	7/9/2011	Asgard was misspelled a number of times, some formatting was incorrect. Some un-ended sentences and incorrect words were also found. Sections that had red text have been modified. Also figure detail added to Interface Requirements.	1.0a
Fixed spelling of Phil's name	4/11/2011		1.0b

3 Introduction

3.1 Purpose

The purpose of this document is to provide a structured set of requirements for the project Stargate: Galaxy. Stargate: Galaxy is under development by a team of university students with a mix of Computer Science, Information Technology, and Computing and Digital Media degree backgrounds as part of a project topic. This document outlines the list of all the requirements that should be found in the final product. If a change does need to be made for the outcome of the final product, then this document should be updated. Any such revisions to the document may be viewed on the earlier revisions page.

3.2 Document Conventions

This document will outline everything with language that makes it non-ambiguous where possible. Many elements may be found in this document to be numbered. All of the numbered elements are specific requirements that will be referenced in further documentation by these numbers. Where elaboration has been deemed as required, the requirements may be written in a paragraph format. Much of the document is in a simple numbered dot point format. For referencing the document, the section number should be used followed by the number/s that define the specific requirement/s.

3.3 Intended Audience and Reading Suggestions

The development of this product has been tailored entirely within the development team, but the stakeholders of the project should also be able to understand the expected requirements from reading this document. Plain language has been used where possible to allow for this. The language used may in some areas assume a certain level of knowledge, but the concept of what is expected should be implicit where this is the case. It is suggested that the reading of this document be done in the specific section/s that individuals are requiring information on. The overall description covers the core information and overview of the requirements. The system features includes diagrams and information pertaining to particular sets of features and their required operations. The external interface requirements section explains the visual interface and storage requirements. The other non-functional requirements section covers a small number of requirements to do with the performance of the system. The art assets and racial requirements sections cover some of the background and development requirements that have not been covered in other sections too.

3.4 Project Scope

The project scope has been defined to allow for development of the working system including documentation, testing and maintenance within a 13 week period. It has therefore been structured so that there are goals within the game that the player can achieve with a small set of means for achieving those goals. It is very important that this project does not scope creep as it could very easily go into too much detail and overwhelm the development team. The scope in particular includes:

- Travel around a galaxy in space ships as seen in the TV show Stargate.
- Battle enemies with a variety of weapons and ships.
- Working with allies for trade and combat assistance.

The rest of this document defines the requirements that make up the scope. It defines the more detailed requirements for the various decided approaches that will be used for the design of the system.

4 Overall Description

4.1 Product Perspective

4.1.1 Existing Products

The most recent games based on the Stargate franchise include simple themed games such has a pinball game, trading card game and role-playing game. Action games based on Stargate that more follow the storyline and style of the franchise have yet to be successful. In fact, the majority of all known attempts have failed prior to release due to lack of funding or legal issues. The few that have made it past the production phase have had decent community support prior to release but were not able to maintain said support from the community. Building a game on such a large franchise comes with high expectations from the supporting community.

4.1.2 Proposed Product

The proposed product shall utilize the franchise's success alongside the knowledge of why other attempts failed to produce a product that shall attempt to fill the void. Stargate: Galaxy is a space based third person action adventure shooter where the player must attempt to make the galaxy a safe place. The player is provided with a ship and is given free reign within the confines of the game with the main objective of defeating all enemies in the galaxy. To do so, the player must defeat non-player controlled units including a range of boss units defined according to the Stargate franchise. The challenge of the game is to devise strategies to defeat units with artificial intelligence while managing the ship's internal power. This style of game has been proven to be successful in the current massively multiplayer online role playing game EVE.

4.2 Product Features

4.2.1 Flight

The general ability that allows the player to move within a defined space without conflict with other entities occupying the same space. Speed of movement will be determined similar to a parabola, where the ship has a period of increasing speed after being at full stop and a period of decreasing speed prior to stopping. However the peak period will extend as far as necessary.

1. To Point in Space

Move directly to and stop at the desired point.

2. To Object

Move towards an object but not collide with said object; stop at a predetermined distance from the object.

3. Orbit Object

Continuously move in a cycle around an object of choice at a predetermined distance.

4.2.2 Hyperspace

An ability that allows for travel between two different points in the game space without the need for 'manual' flight or collision detection.

1. Enter Hyperspace

Selection of a planet visible on the in-game map will provide the ability to warp to the vicinity of the selected planet. A loading screen displayed as the traditional Stargate hyperspace will be displayed during transit.

2. Exit Hyperspace

The player ship will be placed within the general vicinity of the selected planet after the loading screen ends. The actual location will be selected from predefined locations near each planet.

4.2.3   Races

There are five races present that each have their own style of combat and strategy.

1. Goa'uld

The Goa'uld are a stationary fighting race, their ships have shields that allow them to sustain moderate amounts of damage before actually taking any damage directly.

2. Asgard

The only allied race that assists the player within the game and provides the opportunity to trade in salvaged parts for upgrades.

3. Replicator

A race that is unique due to their resilience to being defeated as a whole, a single Replicator ship will be able to reform an army making the replicators a formidable opponent in the long run.

4. Wraith

The Wraith do not posses shields on their ships as each ship is in fact biological in nature and so is able to self-regenerate low amounts of damage taken during combat.

5. Ori

A sort of final race that may be encountered towards end game, their ships are only able to fire in front of themselves but similar to the Goa'uld, posses shields encompassing their ships.

4.2.4   Damage Types

There are two main damage types that cause different levels of damage to either the shield or hull if a ship. The damage type is determined by the weapon currently being used.

1. Projectile

An object directed towards another object with great force with the intent to cause damage. The damage caused is relative to the size of the object and the force at which it was projected. Projectile weapons include railguns, missile batteries and nuclear warheads.

Projectile weapons deal reduced damage to shields but increased damage to the hull of a ship, for example +25% to hull damage by -25% to shields.

2. Energy

Transfer of energy from one source to a object in a specific direction without the use of a projectile. There are two main types of energy weapons, those that use sustained beams and those that send bursts of energy frequently. Beam type energy weapons deal greater damage but take time to recharge before next use. Frequent burst style energy weapons do not have a recharge time as such but also do not deal as much damage per impact.

Energy weapons deal reduced damage to the hull but increased damage to the shields of a ship, for example +25% to shield damage by -25% to hull.

3. Replicator

Replicator weapons are the same as projectile weapons but have an additional feature. On successful impact of the hull of a ship, a status effect will be applied causing a small amount sustained damage to the ship until the effect is removed. Removal is done so according to the player's decision, removal of the replicator status effect causes another short term status effect to be applied. This second status effect is removed after a short period of time and does not require the player to make a decision.

4.2.5　Combat

The main feature is the ability for the player and other units to engage in battle scenarios between two or more ships.

1. Movement during Combat

The ability to move during combat is similar to that of normal travel however distance to an object specifies whether or not combat can actually continue.

2. Escape from Combat

Hyperspace may be used to disengage an enemy at risk of a small amount of time where the player may not be able to defend themselves. Player's may also attempt to travel out of combat but it is unlikely that an enemy would not be able to give chase.

3. Call Allies

Call allies allows the player to request the Asgard come and assist in a combat scenario. The chance of such happening and the amount of ships that come depends on reputation and a little chance.

4. Experience

On the successful defeat of an enemy, the player is rewarded with experience that allows the current player ship to level up and this increases its stats.

5. Loot

Currency used to trade for upgrades may be collected from the wreckage of ships after combat occurs in which the player has participated. The games currency is comprised of scrap parts and two types of power cores which dictate what weapons may be purchased.

4.2.6 Artificial Intelligence

An ability granted to other non player units that dictate what and when to perform certain actions such as fight or flight.

6.1 Selection of Target

Automatic target selection. Target would usually be the player but at times could be other enemies if there are other races invading that region of space.

6.2 Movement to Target

Automatic calculations of movement based on previously determined strategies according to the current situation. Including maintaining combat range when the strategy employed does not determine otherwise.

6.3 Initiate Target

Automatic initiation of selected target when in range.

6.4 Retreat

When a predefined event or series of events occurs the enemy ship will attempt to escape from combat.

4.2.7 Trade

The source of being able to upgrade weapons in order to have the ability to use all types of weapons on a single ship. The Asgard offer the ability to trade in the currency collected in game for automatically equipped weapon upgrades that are dictated by the currency used in the procurement.

4.2.8 Supergate

A single unit that may be activated to travel between its complementary unit. In doing so the player is effectively transported to another section of the game space entirely where the Ori race may be found. Travel back to the original game space can only be achieved though using the unit again.

4.2.9 Planet Discovery

Exploration is a main feature of the game, the details of each planet is not known to the player prior to entering the sector in which the planet exists.

1. Planet Status

Every discovered planet has a current standing which dictates it's allegiance to the player. The colours are red, yellow, green and blue which represent enemy, neutral, friendly and home world respectively. Additional information such as current race controlling the planet is also displayed; all this only appears on the in game map.

2. Planet Capture

Any planet apart from home world may change status according to the previous dominate force occupying the immediate area. A player will "capture" a planet after defeating all surrounding enemies and being the dominate force in the area for a short amount of time. A captured planet has a status of neutral and will alert the player when being captured by a NPC. When a NPC captures a planet from another NPC no alert is given.

4.2.10 Power Management

An ability to manage the power consumption of weapons, shields and engines by increasing or decreasing the needs of each. The total output amount is dictated by the amount of available power to the ship. Ability may be used both in and out of combat.

1. Overall Power

The total power that is available.

2. Weapons Power

Current power level being supplied to the weapons; enables or disables the use of weapons based on the power level.

3. Shields Power

Current power level being supplied to the shields; ability to regulate strength of shields.

10.3 Engine Power

Current power level being supplied to the engines; ability to regulate the maximum travel speed.

4.2.11 Object Models
Refer to section 8.2 for the necessary required models.

4.3   User Cases and Characteristics
The following list identifies the cases that have been identified for this product. This identified cases that are listed are described in section 4 with the System Features. The associated diagrams for each of the cases can be found in Appendix A.

1. Main Menu
2. Ingame Menu
3. Save
4. Load
5. Fly Ship
6. Enter Hyperspace (Player)
7. Exit Hyperspace (Player)
8. Enter Super Gate (Player)
9. Exit Super Gate (Player)
10. Ship At Earth
11. Trade
12. Capture Planet
13. Select Target Ship (Player)
14. Manage Power
15. Fire Weapon
16. Call Allies
17. Loot Ship
18. Projectile Hit
19. Gain Experience
20. Projectile Timed Out
21. Ship at Minimum Hull
22. Replicator Special Attack
23. Enter Hyperspace (AI)
24. Exit Hyperspace (AI)
25. AI Change Target
26. AI Choose Target
27. Engage Targeted Ship
28. Update AI Planet Control

4.4   Operating Environment
Microsoft Windows XP and above versions.

4.5 Design and Implementation Constraints

4.5.1 Data Constraints
Saved content shall be stored in a defined by XML which shall act as the database for the product.

4.5.2 Hardware Constraints
None.

4.5.3 Software Constraints
4.5.3.1 High-Level Languages

- Data shall be defined in XML
- Primary programming language will be C#

4.5.4 Industry Standards
The product shall conform to the necessary IEEE standards that relate to software and in particular to games where possible.

4.5.5 Legal and Regulatory Constraints
Use of images and audio files shall conform to intellectual property rights where necessary and rights of use will be acquired prior to implementation.

4.6 User Documentation

There are two pieces of user documentation that will be required to accompany the deliverable application. The first is a user installation guide; its purpose is to instruct the user of the steps they must take to be able to reach executing the application. The other is a user operations manual that will instruct the user should they need it in the various aspects of the game.

4.6.1 User Installation Guide

The User Installation Guide should fulfil the following requirements:

1. Brief in length and content only providing needed information.
2. Walks the user through any necessary steps that would normally be needed to install or configure the application for operation.
3. Indicates the title, credits, and disclaimer for the game.
4. Provides any other necessary information that would not otherwise be located in the user operations manual. This may include descriptions of how to overcome certain technical issues if they have been found to arise for certain systems.

4.6.2 User Operations Manual

The User Operations Manual should fulfil the following requirements:

1. Should be well presented and informative.
2. Provide an introduction to the game and the setting.
3. Explain the flow of the game and the controls that may be used.
4. Outline the ships and weapons the player may be able to use.
5. Provide some hints for new players.
6. Should be understandable for someone who has never seen the game before to give them enough information to be able to play.
7. Provide credits and disclaimers for the game.

4.7 Assumptions and Dependencies

There are currently no assumptions and dependencies to list here.

5 System Features

5.1 Case Main Menu

Description: The main set of options a user selects from when first starting the program. The player has options to start a new game or continue from a previous save. Starting a new game will delete previous save data if there is any. They may also choose to exit the game from the main menu.

Triggers: Player wishes to continue or start a new game.

5.2 Case Ingame Menu

Description: The player is able to access this menu from within the game, it allows the user to pause the current game and exit the program or return to the main menu in each case any progress will be lost.

Triggers: Player wishes to pause current game or exit.

5.3 Case Save

Description: This use case begins when the user wishes to save the game. When the user wishes to save the game they will select to save from the main menu, if the user is not at Earth when they elect to save then they will receive a message telling them that they can only save while at Earth. If the user it at Earth they will receive a confirmation window telling them that they will be overwriting the current save data, if the user continues with the save then they will overwrite the current save data with new save data.

Trigger: The player elects to save.

5.4 Case Load

Description: This use case begins when the player wishes to load the data that is currently held by the game. If the user is on the main menu then it will load without asking the user for confirmation, if the user is currently in game then it will prompt the user to confirm because they will lose any unsaved progress.

Trigger: The player elects to load.

5.5 Case Fly Ship

Description: The user will need to be able to control their ship; they may target an object and use that to begin the case. If there is not ship selected the ship will halt and any orbit or travel to target options should be hidden. The selected target may have interchanging of the use of travelling directly at the target or orbiting around it at a range. The selection of another target will result in the case restarting to handle the newly selected ship to fly to.

Triggers: Player wishes to fly their ship. They must have control of their ship at the time.

5.6 Case Enter Hyperspace (Player)

Description: The player's means of entering hyperspace require the selection of a planet and the confirmation that the planet or system selected is the desired destination. The player would then be subjected to automatic control by the system forcing their entry into hyperspace. At any point while the map for selecting a planet is displayed the operation can be cancelled by clicking a close button. After the selection of a destination, the player's ship would become invulnerable, their map would be closed, and the effect of opening the hyperspace window with transit would be animated. After the case has finished the loading of the next scene should be imitated and the exit hyperspace player case initiates.

Triggers: The player while in flight wishing to enter hyperspace to move to a different region.

5.7 Case Exit Hyperspace (Player)

Description: The case begins when the player has completed loading the information relevant to a region that they will be entering. The planets are first populated into the scene, followed by the enemies. Then a hyperspace window effect would be created and spawning and exiting animation would occur for the ship entering the system. After the completion of this case, the player should be able to fly freely again.

Triggers: The player finishes loading a system that they are travelling to in hyperspace.

5.8   Case Enter Super Gate (Player)

Description: The case begins when the player wants to travel by Supergate to another location. They would initiate the case by being near the supergate and selecting and confirming that they want to travel by Supergate. After confirming that the player does want to enter the Supergate, the effects for the gate should be displayed. Then the transition into the gate and followed by the loading of the next level.

Triggers: The player wants to be travel by Supergate and is near a Supergate.

5.9   Case Exit Super Gate (Player)

Description: The case begins specifically for when the player is exiting a loading sequence when they have initiated it by travelling via the Supergate instead of by hyperspace. The process will be exactly the same as the case titled "Case Exit Hyperspace (Player)". The difference is that the player's entry into the system is through the matching Supergate. Thus, the scene is populated with planets, and ships. Then the effects and animation of the ship entering the system occur. After the case is over the player is able to fly around as they wish again.

Triggers: The player finishes loading a system that they are travelling to via Supergate.

5.10 Case Ship At Earth

Description: In this case the player begins a loaded state at Earth or will have travelled there. Once they enter the Earth system they are presented with a menu that allows them to select ships that they can change to. If they have more than one ship in their collection they will be allowed to select another ship and confirm they wish to change to that. After confirming they will then have the newly selected ship as the ship they will use when they fly away from Earth. From the Earth system the player may also save and load their game state with the appropriate controls. After a player is satisfied with what they want to do, they may select the star map and choose a destination to travel to.

Triggers: Player enters the Earth system; either from another system coming to Earth or from loading a saved game.

5.11 Case Trade

Description: This case begins when the player enters a system that has allies that can trade with the player. The system determines the items that may be displayed at that time and displays them to the user. They may select items from the list and upon confirmation they will be given to the player. Failing to confirm will result in returning to the trade selection window. When an item or ship is purchased the resources are updated and after the giving of the item both indicate what the system has done with the item or ship to the player. The items are applied to the current ship and the ships are added to the player's collection back at Earth. To exit the case the player may select a cancel or close option that will show the star map to allow for hyperspace travel.

Triggers: The player enters an area that supports trade.

5.12 Case Capture Planet

Description: This use case begins when the player wishes to capture a planet that is not under their or an ally's control. The player right clicks on the planet and selects capture, if the area is clear of enemy ships then the player will receive a conformation window, if they confirm to capture the planet then they will launch a beacon which will capture the planet after a small amount of time. If the ship is not clear of enemy ships then they will receive a message to clear the area of enemy ships.

Trigger: The player elects to capture a planet.

5.13 Case Select Target Ship (Player)

Description: The use case begins when a player wants to target an enemy ship. They may select a target either from the list of targets that exists on the interface panels or click within the 3D space directly on a ship. They may also select a planet as a target. The click of the user will be checked and nothing done if it is an invalid target, otherwise the target is changed to the new target.

Triggers: The player wants to select a ship or planet.

5.14 Case Manage Power

Description: The use case begins when a ship wants to alter their ships power usage. Any of the power supply to the engines, the shields, and the weapons may be adjusted. Decreasing of any individual setting increases the capacity that can be invested in other components. Attempting to increase the power of a component will potentially not be allowed if there is not enough power available, but if successful the relevant attribute shall be increased. Altering the engines power will modify how fast the ship can travel. Altering the shield power will modify the damage mitigation of the shields. Altering the weapon power will allow for more weapons to be brought online.

Triggers: A ship is managing its power.

5.15 Case Fire Weapon

Description: The use case begins when a weapon is ready to fire. The weapon will check that the ship has a target and abort firing if there is not target. The weapon will then proceed to check that the weapon has not fired too recently and abort if it has too. A successful check will result in a calculation of the path for the projectile to follow and the properties of the projectile based on the type of weapon prior to creating and launching it.

Triggers: A weapon is ready to fire.

5.16 Case Call Allies

Description: The player at times may wish to call for assistance in combat. This case begins when the player wishes to call for that assistance from their allies with the call allies button. The pressing of the button too frequently will not help to call allies to the player. The time since the last pressing of the button shall be checked. If it is found that they have requested too recently, the AI will decline to come. Otherwise, the chance of allies coming to the aid of the player shall be calculated. If allies are to go, then a number of allied ships will be determined. A value of 0 ships will be interpreted as the allies declining. Ships that have been set by allies will be brought to the system within a period of time.

Triggers: The player presses the Call Allies button.

5.17 Case Loot Ship

Description: The loot option may be used on wrecked ships that have not already been looted. The case begins when a player selects a wrecked ship. If the player is within a reasonable range of the targeted destroyed ship, then a check will occur to determine if the ship has been looted. If it has been looted the player will be altered. If the ship hasn't been looted a loot button will be available that may be clicked and the resources automatically added to the players own stocks. The ship that has then been looted will be flagged as such and the loot button hidden.

Triggers: The player selects a wrecked ship.

5.18 Case Projectile Hit

Description: This case occurs when a projectile intersects with an object that can be damaged. The amount of damage is determined based on whether the shield or hull is being hit. The shield is decreased if it is enabled and then disabled with possible extra damage being applied to the hull. If there is no shield the damage is directly applied to the hull. The ship reaching minimum health case may be required should the values be too low.

Triggers: A projectile collides with a destructible object.

5.19 Case Gain Experience

Description: This case begins when a ship is destroyed. Increase the player's ship by an amount based on the type of target destroyed. Test if the experience gained gives enough to rise to a new level. If it does: increase the level and stats of the ship.

Triggers: A ship is destroyed that allows for experience gain.

5.20 Case Projectile Timed Out

Description: This case is called from time to time to check if the projectile has been flying for too long and therefore to make it disappear if it has. This will only be triggered if a projectile has not already collided with an object.

Triggers: Periodically to check if a time out has occurred.

5.21 Case Ship at Minimum Hull

Description: This case begins when any ship reaches the minimum hull armour allowed; typically none remaining. If the ship is a player, they are set to their minimum hull strength and immediately forced to enter hyperspace to return to Earth. If the ship is an AI controlled vessel, the AI is disabled and the destruction of the ship is initiated. This includes the creation of explosion effects, changing the ship for a wrecked version, and populating the ship with resources that may be looted. The wrecked ship will remain until the player leaves that system. If the player took part in damaging the ship, then they will gain experience for their ship.

Triggers: A ship reaches minimum health.

5.22 Case Replicator Special Attack

Description: This case begins when a replicator ship uses its special attack to attack another ship. If the ship has active shields then the ship is not infested, if the ship has no shields then there will be a chance of replicators infecting the ship. If the ship is the player's ship then the ship begins suffering from an affliction, this affliction will continue to affect the ship until the player cleanses the ship using an option on the user interface, when the option is selected the players ship will become inert for a period of time after which the ship will return to operational capacity with the replicator infestation removed. If the ship is an AI ship then if it is infected it will fire randomly at other ships in the area, there is a chance that the ship will be cleansed and return to completing its normal actions.

Trigger: A replicator ship fires upon another ship using its special attack.

5.23 Case Enter Hyperspace (AI)

Description: This case is initiated when an AI is allowed to retreat. The conditions for retreat will be checked and if they are all valid, then the AI will become invincible and exit the system through a hyperspace window. The player is altered of the retreat of an enemy and the objects are removed from the scene. After this case is completed, the system will have one less ship in it.

Triggers: The AI wishes to retreat from the battle.

5.24 Case Exit Hyperspace (AI)

Description: This case begins when an AI controlled ship enters an area of space where a player is already there. A hyperspace will open and the ship will be spawned and animated to pass through the effect. After this case has completed the AI will be allowed to engage or search for hostiles.

Triggers: An AI is set to enter a region of space that a player is already in.

5.25 Case AI Change Target

Description: This case begins periodically when an AI will check whether they need to change to a different target to attack. If there is no current target, one should be chosen if one exists. If there is an ally that requires some form of assistance, there should be a chance of switching to attack the attacker of an ally in need. Otherwise the AI will have a small chance of switching target to something different. After the case, the AI should have a target unless there are no targets remaining.

Triggers: Periodic, when the AI wants to check if they can or should change target.

5.26 Case AI Choose Target

Description: This case begins when the selection of a target is required. If there are no enemies, then none would be selected. If there is just one enemy, then that enemy would be targeted. If more than half of the enemies are targeting this AI, then target one of the enemies that are targeting this if an 80% chance is also true. If there are enemies targeting other ships or there are allies targeting enemy ships, and a 60% chance is true, then select from the following:

- If there are no enemies targeting allies, then target a ship that is being targeting by an ally.
- If there are no allies targeting enemies, then target a ship that is targeting an ally.
- If a random 50% chance is true target something that is being targeted by an ally.
- Otherwise target something that is targeting an Ally.

If the above was not used either, then just select a random enemy to target from the collection of enemies. After this case has been completed, the enemy should engage the target that has been selected.

Triggers: The AI wants to select a target to attack.

5.27 Case Engage Targeted Ship

Description: This case begins when a new target has been located and needs to be approached to engage. The ship's relative distance from the other ship and its rotation are taken and used to determine if the ship is ready to fire. If rotation and/or repositioning of the ship to be able to fire weapons, the ship will move to fulfil this goal. Once range has been reached, firing will be initiated.

Triggers: A new target has been selected or the selected target has moved out of range.

5.28 Case Update AI Planet Control

Description: This case triggers periodically by the system to update the state of the galaxy. A list is created to hold the list of attacks that will occur during the next simulation. For each of the races decide on a number of systems to attack, if any. This would be dependent on the number of planets controlled by that race; the desired number would be a mean number. The planets are added to an attack list. Once the attack list is compiled, each of the battles is processed. If the player is in that zone, the battle is postponed, otherwise the battle is decided by random dice chance and updates are made for the control of planets. After all of the battles have been simulated if there was a battle at the location of the player, a number of enemies to spawn is determined and they are entered into the system as an attack force. The player would be altered if this is the case. The end of the case will result in the updated control status of all planets and systems.

Triggers: Periodic system event triggering an update required.

6 External Interface Requirements

6.1 User Interfaces

1. The main menu is the first screen seen by a user after loading the program. As described in the Main Menu use case and use case diagram the user has 3 options. Start new game creates a new game world with all variables set at standard. If the user opts to create a new game with an existing save file they will be warned that starting a new game will over write any existing progress. If available a user can also select to load a previous save, this will load the game world with all game objects as they were when the user last saved. The third option is to exit the program, selecting this closes the main program and reverts the user to their main desktop.

2. The in game menu allows the user to pause the game world, exit to main menu and exit program. When bringing up the in game menu all objects in the game world stop allowing the user to interact with the menu or leave the game unattended. Exit to main menu closes the game world without saving any information and load the main menu, the user will be warned that all progress will be lost before this happens. Exit game closes the program giving the same warning that all progress will be lost.

3. The main GUI in game (figure 1) resembles a Stargate style ship bridge. It contains access to all user ship controls,

   a. Local map/radar Panel 4, Gives the user a view of all ships around the users ship.

   b. Power management: Panel 1 which includes the ability to control power allocation to,

      i. Total power
      ii. Shields
      iii. Weapons
      iv. Engines

Figure 1 : Player ship GUI

   c. Hull level: Panel 2 which shows damage to the users ship as well as shield levels and other ship status details.
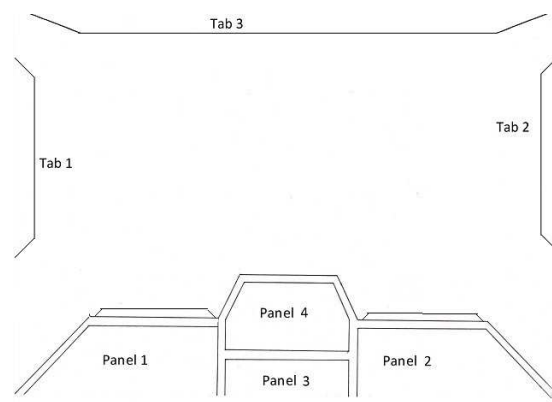
d. Weapons: tab 1 located on the left of the screen which expands when the user selects the tab. The weapons tab allows the user to select weapons to attack with
   i. Rockets
   ii. Energy
   iii. Projectile
e. Ship purge, in the event of a replicator attack a button on panel 2 will become active allowing the user to purge the ship of replicators.
f. Map, tab 3 when selected opens up a full screen map allowing the user to view their current location and where available enter hyperspace to a new location.
g. Game objects, tab 2 shows a list of all available objects around the users ship, giving details on distance away from user name of ship and faction.
h. Target, panel 3 shows details on the users target.

4. The Map screen (Figure 2) displays meta data about each zone giving info on the controlling faction for that area (when the information is available).
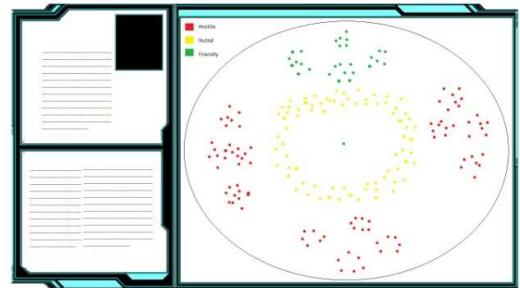
Figure 2 : In Game Map

5. The interface provided at Earth will require:

   a. Selection and change of the player's current ship. Including at least an image and name prior to the selection change.
   b. Interface for saving and loading the game.
   c. Means of exiting the Earth specific interface.

6. The interface provided at the Asgard trade world will require:
   a. Selection of and purchasing of ships. Only those ships that the player does not yet have should appear in the interface.
   b. Selection of and purchasing of weapons. Only weapons that may be equipped to the current ship based on maximum quantities or restrictions should be able to be purchased.
   c. Prices of, names of, and images of items/ships should be visible.

d. Confirmation of purchases should allow the player to withdraw their interest in purchasing.

6.2   Hardware Interfaces

There are no special hardware interfaces required for interfacing with this product. Only the mouse and keyboard may be used to provide interaction. Keys may be bound to particular actions on the keyboard, and the mouse may be used for most of the primary interaction.

6.3   Software Interfaces

The software will not require communication with outside programs other than those that the Unity engine would otherwise communicate with. As these interactions are handled by the engine itself, they are not a requirement concern. The only external interfacing required is the loading of assets and required database information. This includes the elements that will be stored via saving mechanisms.

1. The software will require an external data folder for the applications assets as will be generated when a windows executable is built. This is as defined on http://unity3d.com/support/documentation/Manual/Publishing%20Builds.html.
2. For storing information that shall be loaded, XML will be used as the database storage format.
3. The XML shall be loaded with the use of scripts within Unity and used to populate the worlds.
4. The saving mechanism will require two states.
    a. A state that defines the initial configuration of the entire galaxy; including how the player will initially view planets.
    b. A state that defines the player's single saved state of the galaxy; including a state for the player's most recent view of each planet.
    c. There will only be a single save state for the whole game.
    d. Creating a new game overwrites the existing save game state using the initial configuration state.
5. Information that must be stored in these states will need to include:
    a. The player's current location.
    b. The ships the player owns and their attributes.
    c. The players other attributes including resources.
    d. The state of each planet/system
        i. Player's last view
            1. View type. (This is the attribute that would define the colour that is seen on that particular planet/system).
            2. The enemy type that were seen there (only the dominant race).
            3. The size of the enemy forces.

   ii. Ownership of planet.

   iii. Approximate numbers of enemies that reside at the location.

     1. Multiple races and army sizes should be allowed to reside at each location.

     2. Details of any bosses that reside at a location should be referenced.

6. The state information may only be saved and loaded while either at Earth; and may also be loaded from the main menu.

7. Other information that may be stored in XML in part or full will need to include:

  a. For each planet system:

   i. The planet locations.

   ii. Other specific locations for entities or effects. Like pre-defined hyperspace exit windows.

  b. For each ship:

   i. The properties of that ship as they apply to the player and to the AI.

   ii. The weapons that the ship would normally have.

   iii. The modifications that would occur for levelling up.

  c. For each weapon:

   i. The details about the weapon.

   ii. The damage and weapon type.

6.4 Communications Interfaces

The program's communication with outside sources is very limited; the game is designed as a single player locally hosted game. All external communications, if any, should be handled by the game engine Unity and is therefore outside of our scope.

7 Other Nonfunctional Requirements

7.1 Performance Requirements

The game is to be a real time system, and so it needs to perform under pressure to allow players a stimulating experience without issues causing long waits. The following requirements will ensure the playability of the game.

1. Average minimum frames per second should be 25 to maintain a playable game.
2. The player should not have noticeable visual lag on screen that impacts the game play.
3. The game should run on hardware from at least the last few years.

7.2 Safety Requirements

The game as a software product should not itself pose as a harmful product in any light or allow users to engage with it in such a way that would cause a safety concern. This leads to the main following requirement, under the assumption that the responsibility of the external hardware system is under requirements for safety.

1. There should not be effects and flashing lights that could cause epilepsy.

7.3 Security Requirements

The product will not pose a security threat to anyone. The threats posed come from the stealing of code and assets. Thus, the following requirements are needed.

1. The end product should not include code accessible to the player.
2. The end product should not allow access to game assets if possible.

7.4 Software Quality Attributes

Quality of the product should be handled to ensure a certain level may be seen in the final release. The following requirements have been identified that together provide quality attributes that should make the product of a high quality.

1. Beta testers should respond with 75% or higher average for their overall rating for the game
2. Testing should find that all functional requirements have been met where possible.
3. Tests should find a minimum number of issues with the final product.

8 Other Requirements

8.1 Race Requirements

This section gives an overview of the details relating to each race and how they should exist in relation to others in the galaxy.

8.1.1 Human

Race Overview

- Humans are the player's race in Galaxy. They are relative newcomers to the galactic stage, but have quickly gained knowledge of the Stargates, the ancients and of creating space ships.

Racial Traits

- The human ships are relatively well rounded; they have average shields, hull strength. Each of the human ships also have a multitude of weapons including railguns, rockets and on the Daedalus class ships Asgard Beam Weaponry.

Ships

- Humans have 2 main classes of ship. The X303 ("Prometheus") which will be the players first ship, and the Daedalus Class Ships (""Daedalus", "Odessey"). The Odessey is equipped with an Asgard Computer core which means that it has access to far greater technology than normal human ships.

Relation to Other Races

- Humans have a good relationship with the Asgard.
- In order to gain access to the technology to create ships humanity has made enemies of the Goa'uld.
- The Wraith see humanity as a food source to be culled.
- The Ori believe that humanity has been corrupted by the ancients influence and they wish to either convert or destroy humanity.
- The Ancients are gone from the galaxy; however humans can possess a gene which allows them access to ancient technology.
- The replicators only wish to replicate and they will choose to try and infect more advanced ships over the human vessels.

Other Important Details

- o Human energy weapons have the same colour as Asgard weaponry because they were based off of Asgard technology.

8.1.2 Goa'uld
Race Overview

- o The Goa'uld are a race of parasites who pretend to be gods in order to enslave the populace of a planet, they are served by large numbers of Jaffa Warriors who are loyal to the Goa'uld they serve. Goa'uld society is made up of a number of powerful Goa'uld called System Lords; these System Lords have control over large numbers of minor Goa'uld.

Racial Traits

- o The Goa'uld ships are usually encountered as part of a fleet, there ships have fairly strong shields and weapons but the hull is weaker than most other vessels.

Ships

- o The Goa'uld have one major class of ship, the Ha'tak class vessel. This ship is equipped with energy weapons and shields.

Relation to Other Races

- o Goa'uld are hostile to humans.
- o Goa'uld are neutral towards the Asgard because they fear the technology that the Asgard possess.
- o The Goa'uld have not encounter the Wraith before but the Wraith view them as a food source just as they do all other races.
- o Replicators can quickly increase their numbers aboard Goa'uld vessels because of the elements that make up the ship.
- o The Ori are hostile to the Goa'uld because the Goa'uld are posing as gods, while the Ori wish everyone to worship them.

Other Important Details

- Each of the Goa'uld System Lords would have their own more powerful vessel; these vessels may have increased firepower or shields. An example of one of these unique vessels would be the vessel of Anubis which had a unique energy weapon.

8.1.3 Wraith
Race Overview

- The Wraith are an ancient race who sleep for long periods while human populations grow, when they awake they cull the populations in order to feed until they enter back into hibernation when the population numbers fall too low.

Racial Traits

- The Wraith are unique in that they do not possess any kind of shielding technology, this is made up for by the incredible strength of the hulls of their ships. Wraith vessels are equipped with energy weapons.

Ships

- Wraith have three two main types of ship which are larger than fighters, these are the Wraith Cruiser which is a smaller vessel that acts as an escort for Hive ships, but can also act independently. The second wraith vessel is the Hive Ship, this ship is very large and much stronger than the wraith cruiser.

Relation to Other Races

- The Wraith see all other races in the galaxy as a potential food source to be culled, this makes them hostile with all other races, they may make alliances of convenience in order to take down a stronger foe but will likely betray their ally after their victory.

Other Important Details

- The wraith also have a unique ship, the super hive, which is a hive ship powered by a ZPM and as such has better sensors, weapons and hyperdrive than normal hive ships and is also much more difficult to destroy.

Race Overview

- o The Replicators are a race of robotic creatures whose primary directive is to create more of them; they complete this by consuming resources. They were originally created by the ancients.

Racial Traits

- o The replicators are very difficult to destroy as they only require a single replicator to survive in order to rebuild, they also feed on energy weapons.

Ships

- o The Replicators only have a single ship which is made up of replicator blocks, this ship has energy shields and fires replicator blocks at opposing vessels. These replicator blocks are used to infest enemy ships with replicators which take control of them from the inside.

Relation to Other Races

- o The replicators are hostile to all other races as they only wish to consume resources to replicate, the replicators do tend to attack more advanced vessels first as they contain better materials.

Race Overview

- o The Asgard are a very advanced race who enforce treaties that various planets make with the Goa'uld in order to avoid occupation.

Racial Traits

- o The Asgard ships are very advanced but small in number; they have very strong shielding and weaponry.

Ships

- o The Asgard have two main classes of ship, the Biliskirnir class ship and the O'Neill class ship. The O'Neill class ship is more advanced than the Biliskirnir but both ships are equipped with advanced shielding technology and weapons.

Relation to Other Races

- o The Asgard are friendly with humans.
- o The Asgard are neutral with the Goa'uld although they dislike what they do to the populations under their rule, the Asgard will not hesitate to defend planets protected by treaties though.
- o The Asgard will be hostile towards the Wraith because they will cull planetary populations.
- o The Asgard are hostile towards the replicators and do whatever they can to contain a replicator threat once it has emerged.
- o The Ori would be hostile to the Asgard as they were allied with the ancients.

8.1.6 Ori

Race Overview

- o The Ori are ascended beings who use human followers to gather more worshipers in order to increase their power.

Racial Traits

- o The Ori ships are very advanced and are controlled by a Prior. They have above average shields, and weaponry.

Ships

- o Ori only have one class or ship which is armed with an energy projector type weapon.

Relation to Other Races

- o The Ori will destroy all races who do not convert to follow Origin.

Race Overview

- The ancients were the race that created the Stargates, they also created powerful defensive weapons and other types of technology, however they all ascended to a higher plane of existence and do not interfere with "lower planes".

Racial Traits

- Ancient ships are some of the most powerful ever to exist, containing powerful weapons including guided rockets, they also have good shields.

Ships

- The only ancient ship encountered in the game will be Destiny, if it is included. Destiny is a ship which followed a series of ships that were setting up stargates on planets they encountered. Destiny does not travel through hyperspace but has incredibly advanced engines that allow it to travel faster than light. Destiny has adaptable energy shields and numerous energy weapons including smaller point defence weapons and a large main battery. Destiny can recharge its power by flying through stars.

Relation to Other Races

- The ancients no longer exist in the universe but their technology endures. In life the ancients were enemies of both the Ori and the Wraith. They were also the creator of the replicators.

8.2   Graphical and Audio Requirements

8.2.1   Model Requirements

Model requirements pertain to the elements that must be included with and the details of each model. The requirements that follow will ensure that models are created with a level of quality and are useable for the product.

1. Models should appear aesthetically appealing.
2. Models should be detailed enough to make it obvious what they are representing.
3. Should include any required animation, and skinning.
4. Bones for the locations of weapons on ships to indicate where weapons exist.
5. The models are not individually more than 1000 polygons or vertices unless there is a good reason.
6. A simple collision mesh should be included that can be used to check for collisions.
7. A destroyed version of the ship that is split into a couple or more parts for each ship is needed.

8.2.1.1   Models Required

The list of models that follows lists models that will be required for the product. Following the summary list is an expanded list showing an image of each of the models to reinforce what the individual models should appear to be.

1. Asgard Mothership
2. Ori Mothership
3. Goa'uld Ha'tak
4. Goa'ul Anubis's Flagship
5. Wraith Cruiser
6. Wraith Hive Ship
7. Human Prometheus class battle cruiser (X-303)
8. Human Daedalus class battle cruiser (BC-304)
9. Replicator ship
10. Super gate
11. Planet/Star

1. **Asgard Mothership**

http://www.gateworld.net/omnipedia/ships/a/asgardmothership.shtml

2. **Ori Mothership**

http://www.gateworld.net/wiki/Ori_warship

3. **Goa'uld Ha'tak**

http://www.gateworld.net/wiki/Ha%27tak

4. **Goa'ul Anubis's Flagship**

http://www.gateworld.net/wiki/Anubis%27s_mothership

5. **Wraith Cruiser**

http://www.gateworld.net/wiki/Wraith_cruiser

6. **Wraith Hive Ship**

http://www.gateworld.net/wiki/Wraith_hive_ship

7. **Human Prometheus class battle cruiser (X-303)**

http://www.gateworld.net/wiki/Prometheus

8. **Human Daedalus class battle cruiser (BC-304)**

http://www.gateworld.net/wiki/Daedalus

9. **Replicator ship**

http://www.gateworld.net/wiki/Replicator_cruiser

10. **Super gate**

http://stargate.wikia.com/wiki/Supergate

11. Planet/Star

http://www.ironstarmedia.co.uk/2009/12/free-game-assets-05-planet-sprites-and-textures/
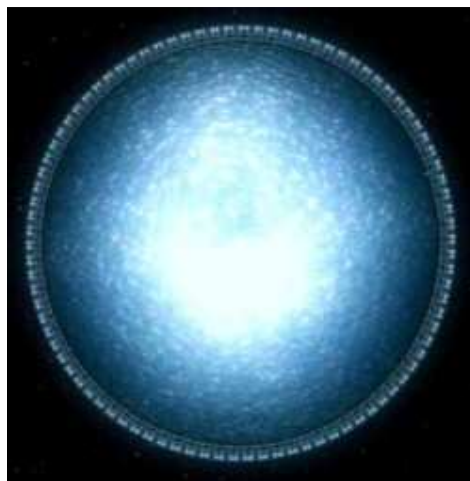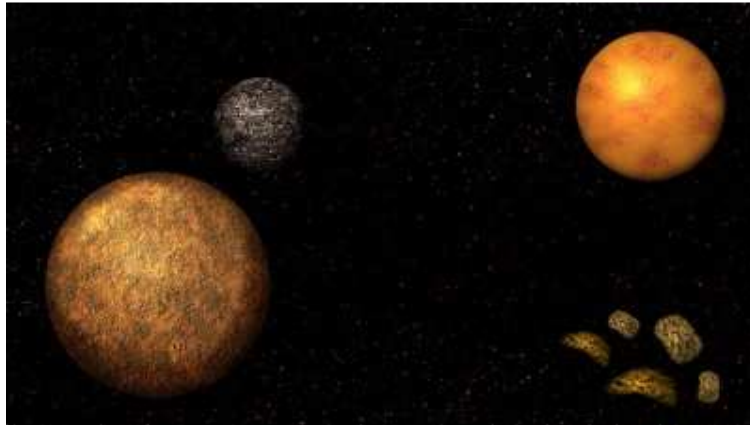
8.2.2   2D Art Requirements

For further details regarding more specific requirements of the HUD please see the interface requirements section. This section will just cover the types of elements that will be required.

1. Main Menu
    1. Buttons Images
    2. Logo or Icon for game
2. HUD elements
    1. Star map
    2. Increase/Decrease buttons
    3. Progress bars for power
    4. Button for each type of weapon
    5. Background for interface
    6. Button for Call Allies and other actions
    7. Buttons for trading
    8. Icons for items/ships that may be purchased
3. Skybox texture for basic star background

8.2.3 Effect Requirements

There are a number of main types of effects. The main one that splits into a variety of other types of colours and effects are the weapons. Effects, where possible, may be used from external sources that have been already created and used under allowable licensing or through contact with the creator/s.

The required effects and some of their required attributes include:

1. Weapons
    a. Continuous Beam
        i. Blue beam
        ii. Yellow beam
    b. Small/Large energy "ball" fire (each with a white centre)
        i. Blue
        ii. Yellow
        iii. Purple
    c. Rail Gun burst (yellow beam)
    d. Rocket
        i. Fire and smoke effect out back
        ii. Model for rocket
2. Engines
    a. Blue or yellow glow/fire
3. Hyperspace entry/exit
4. In Hyperspace/Star travel
5. Stargate

8.2.4 Optional Elements

These elements are not part of the core game, but if the game is completed, these elements may be added as hidden or rare features that the player may encounter. It is not a requirement for their inclusion. But they have been included as a requirement in the document so that they have been documented prior to their potential inclusion.

1. Destiny
2. Llama Space

8.2.5 Audio Requirements

Audio is a vital part of any game and heavily complements the visual component. To increase the depth of the game and enhance the users experience music will be added to each scene of the game. The type of music will depend on what events are taking place for example; the music playing when the player is simply navigating through space would differ from the music playing during combat. The music is there to complement and enhance what is visually happening in the game environment. The game being created is based on the TV series Stargate; the music selected must conform to the same genre and overall theme.

The game will have scenes ranging from tranquil exploration to fast paced combat each requiring different sound tracks. Scenes used will include combat, exploration of new zones, flying through friendly zones as well as hostile and neutral zones. Loading between scenes and the main game menu can also be enhanced with audio.

When adding in sound effects much more detail must be applied. Many of the sounds used in the Stargate series have very unique qualities and must be replicated thought the game. Examples include weapons fire; the sounds from ships such as the Goa'uld ships such as the Ha'tak have a very distinct sound and would be instantly noticeable. More general sounds such as explosions, button clicks and ambient sounds can be used from other sources where sounds such as ship engines, weapons fire, game voices and shields must sound as they do from the TV series.

9 Conclusion

This document has provided a set of requirements that shall be applied during the agile development. The overall game software product shall be developed as defined by this document. During the course of development minor modifications that need to be changed must be changed here in this document too and noted on the revisions table. The remaining sections display the various diagrams in appendix B that depict the use cases using mostly Activity Diagrams.

10 Appendix A: Glossary

HUD = Heads Up Display

11 Appendix B: Analysis Models

See the section on System Features for further details matching each of the activity diagrams.

11.1 Main Menu
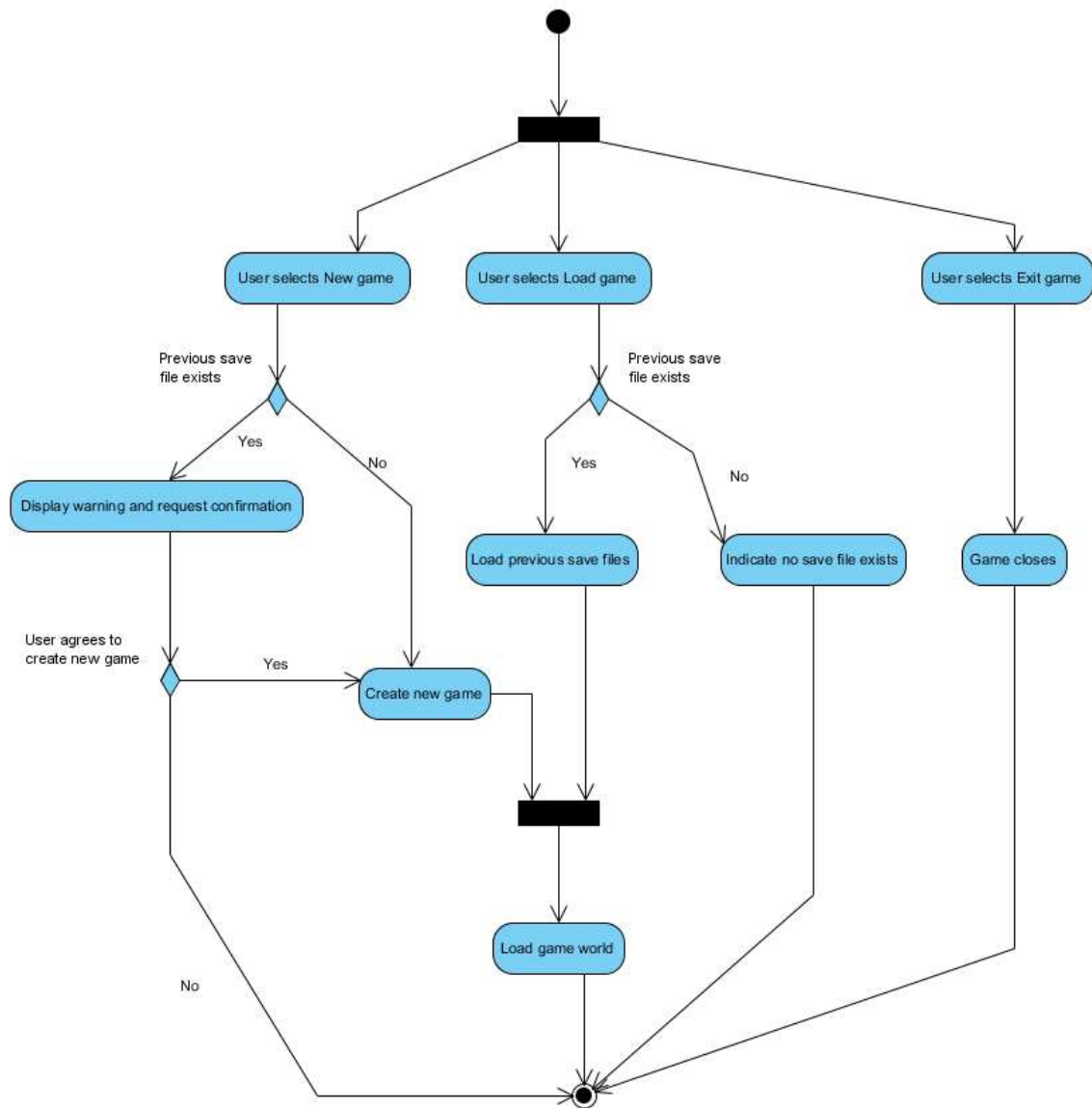
Figure 3 : Main Menu Activity Diagram

11.2 Ingame Menu

11.3 Save
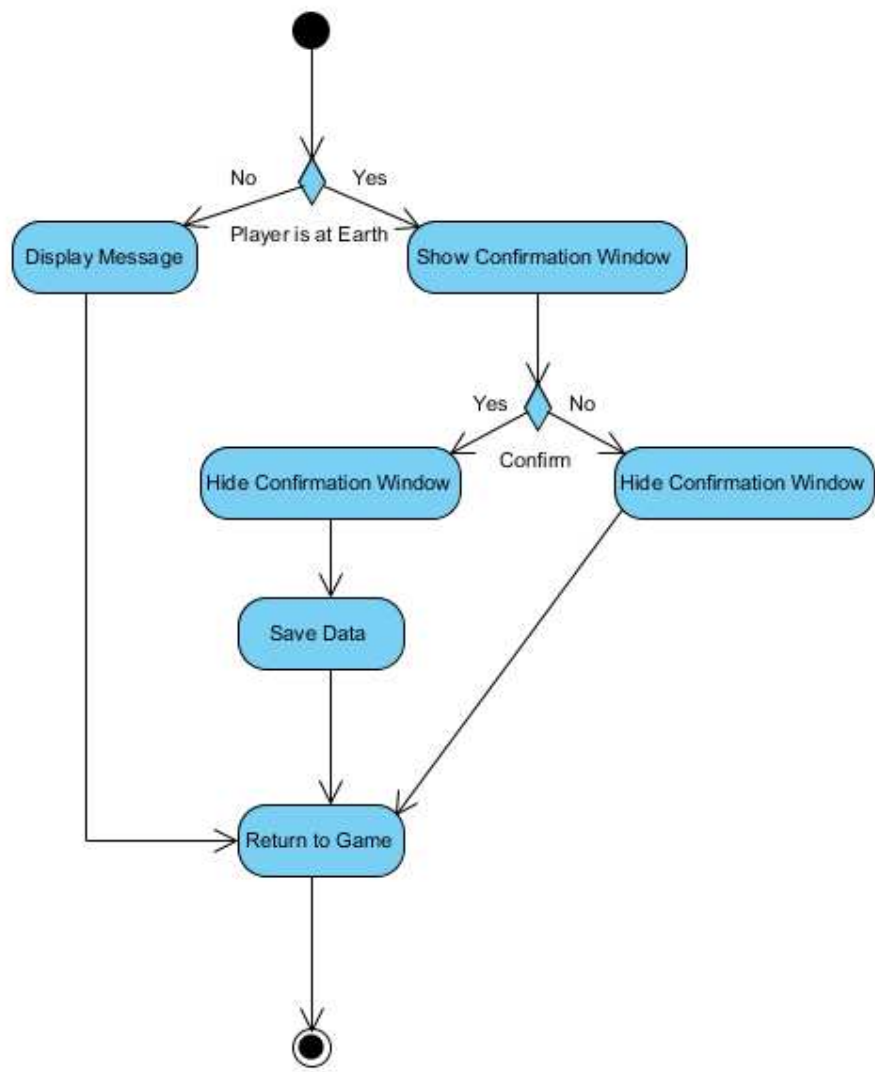
Figure 5 : Save Activity Diagram

11.4 Load
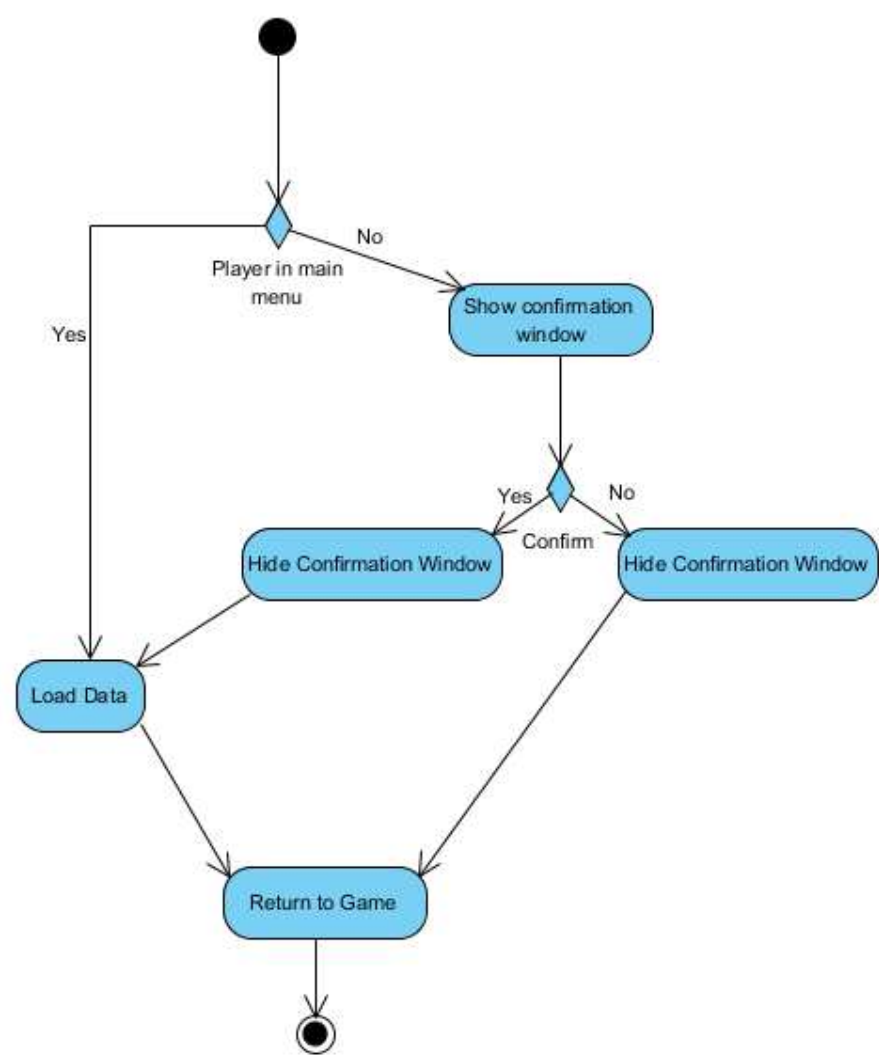
Figure 6 : Load Activity Diagram

11.5 Fly Ship
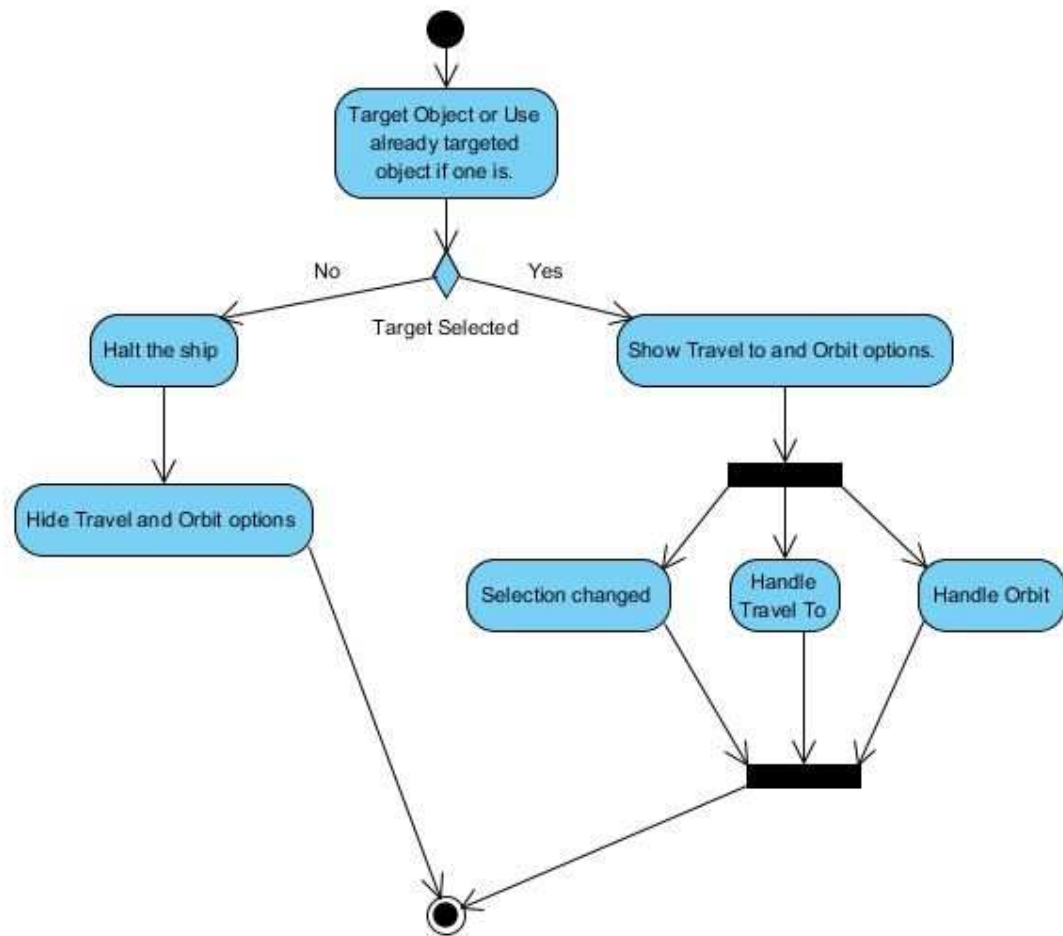
Figure 7 : Fly Ship Activity Diagram

11.6 Enter Hyperspace (Player)
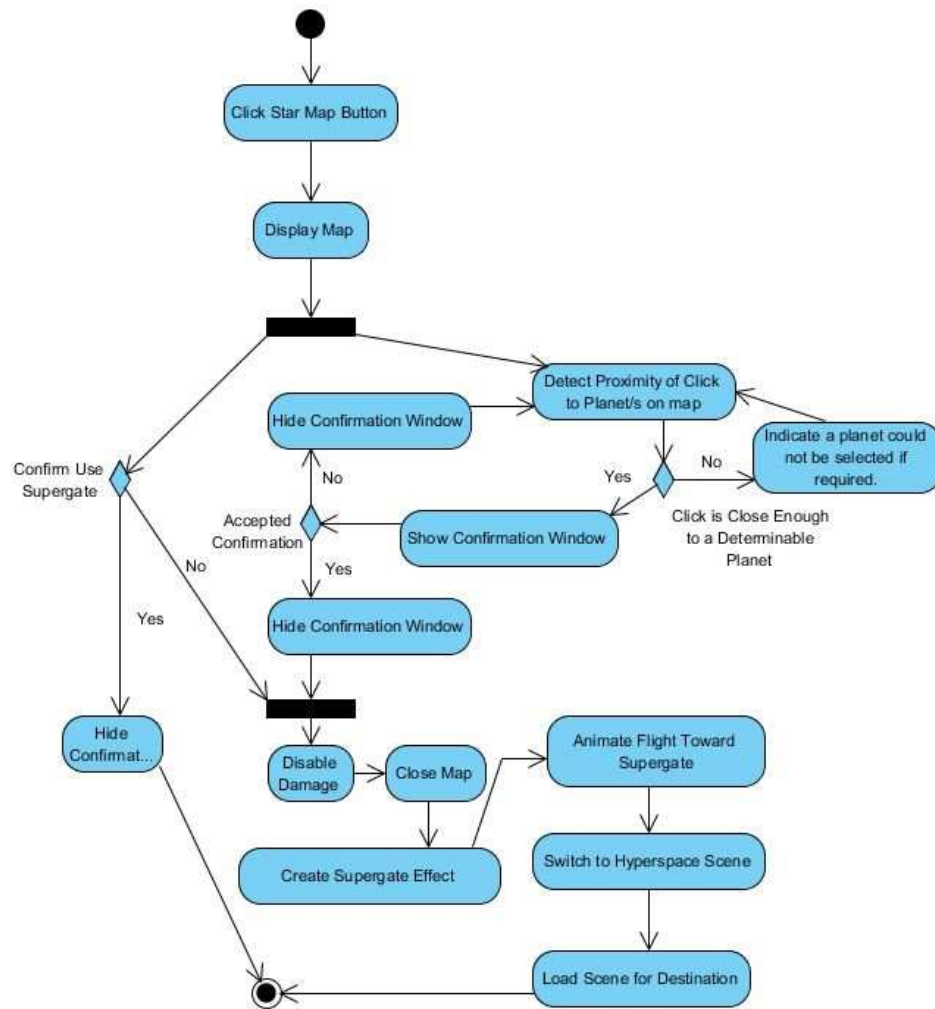
Figure 8 : Enter Hyperspace (Player) Activity Diagram

11.7 Exit Hyperspace (Player)
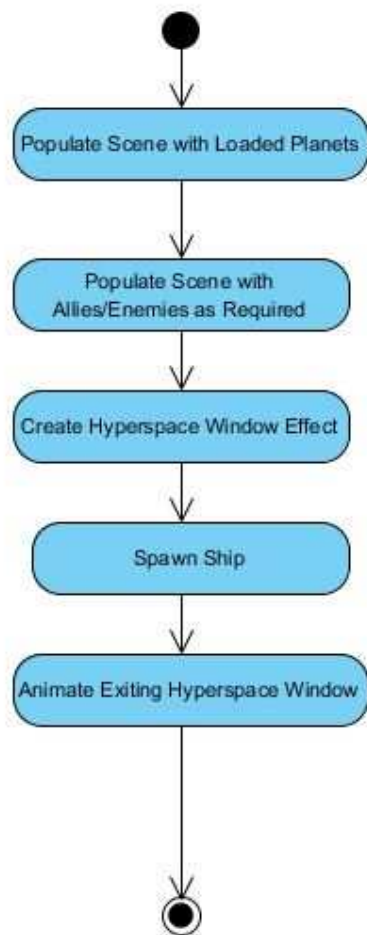
Figure 9 : Exit Hyperspace (Player) Activity Diagram

11.8 Enter Super Gate (Player)
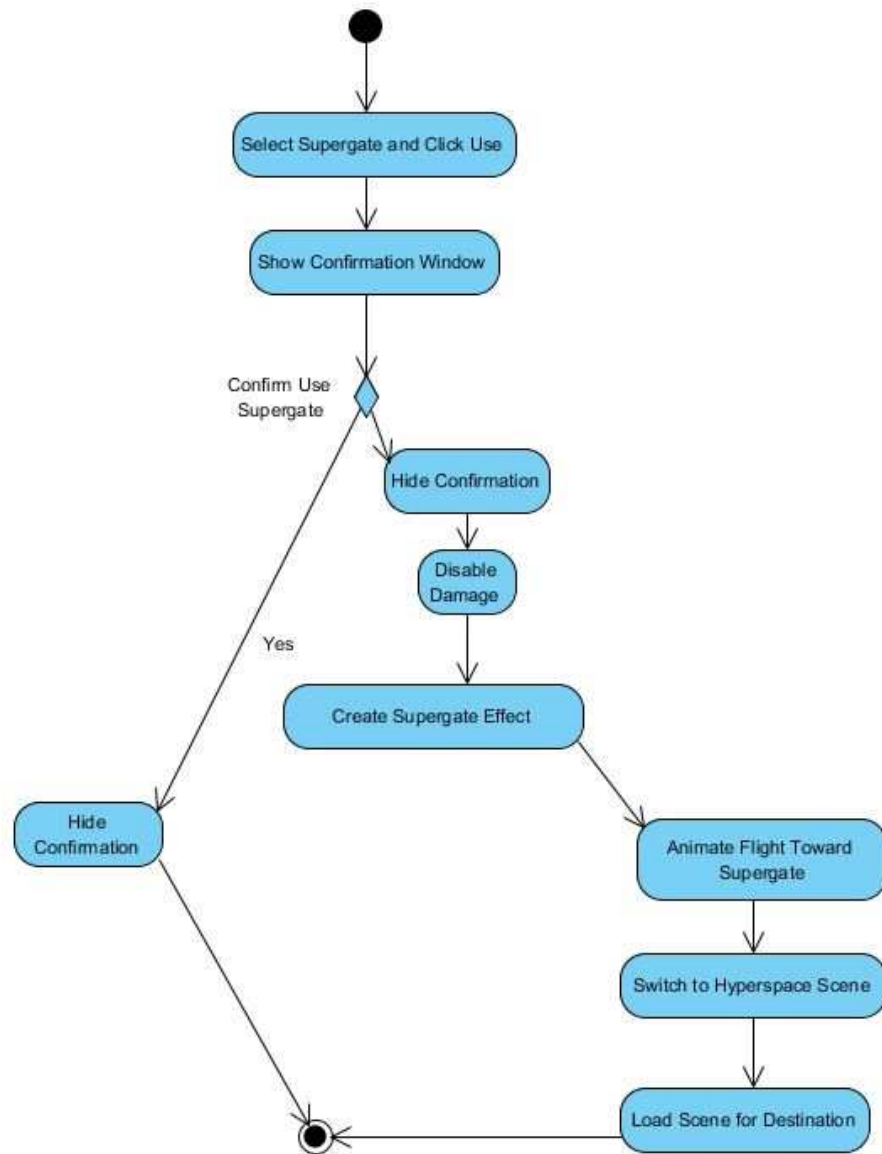
Figure 10 : Enter Super Gate Activity Diagram

11.9 Exit Super Gate (Player)
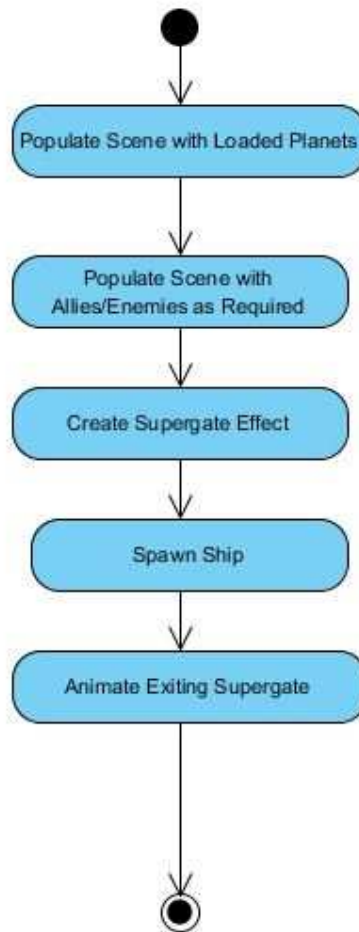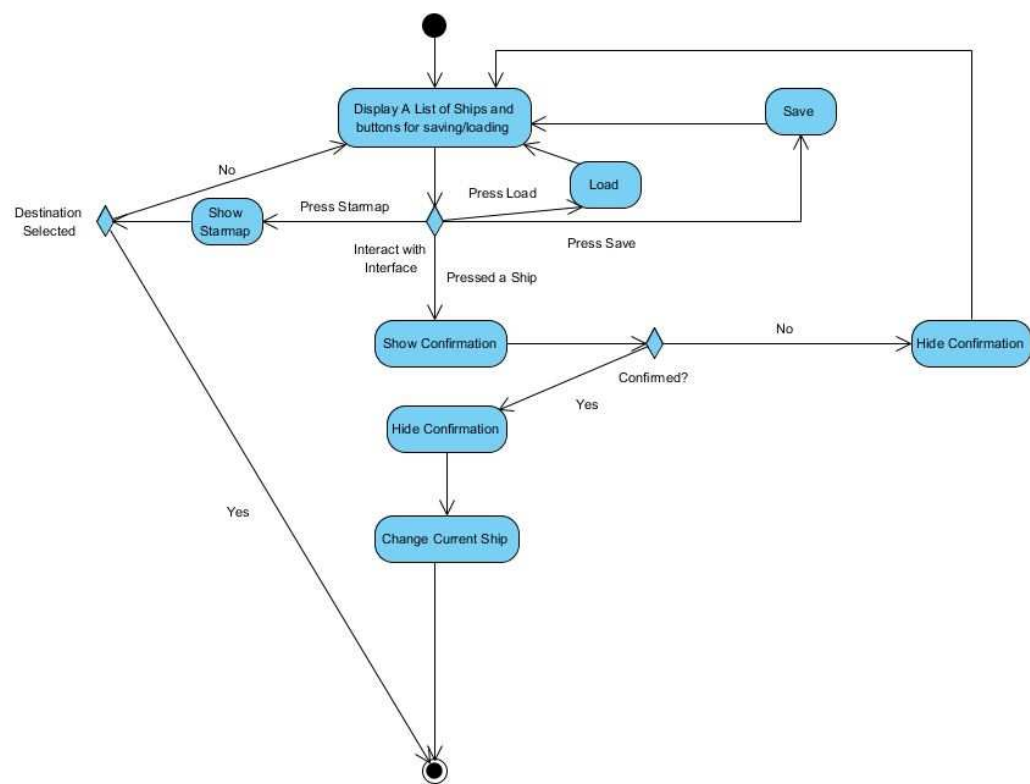
11.10 Ship At Earth

Figure 12 : Ship At Earth Activity Diagram

11.11 Trade
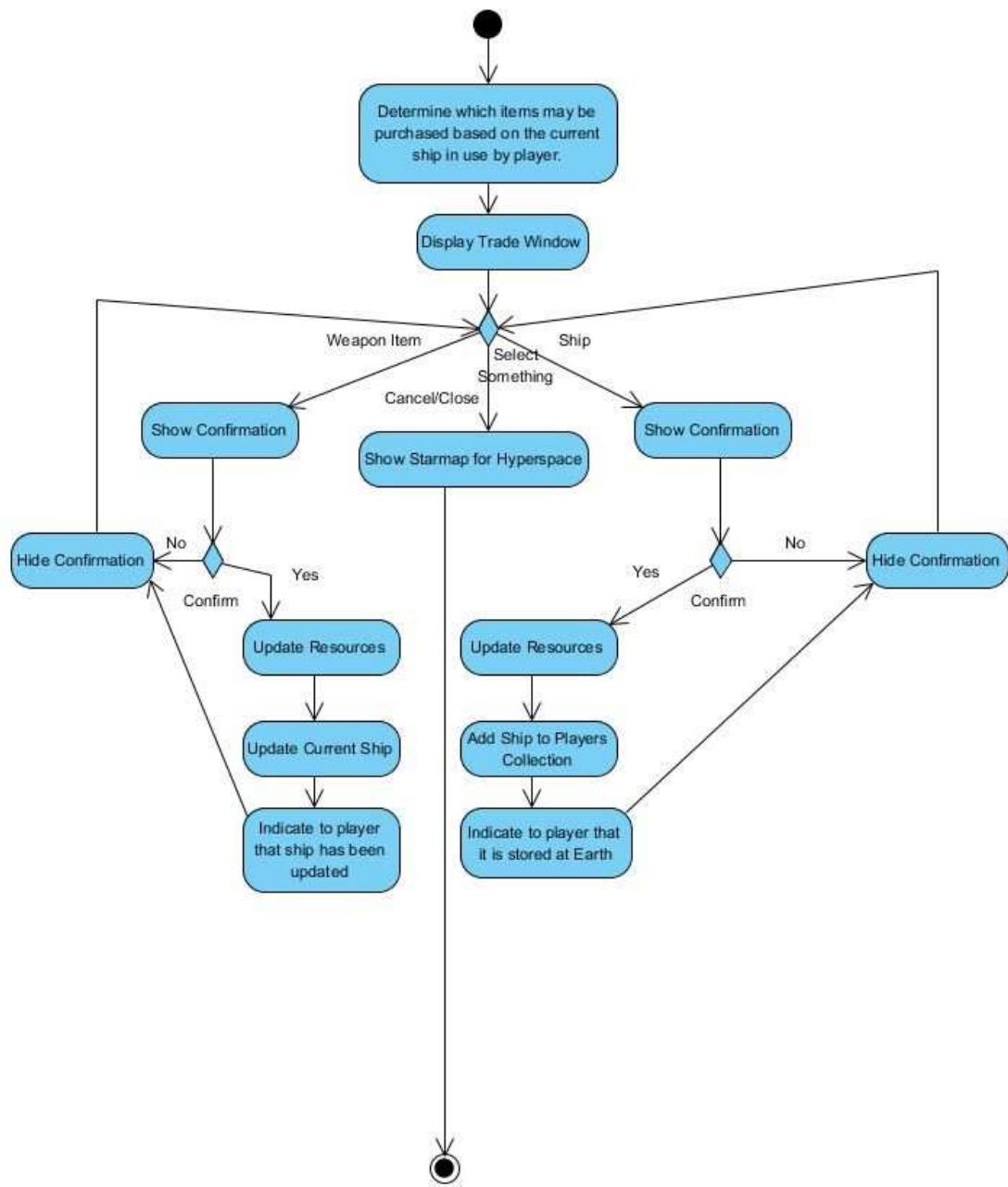
Figure 13 : Trade Activity Diagram

11.12 Capture Planet
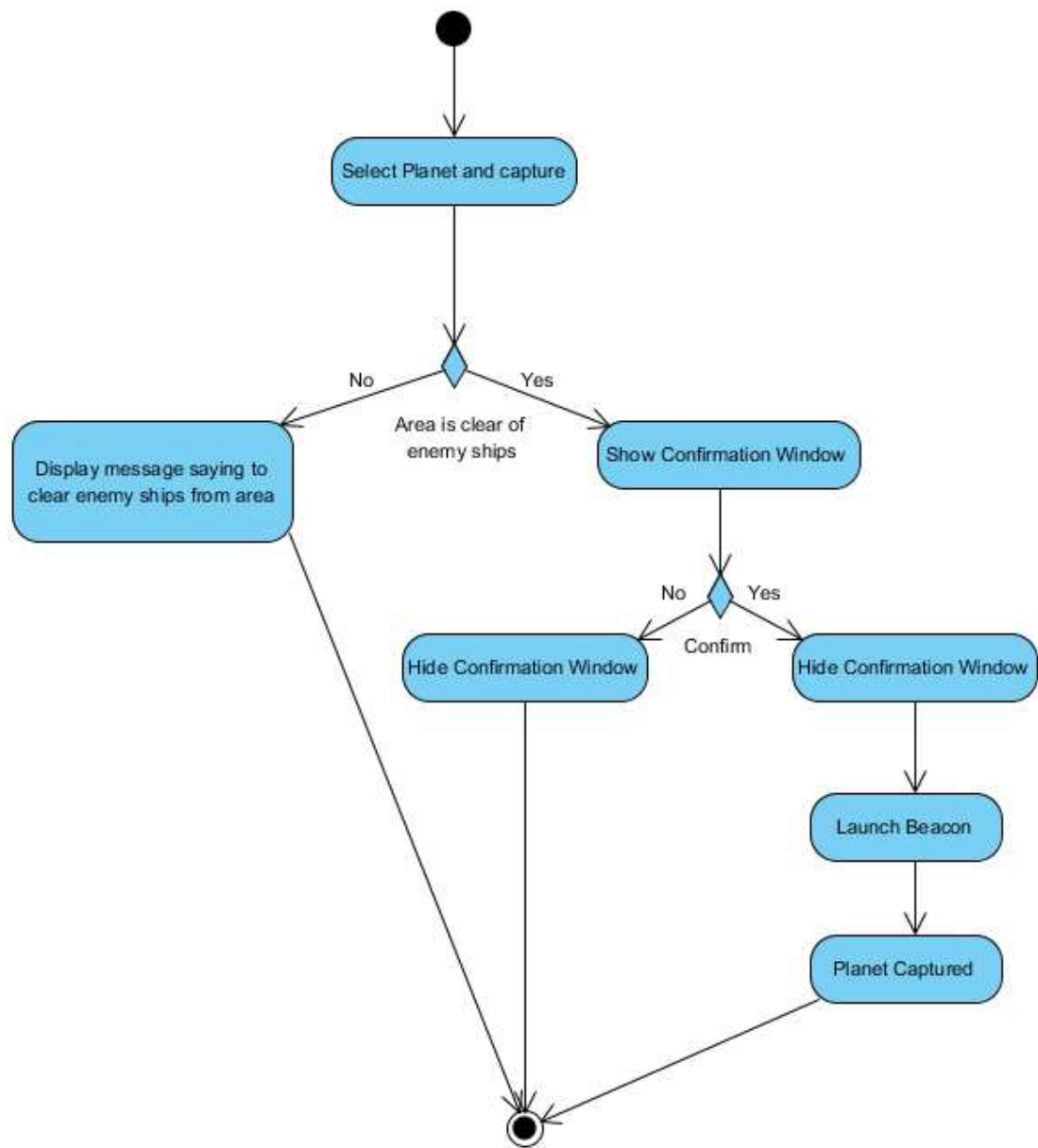
11.13 Select Target Ship (Player)

Player clicks on either a planet or ship from either the menu listing those in the area or by directly clicking in the interface.

Yes / No

Target Valid

Set current target to the selection

Do Nothing
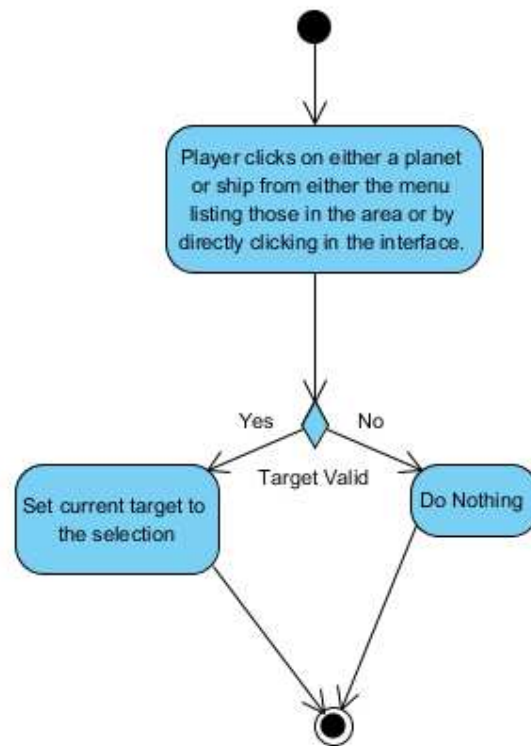
Figure 15 : Select Target Ship (Player) Activity Diagram

11.14 Manage Power
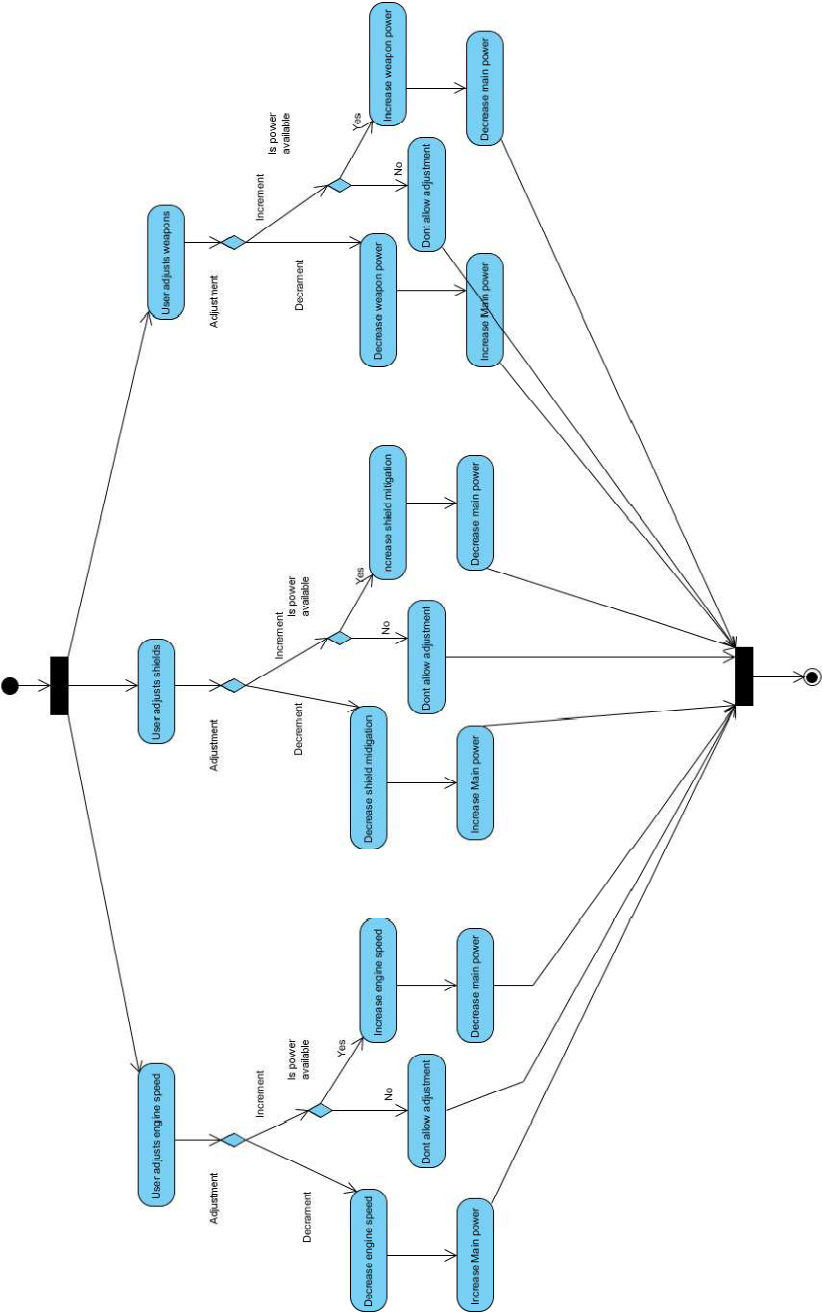
Figure 16 : Manage Power Activity Diagram

11.15 Fire Weapon
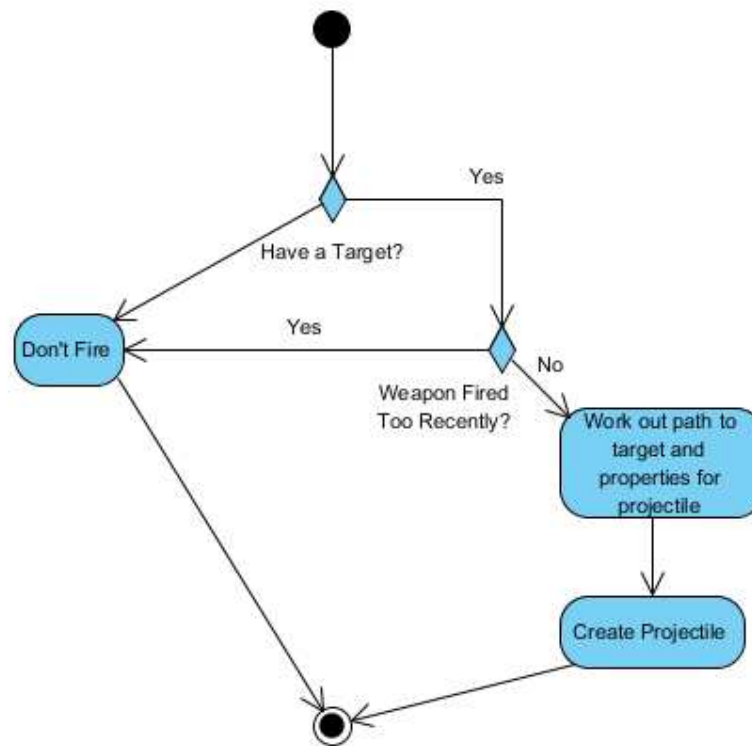
Figure 17 : Fire Weapon Activity Diagram

11.16 Call Allies
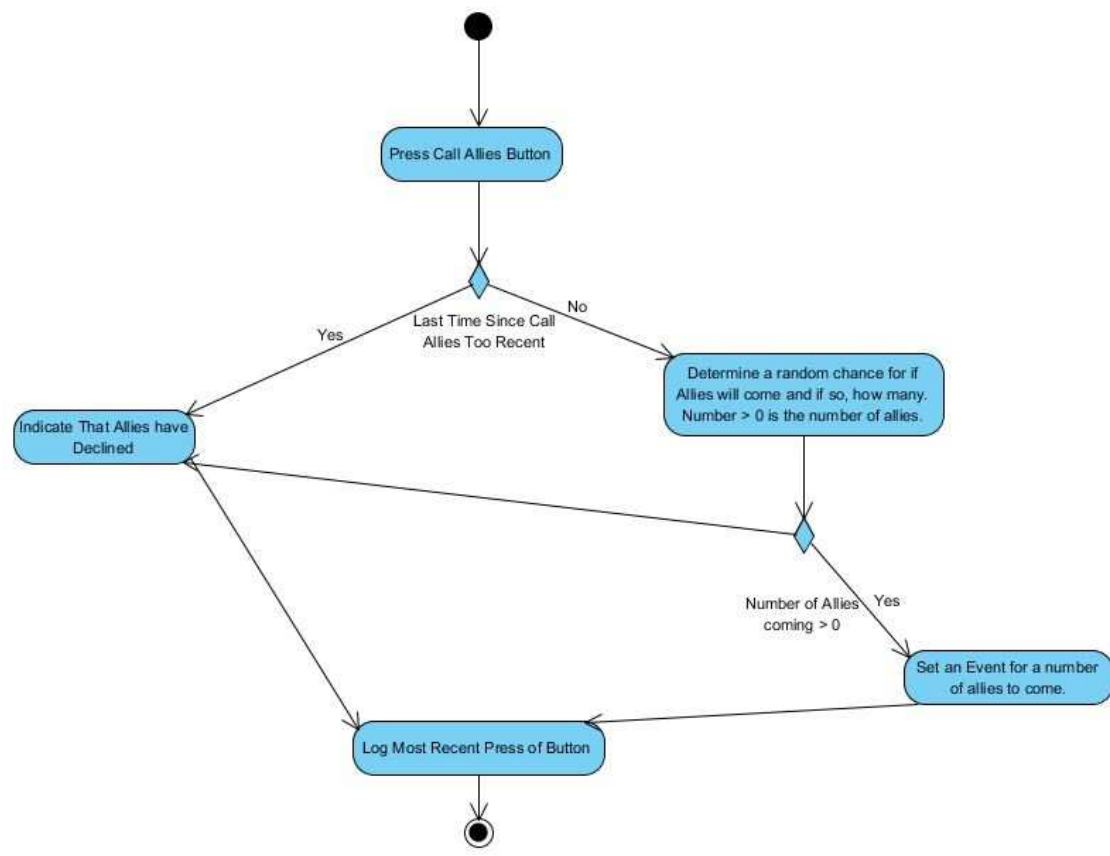
Figure 18 : Call Allies Activity Diagram

11.17 Loot Ship
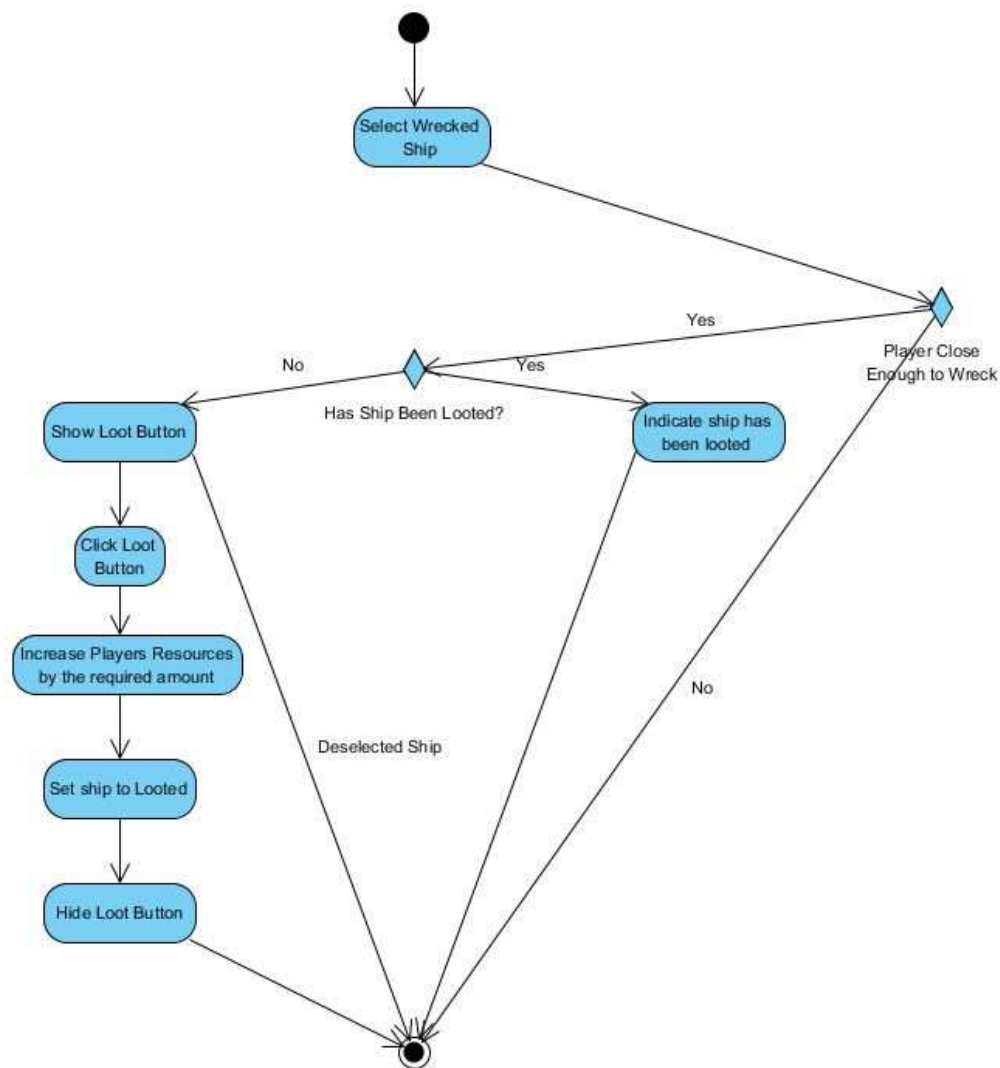
11.18 Projectile Hit
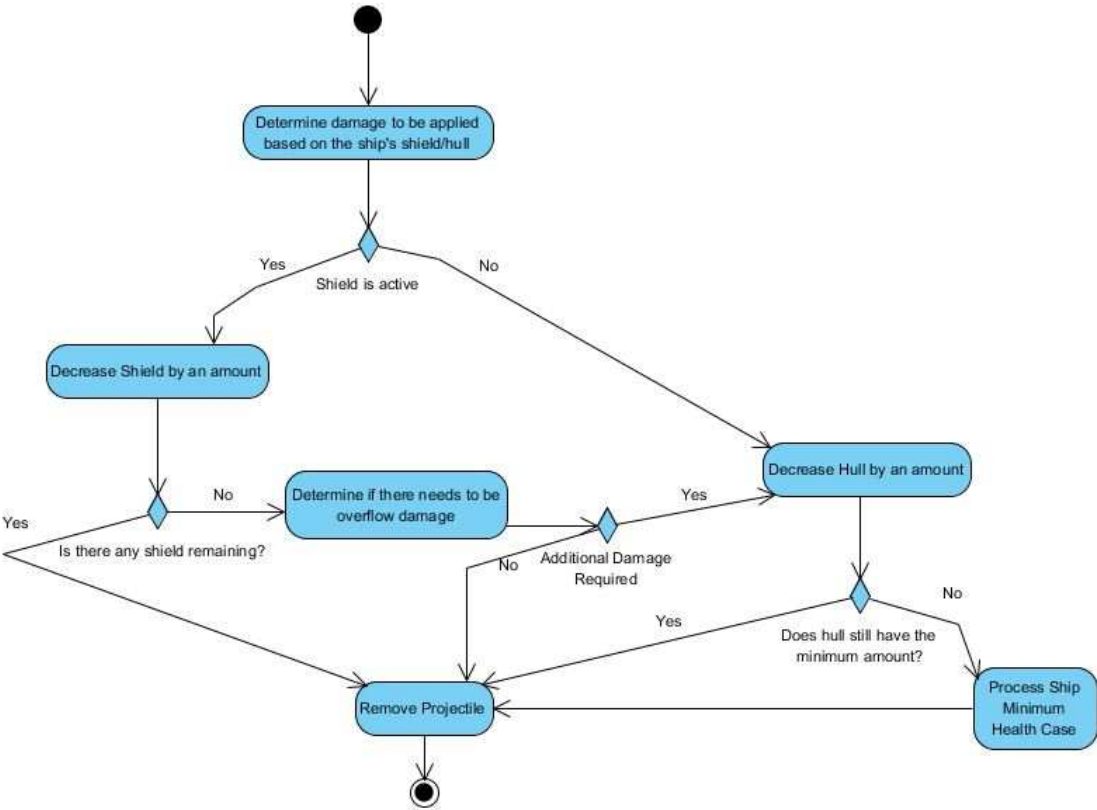
Figure 20 : Projectile Hit Activity Diagram

11.19 Gain Experience

Increase the Player's Ship by an amount based on the target destroyed.

Is ship's experience high enough for new level?

Yes

No

Increase Ships Level

Increase the Stats of the Ship
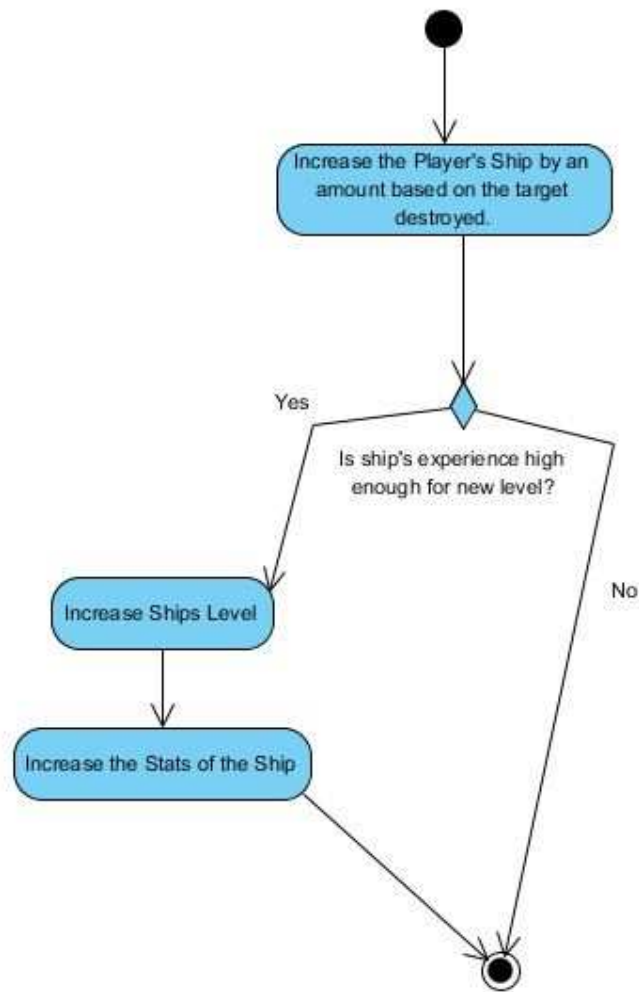
Figure 21 : Gain Experience Activity Diagram

11.20 Projectile Timed Out
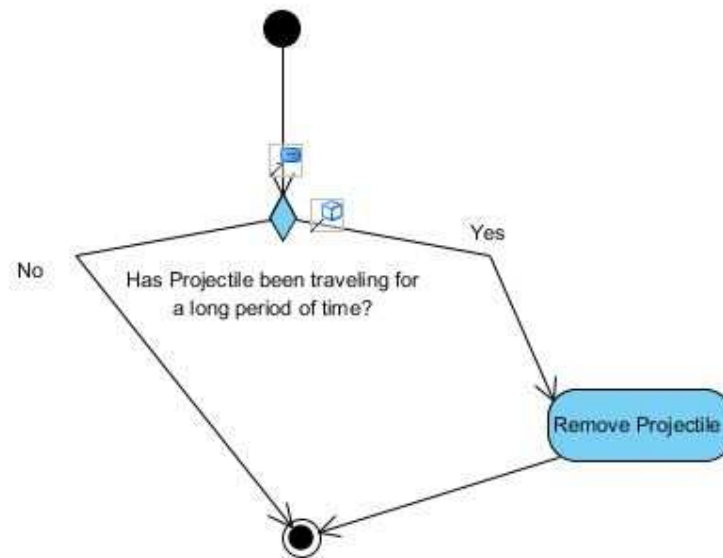
11.21 Ship at Minimum Hull
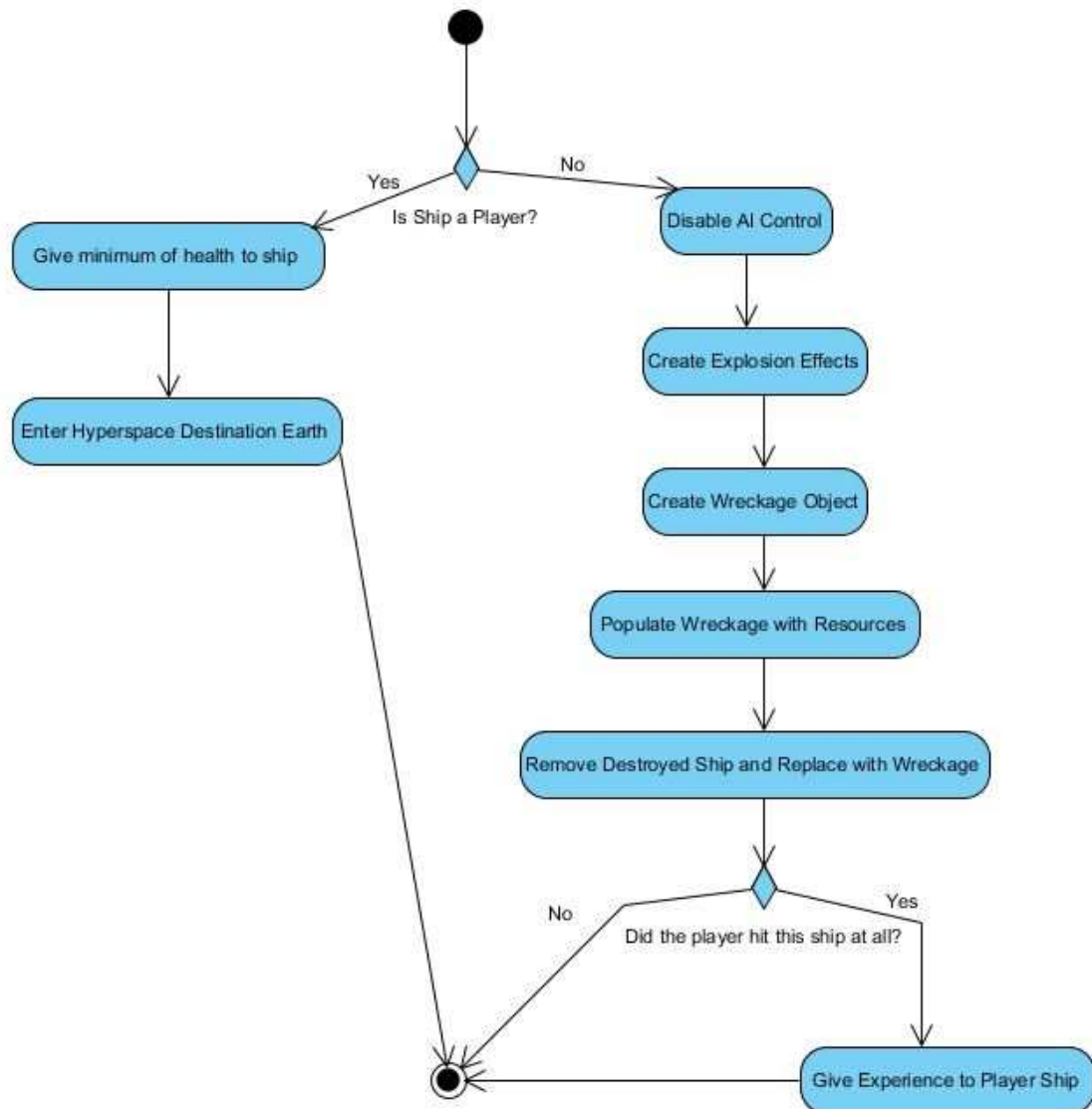
Figure 23 : Ship at Minimum Hull Activity Diagram

11.22 Replicator Special Attack
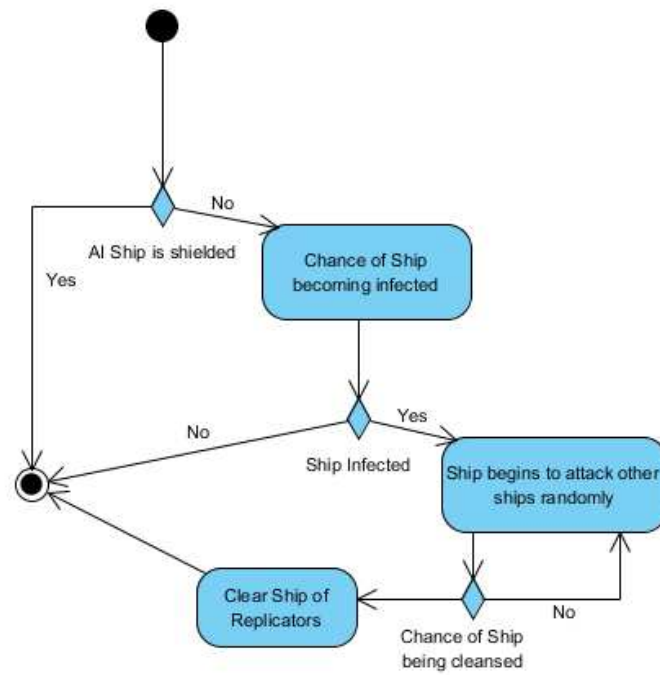
Figure 24 : Replicator Special Attack (AI target) Activity Diagram
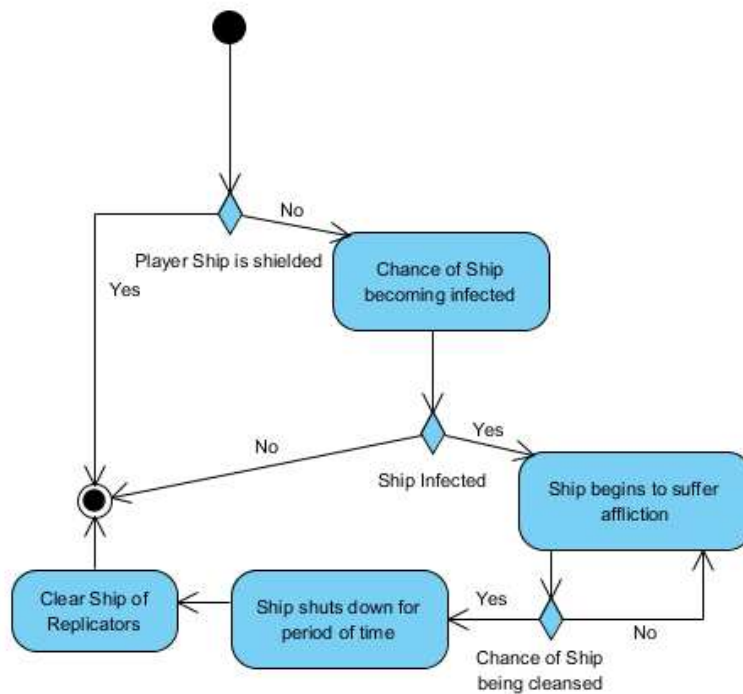
Figure 25 : Replicator Special Attack (Player target) Activity Diagram
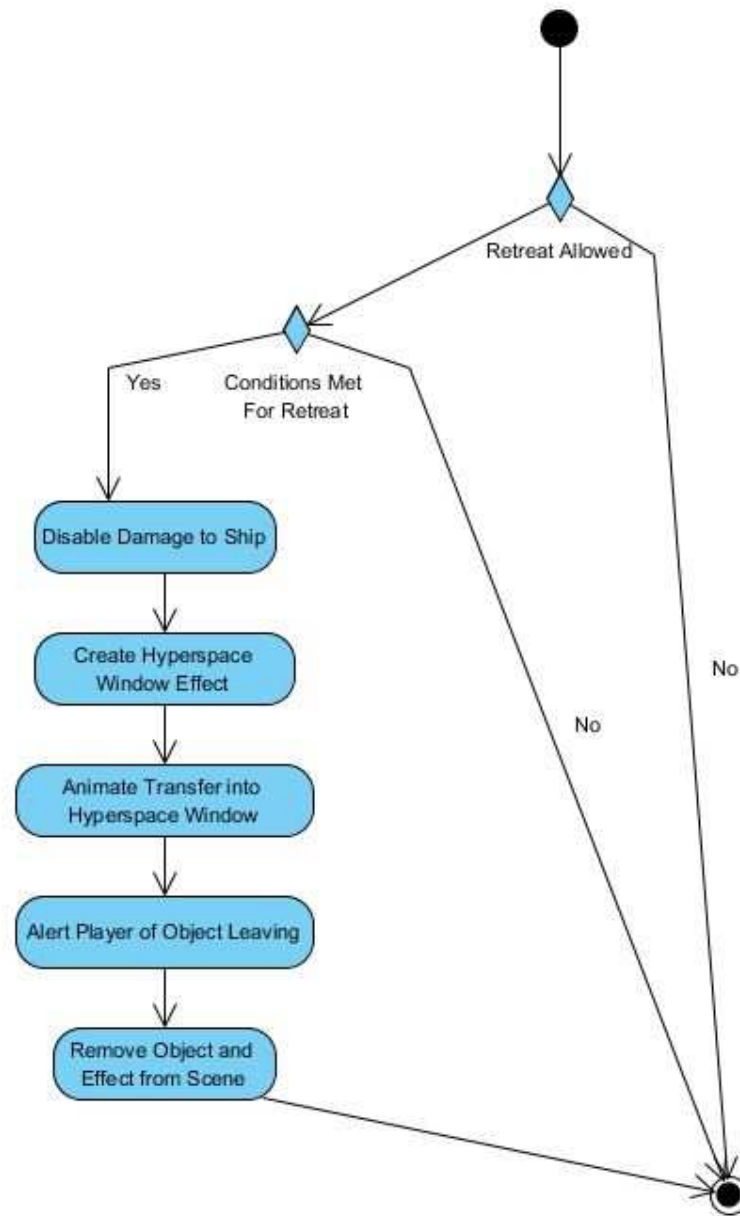
11.23 Enter Hyperspace (AI)

Figure 26 : Enter Hyperspace (AI) Activity Diagram

Conditions for retreat may be determined based on the table over page.

Can a ship retreat

ID	Is Boss	Number of Planets	Hull HP	Random Chance	Retreat
1	No	X	X	X	No
2	Yes	<= 1	X	X	No
3	Yes	X	>60%	X	No
4	Yes	> 1	<= 60% && >40%	1/3 = true	Yes
5	Yes	> 1	<= 60% && >40%	1/3 = false	No
6	Yes	> 1	<= 40% && >20%	3/5 = true	Yes
7	Yes	> 1	<= 40% && >20%	3/5 = false	No
8	Yes	> 1	<= 20%	5/6 = true	Yes
9	Yes	> 1	<= 20%	5/6 = false	No

Table 1 : Can a Ship Retreat
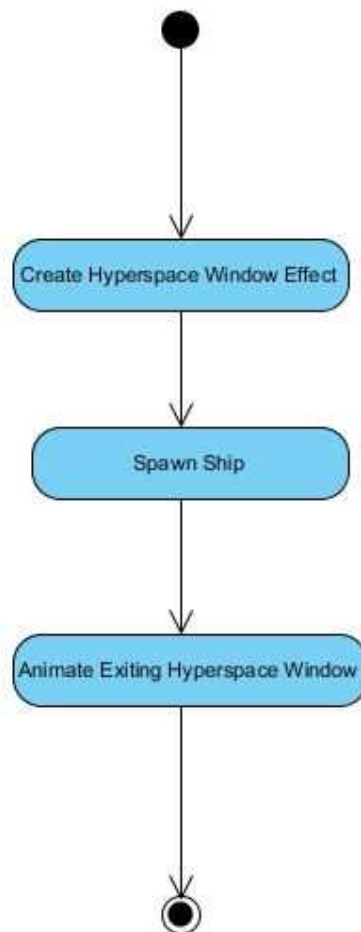
11.24 Exit Hyperspace (AI)

Figure 27 : Exit Hyperspace (AI) Activity Diagram
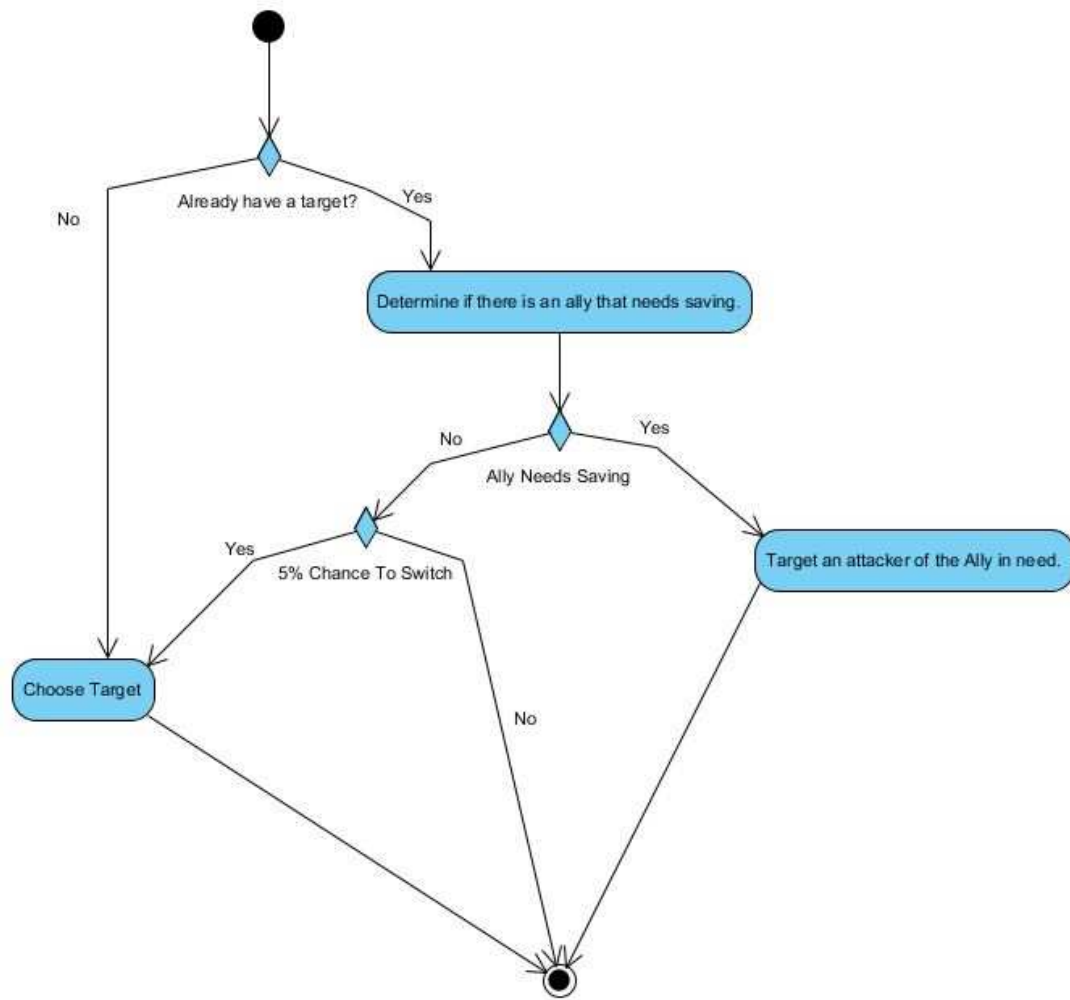
11.25 AI Change Target

Figure 28 : AI Change Target Activity Diagram

11.26 AI Choose Target

The following definitions define what each of the letters represents that is used in the logic of the activity diagram that follows.

A = everyone = FULLSET - U

U = you

X = set of enemies = { A | A.type != U.type AND ((U.type == Human AND A.type == Asgard) OR (U.type == Asgard AND A.type == Human)) }

Y = set of allies = A - X

W = { X | X.target == Y } = X.target is in set of allies

Z = {X | Y.target == X} = allies targeting
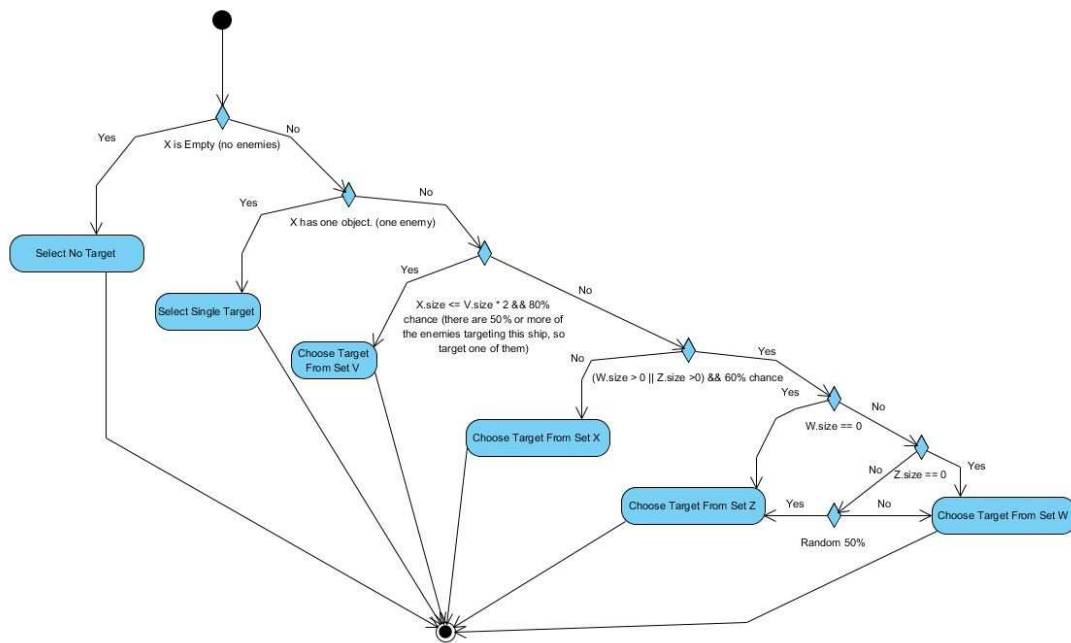
V = X.target is self = { X | X.target == U }

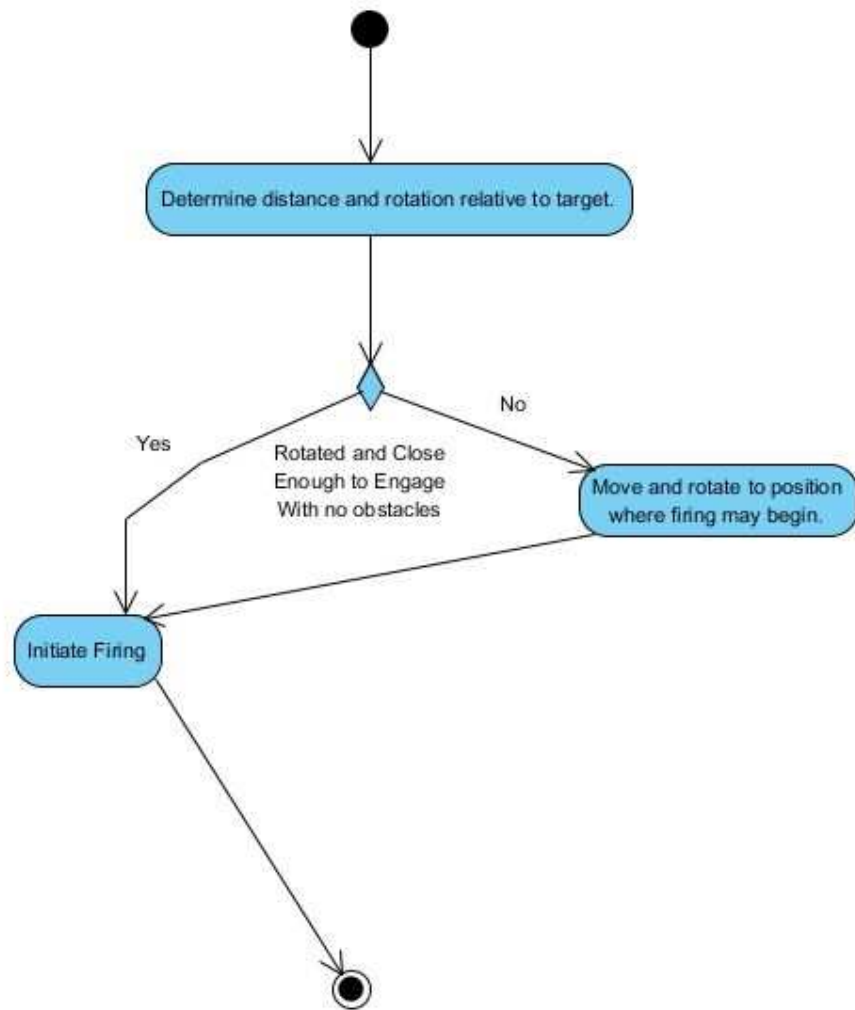Figure 29 : AI Choose Target Activity Diagram

11.27 Engage Targeted Ship

Figure 30 : Engage Targeted Ship Activity Diagram

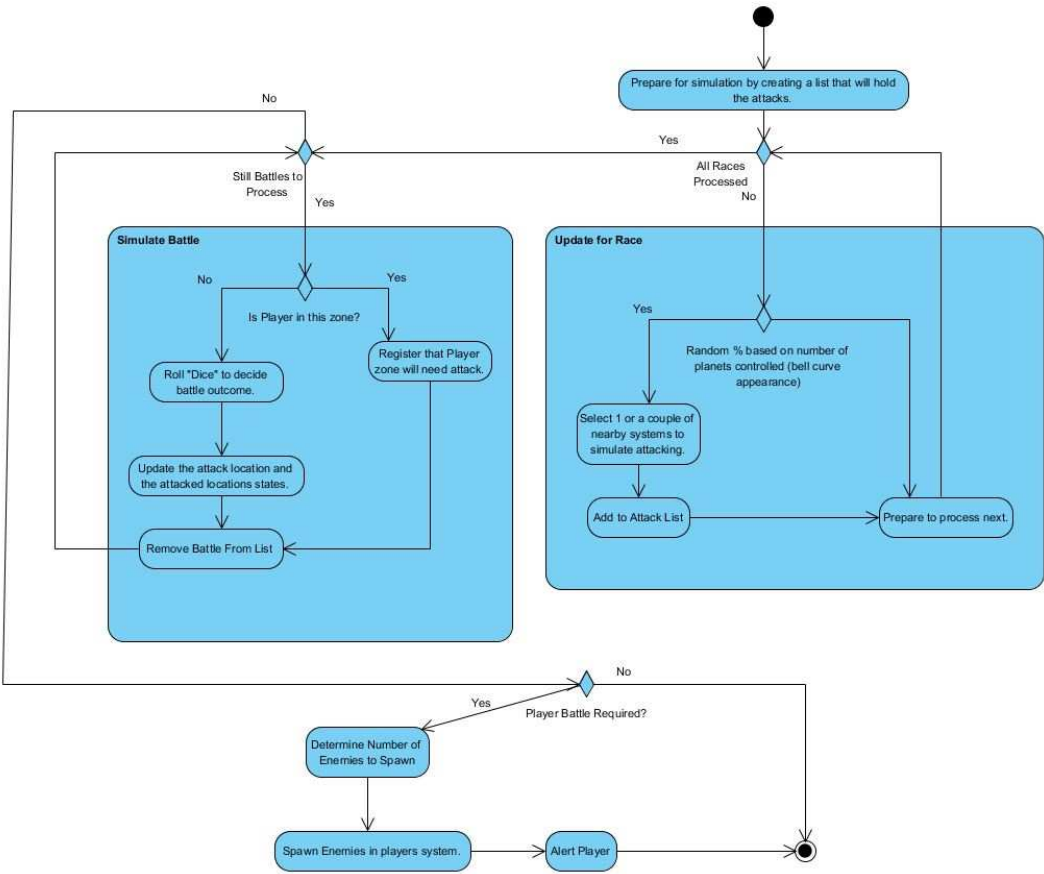# 11.28 Update AI Planet Control

Figure 31 : Update AI Planet Control Activity Diagram

12 Appendix C: Issues List

There are no current known issues.