

Robot Factory Rulebook

Sam Hopkins

The Vrinski Accord gave each of the mining cartels their own chunk of the great beyond... so why is Titus 316 reporting unidentified robotic activity? No time for questions now: the only choice is to convert this peaceful mining operation into a war machine!

Number of players: 2

Length: 20-30 minutes.

1 Overview

Robot Factory is a strategy game for two players. The object is to destroy your opponent's factory by amassing an army of robots.

Because of the great distances involved in remote mining, cartels must send ahead instructions to their factories and wait hours to observe the outcome. Thus, the central mechanic of the game is *simultaneous worker placement*. Each turn players secretly make plans to develop their factories and then reveal and process the results at the same time.

2 Components

- 2 Factory boards
- 2 sets of 5 Robots (each represented by two six-sided dice)
- 2 decks of 25 Module cards
- 2 sets of 8 Upgrade cards
- 2 Basic Software cards
- 9 Advanced Software cards
- A divider
- Minerals in denominations: 50❖ (black); 100❖ (green); 500❖ (purple)
- Crystal cubes (yellow)
- Markers (red)

3 Game terms

Each player's **factory** is the base from which he gathers minerals, builds modules and upgrades, and produces robots. Players must protect their factories because if a factory's shields ever go below 1 it is destroyed and its owner loses.

Robots are workers who carry out all of the actions in a factory, as well as wage battle against the enemy factory. Each robot is represented by two dice, one which denotes its power (★) and the other its armor (▼). Power determines how much damage a robot does in battle, while armor determines its susceptibility to being destroyed by other robots. Both the power and armor of a robot are always integer values between 1 and 6. A player can only have five robots built at one time, but may rebuild robots if they are destroyed. Timing when to produce robots is vital to success.

Upgrades improve the robots produced by a factory. Upgrades may increase the power or armor of robots produced or lower their cost of production. For instance, Titanium Plating is an upgrade that reads "+1 ▼ -100 ♦", and thus it makes the robots you produce 100 ♦ cheaper to produce and come out with +1 ▼. There is no limit to the number of upgrades a factory can have. Importantly, upgrades only improve robots produced after the upgrades are built; they do not improve a player's already-existing robots.

Modules are extra rooms for a factory. Modules can have a wide range of effects, including modifying costs, dealing damage indirectly, allowing robots to take unique actions, and so on. Players each have an identical deck of module cards, but each turn have access to a limited number of these in their hands. A factory only has three slots for modules; if a player wishes to build a new module in excess of three, he may demolish one module and build another in its place. Choosing the right modules and the right order to build them in are key decisions in the course of the game.

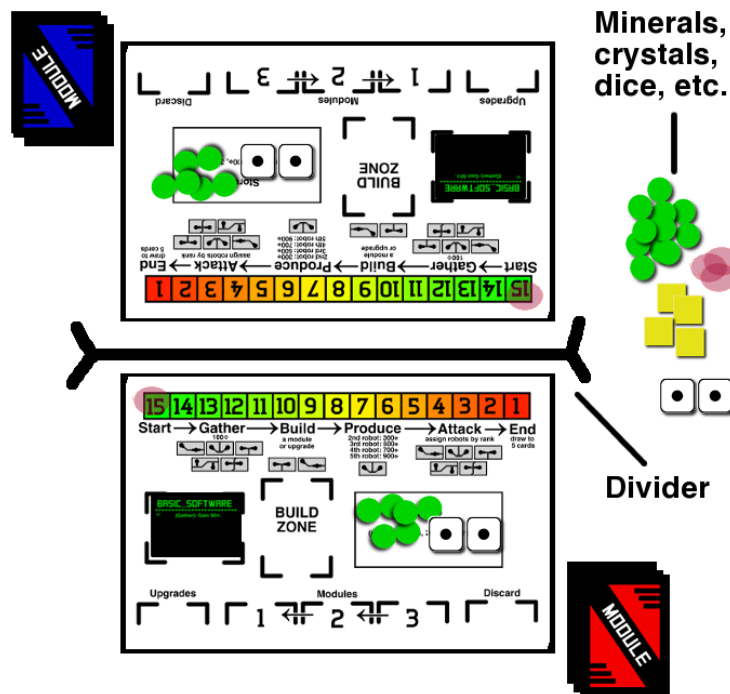
Software modify your factory like modules. However, unlike modules, you start the game with one and cannot build others throughout the game.

Minerals, symbol ♦, are the common currency of the game, used to build upgrades and modules and produce robots. **Crystals** are special resources which some modules yield that allow players to double the actions taken by their robots. Robots, minerals, and crystals are kept in a factory's **store**. The store can hold at most 5 robots, 2000♦, and 3 crystals.

Each player has a **draw pile** from which he draws new module cards. Each player's **hand** consists of those modules he can build, as well as all of his unbuilt upgrades. A player always has access to all his upgrade cards.

There is also a face-down **discard pile** where players put modules that have been destroyed or discarded. This discard pile should be removed from the playing area (so as not to confuse it with a deck).

4 Set-up



Give each player a factory board, a deck of module cards, and a set of upgrade cards. Each player shuffles his deck of module cards and draws five of them into his hand, setting aside the rest to become a draw pile. Give each player 600♣, one 1★1♥ robot, a Basic Software card, and a marker for factory damage. Players should put a marker on the 15 spot of the shield track of their factories. When their factory takes damage, they move this marker down that many points of damage. Players put the Basic Software card into their factory's software slot. Set aside the rest of the dice, minerals, crystals and markers to form a communal bank for these components. Place the divider between the two factories and begin the first turn of the game.

5 Gameflow

The game takes place over a series of turns where players make simultaneous decisions and then compute the results of those decisions.

Each turn is divided into two phases: programming, where players secretly assign their robots to actions, and execution, where all of the actions are resolved. Before the programming phase, place the divider between the two factories, and, once players have completed programming, remove the divider and move on to execution.

5.1 Programming

During programming, each player locks in the choice of actions he will take at execution. These choices a player can make are the following:

Assign robots. Each player may assign each robot he owns to some action in his factory by placing that robot on the appropriate assignment slot. This action can either be one of the default action built in to the factory (Produce, Build, Gather, or Attack) or an action granted by one of his factory's modules. Robots cannot be assigned to actions in excess of the number of slots available for that action. For instance, considering only the default actions of the factory, only one robot may be assigned to produce, and only two to build, while all five may gather or attack. When assigning robots to attack, robots must be assigned in order of their rank, meaning that the first slot must be occupied by the highest ranked attacking robot, the next slot by the next highest ranked and so on. Ranking of robots is power first, and then armor. So, for example, a 3★2▼ robot is of higher rank than a 3★1▼ robot, and both of these are of higher rank than a 2★5▼ robot. A player may leave any number of his robots unassigned in the store.

Place cards in his build zone. When assigning robots to build modules or upgrades, a player also must choose the appropriate cards from his hand and place them in the build zone of his factory. Place cards in the order they are to be built, with the first to be built going on top.

Demolish modules. A player may set any module in his factory to be demolished by turning it facedown during programming. Facedown modules are inactive; their effects are ignored.

Choose to activate optional payment-based effects of modules. Some modules have payment selectors that allow the player to optionally choose to pay minerals or crystals for certain effects. During programming, a player may move a marker onto one of the options in a payment selector of one of his modules to indicate his choice for the turn: this may be either

“Don’t Pay” or “Pay 100♠,” etc.

Apply crystals. Crystals are special resources that allow a player to double the action of a robot. During programming, a player may place up to one crystal from his store on each robot he has assigned to an action, except for robots that have been assigned to attack. **Crystals cannot be used on the attack action**, but they can be used to double any other robot action. Crystals being applied to robots do not count as being in a factory’s store.

Discard module cards from his hand. Finally, during programming, a player may place any number of modules from his hand face-down into the area marked “Discard” on his factory. At the end of execution he will draw up to 5 module cards in hand, so this is a way for him to get new cards.

5.2 Execution

After both players have made all their choices for the turn during programming, the divider is removed and execution begins. The execution phase is broken into six steps that are resolved in order. Within any individual step, all effects occur simultaneously unless otherwise noted. *Example: If a module is destroyed in the same step that it has some effect, that effect will still happen.* The steps are the following:

The first step is **Start**, where modules or software may have effects.

The next step is **Gather**. During this step, each robot a player has assigned to gather will yield 100♠ in that player’s store.

The next step is **Build**. During this step, first resolve all demolitions by destroying all facedown modules. Whenever a player’s module is destroyed he moves it to his discard pile and slides his other modules over so they occupy the lowest indexed slots. Then, each robot a player has assigned to build will give the player a build action, allowing him to build a single module or upgrade. Players build the cards in their build zones sequentially from top to bottom. Process each build as follows:

- Verify that the player has a build action for the build. If he lacks a build action, the build is canceled.
- Verify that the player has enough minerals in his store to pay for the build. If he lacks the appropriate minerals, the build is canceled.
- If the player is building a module, verify that the player has an empty slot in his factor for the module. If he lacks this slot, the build is canceled.

If a player's build is canceled, he returns that card to his hand. Otherwise, he spends the appropriate minerals (returning them to the communal bank), uses one of his build action and builds the card. If he built an upgrade, he places it into his factory's upgrade pile (where order is unimportant). If he build a modules, he places it in his factory's lowest indexed empty module slot. Note that the order of building can matter because modules are active as soon as they are built. *Example #1: Sam builds a Workshop and then a Gamma Cannon on the same turn. The cost reduction from the Workshop is applied to the Gamma Cannon. Example #2: Sam builds a Forge and then a Gamma Cannon on the same turn. The ability from the Forge is activated, causing another Gamma Cannon to be installed.*

The next step is **Produce**. During this step each robot a player has assigned to produce will yield a new robot for that player, at his factory's current level of upgrades, and at a cost modified by his upgrades and modules. By default, the robot produced is $1\star 1\blacktriangledown$, and the cost to produce a robot depends on the number of existing robots:

Number of existing robots	Cost to produce next robot
1	300 \blacklozenge
2	500 \blacklozenge
3	700 \blacklozenge
4	900 \blacklozenge

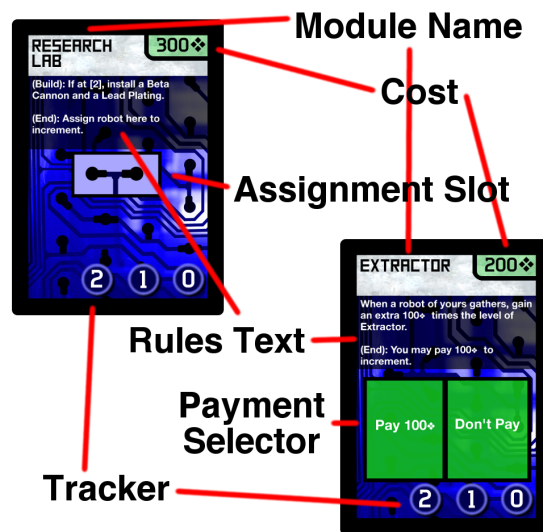
The stats of a robot max out at $6\star 6\blacktriangledown$; if a produced robot's \star would be greater than 6, it is 6 instead, and similiary for \blacktriangledown . *Example: Sam has as upgrades two Beta Cannons (which each give $+2\star$) and two Lead Platings (which each give $+1\star/+2\blacktriangledown$); a robot he produced would be a $6\star 5\blacktriangledown$.* If a player produces more than robot in a single step, he pays for each in sequence at the appropriate cost. *Example: Sam, who has two robots, assigns both to produce. He pays $500\blacklozenge + 700\blacklozenge = 1200\blacklozenge$ to produce two new robots.* If a player who assigned a robot to build cannot afford to produce that robot, he skips its production and pays nothing. Recall that a player can never have more than five robots. Produced robots go to the store.

The next step is **Attack**. During this step, opposing robots in the same slot fight one another. Recall that robots must be assigned to attack by rank; if they are not, rearrange the attacking robots so that they are ordered by rank. Thus the highest ranked attacking robots will fight each other, and so on. Resolving fights between two robots is simple. Suppose robot A is fighting robot B. If A has power strictly greater than the armor of B, reduce the armor of B by one; if B has power strictly greater than the armor of A, reduce the armor of A by one. If a robot's armor would be reduced below 1,

instead it is destroyed. Thus both, one, or none of the robots in a fight may be destroyed. *Example #1: Sam assigns his 3★2▼ robot to attack and Cole assigns his 4★1▼ robot to attack. Sam's robot becomes 3★1▼ and Cole's robot is destroyed.* If a robot is assigned to attack and has no opposing robot in the same slot, it deals damage equal to its power to the opposing factory instead. *Example #2: Sam assigns his 3★2▼ robot and his 2★1▼ to attack and Cole assigns his 2★3▼ robot to attack. Sam's 3★2▼ robot will fight Cole's 2★3▼ robot; the result of this fight is that Sam's 3★2▼ robot becomes a 3★1▼ robot, while Cole's 2★3▼ robot is unchanged. Sam's 2★1▼ will do 2 damage to Cole's factory.*

The next step is **End**, where modules or software may have effects, and where cleanup occurs. During cleanup, each player draws enough cards from his draw pile so that he has 5 module cards in hand. If there are not enough cards in the draw pile, he draws as many as he can. Note that **a player's discard pile is never shuffled back into his draw pile**. Also during cleanup all crystals used to double robot actions are returned to the communal bank, and modules that have been destroyed or discarded get moved to the face-down discard pile.

6 Module and software effects



Some modules have effects that occur automatically at various steps of execution. Treat all these actions as occurring simultaneously with each

other and with any default actions that occur during that step of execution. For instance Auto-Miner says “(Gather): Gain 100❖.” This effect happens at the same time as robots gather minerals.

Some modules destroy themselves or other modules. Whenever a player’s module is destroyed, he moves it to his discard pile and slides his other modules over so they occupy the lowest indexed slots. If a module is destroyed at the same step that it would have an effect, it does have that effect and is then destroyed. If multiple module-destroying effects activate in the same step, carry them out sequentially in the way that maximizes the number of destroyed modules. *Example: Cole’s Forcefield destroys itself during the End step, but the module-destroying effect from Sam’s Explosives Depot also activates during this step. The result is that the Forcefield as well as Cole’s module in his second slot will be destroyed.*

Some modules install upgrades. A player installs an upgrade by moving it from his hand to his factory’s upgrade pile. Installing does not count as building. If a player is instructed to install an upgrade he has no more copies of in his hand, he does nothing.

Some modules let you assign robots to them to activate an effect. During programming, these slots are treated exactly like the default slots of the factory. Again, you can only have a number of robots assigned to a module equal to the number of slots on that module. The module will indicate in which step that the assignment effect occurs.

Some modules have trackers. When a module with a tracker is built, immediately place a marker on its lowest value. To increment such a module is to move its marker one up on the track. Note that you cannot move the marker further than the maximum value on the track.

Some modules have payment selectors that let you optionally pay minerals or crystals to activate an effect. When a module with a payment selector is built, immediately place a marker on the “Don’t Pay” option of this module. During programming, a player may move the marker onto any option. The module will indicate in which step of execution the payment effect takes place in; during this step, if he selected the payment option, the player pays the appropriate resources and activates the resulting effect. If a player does not have enough resources in his store to pay for all the effects he has selected to activate in a given step, he pays nothing and the effects do not activate.

When a robot’s action effect is doubled by a crystal, treat this effect as happening twice in sequence.

7 Game end

The game ends immediately after any step of execution where one factory has 0 shields; the player whose factory lost all its shields loses. If both factories go to 0 shields in the same step the game ends in a draw.

8 Software

You start the game with one software card and cannot build any others. They do not occupy module slots and cannot be destroyed. We recommend players play their first game with only the Basic Software cards. Software have effects just like modules: a player with Basic Software will automatically gain 50♦ each gather step.

After players have a hang for the game, we recommend they play with Advanced Software instead of Basic Software. At set-up, shuffle the deck of Advanced Software cards and deal three face down to each player. Players secretly choose one of the three to play with and simultaneously reveal their choices. They put the chosen software cards into the software slots of their factories (and reveal this choice before the first turn of the game).

9 Card FAQ

Armory - This can never increase a robot's stats above 6★6▼.

Auto-Miner - This does not count as gathering (for purposes of modules like Extractor or Radiological Facility).

Explosives Depot - If at [2], this will reset to [0] even if the enemy has no modules.

Extractor - The “level” of Extractor means the number on its tracker which is currently being marked; e.g., immediately after being built, the Extractor will be at level zero and hence will cause no extra minerals to be gained from gathering. A robot whose gathering action is doubled via a crystal counts as gathering “twice” for purposed of Extractor, and consequently the extra minerals from Extractor will also be gained twice in this case.

Forge - Installing does not count as building.

Fortitude - Effects that remove damage from a factory will allow a player with Fortitude to go back above 15 shields.

Fusion Core - This counts as the robot itself dealing damage. But it does not count as the robot attacking.

Ingenuity - Installing does not count as building.

Jetpack Hangar - This counts as the robot itself dealing damage. But it does not count as the robot attacking. A player may double this effect with a crystal. If this effect is doubled, it counts as the robot dealing damage twice (e.g., for purposes of cards like Explosives Depot or Aggression).

Laser Chamber - This payment selector allows payment of *either* 300♦ or a crystal to activate its effect. If he wishes to activate the effect, the player must indicate which of these payment options he is selecting during programming.

Missile Silo - This does not count as a robot dealing damage.

Rocket Launcher - Up to three robots may be assigned to Rocket Launcher, and each one assigned will cause 2 damage to be dealt to the enemy's factory. This does not count as the robot dealing damage or attacking. A player may double this effect with a crystal.

Radiological Facility - If a crystal has been applied to a robot assigned to gather, that robot still only counts as a single robot assigned to gather for purposes of Radiological Facility.

Refinery - This does not count as gathering (for purposes of modules like Extractor or Radiological Facility).

Research Lab - Installing does not count as building.

Salvage Bay - The gaining is not tied to the destroy. If the robot is not destroyed, the player will still gain the minerals. For example, if a player doubles this action via a crystal, he will gain 800♦.

Security Center - The Security Center will only ever decrease a robot's ★ by 1 each turn, even if that robot dealt damage multiple times that turn.

Strength - This can never increase a robot's stats above 6★6▼.

Training Room - This can never increase a robot's stats above 6★6▼.