# GTX (Omicron)

*From: JasonP - Under Directory: Game Design*

This system is a high crunch, simulation-minded engine for roleplaying wicked cool transformers, inspired by Transformers. GTX is short for: **Gaming Transformers eXperiment**. The name of this working version of the game is Omicron, denoted for both its common usage in science fiction and that fact that it means 70 in greek, which is the amount of kilograms of alpha nanites each machine starts with.

## Goals

The ideal system here is complicated, but not impossible to handle. It will allow you to use a point-buy system to design and create your transformer, with stats for all features it may have as a machine.

## Die Codes

GTX uses a progressive array of die codes to value: ability, worth, and strength. Here are the die codes the game uses in their order, with standardized base costs. The higher the step and die code, the more value you have in a given rating, and the die cost is higher. The primary cost of a die is its number of faces, so that a d6 costs 6. You pay this for a die bought for primary uses. The secondary and tertiary costs are used for buying additional die codes for a given module.

| Step | DIE CODE | COSTS: | Primary | Secondary | Tertiary |
|---|---|---|---|---|---|
| 1 | d3 | | 3 | 2 | * |
| 2 | d4 | | 4 | 3 | 1 |
| 3 | d6 | | 6 | 4 | 2 |
| 4 | d8 | | 8 | 6 | 3 |
| 5 | d10 | | 10 | 7 | 4 |
| 6 | d12 | | 12 | 8 | 5 |

## Modules

Each transformer is a system of modules, a network that integrates a core module and then interfaces with additional modules. Beneath this level, they are made up of something just call alpha, which are living and replicating nanite machines too small to see. This game is about Omicron machines, and these start out with a total of 70 alpha and their core module is always a transforming gyro. In their world, they are the most evolved machine lifeform.

Modules are bought based on a total amount of alpha. This amount is determined by a somewhat complex system of rules. However it can be summarized as so:

- **Cost** = (Primary cost for Use Die Code x Use Multiplier + Secondary cost for Versatility Die Code x Use Multiplier) / (Power to use + 1).

Lets provide an example. A simple drive gyro has a use multiplier of x2. To buy a simple drive gyro of d8 with 1 power to use: d8 has a primary cost of 8 and we multiply that times the use multiplier of x2 = 16. We have no versatility die, so we cost is 0 x 2 = 0. Finally we put the cost together as (16 + 0) / 2 = 8. So its 8 alpha for that simple drive gyro.

# Primary Uses

Each module is designed for a primary use. There are many of these built into this basic form the of game: **gyro**, **threat**, **mobility**, **response**, **emitter**, **intelligence**, **sensor**, **control**. The first: gyro, is the most complex and the core device of any omicron machine. The next three: threat, mobility, and response are all core devices they must at least possess one of in order to function. This means the first four comprise a core set of uses backed by one module each: gyro, threat, mobility, and response. Lets define what these uses mean and their use multiplier:

- **Gyro**: This is the complex and powerful mechanism at the very core of an omicron machine. Gyro splits into two die codes: Power and Pulse. It allows it to transform into one or more other non-human modes, and stores its energy. The use multiplier depends on type of transformation the gyro offers:
  - *Simple Drive*: This is a simple wheeled motor transformation (autobot, car type). Its use multiplier is x2, but you can only run on surfaces.
  - *Fluid Thrust*: This is a simple vector thrust transformation, allowing travel through mediums of gas and water. Its use multiplier is x3, but you can only operate in a fluid/gas of some type.
  - *Ion Thrust*: This is a high-energy thrust transformation, allowing travel pretty much anywhere regardless of medium or lack thereof. Its use multiplier is x4.
  - *Basic Shift*: This is a very simple transformation into a non-mobile form. Its use multiplier is x1, but you can't move under your own own in transformed mode. However, you may pay half-cost for up to 30 alpha of modules that only operate in that form.
- **Threat**: This is how a machine attacks, by using threat. However there are generally secondary purposes for threat too, as it is the most basic way the machine can influence its environment. Simply put, as a core module this is the machines strength and prowess in humanoid form. Threat splits into two die codes: Strength and Prowess.
- **Mobility**: This is how a machine moves, by using mobility. This is usually also assisted by the gyro in transformed modes (not shift types obviously). Mobility splits into two die codes: Speed and Agility.
- **Response**: This is how a machine resists injury and attack, by using response. Response splits into two die codes: Absorb and Deflect.
- **Emitter**: This is how a machine projects and emits energy. The exact purpose of each emitter has to be created by its designer and discussed with the group. Each emitter splits into two die codes: force and operation.
- **Intelligence**: This is how a machine gains intellect. Each intelligence is a coexisting mind within the machine, not unlike what we would call an AI in computer terms. Each intelligence splits into two die codes: Savvy and Recall.
- **Sensor**: This is how a machine detects and interprets the world around it. The exact range and purpose of each sensor has to be created by its designer and discussed with the group. Each sensor splits into two die codes: Range and Detail.
- **Control**: This is how a machine coordinates its modules. The network of a machine can manage its modules, but a control lets them work together in an enhanced way. When a control links a system you can merge dice upward, and gain other additional abilities. The step of a controls

die code is most important, as it limits the amount of modules it can manage. As such, a control is never a d3 module and must be bought at d4 or higher.

## Core Modules

## Modules & Features