Benjamin Suh

Nunya Business

Space Battle Rulebook

v0.1

Contents

[Introduction 2](#_Toc473035908)

[Setting Up the Game 3](#_Toc473035909)

[Playing the Game 4](#_Toc473035910)

[Some Quick Notes Before Beginning 4](#_Toc473035911)

[Deployment 4](#_Toc473035912)

[Game Phases 5](#_Toc473035913)

[Ship Profile 5](#_Toc473035914)

[Game Phases 7](#_Toc473035915)

[Command Phase 7](#_Toc473035916)

[Operation Phase 7](#_Toc473035917)

[Resolution Phase 9](#_Toc473035918)

[Assault 11](#_Toc473035919)

[Appendix 12](#_Toc473035920)

# Introduction

**V0.0**: I’ve been interested in making a video game for a while. The thought process that eventually led to this one is far too long to be written in this small introduction, and I suspect that it isn’t terribly interesting to most readers. In any case, if you are interested in learning more about this game, or if you have any comments or questions, you can reach me at [bsuh42@gmail.com](mailto:bsuh42@gmail.com). This is probably the best way to get in touch with me and probably the method which is most likely to elicit a response. Anyways, I should probably describe what this game is all about.

This is a game designed to simulate a space battle between two opposing fleets. Sometime in the far future, humanity has colonized much of the galaxy and of course, they brought many of their prejudices and grievances with them (well, in the lore I have planned, most of the casus belli would be new-found grievances). This of course led to combat, some of it involving spacefaring vessels. Part of the goal of this game is to teach me how to use Python, and so many of the mechanics will be fairly basic. I don’t imagine that I will revolutionize the turn-based strategy genre with this, and I am not aiming to do so. In any case, I hope that you enjoy this little game.

N.B. I will be leaving comments as I write this rulebook. I will try to be consistent with the method by which I leave these comments, and let’s say I will be leaving them in italics, *sort of like this*.

**V0.1**: I don’t know if I’ll copy each introduction over between each iteration of the rulebook. Probably not because it’ll start to get pretty long. Anyways, what’s changed between this and version 0.0? I’m going to try my hand at formatting and making sure everything sounds uniform. The base rules will likely remain the same as v0.0 unless I find some mistakes or think of a better version of the rules. In any case, let’s get started with v0.1. *You’ve probably already noticed the cover page and the table of contents. Those are the sorts of changes I meant when I said I was going to update the formatting.*

# Setting Up the Game

There are currently two versions of the game which are quite intimately linked. There is the digital version which I plan to spend more time and energy on since that is ultimately the final product. This rulebook is, however, for the other version. The physical version of the game is what will be used during the first phase of designing this game to playtest ideas before implementing them in my code. In theory, most of the play will be exactly the same between the two versions.

In order to play Space Battle, you will need the following materials: *I’m re-reading this sentence, and man. We really need a better name for this game than “the game” or “Space Battle”.*

* Two players. While you could play this game alone, that takes out most of the fun, doesn’t it?
* A 14x14 board. I can’t think of any games off the top of my head that natively use a 14x14 board, so you’ll probably have to make it yourself. Alternatively, I will attach an empty board at the end of this document if you just want to print it out.
* Six-sided dice. In this document, six-sided dice will be referred to as D6. If two dice need to be rolled, it will be referred to as 2D6. Three dice will 3D6 and so on. At the moment, the most dice that are being rolled simultaneously is three. So having at least three dice would be optimal. A quick note on convention: I refer to the total value of the dice as the “result”. Many times, you will be asked to compare a result against a value.

For example, when a ship takes a Repair action, the player is asked to roll 2D6 and check to see if the result is less than or equal to the ship’s Crew Skill value (in this case, let’s say 6). The player who controls the ship will roll two six-sided dice (which turn 4 and 1). The result is 5, which is less than the Crew Skill value of 6, so the action succeeds and the ship repairs some Hull points.

* Figures to represent ships. This can be anything from beans to little sculpted figures of ships. It really doesn’t matter what you use so long as you are able to accurately distinguish between ships and different types of ships.
* Paper and pencil. During the turn, players will often be taking actions at the same time, which means that actions will need to be written down so as to keep them hidden from the other player. Paper and pencil are also necessary to mark damages and other effects.

Choosing which player is Player 1 is not particularly important since the board is completely symmetric and there are no advantages associated with being Player 1. In fact, both players are completely equal. They give orders at the same time, order of actions doesn’t actually matter.

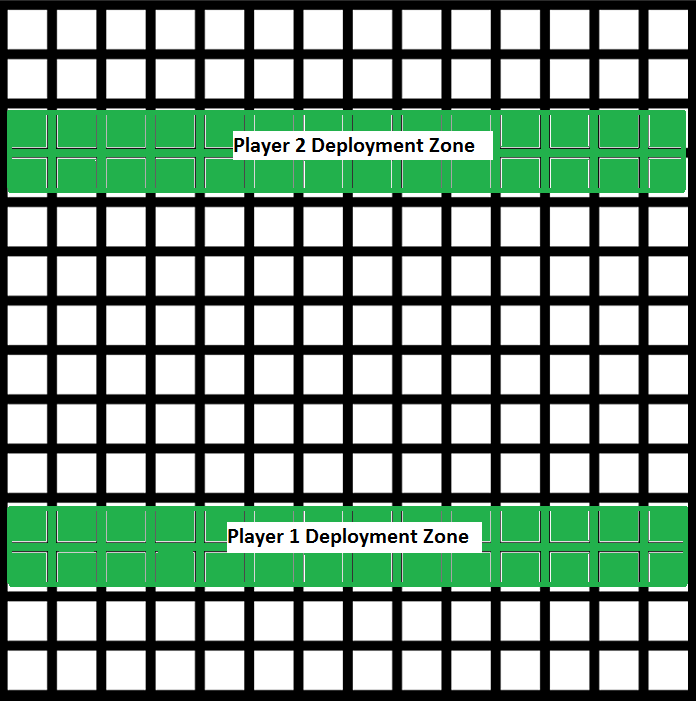
# Playing the Game

## Some Quick Notes Before Beginning

At the moment, there is no system in place for players to choose what ships to bring, so for right now, talk with your opponent about what ships each player will bring. In this rulebook, sometimes there will be a rule that contradicts a special rule that a ship has. In this case, the specific rule will always trump the general rule. For example, if the rulebook says that all ships may fire all their weapons each turn, but a ship has a special rule that it can only fire one weapon per turn, the ship’s special rule will take effect and the general rule is ignored for that ship. The ship may only fire one weapon per turn.

Since the board is completely symmetrical and actions take place at the same time, player numbers do not matter. For the purposes of this book, we will suppose that Player 1 is on the bottom table edge, and Player 2 is the upper table edge.

## Deployment

The Deployment Phase only happens once at the beginning of the game. During this phase, each player will choose where to place their ships on the table. First, players choose opposite table edges, then they must choose how many of their ships will start on the table. Player’s deployment zones are 3-4 rows away from their respective table edge. Deployment occurs simultaneously. *At the moment, all ships must start on the table. There are no rules for reserves. The best way to deploy simultaneously is probably to write down the coordinates of where you want your ships to go then reveal and place ships on the written coordinates.*

This figure shows each player’s deployment zones. The players may deploy their ships on any of the green squares within their respective deployment zone.

*Additional Rule: Reserves. Quick note, these are optional rules that I want to test (and add if they go over well) at some point. I will also add a quick measure of how likely each additional rule is to make it into the game. This rule is very likely to end up in the game. During the Deployment Phase, players may choose not to place all their ships on the board. Rather, players may declare that some of their ships are being placed in reserves. This represents the strategy of not committing your entire forces to a battle. During any turn starting on turn 2, a player may choose to call in reinforcements from their reserves. In order to bring in reserves, during the Command Phase, a player may give an order to their ship to deploy. The square may not be occupied by another ship, friendly or enemy. During the Operation Phase, for each ship being brought in from reserves, roll 2D6.If the result is equal to or less than the Crew Skill value of the ship being brought in, you may place that ship on the board on a square adjacent to a friendly ship that was not deployed this turn and is not in an assault. The incoming ship may not be placed on a square that is occupied by another ship. If the result is greater than or equal to the Crew Skill value, the ship fails to deploy due to engine trouble or still loading or something. For each failed deployment roll on a ship, subtract 2 from each subsequent deployment roll for that ship. So the first deployment roll will be taken at straight dice, the second deployment roll is at 2D6 – 2, the third is at 2D6 – 4, and so on.*

## Game Phases

Once the players have finished deploying their ships, turn one begins. Each turn consists of three phases. See the Phases chapter for details on how each phase proceeds.

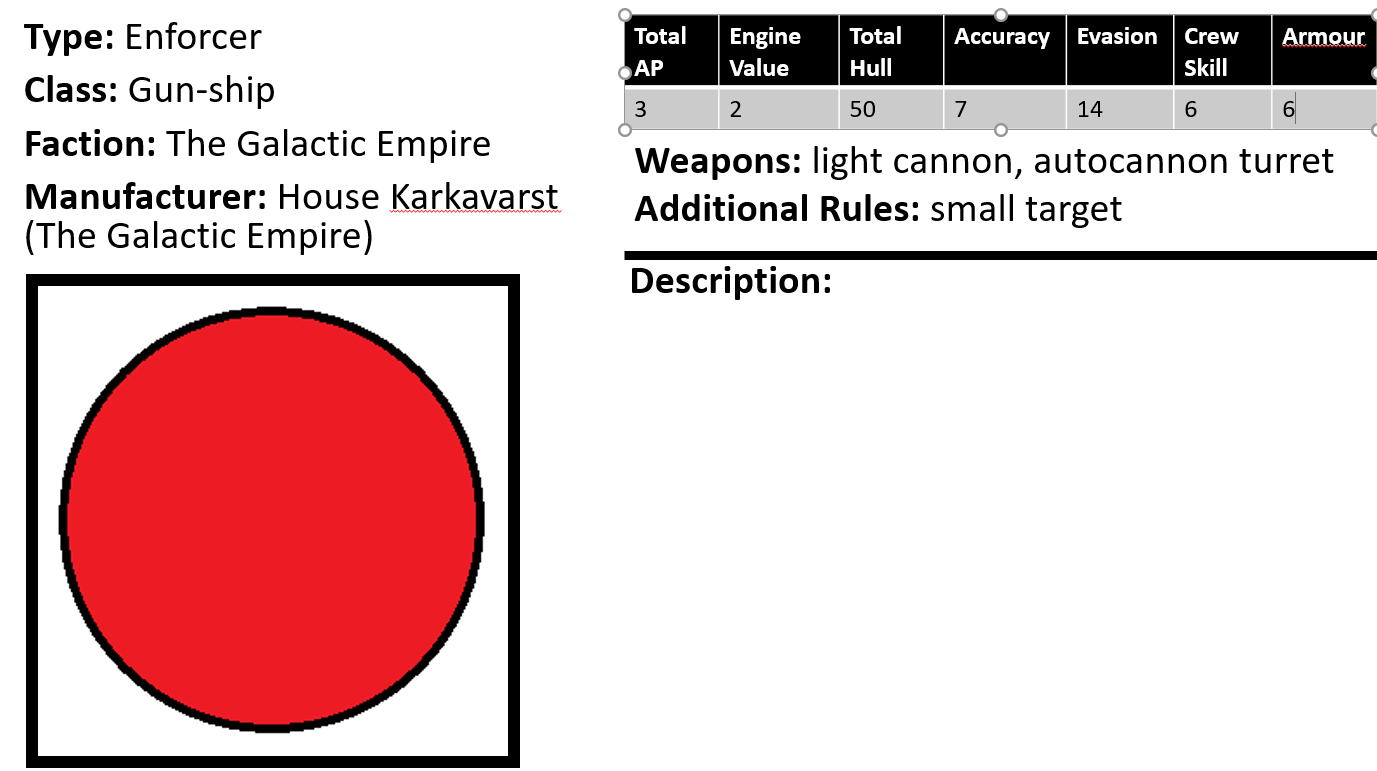
1. Command Phase: during the Command Phase, players decide what actions each ship will take. Each player should write down what actions each of their ships should take then reveal them simultaneously when both players are ready.
2. Operation Phase: during the Operation Phase, ships will execute the actions that were prescribed them during the Command Phase. The order in which actions are undertaken does not matter.
3. Resolution Phase: during the Resolution Phase, players check to see if ships are destroyed, ongoing effects, and others

*Additional Rules: Turn Limit. Fairly likely to be included. Battles do not last indefinitely. Men get tired, ships start to run low on fuel and ammunition. Thus, the game will automatically end on turn 5. Defender will be declared the winner. For now, since there are no defenders or attackers, the game is a draw.*

## Ship Profile

Ships are the primary characters of the game. You take the role of a commander, overseeing a fleet consisting of some number of ships. *Note that even though you represent a physical character, there is no commander character in the game. That is, even if your flagship is destroyed, you will still be in command of the fleet. This is to prevent the game from devolving into an assassination attempt where both players attempt to destroy the enemy flagship to win. Actually, that could be a pretty interesting concept with layers of counter-play and additional strategy. We might to investigate this once we have ships larger than gun-ships.*

Ships come in a vast array of shapes and sizes, ranging from a squadron of fighters to the gargantuan Dreadnoughts whose populations rival that of small planets. Regardless of their size, each ship will take up one square. This is because the square represents not a physical space, but more of a sphere of influence.



This is a sample ship profile for the *Enforcer*, the main gun-ship of The Galactic Empire.

**Type:** the design of the ship. Also known as **Name** in the digital version because.type is already a thing in python.

**Class:** the size of the ship. Generally, tells how many crew are present on the ship.

**Faction:** the faction that most commonly employ this ship.

**Manufacturer**: the primary manufacturer of the ship. *We might want to remove this due to flavour reasons.*

**Total AP:** the number of action points that the ship’s engines are able to produce. At the beginning of each turn, the ship will generate a number of Action Points equal to this value. Any excess Action Points are discarded during the Resolution Phase. Action Points may be spent to perform certain actions such as accelerating or performing evasive maneuvers.

**Engine Value:** the number of action points that the ship must spend in order to accelerate one square.

**Total Hull:** represents the amount of damage a ship can sustain before being destroyed. A ship can never have a hull value larger than the Total Hull.

**Accuracy:** represents gunners’ skill at hitting targets. The Accuracy Value is shared amongst all weapons on a ship regardless of how many different individual gunners there are.

**Evasion:** represents the pilot’s skill at dodging incoming shots. *This number might need to be reduced due to the addition of the small target special rule*

**Crew Skill:** represents the crew’s skill at performing special actions.

**Armour:** represents the ship’s ability to deflect incoming shots. *For how this would work, thing of how tank armour works. When a shell is fired at it, if the shell hits at a certain angle, it will slide off the armour.*

**Weapons:** a list of available weapon systems. Each weapon can be fired once per turn.

**Additional Rules:** any additional ship rules.

# Game Phases

In this chapter, we describe in depth the game phases and all the actions that may take place during those phases.

## Command Phase

Ships may take the following actions during the command phase

**Accelerate:** During the Operation Phase, a ship will move based on its velocity vector. *Hopefully everyone knows how vectors work*. At the beginning of the game, all ships start with a velocity vector of (0, 0). In order to change a ship’s velocity, a ship must first accelerate. If a ship chooses to accelerate, it must first spend a number of Action Points equal to its Engine Value. Accelerating will change either the x-component or the y-component of the ship’s velocity vector by 1.

If we take the Enforcer, whose ship profile has been shown previously, we see that it has a Total AP of 3 and an Engine Value of 2. During the Command Phase of Turn One, player decides to accelerate the Enforcer. Its current AP is 3 since it has not taken any other actions this turn. The ship spends 2 AP, leaving it with 1 more AP to perform any other actions. The ship’s velocity vector changes from (0, 0) to (0, 1), (0, -1), (1, 0), or (-1, 0).

**Fire Weapons:** During the Operation Phase, a ship may fire each of its weapons once. The player must choose a target for each weapon. A player may not choose to target a friendly ship or a ship locked in an assault. Once a suitable target has been chosen, the player may then choose to spend 1 AP to aim the weapon, which gives a +2 bonus to the to-hit roll for that weapon.

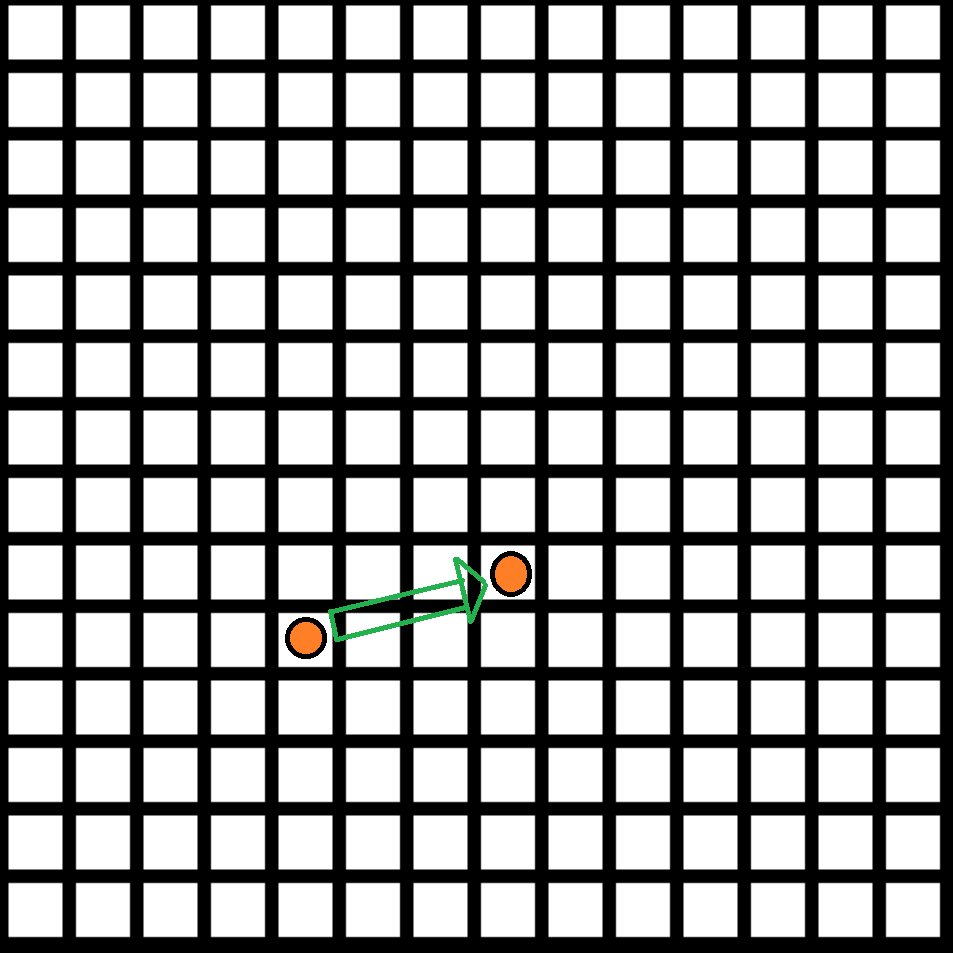
**Evasive Action:** During the Operation Phase, a ship may choose to take evasive action. In order to take evasive action, spend 1 AP. Evasive Action may be taken multiple times each turn on the same ship.

**Repair**: During the Operation Phase, a ship may choose to repair hull damages. In order to repair, spend 1 AP. Repairs may be taken multiple times each turn on the same ship.

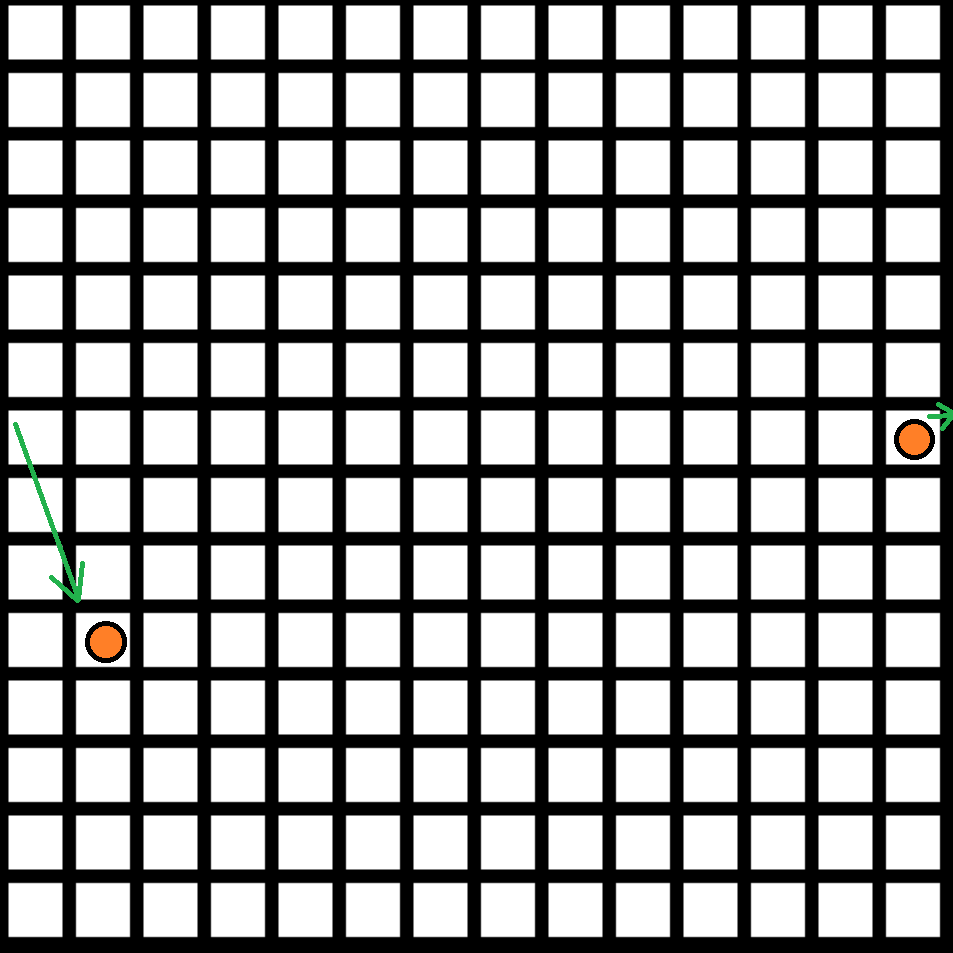
## Operation Phase

During the Operation Phase, all the orders that ships were given during the Command Phase are undertaken. The order in which orders are undertaken does not matter, except for acceleration movement, which always happens first.

**Moving Ships:** A ship moves a number of tiles in the vertical direction equal to the x-component of its velocity vector and a number of tiles in the horizontal direction equal to the y-component of its velocity vector. If a ship encounters the edge of a board while moving, it will wrap-around to the other side of the board. This is done to represent that space battles are not happening on a plane in deep space, but rather in a sphere orbiting a planet. During the Operation Phase, ships are allowed to occupy the same square. There are additional rules for this in the Resolution Phase



In this example, we have a ship at the coordinates (4, 4) where (0, 0) is defined to be the bottom left box. The ship has a velocity vector of (3, 1). During the operation phase, the ship will move from its present position to the coordinates (7, 5). *This convention might not actually be correct because Python actually defines the upper left box to be (0, 0) and down is the positive y-direction. In any case, the spirit of this example should be clear. Hopefully. Also, these are horrible illustrations. But then again, they were made in paint on a trackpad*



In this example, we have a ship at the coordinates (13, 7). It has a velocity vector of (2, -3). During the operation phase, it will normally move to the coordinates (15, 4). However, since (15, 4) is outside of the board, we subtract the board size (14) from any position component equal to or greater than the board size (14) to get the final position of the ship, which in this case will be (1, 4).

**Weapons:** For each weapon that was fired, figure out the distance between the firing ship and the target. *Do not take diagonals into account. For example, in the first movement example picture, the ship would be a distance of 4 away from where it started. Three in the vertical direction and one in the horizontal direction. In the second example, it would be a distance of 5 away from where it started. Two in the vertical and three in the horizontal.* If the distance is greater than the range of the weapon, the shot is considered out of range, and thus suffers a -3 to hit.

To figure out if a shot hits the target, roll 2D6 and add the following modifiers

* +2 is the weapon is being aimed
* -2 per evasive action that target ship took this turn
* -3 if target ship is out of range

Add the total to the firing ship’s accuracy value. If the result is greater than or equal to the target ship’s evasion value, the shot connects. Otherwise, the shot misses and there is no effect.

Just because a shot hit the target doesn’t necessarily mean that it penetrated the ship’s armour. To determine if the shot penetrates and deals damage, roll 1D6 and add to the weapon’s penetration value. If the total is greater than or equal to the target ship’s armour value, the shot penetrates the armour and deals damage to the target ship’s hull equal to the weapon’s damage value.

**Repair:** For each repair action, roll 2D6. If the result is less than or equal to the ship’s Crew Skill value, the repair is successful, and the ship repairs 1D6 hull points. Otherwise, the repair is unsuccessful and the crew fails to restore any hull points. *Might want to make this a 1D10.*

## Resolution Phase

During the Resolution Phase, players check the following in order:

**Remove destroyed ships:** For all ships, check current hull values. If a ship has a hull value of 0 or less, remove that ship from the game.

**Check over-repaired ships:** For all ships, check current hull values. If a ship has a hull value greater than its total hull value, set the ship’s current hull value to its total hull value.

**Move friendly ships:** For all ships, check to see if any friendly ships are occupying the same square. If they are, move the ship with the largest speed one square in the direction of its largest velocity component. Repeat until no friendly ships are occupying the same square. *To calculate the speed of a ship, figure out the amplitude of the velocity vector. If a ship has velocity (4, 2), it moves to the right by one square. If it has velocity (1, -3), it moves 1 square down.*

**Check Assault:** For all ships, check to see if any enemy ships are occupying the same square. If they are, they are considered to be in an assault. This represents soldiers latching on and attempting to board the enemy ships and take control of it. Repeat until no enemy ships are occupying the same square. *Assault rules have not been fully formed, and so will be its own section.*

**Check victory conditions:** Check to see if either player has met the victory conditions. As of right now, the only way to win is to destroy all enemy ships. Note that this only counts ships on the board. If a player has ships not on the board (for example, in reserves), they are considered not part of the game for this phase only. *That is to say, if a player’s entire fleet is in reserves during the Resolution Phase, that player loses.*

# Assault

During an assault, ships stop firing at each other and instead begin to deploy marines to kill or capture the enemy crew and take over the enemy ship. Assaults are bloody affairs and can only end when one ship is destroyed. In order to participate in an assault, a ship must have marines. If one ship no longer has any marines, but the enemy does, it is considered destroyed. This is not to say the ship is actually destroyed, but rather it has been disabled in some way and can no longer participate in the battle. If neither ship has marines, follow the rules for moving friendly ships that occupy the same square. If both ships have marines, then the real fun begins.

*Here, the rules for an assault will differ between the digital and the physical versions. In the digital version, I can do hundreds of calculations in no time, but I don’t want to subject the player to that.*

Each player will look at how many marines are on their ship. The player with more marines will roll three dice, and the player with fewer marines will roll two dice. The player who rolled three dice will discard the lowest number rolled then compare the largest number to the largest number the player with two dice rolled. Then compare the remaining die. For each die pair, the player who rolled the lower number removes one marine. The player who rolled two dice wins ties. Repeat this entire process until one player no longer has any marines.

# Appendix

