Design Example - Steel Tactics Online

Design Analysis of Existing Game

Steel Tactics Online was a real-time player vs player tactics game with an emphasis on building Mechs, 6 of which are taken into a match against another player. When I joined Artplant the project was still in development and, ultimately, was never published.

While I was working on another project the STO team was struggling with the game not coming together as a satisfying experience. For example they were having difficulty with the user interface for building custom Mechs. Due to my UX background I was asked to evaluate and redesign this portion of the game. As I began testing and investigating the UI it quickly became apparent to me that the issues they were having were rooted deep within the game design itself and the poor interface was just a symptom of those problems.

Major Issues

I identified several major issues, including;

1. Difficult to design and assemble Mechs
2. Combat design dictated that heavy, high damage mechs were always superior
3. Mechs had many, sub-systems and targeting them was done through UI buttons

4. Combat was extremely fast paced with engagements being determined in seconds; making matches unsatisfying.
5. Levels were non-interactive and uninteresting

The Redesign

The redesigned concept is the proposal I created to address these and other issues. Below I outline my thought process and the reasons behind each decision.

Difficult to design and assemble Mechs

To address the difficulty of designing Mechs I drastically simplified their stats, number of components, and created a Hardpoint-Slot system as described in the design.

This allowed even novice players to have a grasp of how Mech pieces fit together and what their relative power is. I also added an equipment-ability system to reintroduce some depth back into the game but in a more discrete, controlled, and understandable manner.

Combat design dictated that heavy, high damage mechs were always superior

Probably the largest problem with the previous design, there needed to be a combat role for lighter Mechs. Since they couldn't match the firepower of a heavier Mech I created the Detection System to add a layer of information warfare that Light Mechs could excel in.

This encouraged players to bring a balanced group of mechs, such as scouts and artillery, rather than just heavy tanks into a match.

Mechs had many, sub-systems and targeting them was done through UI buttons

Mech sub-systems are core to the genre and necessary for a tactics game with 6 units per player. For this reason I wanted to keep them. Rather than simply be menu buttons sub-system damage would be determined by relative battlefield positioning between Mechs.

This allowed players to take appraisal of which mech sub-systems were at risk simply by looking at the main game window as well as adding counter-play for defending units. Where before players were mostly engaged with the UI now they were engaged with the game itself.

Combat was extremely fast paced with engagements being determined in seconds; making matches unsatisfying.

Combat was over extremely quickly and favored superior firepower such that the optimal strategy for every match and situation was to create Mechs with the most damage possible and use them all to simultaneously “Alpha Strike” each of the opponents Mech’s in turn. To alleviate this problem I introduced damage zones and interdiction.

This meant that Mechs could no longer fire “through” other mechs and hit any sub-system they wanted. Now positioning correctly became a critical combat factor and maneuvering Mechs into slightly advantageous 2v1 situations became a fundamental part of victory. Slowing down combat in this manner added more tactical layers and improved decision points.

Levels were non-interactive and uninteresting

Previously the levels were uninteresting and didn’t play a strategic role in the match. The previous design team attempted to address this by adding new features such as capturable repair stations to the levels.

I identified that the levels themselves weren’t the core problem but that the combat systems themselves didn’t allow players the opportunity to exploit even basic level design elements that were present, and this was the primary source of the issue. As such I focused on the problems described above first. Once positioning, scouting, and interdiction became core combat features the levels came alive without much additional work.

Steel Tactics Online Redesign Concept

Steel Tactics is a competitive real-time tactics game set in the future. Players are mercenaries with a crew of mechs that take contracts to battle for land. Through a combination of clever set up and perfect execution players are able to defeat their opponent and take control of the battlefield.

Mechs

Combat Systems

Detection System

Environmental Effects

Pilots

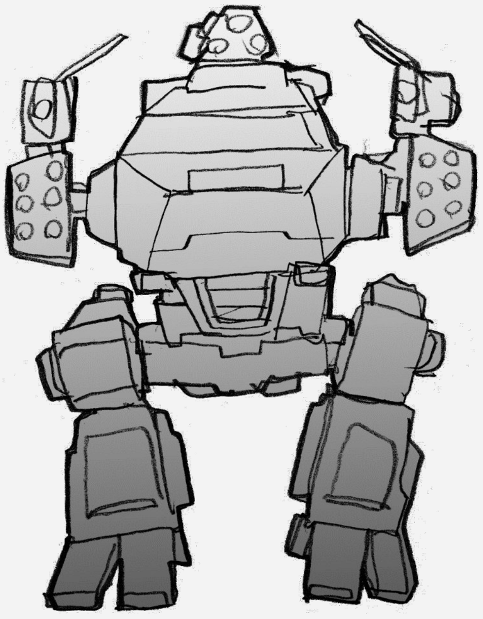
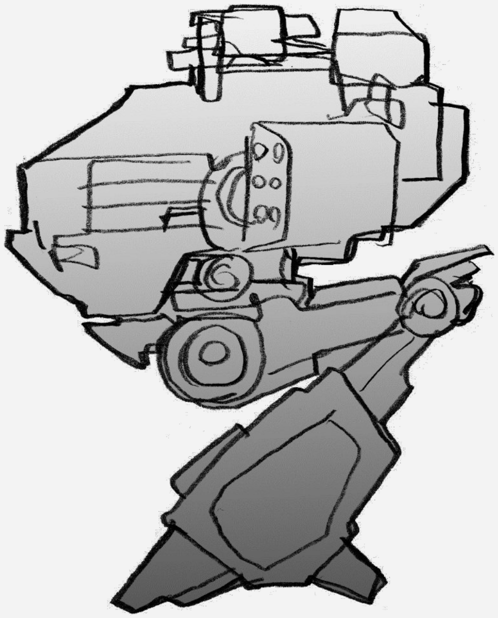
Research

Matchmaking

Mechs

Mechs are walker style armored battle vehicles controlled remotely or by an internal pilot. These machines vary greatly in size, shape and appearance. They are the basic unit players use on the battlefield. Mechs are highly customizable and are assembled by the player in their garage by combining 3 main components to assemble a chassis and then installing weapons, armor, and equipment on that chassis.

A Mech's construction, loadout, and combat effectiveness are constrained by weight and heat generation.

	
Front View - Chassis	Side View - Chassis

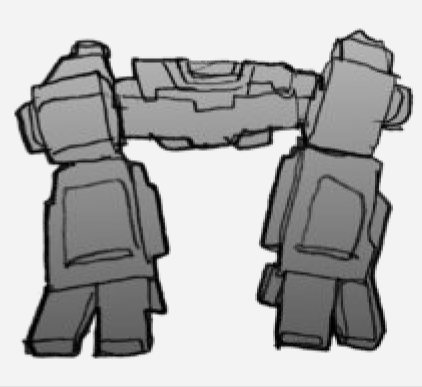
Components

A Mech consists of 3 major components; legs, torso, and arms. Together these 3 components for the Mech chassis which is the object that all other equipment is built against.

The total integrity of the Mech is determined by the sum of the sub-systems of these 3 components. A Chassis is destroyed when it's total Integrity reaches 0 regardless of the state of its sub-systems.

Legs

Legs determine the maximum weight of the chassis as well as its movement characteristics. Legs also have a health and mitigation value. As a general rule faster legs are able to carry less weight and are more vulnerable to damage. Legs that are able to carry more weight or are tougher have their movement attributes compromised.



Legs Component

Legs are also their own sub-system.

Leg Stats

Stat	Unit	Description
Leg Integrity	v	Max Health Leg Sub-system
Density	v	Mitigation Rating
Tonnage	v	Max Weight
Forward Speed	m/s	Rate at which a Mech moves forwards
Reverse Speed	m/s	Rate at which a Mech moves backwards
Turn Rate	d/s	Rate at which a Mech can turn
Accessories	v	List of Device Mounts by size
Fittings	v	List of Armor Mounts by size
Armor Capacity	v	Total Armor Value

Torso

The Torso component contains the Mech's reactor and therefore its cooling system, thus creating an effective limit for weapon firing rates. The Torso also contributes the bulk of a Mech's integrity by having the largest portion of the health pool.

Reactor cooling systems and tougher structure both cost weight. Lighter torso's will therefore provide less cooling to weapons and be more susceptible to damage.

The torso also contains a special weapon mount referred to as the turret.

2 subsystems are contained within the torso, the turret and pilot housing.

Torso Component

Torso Stats

Stat	Unit	Description
Pilot Integrity	v	Max Health Pilot Sub-system
Turret Integrity	v	Max Health Turret Sub-system
Density	v	Mitigation Rating
Weight	v	Weight cost
Cooling Rate	v/s	Rate at which a Mech vents heat
Vent Time	m/s	The length of time a Mech shuts down when venting
Rotation Rate	d/s	Rate at which a torso can rotate
Hardpoints	v	List of Weapon Mounts by size

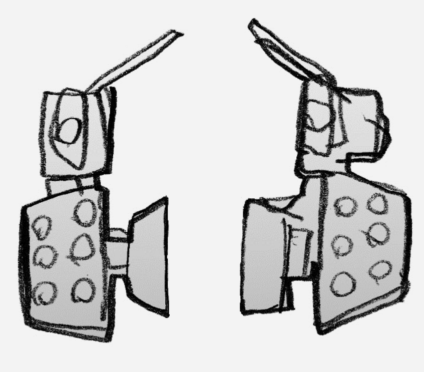
Accessories	v	List of Device Mounts by size
Fittings	v	List of Armor Mounts by size
Armor Capacity	v	Total Armor Value

Arms

The arms of a Mech contain most of its weapon mounts. If an arm is destroyed any weapons mounted on it will cease functioning. Weapons themselves do not contribute any weight towards the total limit, but their combat effectiveness is limited by their heat generation vs the torso's cooling rate.

The weapon mounting system is a flexible slot based system in which a weapon occupies a slot, but that slot may only accommodate weapons up to a certain size.

Arms are added individually but are ambidextrous. An arm may be mounted in either the left or right arm slot on a Mech. Each arm is its own sub-system in battle.



Arms Component

Arm Stats

Stat	Unit	Description
Arm Integrity	v	Max Health of Arm Subsystem
Density	v	Mitigation Rating
Weight	v	Weight cost
Hardpoints	v	List of Weapon Mounts by size
Accessories	v	List of Device Mounts by size
Fittings	v	List of Armor Mounts by size
Armor Capacity	v	Total Armor Value

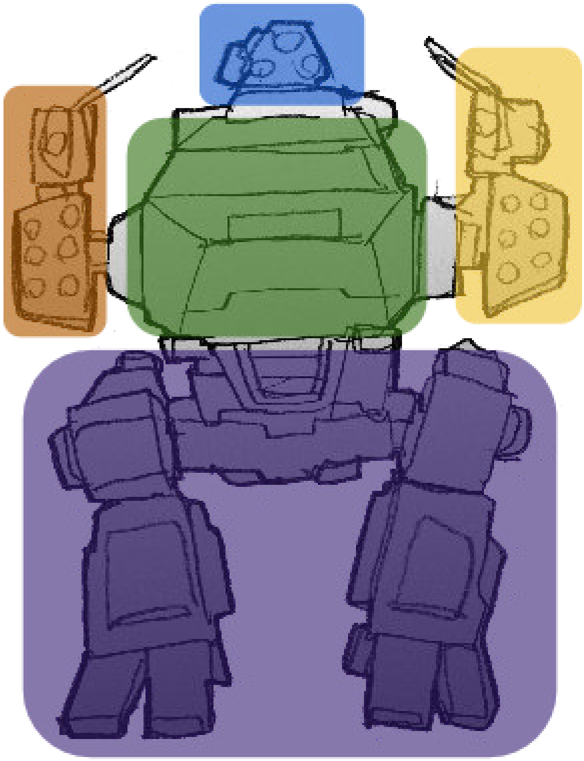
Sub-Systems

Sub-systems are parts of the Mech that control certain aspects of its functionality. Sub-systems are linked to one of, but not exactly the same as, the 3 main components.

The Legs component contains the legs sub-system and controls all Mech movement. If this sub-system is destroyed the Mech is rendered immobile.

The torso component contains the Pilot Housing and Turret sub-systems. The Pilot provides passive perks to Mech operation, and the Turret sub-system controls the shoulder mounted weapon. If the Pilot housing is knocked out the perks cease functioning, and if the turret sub-system is destroyed the attached weapon is no longer usable.

Finally the Arms component contains the left and right arm sub-systems. If these are destroyed any weapons attached to that arm become unusable.



Sub-System	**Component**	**Functionality**
Leg	Legs	Mech Movement
Pilot	Torso	Passive Perks
Turret	Torso	Shoulder Weapon
Right Arm	Arms	Right Weapons
Left Arm	Arms	Left Weapons

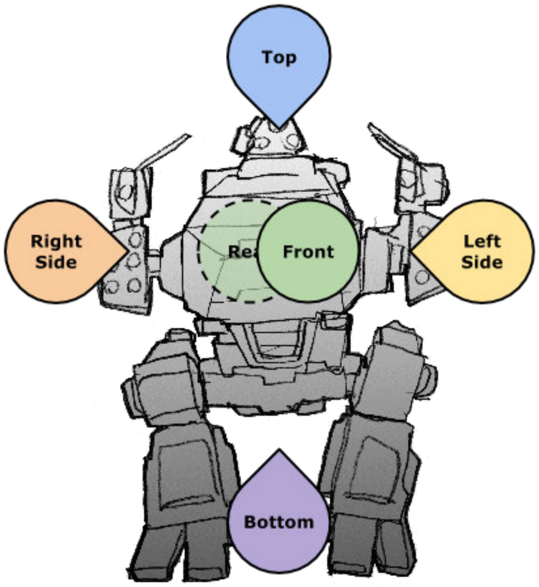
Damage Zones

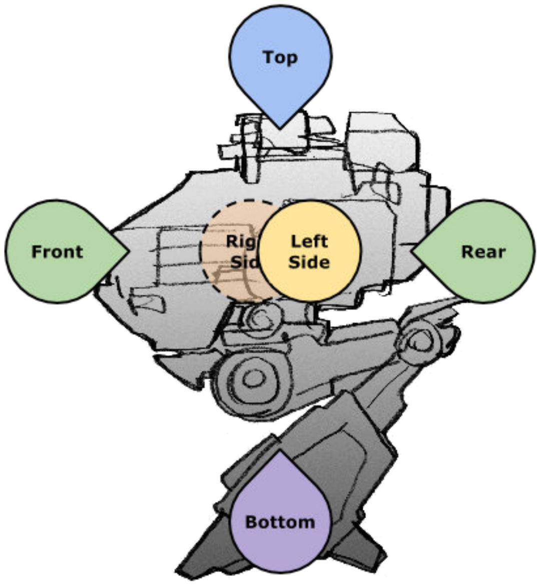
The game determines which sub-system takes damage based upon where the Mech is hit. This is not randomized in any way or based on a % chance. What you see is what you get.

If a Mech is hit it takes damage in the sub-system associated with that location as well its overall integrity. If a subsystem is destroyed, the Mech will continue to take overall damage when hit in that location.

There are 6 damage zones whose position resembles the face of a cube. Direct Fire weapons do damage to 1 of the 4 sides of a Mech, indirect weapons damage either the top of a Mech if the shot comes from above, such as with artillery, or the bottom of a Mech, such as with a mine.

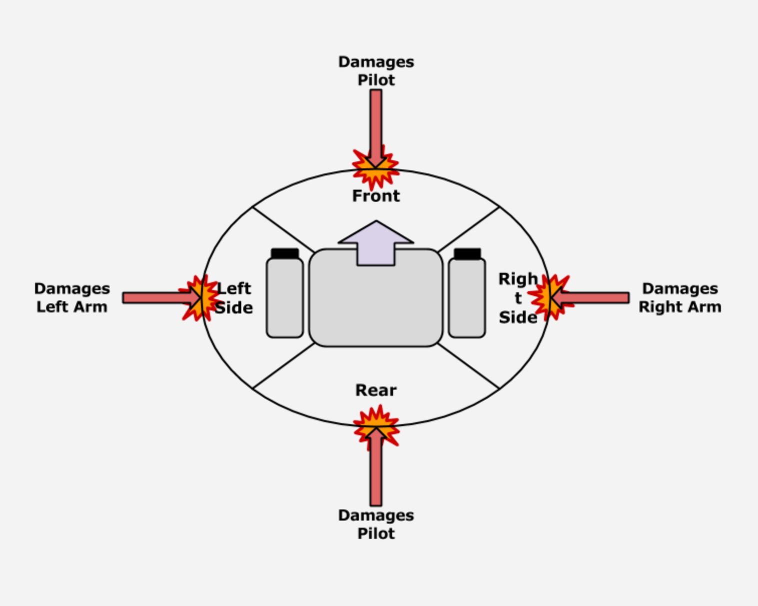
Each damage zone is independently armored. It is up to the player to determine how much armor to place in each zone and of what type.



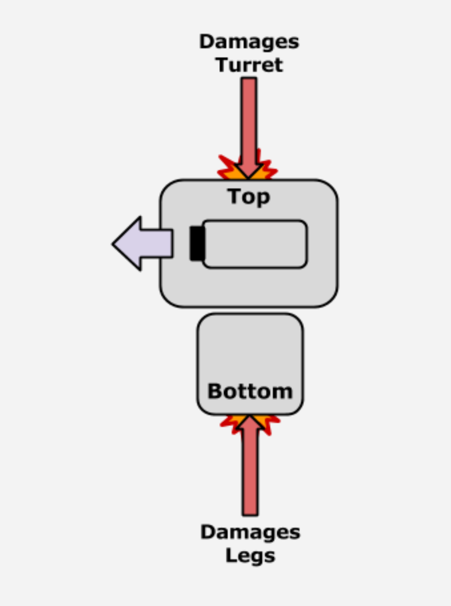


Damage Zones

Zone	Sub-System
Bottom	Leg
Front	Pilot
Rear	Pilot
Top	Turret
Right Side	Right Arm
Left Side	Left Arm



Direct Fire



Indirect Fire

Equipment System

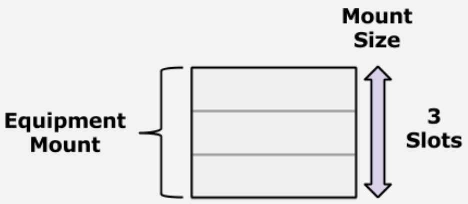
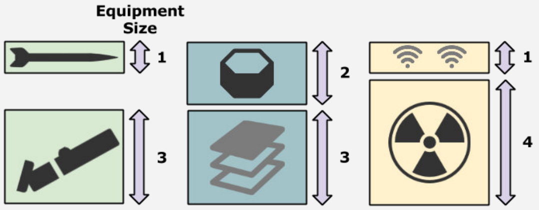
Equipment is a generic term for things the player slots into a chassis to give it functionality.

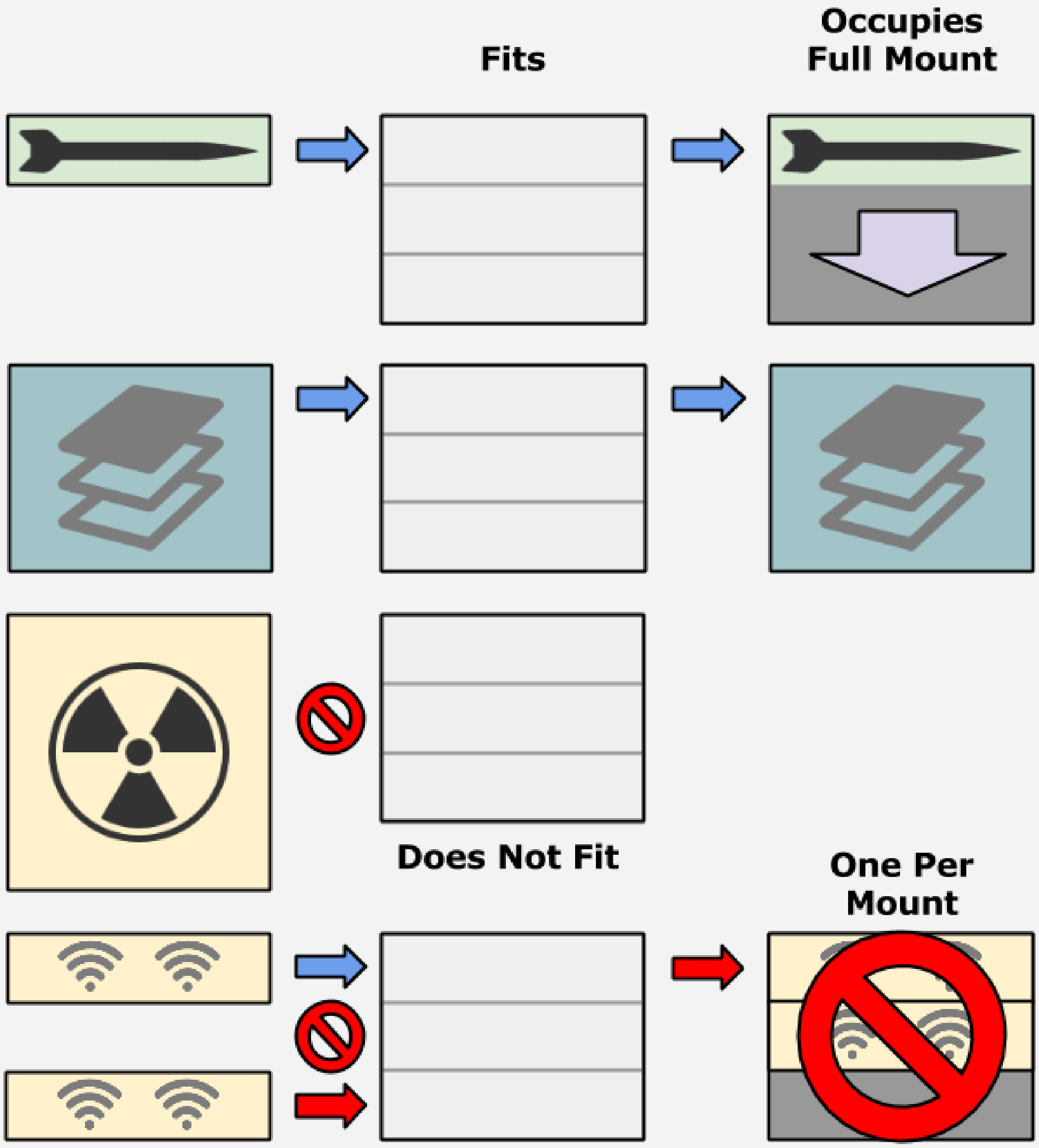
There are 3 major types of equipment; weapons, armor, and devices. Weapons provide the functionality and stats for a Mech's basic attacks. Armor is manually installed plates that provide different types of damage reduction in a defensive scenario. Devices are special active abilities the player initiates manually. Devices can vary significantly in function.

Mounting

Equipment is installed via the mounting system. It applies to all equipment that is mounted onto Mech components. Equipment such as weapons, armor, and devices does not cost weight, but instead has a size that must be able to fit into a slot. Only one piece of equipment may occupy a slot at once.

This type of mounting system allows the player to determine up front, based on components, whether the Mech will lean towards speed, attack, or defense. This enables her to more easily customize a specific load outs as the player will not have to juggle weapons vs armor vs devices. Instead they will only have to juggle weapons against other weapons and so on.

	
Components contain Mounts of a size	Equipment also has a size

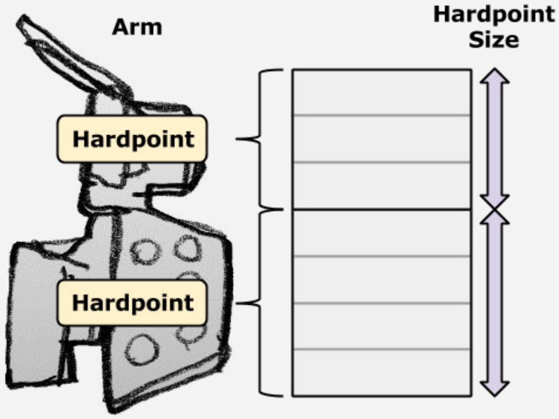


Equipment must fit in mount, only one per mount

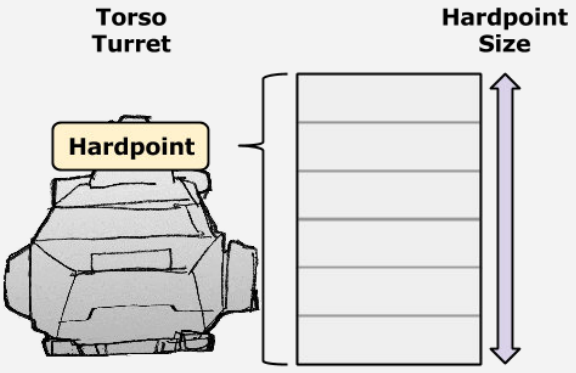
Hardpoints

Weapon mounts are called hardpoints. Weapon hardpoints are located primarily on arms but may also be available on the torso. Weapons, like all equipment, must fit into one of the hardpoints located on the arm or turret. An arm or turret may have several hardpoints of varying sizes.

A weapon occupies the full hardpoint, even if it is smaller than the maximum size of the hardpoint, two weapons may never occupy the same hardpoint. A weapon that is larger than a hardpoint may not be mounted onto it.



Components contain Hardpoints

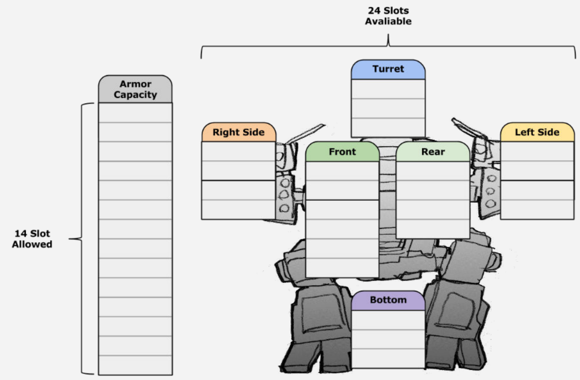


Weapons must fit into a Hardpoint

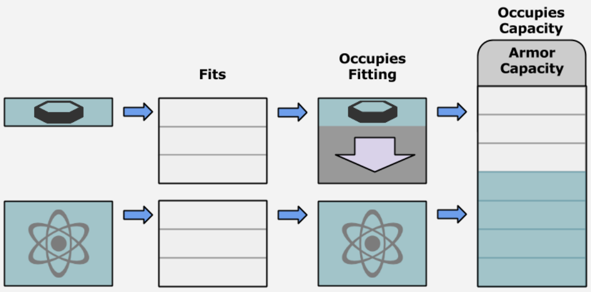
Fittings

Armor mounts are called fitting. Each damage zone has a certain size and number of fittings which vary among torsos. Like all equipment armor must be the same size or smaller than the fitting in order to be installed.

Armor differs slightly from weapons in that there is a maximum amount of armor that can be attached to a chassis. This allows players to customize the distribution of armor protection among the various damage zones to fit their strategy.



Components Armor Fittings



Armor occupies Capacity based on Size

Accessories

Devices are the third type of equipment, and these fit into an accessory slot. Devices provide active abilities to a Mech but are not weapons or armor. Devices use the same mounting system as weapons with each component having (or not having) an accessory slot or slots of certain sizes. A device occupies the entire slot and must be equipped in order to function.

Devices have charges and/or a cooldown in order to govern their use.

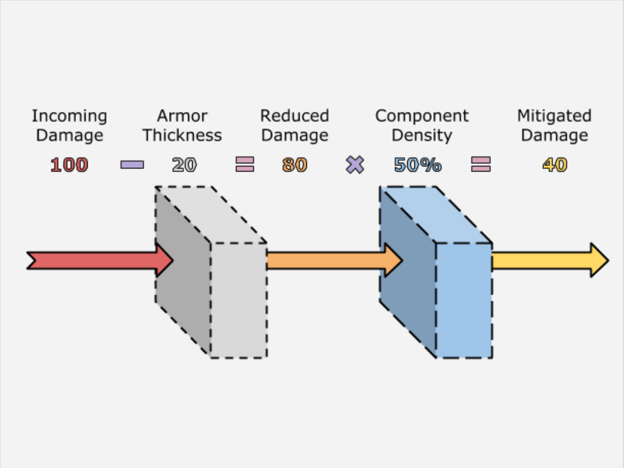
Device Examples

Device	Description
Active Scanner	Increases Sensor Power and Signature
Jump Jets	Allows a Mech to move instantly and indirectly
Liquid Helium Injector	Instantly Removes All Heat
Energy Shield	Diverts Damage to Heat While Active
Enhanced Pnuematics	Temporary Speed Increase
Cluster Mine	Group of Mines in an area
Concealed Mine	Mine with High Concealment
Active Sensors	Placed Sensor with High Detection but Low Concealment
Stealth Sensors	Placed Sensor with Low Detection but High Concealment
Nuke	Calls down a devastating orbital strike

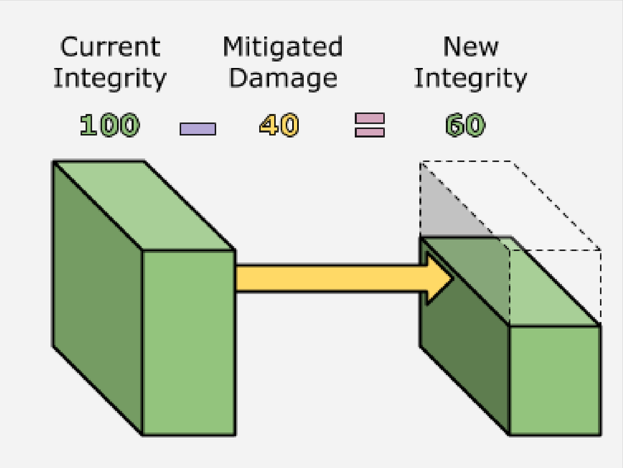
Combat Systems

The combat system is designed to be simple in construction but create many types of emergent game-play through its simplicity. In this system there is no % chance to hit or % effectiveness. If a weapon strikes another Mech then it goes through the following process.

In brief, armor reduces damage based on its thickness, that reduced damage is then mitigated based on the density of the component it struck. That now mitigated damage value is used to decrease the integrity of the Mech. When a Mech's integrity reaches 0 it is destroyed.



Damage is reduced by Armor and Chassis

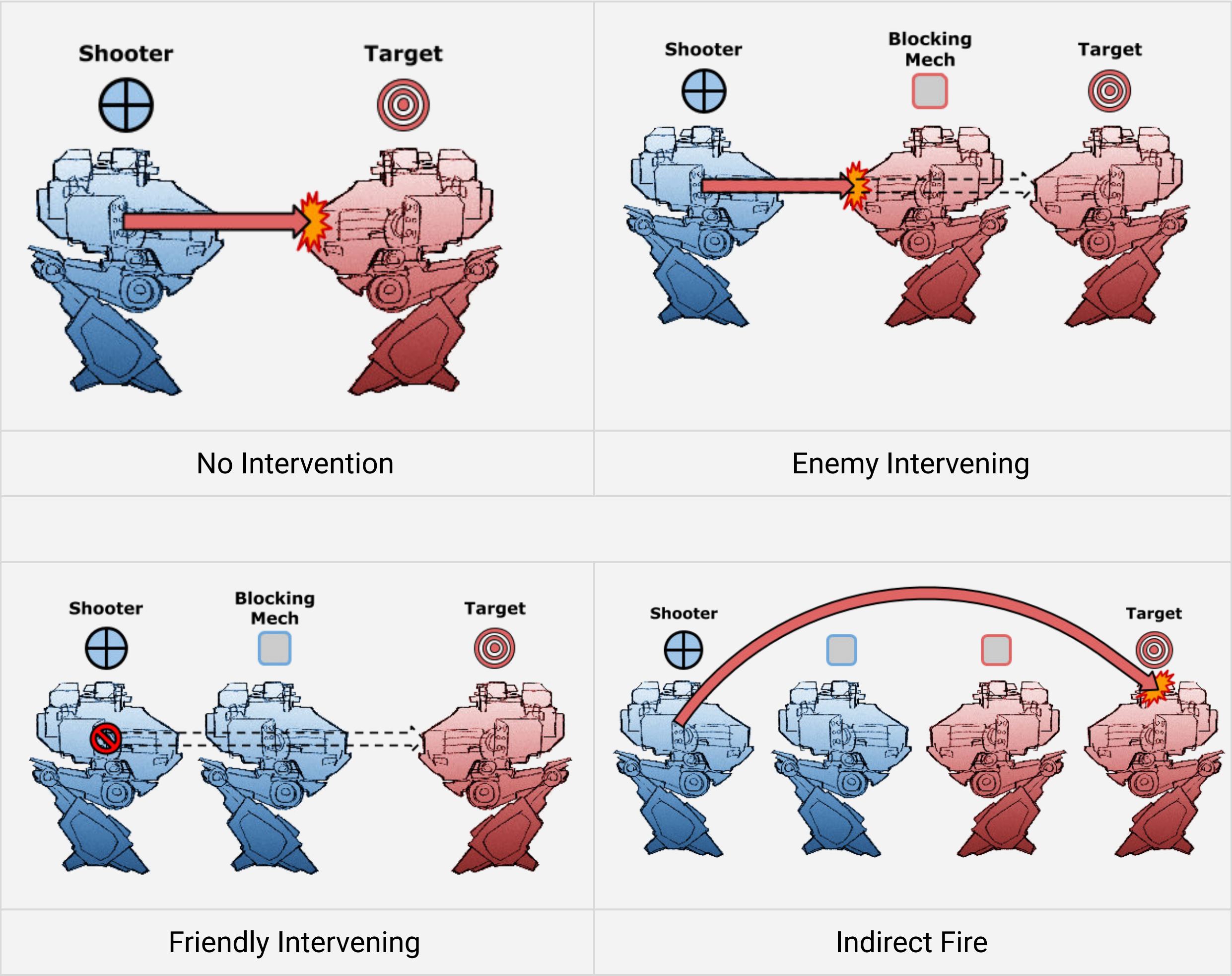


Resulting Value decreases Mech Integrity

Weapons

A weapon inflicts damage to blocks or entities in the world. There are 2 main types of weapons, direct fire weapons and indirect fire weapons. Direct fire weapons must have a clear line of sight to the target, if an enemy Mech is between the shooter and the target then the intervening Mech will be stuck instead. If a friendly Mech is between the shooter and the target, then the Mech will not fire at all.

- **Direct Fire:** Requires Line of Sight
- **Indirect Fire:** Does not require Line of Sight



No Intervention

Enemy Intervening

Friendly Intervening

Indirect Fire

Heating & Cooling

Weapon usage is constrained by heat generation and cooling. Each weapon generates a certain amount of heat per cycle, and the Mech counters this by reducing a certain amount of heat per second.

Players can turn weapons on and off to micromanage their heat use or may also decide to build a "heat neutral" Mech whose weapons do not generate heat faster than the Mech can cool it. Pilot Perks may passively assist in cooling and some devices might have the ability to counter it as well.

If the heat generated by weapons/devices reaches a critical point the Mech has **overheated** and must **vent**. When venting a Mech is completely shut down. It is immobile, unresponsive to orders, and cannot fire or maneuver. In this state a Mech is extremely vulnerable and players should not let their Mechs ever enter this state during combat. Players may also manually initiate a vent procedure to clear all residual heat. This is best done well outside of combat.

Venting removes all heat on the Mech and takes a certain amount of time regardless of the actual amount of heat being vented. This encourages players to "walk the line" and not vent too frequently.

Armor

Armor reduces the amount of damage a weapon can inflict by a flat amount. This allows more heavily armored Mechs to be nearly impervious to light weapons with low amounts of damage no matter how quickly they can fire.

There are 3 types of armor:

Type	Effectiveness	Description
Plate	Low - Moderate	Maintains Constant Effectiveness for Full Battle
Ablative Armor	Moderate - High	Loses Effectiveness as Damage is Taken
Reactive Armor	High - Very High	Limited Charges - 0 Effectiveness when Gone

Due to the name and function of the armor, each type is useful in different circumstances. Standard Plate Armor is good for Mechs and Zones that will take prolonged damage over the course of an battle. Ablative armor is best used in areas that will take some, but not very prolonged, damage during a battle. Finally, reactive armor is useful against rare but incredibly heavy damage such as direct artillery strikes.

Each armor type comes in several versions of ascending heaviness, with heavier armors being more effective but consuming more armor slots.

Armor costs weight and is installed independently on each damage zone so that players can customize the vulnerabilities of the Mech.

Armor Stats

Stat	Unit	Description
Size	v	Slot Size of Armor
Thickness	v	Amount of damage reduced
Durability	v	Health(Ablative), Charges(Reactive)

Chassis

Beyond armor a Chassis has inherent sustainability and damage resistance based on the type of metal the component is made from. Tougher metals are heavier, but provide more protection. This toughness is represented by Density and Integrity, and each component has its own value of each.

Density	A Rating that is converted into a Mitigation Value
Integrity	The Health Pool of a Component

Density

Each component has a density rating. This rating converts into a Mitigation Value that reduces the amount of damage inflicted on a Mech's integrity by a percentage of damage inflicted (to a minimum of 1).

Mitigation Multiplier	=	100 / (Density + 100)

Integrity

Integrity is the health of a Mech and its subsystems. The integrity of a sub-system is specified on the component it belongs to and the total integrity of a Mech is the sum of the sub-systems.

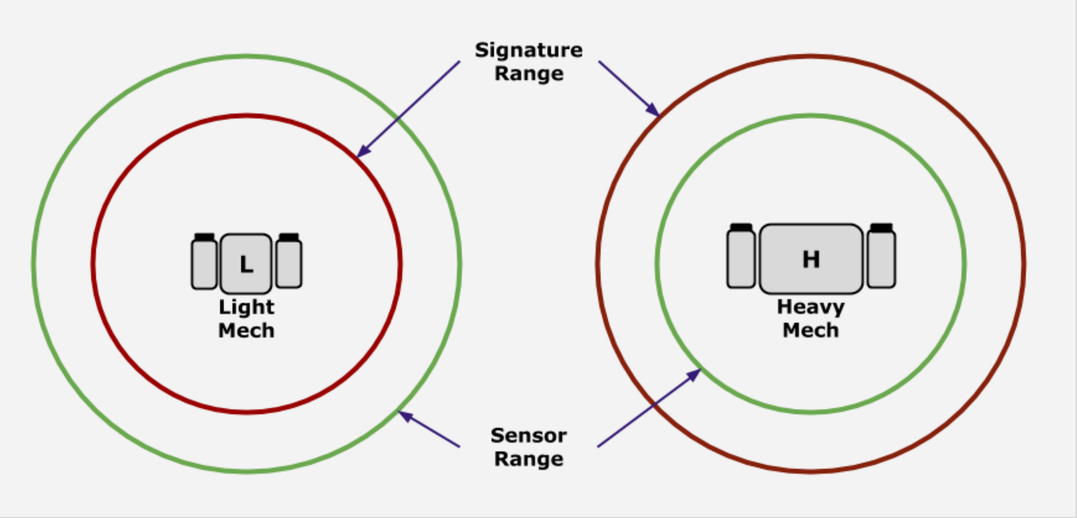
When a Mech is struck any portion of damage not stopped by its defenses will reduce the integrity of both the sub-system that was hit and the Mech overall. If a Mech is struck in an area with a destroyed sub-system then only overall integrity of the Mech is still reduced.

Detection System

Battlefield detection is determined by the concealment system. Unlike most vision systems, detecting a Mech with another Mech is two-sided. This is more involved than a simple fog of war with a sight radius, giving intelligence an increased role and making ambush strategies more effective.

Each Mech has sensors, which do the detecting, and a signature, which allows it to be detected. When an observer Mech's sensor range overlaps another Mech's signature range the observed Mech becomes visible on the map.

This can be visualized as two circles around each Mech, if those circles overlap then the sensing Mech can see the other Mech. It is very possible in this system for one to remain undetected while observing another Mech.



Sensor & Signature Ranges

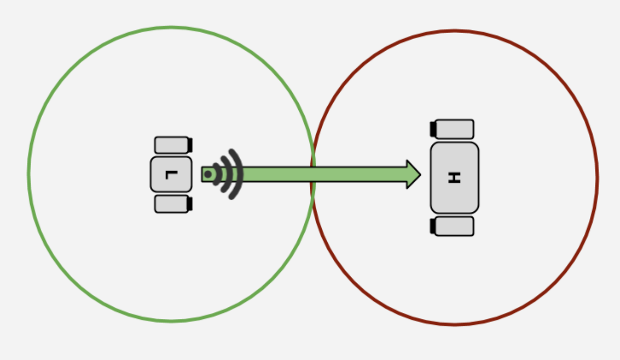
Sensors

Sensors are a single stat but represent several detection methods such as motion sensing, heat signatures, RADAR cross-sections, and normal optics.

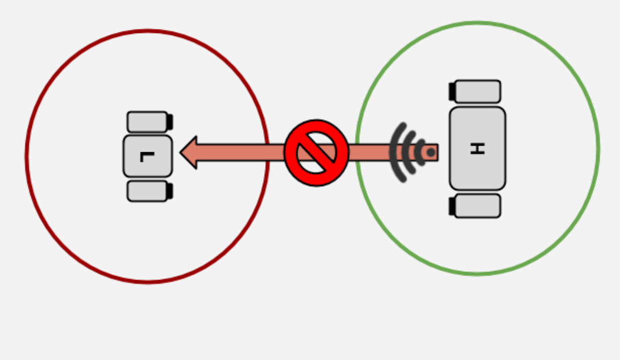
Base sensor range is a stat on the Torso of a Mech. This can be increased passively with Pilots and actively by using certain accessories.

There are 2 levels of detection. A **contact** appears first, at the edge of a Mech's sensor radius. A contact is indicated by a simple blip and contains no further information. It only indicates to the player that "something" is present but not what.

At a certain threshold a contact enters **identify range**. An identified contact reveals the entire Mech and its status.



Light Mech Observes Heavy Mech



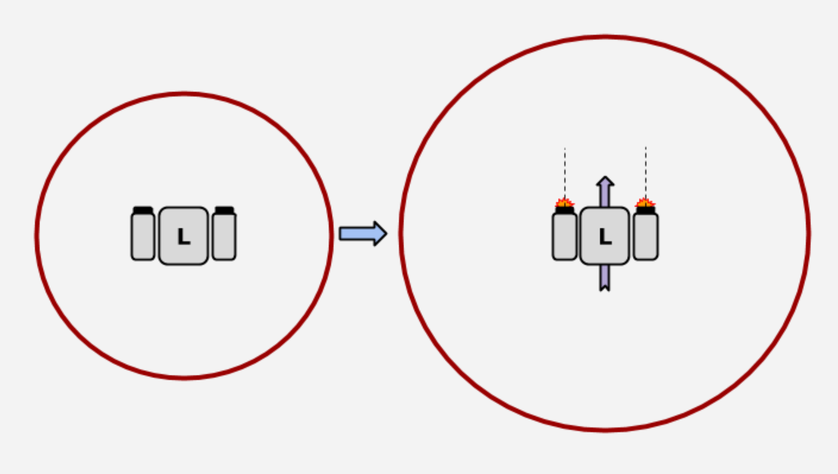
Same Distance:
Heavy Mech does not Observe Light Mech

Signature

A Mech's base signature is determined by its weight as the larger the Mech the easier it is to detect. The base signature generated by weight represents the radius of a Mech when standing still on open terrain and not firing any weapons or using any devices.

Performing actions can increase a Mech's signature, making it more noticeable. Moving increases a Mech's signature as does the number and size of the weapons being fired matters, with each weapon increasing signature based on its size.

Finally the current heat percentage of a Mech will also make it easier to detect. Very hot Mechs are easy to spot in the infrared spectrum.



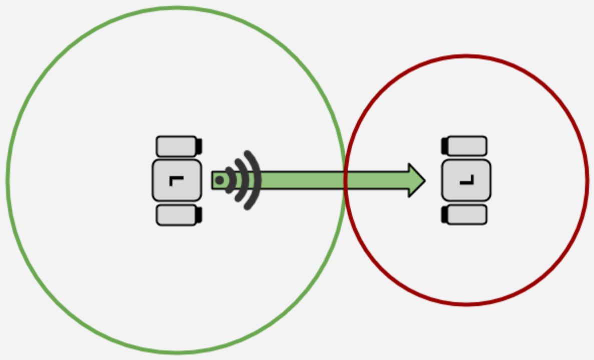
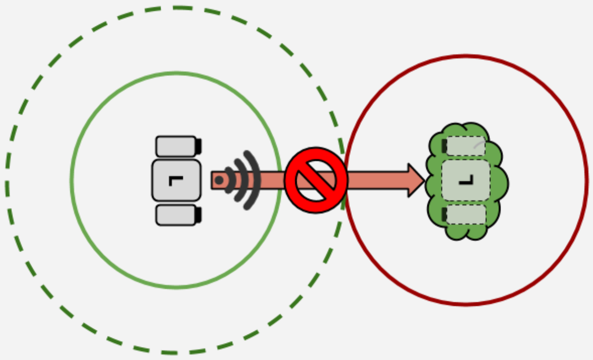
Moving & Firing Increase Signature

Concealment

By using terrain modifiers and devices a Mech can actively conceal itself. Concealment can work in two ways, either by reducing a Mech's current signature or reducing the effectiveness of the enemy's sensors.

By allowing concealment to foil normal sensor operation, a Mech is able to remain undetected even if it is within the enemy sensor range. Springing a trap from such a close range can make ambushes very deadly.

Concealment modifiers stack multiplicatively and do not increase the effectiveness of enemy sensors, they only decrease them. This one way street and multiplicative stacking are to keep the system relatively simple and understandable to players.

	
Without Concealment - Detected	With Concealment (Same Signature) - Not Detected

Calculating Detection

To calculate the detection, and what kind of detection, we use the following process.

First, a detection check is triggered when I sensors radius intersect an enemy signature. When this happens, the effective sensor and effective signature range, taking modifiers into account, are calculated.

Effective Sensors	=	Sensor Range * (1 - Π(Modifiers))
Effective Signature	=	Current Signature * (1 - Π(Modifiers))

The Effective Sensor range and Effective Signature range is then added together and the current distance between the 2 Mechs is subtracted from it generating a Raw Signal. The Raw Signal is then divided by the Effective Sensor Range to determine the Signal Strength.

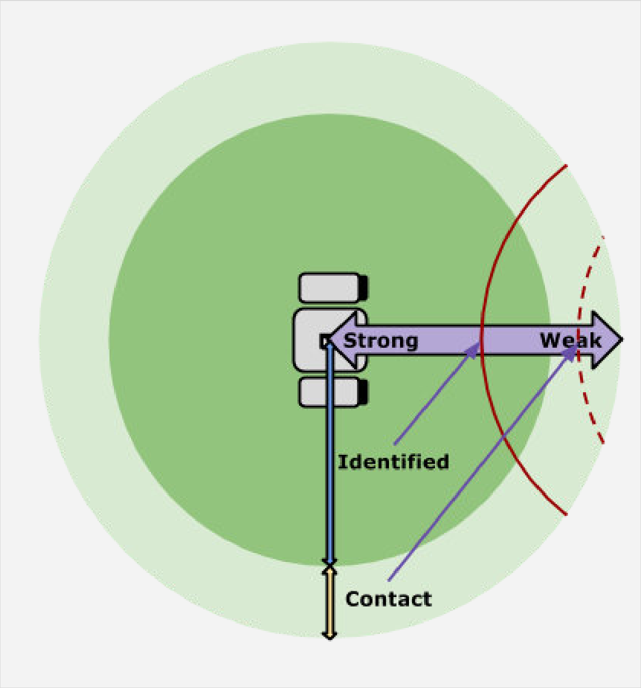
Raw Signal	=	Effective Sensors + Effective Signature - Distance
Signal Strength	=	Raw Signal / Effective Sensors

or

Signal Strength	=	((Sensors * (1 - Π(Modifiers))) + (Signature * (1 - Π(Modifiers))) - Distance) / (Sensors * (1 - Π(Modifiers)))

Signal Strength value is used to determine whether a Mech detects another Mech and at what level. If Signal Strength is less than 0, then the Mech is not detected, if the Signal Strength is between 0 and the less than the Identify Threshold, then it shows up as a contact, and finally if the Signal Strength is greater than or equal to the Identify Threshold it is fully identified.

No Detection	Signal Strength < 0
Contact	0 <= Signal Strength < Identify Threshold
Identified	Identify Threshold <= Signal Strength



Signal Strength Diagram

Environmental Effects

Environmental Effects are stat modifiers that exist on the Battle Map. These can manifest in one of two ways; globally or locally.

Global Effects are present everywhere on the map at the same time. For example, maps such as an ice planet or a fire planet would both have modifiers that affect heat and cooling throughout the entire map. The player will get a briefing about the nature of a planet before they enter a mission so they can customize their Mechs to take advantage of, or mitigate, any environmental effects that might be present.

Local effects behave in a similar manner, but are associated with specific points on a battlefield and are not universally present. For example, a pool of water on the battlefield would increase the cooling ability of any Mech standing in it, but that effect ceases when they exit the water. Global and local environmental effects stack, so standing next to a lava flow on Io would increase heat generation significantly, firstly just from being on Io and secondly from the local effect of the lava flow.

Environmental Effects Examples

Effect	Description	Global	Local
Cold	Increases Cooling	Yes	Yes
Hot	Increases Heat Generation	Yes	Yes
Acid Rain	Reduces Armor Effectiveness	Yes	No
Electrical Storm	Impairs Sensors	Yes	Yes
High Gravity	Reduces Movement	Yes	No
Cover	Increases Concealment	No	Yes

Pilots

Pilots provide passive perks to Mech operation that come in the form of stat bonus or special rules. Although a Mech can operate without a Pilot, it will be severely inhibited without access to these perks and bonuses. Therefore Pilots are a critical subsystem, worthy of targeting and protection.

Rather than gain flat experience which the player spends like a currency, pilots upgrade via a discrete achievement style system. This way the perks a Pilot gains are directly related to their battlefield actions rather than the player arbitrarily picking them. For example, a Pilot might gain the "Artillery Commander" perk, which increases Artillery Damage, after successfully killing 5 Mechs with Artillery, or perhaps after doing a certain amount of damage with Artillery weapons overall.

Perks have several levels with increasing difficult requirements as a form of vertical progression. A pilot may only have a limited number of different perks, but if she gains one the player does not not desire for a particular build they can be “forgotten", making room for new perks.

Perk Examples

Perk	Effect	Action	Tier 1	Tier 2	Tier 3
Artillery Commander	Increased Artillery Damage	Destroy Mechs with Artillery	10	100	1000
Survivor	Increased Chassis Integrity	Endure Battles without being destroyed	10	100	1000
Ghost	Does not Register as Contact	Spend a total of x Minutes inside an enemy's Sensor range without being detected	30	300	3000
Technician	Increased Sub-system integrity	Endure Battles with all subsystems intact	10	100	1000

Observer	Increased Sensor Range	Discover Hidden Mechs, Mines, or Sensors	10	100	1000
Engineer	Increased Chassis Toughness	Absorb 10k Damage in a Battle Without Being Destroyed	10	100	1000

Research

Research allows for certain classes of components, weapons, armor, and devices to be used when constructing a Mech, and also unlocks that equipment for purchase in the shop.

Research Points are the currency used by players for this unlocking process. They are gained as a reward for missions, and other activities, much like normal currency is. Those research points are spent into a component, weapon type, armor type, or device type which then allows anything in that class of component or equipment to be used.

Small differences between components and equipment can still exist, but if they belong to the same unlocked class the player is able to freely swap between them.

Components and equipment unlocks are not linked in a tree structure. The players may unlock anything, at any time, provided they have the adequate number of research points to spend.

The only exception to this perfectly flat research system are research Tiers. Each class of equipment has 3 Tiers, with each tier being significantly more expensive than the previous one to unlock. Tiers represent vertical progression through the game, whereby players unlock more power. In order to advance to tier 2, a player must unlock tier 1 and so on. The weight class system in matchmaking will ensure that players are matched appropriately for their tech level.

Research is divided up into 3 broad categories; Offense, Defense, and Recon. The Offensive section contains all weapons, components that focus on hardpoints and cooling, and devices that are offensive in nature such as the Nuke. The Defensive category has all armor, components that focus on fittings and density, as well as devices that are defensive in nature such as the shield. The recon section contains components that focus on lightness and sensors, as well as accessories involved in information or ambush warfare such as sensors and mines.

Category Examples

Object	**Offense**	**Defense**	**Recon**
Legs	↑-Weight : ↓-Defense	↑-Weight : ↑-Defense	↓-Weight : ↑-Speed

Torsos	↑-Cooling : ↑-Hardpoints	↑-Density : ↑-Fittings	↑-Sensors : ↑-Accessories
Arms	↑-Hardpoints	↑-Density	↓-Weight
Equipment	Weapons	Armor	Sensors
Devices	Nuke	Shield	Mines

Matchmaking

Matchmaking will use a combination rating and weight class system.

The combined maximum total weight of a Mech squad is used to determine that squad's weight class. It is important to use the total maximum allowed weight, and not the actual equipped weight, as we want players to construct the best possible Mechs they can within a weight class, and not try to "cut weight" to make it into a lower one.

Within each weight class players will preferably be matched using the Glicko-2 system. Glicko-2 is the basis for a variety of modern matchmaking systems including Microsoft Trueskill.

Weight Classes Example

Class	Tonnage
Light	-50
Middle	-100
Heavy	-200
Super Heavy	+200

The weight class system takes into account both time played and vertical progression by upping the total weight on higher tiered items and having stats cost more. At the same time, this allows some flexibility for higher tiered components and equipment in the lighter weight classes.

If needed or possible we will implement a "rookie" league where new players only play against other new players.