

Carnage Heart
(fill in with sweet ascii art later)

Table of Contents

1. Overview	[orvw]
2. Controls	[ctrs]
3. Story.	[stry]
4. Main Menu.	[mamu]
A. Start Game	
B. Load Game	
C. Vs. Battle	
5. Difficulty Campaigns	[dfcp]
A. Europa (practice)	
B. Europa	
C. Ganymede	
D. Callisto	
6. How to Play Campaign Mode	
A. Tactics	
B. Design	
C. Negotiation	
D. System	
E. End Turn	
7. OKE Overview	
A. Two Legged Type	
a. Prowler	
b. Jujuman	
c. Vypor	
B. Four Legged Type	
a. Livewire	
b. Manticore	
c. Rogue	
C. Tank	
a. Kouger	
b. Ronin	
c. Jackhammer	
D. Flying	
a. Valiant	
b. Iron Death	
c. Raptor	
8. Hardware	
A. Engine	
B. Main Weapon	
C. Sub Weapon	

- D. CPU
- E. Fuel
- F. Armor
- G. Optional
- I. Paint
- 9. Software
 - A. Overview
 - B. Chip Explanation
 - C. Logic flow overview
 - D. General Strategies
- 10. Hardware Complete list

=====

Intent of this FAQ

This article is designed to be a replacement/ addendum to the original manuals that are not available to most people who would want to try and play the game now. It's a game with a very steep learning curve and would be very difficult to enjoy without some explanation. Artdink was so convinced of its complexity they packaged a second whole CD with the game that only contained a fmv overview of how to play the game, and two seperate manuals. The first explaining how to play the game and the second going into great depth of how to design the AI and build tactics. The AI manual is 58 pages long alone so needless to say they wanted people to understand the game to be able to enjoy it.

This is not a step by step walk through and I will not be giving you prebuilt OKE designs so you can beat the game easily. I feel that to play this game you really must experience the joy of developing a winning strategy and implementing it yourself. To remove that from you by simply giving you winning, proven AI designs would ruin that joy. Much like solving a puzzle for you. Instead this is here to explain all the details and hidden rules of the game so that you may enjoy it to a fuller extent.

1. Overview

[orvw]

Carnage Heart is a mech themed, turn based strategy game where the mechs are controlled by a player designed AI system. Each map starts with a certain amount of bases in your control and the goal is to take control of the enemies bases. This is done by capturing them with teams of OKE (OverKill Engines, aka the mechs). You, as commander, must buy and research hardware, design and program OKE units, create them, and deploy them effectively on the field. When two teams meet on the field they fight until either one team is destroyed or the timer expires.

The OKE teams are controlled in battle by the programming you supply them. So a great deal of development is designing the AI program that controls each OKE before they are sent into battle. Honestly the main strategy in the game is in the surprisingly deep AI development since the other aspects of the strategy are much more shallow when compared to contemporary games. The core strength of your strategy lies solely in how well you can develop the fighting ability of your OKE teams. The game provides some prebuilt OKE designs and will auto fill in a AI program if you want but it's incredibly weak and will not last very long in battle. The core of the game is simply designing AI to fight in teams against other teams. If you can create a strong team then winning is inevitable. If you cannot, then you are doomed.

=====

2. Controls

The game is menu based so most interactions using menus. Screens are either rigid menus or free moving cursor.

Direction Pad - moves the cursor around or moves through menus. If you hold down Triangle will in cursor mode it will quick move to buttons, use to navigate quicker

Select - if you are testing your OKE it will open up the testing menu, otherwise it will open up the quit game menu

Start - if testing OKE it will cancel the test, otherwise it's not used.

Triangle - hold it down and use direction pad to quick move the cursor

Circle - cancel, if editing AI chip it will bring up the exit menu if everything is canceled out already

X - Select

Square - not used

L1 - in chip editing mode, hold down and use left/ right to change current chip red arrow direction

L2 - in chip editing mode, hold down and use left/ right to change current chip green arrow direction

- in Tactics mode it will cycle through the base list

R1 - hold down and use direction pad to move the chip area around

R2 - in chip edit mode brings up the chip menu

=====

3. Story

In the future, mining rights are being fought over on the moons of Jupiter. You command the forces of the World Federation against the private corporate forces of the Drakken group. You are not a corporate man but a simple soldier that might be in over his head.

It doesn't matter, the instructions have 4 pages of back story but all you need to know is you fight the enemy. Simple enough.

=====

4. Main Menu

A. Start Game

Starts a new single player campaign. Select the difficulty here.

B. Load Game

Load a previous game. A card can only hold 3 separate saves. It takes 5 units of space per save file, so pretty heavy save requirements.

C. Vs. Battle

Load up OKE units from save files and set teams to fight against each other. The stages are from the development test stages.

=====

Definitions of Terms

OKE - An acronym for Over Kill Engine, this games term for the mechs. Used in this FAQ as a general reference to any mech in the game.

AI - artificial intelligence, in this game referring to the programmed commands that control the OKEs in combat.

Card - a specific slot that different OKE designs are saved under, there are 28 card slots but only 25 that can be edited

Master Card - when you are satisfied with a card design you change it to be a Master Card so that it can be produced, once a card is set to master it cannot be edited

Unit - a single produced mastercard OKE, can be moved around between teams

Team - a group of one to three units that can be commanded to move across the map and enter combat. Units must be formed into teams before they can be move, attack, or defend. There are 32 total team slots available.

Combat - when two teams meet on the map or when a team either attacks or defends a base they enter a timed battle where they fight to either death or until time runs out. The units act solely based on their AI and the only player input is camera control

Battlefield - this is the arena that combat takes place in. These are square, flat areas filled with various obstacles.

Starter OKE - there are three predesigned cards, card 26, 27, and 28. They cannot be edited, they do not upgrade, and every base starts with one random starter OKE unit for each line it can handle. They are free but are generally so weak they are not worth using.

Campaign - there are four difficulty settings: practice, easy, medium, and hard. Each setting is a series of unique maps and enemies that can be played through. None of the designs transfers between campaigns and starts with nothing.

Turn - the game is played in turns consisting of two phases, strategy and action. The first phase is when all the negotiations, card designs, and team movement commands are completed. When you are satisfied with your planning you can end your turn and the second phase will begin. This is when all the team movement, combats, and base capture will take place. Other than hitting circle to move through the steps you do not get to directly control any part of the action phase. If at the end of the turn one side is in control of all bases then the map is over, otherwise the next turn begins.

Map - each campaign consists of a series of maps. Each map is a physical hex map that shows all bases, teams, terrain, and roads. Basically a map is a single stage.

Base - each map has various bases that you and the enemy operate from. These are where you produce units, form teams, and what you need to attack/ defend. Each map has the same win conditions which is take control of every base.

Level - every card design has a production level, and an assembly line must be upgraded to or higher than that level to produce that card. An increase in an assembly lines level will also increase its production amount per turn. An assembly line can be upgraded to a maximum of 9.

Line - every base contains assembly lines that can be used to produce different master cards or ammo. Every base starts with only two open assembly lines, but can support 3 to 6 total that can be purchased. Each open line can be upgraded by levels as well to produce units and ammo at a faster rate.

Process (PC) - every master card has a process amount that is the summation of it's individual hardware components. This total amount is how much is required to complete a new unit. Each line has a process per turn amount that is

determined by its level. For example if a master card requires 200 process and a line can produce 100/turn then that line can complete one unit in two turns, as long as you have the funds.

Funds - this is the money in the game. It is not every given a specific name in the game so it will be referred to as funds as much as I can remember to.

Hardware - an OKE is build from various types of hardware. Choosing the hardware is the first part of a card design.

Software - this is the AI design part of card design. It controls a unit in combat and is the most complicated part of the game.

Chip - an individual part of the software. There are 40 different types of chips available and how you combine them forms the AI itself.

Chipset - refers to the complete software build for a card

Chassis - there are 4 different types, and 3 designs of each type, 12 total chassis

Main Weapon - an OKE main weapon will be either shotguns, assault rifles, or lasers, the ammo capacity is very high and will be used to do the majority of damage.

Sub Weapon - there are 3 types of projectile, rockets, missiles and large missiles, and 2 types of mines, floating and scatter. Not every chassis can carry every time and the ammo capacity is determined by the chassis unlike main weapon ammo. The ammo is very low compared to the main weapon but the damage is usually much higher so special care should be taken to not waste them.

Fuel - one of the two commodities in the game, all units run on fuel and can be refilled at any base that has fuel reserves. A bases fuel supply is set at the beginning of a stage and it cannot be replenished. If a unit runs out of fuel in combat it will simply stop moving so care should be taken to not let that happen.

Ammo - the other commodity, this can and must be produced in lines so that when teams are formed they have ammo to use in battle. It is a universal ammo type, so no matter what weapons a unit carries ammo will resupply it. A team carries a supply of ammo but a unit can only carry into combat the amount the card design allotted by the hardware. At the end of a turn all units will refill their supplies from the team supplies if the team still has any. A team automatically fills up it's reserves if it enters a friendly base that has ammo.

=====

5. Difficulty Campaigns

There are 4 difficulty settings. Each one is a different campaign that has various differences that adjust how hard it is. The radius, mass, and density are just fun details. Level: difficulty lets you know the difficulty of the enemy OKE, the rate at which they are constructed (hard will build a lot more units every round), the rate they use better equipment, the amount of bases you have to capture, the amount you start with, the factory lines you have access to, ect.. Funds is the starting amount of money you have on turn 1. Maps is the amount of maps (stages) you have in the campaign.

A. Europa (practice)

Radius $1.6 \times 10^3 \text{ km}$
Mass $480 \times 10^{20} \text{ kg}$
Density 3.0 g/cm^3
Level: Easy
Funds: 50,000
1 map

Very easy, enemies are kouger tanks. You have a lot of funds, 1 base, 6 lines, a place to dip your toe in. This mode also has little pop ups that help explain what everything does. It's so easy that the starter OKE teams are strong enough to win this map just sent out to take over the enemy bases.

B. Europa

Radius $1.6 \times 10^3 \text{ km}$
Mass $480 \times 10^{20} \text{ kg}$
Density 3.0 g/cm^3
Level: Easy
Funds: 100,000
5 maps

The enemy units are not very hard yet. The maps are not hard yet, and the battle stages are minimally populated with objects.

C. Ganymede

Radius $2.6 \times 10^3 \text{ km}$
Mass $1482 \times 10^{20} \text{ kg}$
Density 1.9 g/cm^3
Level: Medium
Funds: 200,000
7 Maps

Difficulty is much higher from the first stage on. There are wider variety of enemies and terrain. The maps are harder and you have less resources.

D. Callisto

Radius $2.4 \times 10^3 \text{ km}$
Mass $1077 \times 10^{20} \text{ kg}$
Density 1.9 g/cm^3
Level: Hard
Funds: 300,000
8 maps

Very difficult. The enemies will use much more powerful tactics and weapons. The maps will start with you at a greater disadvantage. The enemy will create a lot of units every round. The combat stages will have a lot of obstacles that have to be avoided and maneuvered around. From the very first round you have to really understand the tactics and the design or you will have a hard time being successful in this campaign.

=====

6. How to Play Campaign Mode

Campaign mode is broken down into individual maps. The win conditions of each map is the same: to be in control of every base on the map at the end of the turn. You do not have to eliminate every enemy OKE team on the map so keep that in mind. Each map is turn based and a turn is not over until you select End Turn. Each turn is broken into two sections: Strategy and Movement.

Strategy

This is when you will do all of your OKE design, hardware research and purchasing, and on map unit building and deployment. Once you have done everything you want to you end your turn.

A. Tactics

This is where you deploy teams on the map, build and upgrade construction lines, build units and ammo, and view the current map conditions. You can view only a certain distance on the map around your bases and current teams on the map so you might not be able to see all of the enemy movements until they are close to your base or teams. There is no hidden parts of any maps or surprise bases, the whole map including terrain and bases are known at the start of a map. Enemy team movement is the only unknown.

You can direct your teams with several commands: move, attack, defend, patrol, and standby. Move simply commands a team to move to a specific hex, if that hex is an enemy base they attempt to capture it. Attack can only be directed at

enemy bases and the team will move across the map and attempt to capture it. Defend will cause a team to move to defend a selected bases from enemy teams. Patrol sends a team just out onto the map to wander. And standby stops a team, it is also the default condition of a team once it has moved to its designated hex.

The strategy here isn't very deep. You can just use the move command for every team deployment. Attack is basically the same action but you can only select enemy bases. Defend is not important since a team will always defend a base they are in if an enemy teams attacks (and sometimes this will move them away from a base to intercept but them miss and leave the base open to attack, missing the whole point). Patrol is useless because attacking is always a better option. And standby is the default.

A team will only move one hex per turn at all times.

One thing to talk about here is how the game handles map movement. All units under your control go first, then all of the enemy units. Once all moves and combats are resolved then the turn is over. Which team moves first is based on the team slot number. There are 32 slots available and no more teams can be active at any time. So team slot 1 performs its move and then it's combat if one is initiated, then team slot 2, and so forth. If two seperate teams are trying to move into the same hex the team with the lowest team slot number will move there first and the second team will move in a random direction afterward. That's the catch, they always move, even if it's away from the direction of their objective. This becomes an issue often when the map has pinch points of passage and you have teams from multiple bases trying to move through. They will bump into each other often and clog the path. Managing this becomes important in some maps and you need to be aware of it, or it will be incredibly frustrating watching the teams wander off instead of advancing cleanly.

Another thing to note is unless you micromanage the movement with constant move commands the pathfinding does its own thing. You can see visual representation of roads on the map and if given the freedom the pathfinding will always try to move via roads. Even if it takes longer to travel to the destination. Sometimes you will have to micromanage the movement to avoid this. Since the attack command only allows you to select bases using move is almost always preferred.

Lines

To produce units and ammo you need to instruct lines to build them. Each base starts with two open lines but can be expanded to the bases max available. Every base can house between 3 and 6 lines max. Extra lines must be purchased in your starting bases but captured bases will have whatever the cpu had developed when captured. Each line has a specific level that dictates how much it can produce in one turn and what that max hardware level it will allow. Every master card has a total process amount required to make it, a funds cost, and a minimum line level to produce. If a card is a level 3 then to produce it you must level up a line to level 3, have the funds available, and wait the amount of turns required to meet the process amount. Simple enough. You can only build master cards and if you try to edit a card after it has been

produced the game will warn you that it will scrap all units based on that card. The penalty of design errors is high.

Ammo is the only item that can be produced in the game besides OKE units. Ammo is stored in bases and is transferred to teams when they are formed or when they enter a base. Ammo in the game is handled less than intuitively so I'll try to explain best I can. Ammo Units, or AU, are used to refill individual units main and sub weapon ammo between turns. 1 AU is enough to fill a full team (3 OKEs) 1 time. Ammo Units are universal and can replenish any type or quantity of ammo. When a team is formed it will automatically add 10 AU to the team reserves and 1 AU to the team itself, for a total of 11 AU. You can adjust of reserve AU a team has from 0 to 50 AU in the Deploy Units menu. If a base has no ammo then the team will be formed with none and they will enter combat with 0 ammo and will most likely die. At the end of each turn if any of the units have used ammo in combat then 1 AU will be used to refill all of the team units ammo. A team will only refill the ammo at the end of a turn so if a team has to engage in multiple combats they will enter each combat with the ammo remaining from the last combat, and could easily run out of ammo. The system is mostly automatic but there are cases when you want to adjust the amount of ammo a team carries. It will always default to 10 reserve AU for a team if a team is in a base.

Fuel is the other resource in the game but it is handled a little different. At the beginning of every map each base has a preset amount of fuel in it but every turn it will automatically refine fuel at about 500L per turn. When you form a new unit from available units it will start with 1000L in reserve and enough to fill the tank of each unit (if enough fuel is available). For example if a each unit has a 50l tank, when 3 are formed into a team then 1150L of the bases supply. When a team enters combat they will carry into it the amount of the OKEs fuel tank and use it as they battle. If they run out of fuel then they will stop moving but will still be able to fire. Flying types will fall from the air and explode though so be careful about fuel allotment on those types. If a team faces multiple combats per turn they will enter subsequent combats with the remaining amount from the previous combat. Any units that survive the turn will fill up their tanks from the team reserves until those are depleted. When a team enters a base it will automatically fill up the team reserves to 1000L again. For the most part this is a fully automatic system and you will not need to pay attention to it. If you need to for tactical reasons you can edit the amount of fuel a team carries while it is in a base from the Tactics/Deploy Units menu. You can change the amount from 0 to 5000L if there is enough in the base supply of course. Even if a team has 0L in its reserve the units will still fill up if there is fuel available. It is also important to note that if a team has 0L of fuel and every unit is also on 0% of its fuel supply the team can still move across the map. It simply will not move in combat and probably die). Be careful though if you edit a teams fuel supply but do not deploy it, the next round the game will reset it to 1000L again.

You can actually use teams to carry supplies from one base to another if need be by simply filling the supply of a team up to its max of ammo and fuel and moving it to another base. Then either editing the number lower or removing all units from the team. For example if there is a base closer to the enemy but has little ammo and fuel you can load a team up with max supplies and send it there so you do not have to waste turns building ammo instead of units from the lines. Very helpful with those 3 line bases on the front lines.

B. Design

This is where you develop your oke cards to use in combat. You choose the hardware, design the software, and test it in special testing arenas until you are satisfied then you set a design to be a master card that can be produced. Each design is attached to a card slot. There are 28 card slot available but card 26-28 are fixed as the 3 starting unit designs and cannot be altered, updated, or deleted. Leaving you with 25 available slots to design in.

Hardware

This is where you choose the hardware that makes up the OKE design. There is a detailed section devoted to this later.

Software

After you have set the hardware this is where you develop the ai for the design. You will spend a lot of the game in this area so be one familiar with its control shortcuts. Great length and detail is further down in its own section. Suffice to say this is the part of the game that makes Carnage Heart what it is.

Testing

Once you have created hardware and software you can test it in simulated combat. The test card is always the card you are currently editing, but you can add up to two allies and also choose up to 3 enemies to face. You can also choose one of five test battlefields. They have various amounts of obstacle placement but they are are symmetrical. Map 2 is useful because it is simple a blank battlefield with no obstructions at all. Very useful for testing. You can only choose cards to test with that are in you card inventory. So unless you program specific enemies to test against you are mostly going to be testing against the starter designs or your previous designs. It's not as good as sending them onto the map to see how they do against live enemies but it's a great place to troubleshoot and design.

At any time you want you can quit by hitting Start. Select brings bringing up the menu. Where you can change camera view, music, sound effects, ect.. I

prefer the following camera but free roaming is also useful (it has its own controls to move around the battlefield freely), while random is weird and pretty useless.

It is worth noting there is a secret cheat code to enter manual control mode. If you hit Select 11 times in a row an additional option to control an OKE manually is added. Frankly the control scheme to control is very complicated and you'll most likely find it far easier this way but it's fun for testing purposes.

C. Negotiations

This is the games R&D section. There are 6 vendors that develop, sell, and upgrade hardware. You can also ask a vendor about any rumors they might know about enemy purchases to help with strategy decisions. As the game progresses new hardware will first come up for research. You can spend funds to speed up development so they can be purchased sooner but if you don't they always become available anyways. If you did put funds into research of a specific item the vendor will send you a message to pop up at the beginning of the turn it becomes available to purchase, which is nice. The prices on items fluctuate based on how much money you funded into it, if the enemy has also purchased it, how much stuff you have purchased from that vendor, if they are trying to get your business from another vendor, etc... It is a surprisingly deep system that ends up having no bearing on the game at all. Mostly because funds given to you in game are game breakingly high. The gossip system appears cool but they mostly just tell you basic things like "I heard they bought a powerful missile from McCormell." which is sort of helpful but not really in practice.

The other thing to note is every chassis, engine, and sub-weapon in game can be upgraded. Every upgrade follows the same formula. For some funds, the item will be upgraded in a predetermined amount of time, it will increase good stats, lower process requirements, and cost more to produce. The price and time to upgrade is fixed for each item as well. These upgrades usually are not huge, but every bit of added weight capacity, damage output, and durability helps in the game. The original item is still available in your inventory and can be used for new card designs. It does not update already existing designs so if you want to upgrade a previous design you either have to build a copy of that card with the new hardware or destroy all of the currently produced units of that card. So unless it's the first turn of a new map just build a copy card.

The other important note is the time it takes for a new item to be available to research. This is on a fixed timer for every campaign difficulty and is counted in turns from the beginning of the campaign. For example the Manticore is always available to research on global turn 40, if you are on turn 40 at map 2 or map 4 in the campaign it won't matter. So the faster you go the less advanced hardware you have available. This is true for the CPU as well because they can only use hardware available from the vendors at any given time. This is why you often won't see the very late game hardware in single player, it simply never becomes available.

System

The menu for loading, saving, and adjusting music and sound effects pretty self explanatory.

End Turn

If you are satisfied with all your choices for a turn this will end the controlled portion and enter into the second phase of a turn. All units will move and all combats will be resolved, all bases updated, all building calculated, all items available updated, and if one team is in control of all of the bases after all that the map will end.

Every 15 turns command will access your progress and give you funds based on how well they think you are doing. This can be anywhere between zero and 40k funds. When you complete a map they will give you funds based on how fast you completed it. This is where the money system in the game breaks down. If you complete any map in a reasonable amount of turns the funds awarded are uselessly high. Money doesn't matter in the game after the first map of each campaign. The starting funds for easy, medium, and hard are 10, 20, and 30k respectively but completing map on in a decent amount will usually net somewhere around a million. It's just functionally breaks all money strategy in the game. So just buy whatever you want and build as many high level lines as you want. Money is a joke aspect of the game after the first map.

=====

Hardware

There are two types of hardware: chassis specific and universal. When designing you select the chassis first and then fill in the rest. Chassis specs will be given as the base version and upgraded version in parenthesis.

Chassis

2 legged

These are upright walking, 2 legged OKE, The starting OKE is a 2 legged design so everyone begins with working with it. It's fast, very maneuverable, has decent weapon loadout options, can carry a reasonable amount of armor, and can easily be the only design you use in an entire campaign. They are my personal favorite due to their tactical flexibility, dodging strength, and offensive capability.

Prowler

Durability: 520 (530)

Engine efficiency 50% (53%)

Main Weapon: shotgun, assault, laser

Main Weapon Forward firing, Single Shot, grapple

Sub Weapon: forward firing

Sub weapon loadout:

4, 8 rocket pod

4, 8 missile

1 large missile

This is the starting OKE. You begin with all of the hardware but can research the mk2 version for a small fee. Of the 3 2 legged types its the weakest chassis, slowest go move, and carries the lowest amount of sub ammo. It's easy to design for, use, and really is a nice entry point to begin with. Once you master control of this design you will be able to move easily onto other designs that are harder to handle.

One nice thing about this chassis is simply how neutral it responds, it's easy to avoid by a single jump to the side, the is very very hard to overheat, can still be somewhat strong, and with 8 available rockets or missiles can hold its own still.

Jujuman

Durability: 540 (550)

Engine efficiency 55% (58%)

Main Weapon Types: shotgun, assault, laser

Main Weapon Forward firing, Single Shot, grapple

Sub Weapon: forward firing

Sub weapon loadout:

8, 12 rocket pod

8, 12 missile

2, 4 large missile

6 floating mines

6 scatter mines

The thing with the 2 legged types is they are all basically the same, just upgraded each time. This chassis is slightly stronger, more sub weapon ammo, faster to maneuver, and still not hard to program for. You can copy any successful prowler design strait over and it works just fine. It's a strait upgrade on every front.

Vypor

Durability: 570 (585)

Engine efficiency 60% (65%)

Main Weapon: shotgun, assault, laser

Main Weapon Forward firing, Double Shot, grapple

Sub Weapon: forward firing

Sub weapon loadout:

8, 12 rocket pod

- 8, 12 missile
- 4, 6 large missile
- 6 floating mines
- 6 scatter mines

Continuing the same approach, it's faster, stronger, higher ammo count, can carry better armor, and fires 2 shots for every main weapon action. This makes it the ultimate 2 legged OKE. It shows up later game but not so late that it won't be used I. The longer campaigns and that makes it special. It's the only 2 legged chassis with overheating issues so be careful with the dual firing lasers. I would rate this as the single best chassis in the game due to its ease of programming a strong program for, it's strength, it's firepower, and that is actually useful in the single player. It's also a fan favorite for pvp play and is probably the most widely used chassis that I've seen. A must try.

4 legged

A hybrid between the tank and the 2 legged types, these tend to be very jack of all trades. Not as armored or loaded as tanks, not as maneuverable as 2 legged type. They still can grapple strongly (the manticore is designed around it) and have higher sub weapon capacity. These are often some of the strongest enemy types due to their resilience and heavy missile loadouts. Very versatile, very strong, and with the turret style top you are never caught off guard.

Livewire

Durability: 720 (730)
Engine efficiency 70% (73%)
Main Weapon: shotgun, assault, laser
Main Weapon Turret firing, double Shot, grapple
Sub Weapon: Turret firing
Sub weapon loadout:

- 8, 16 rocket pod
- 8, 16 missile
- 2, 4 large missile
- 6 floating mines
- 6 scatter mines

Available to research at the beginning of a campaign makes this the best 4 legged type to use single player. It has a high sub ammo capacity, brutal grapple skill, good avoidance, stout armor and durability (best of the 4 legged types), and access to all main weapons. Converting from 3 legged to 4 legged AI programs is pretty simple and if you are struggling this might be the push you need to really push through those tougher maps.

Rogue

Durability: 680 (690)
Engine efficiency 78% (82%)
Main Weapon: shotgun, assault, laser
Main Weapon Turret firing, Double Shot, grapple
Sub Weapon: Turret firing
Sub weapon loadout:
 6, 12 rocket pod
 6, 12 missile
 2, 4 large missile
 6 floating mines
 6 scatter mines

Not much different in terms of function than the livewire design wise but has similar stats (less durability but higher engine efficiency so kind of a wash). It's fine I guess but not sure if it's an upgrade from the livewire even though it's later in the development cycle.

Manticore

Durability: 800 ()
Engine efficiency 72% (78%)
Main Weapon Turret firing, Quad Shot, powerful grapple
Sub Weapon: forward firing
Main Weapon: assault, laser
Sub weapon loadout:
 8 rocket pod
 8 missile
 2 large missile

This chassis looks great. It is easily the most intimidating design in the game with its large fanned head and comical clobbering arm. And carries the durability and engine efficiency stats to back it up. That giant claw is meant for action too. If you can get in close enough to grapple it's will do a heroic amount of damage to any OKE. The counterpoint is its low sub weapon ammo count (2 large missiles is pretty weak). Another solid 4 legged choice.

Tank

Literal tanks in the standard video game sense of the meaning, these carry the heaviest armor, have the highest standard durability, carry the most sub weapons, and can take the most punishment of any type. The only other available OKE at a campaign start is the kouger that boasts a 95% engine efficiency and a 1060 durability, it can carry the 35mm armor with starting engines. And while they lack in avoidance maneuverability like the legged and flying types they can still reverse fast enough to avoid incoming missiles. While they never get

a shotgun (which they would dominate with if they could) they do get the highest sub weapon amounts with the Jackhammer holding and destructive 24 rockets or missiles or 8 large missiles. Very little can survive that amount of firepower. Other than maneuverability the only other major concern is overheating with laser shots or taking damage. The choice of optional cooler or jammer are most prominent here since most chassis fire double main weapon shots but the red line for heat is very low. A great brute force design especially if you can master firing from the sides while dodging back and forth.

Kouger

Durability: 1060 (1120)
Engine efficiency 98% (100%)
Main Weapon Turret firing, double Shot
Sub Weapon: rear firing only
Main Weapon: assault, laser
Sub weapon loadout:
 14 floating mines
 14 scatter mines

These are the classic idea of a tank. Treads, turret top, camo green, very heavy armor, and high resilience, these are the most defended OKEs in the game. The main turret fires two rounds per attack action and can rotate 360 degrees. Lasers make a great main weapon choice but overheating limits fire rate, one of the few times I might recommend assault rounds. Sub weapon choices are limited to both mine types and they only deploy out of the rear. Really it's the main drawback to this chassis. It's also the mine laying champion carrying the most of any other chassis so if that's your thing (it should not be you're thing) then you can't beat it.

It's the only other chassis available at the start of a campaign and the computer uses it almost exclusively so be prepared to see it a lot. You'll see pretty quickly how such a simple strategy of keeping the enemy to the side and constantly moving forward combined with heavy armor can be. If you can develop a winning strategy with this design you can steamroll through the entire game from the start.

Jackhammer

Durability: 970 (1040)
Engine efficiency 96% (100%)
Main Weapon Turret firing, triple Shot
Sub Weapon: Turret firing
Main Weapon: assault, laser
Sub weapon loadout:
 18, 24 rocket pod
 18, 24 missile
 8, 8 large missile

12 floating mines
12 scatter mines

The first tank that gives you projectile sub weapons, this is everything you wish the kouger was. It has the most ammo of any oke design, second most durability, and as fast as the kouger. It's a true force to deal with as an enemy and can be a powerful force on your side

Ronin

Durability: 860 ()
Engine efficiency 85% (90%)
Main Weapon: assault, laser
Main Weapon Turret firing, triple Shot
Sub Weapon: turret firing
Sub weapon loadout:
 16,24 missile
 4, 6 large missile
 10 floating mines
 10 scatter mines

The last of the three tank types, it's very similar to the Jackhammer but carries less ammo, lower durability, with the trade of being much faster. It still can only use tank movement but can speed put of the way of incoming missiles and lasers with ease. It can still hold 24 missiles but no rockets and only 6 large missiles. It is a fun design but of the 3 is my least favorite of the tank types. The Jackhammer does everything this does but better and earlier in the game.

Flying

These fill a very special niche in the array of designs. Three very different chassis each needing different tactics and approach. The valiant is nimble, powerful with its high weapon loadouts (18 rockets!) and dual fire main weapon but fragile with low armor available, while the iron death and raptor are more akin to flying tanks. These are the only designs you need to be aware of fuel tanks and frankly the last two come so late in the development cycle you've probably beaten Callisto, but they are fun to play with for pvp. Mind the overheating, and enjoy just piloting over ground obstacles while dropping death spheres from on high.

Valiant

Durability: 380 (410)
Engine efficiency 32% (26%)
Main Weapon: assault, laser
Main Weapon turret firing, double Shot
Sub Weapon: forward firing

Sub weapon loadout:

12, 18 rocket pod

12, 18 missile

4 large missile

The first available flying type, this chassis delivers on the grand idea of raining firepower down from upon high. It's shockingly high rocket/ missile capacity and it's dual fire turret main weapon means you always can attack and once you get an enemy lined up, the downward angle makes rockets more devastating than usual. If only it could carry shotgun ammo it would be perfect.

The biggest drawback, and to all flying types, is it's low durability and low engine efficiency. This chassis is actually the lowest in the game on both fronts making it the definition. Of a glass cannon. Combined with its tendency to overheat easily and the cooling unit/ jammer choice is most exaggerated here.

The freedom of movement you get the first time you use this chassis is wonderful. No longer constrained by the majority of ground based obstacles, you simply float on over, avoiding all scatter mines and short rocks, and drop fire from above. Don't be fooled by the cpu poor use of these in combat. This can be incredibly powerful, especially when combined with a ground unit or two that can cover the angles. Easily my favorite of the flying types.

Iron Death

Durability: 430 (455)

Engine efficiency 34% (38%)

Main Weapon: assault, laser

Main Weapon Forward firing, double Shot

Sub Weapon: forward firing

Sub weapon loadout:

6, 12 missile

6 large missile

6, 12 scatter mines

Honestly, iron death and raptor come so late in the development cycle I have never used them in game.or seen them used by the cpu. So my only experience is by playing around in pvp.

Raptor

Durability: 580 (600)

Engine efficiency 38% (42%)

Main Weapon: assault, laser

Main Weapon Forward firing, quad Shot

Sub Weapon: forward firing

Sub weapon loadout:

6, 12 missile
6 large missile
6, 12 scatter mines

Same as iron death, it is the last to develop chassis in a campaign and the likelihood of actually using it is very slim. It is the most resilient of the 3 flying types but basically a beefed up version of the iron death.

Universal Components

These are the universal components that can be fit onto any chassis. For the most part, as the game progresses better options will be available as these items are developed. Almost all of these items have a upgraded version that is more powerful and costs less process to build an more funds to build. Each item has an associated process cost to build, a funds cost to build, and a weight, the rest of the specs are item specific as discussed below.

A. Engine

After you choose the chassis type you want to use for a card, you are required to select an engine for it. Each engine has a weight (basically how powerful it is) that when multiplied by the chassis engine efficiency determines how much weight the OKE can carry. The development curve is pretty linear so don't expect any major engine strats.

B. Main Weapon

There are 3 different types of main weapons, shotguns, assault rifles, and lasers. There are 4 strengths of each type, each at 5 firepower increase for each level.

Shotguns

Standard style short range spread weapon. The firepower is rated in a weird way because while it is given with the same stat as other weapons it's a rough gauge. Each pellet counts as a single hit so obviously close is better here. The general range is going to max out at around 150m because the pellets fall to the ground. But oddly enough it also tapers off at closer than 20m, something to do with the spread, I guess, not sure. I think the way the firepower calc works is that each pellet counts for that strength, so the more pellets that hit the more damage it does. Against my test dummy I was able to use the AP shot (the starting ammo type) and score single shot damages of up to 26%, which is way higher than the other starter ammos for sure, but the catch is that is at a range of 20m and practically you will never intentionally be that close. The averages tend to get higher damage wise as you get close but anything effectively beyond 110m is just there to trigger the enemies projectile avoidance and not really do damage. The sweet zone is 60m to 110m

which means you must be much more in the danger zone than with other main weapons. The other odd thing is the total ammo capacity is half of the other two types at 100 max, which is a joke really, and makes repeated combats much more difficult.

Every newer shotgun type simply goes up by a firepower rating of 5, ranged from 25 for starter ammo to 35 for end game ammo. Not a huge swing really. Pellet count and range don't go up either, it's just damage per pellet.

Strategically shotguns are great for keeping an enemy on the defensive. If the enemy has high reaction to projectiles these tend to keep them on the defensive since it will constantly trigger them to avoid. Against jumping OKE usually the jump won't be enough to get out of the spread and they will fall anyways, causing them to just take shot after shot. Getting into range is the real issue with this weapon since the other types will make it difficult. These work best against 2 legged and 4 legged types, and very well against flying types if you can get in range. Against tank types its much harder since they don't flinch, fall down, or have active avoidance in the AI much. Highly recommended for pvp builds, and works great in single player as well, but watch out for tanks that tend to not be as phased by the shear wall of firepower these throw down. Another nice feature is the low internal heat generated so you don't usually need to make a heat check in the chip design. Tank types do not get to carry shotguns though, and frankly that's for the best. If there was a Jackhammer with a 200 round shotgun and a 24 rocket pod the game would be broken.

Assault Rifle

Ugg.. ok, here's the deal, these are not great. They have roughly the range of the shotgun, less damage than the laser at each development stage, odd ammo amounts (80?, 140, 200), and don't do the heat damage the other types do. They don't weigh as much, or cost as much and have the same level of accuracy as lasers at the same distance, but the falling of it means that by 160m they generally already hit the ground. They generate almost no heat in the OKE to fire so that is nice but when give the option of lasers or assault I take lasers every time. If you like them or have figured out a better niche for them please feel free to use them but I don't know of any and don't really recommend using them. It's also worth noting that in testing these do consistent damage until at their very far range they seem to do slightly less damage, at about 170m they lose 1% of damage to the test OKE.

Lasers -

Lasers are really the other long range option in my opinion. They don't fire in an arc so the range is effectively 200 since the fire chip won't aim at anything further away. They have the same accuracy at range as the assault rifle. The starting damage is 50 firepower so they are always 5 more powerful than the equivalent assault rounds, topping out at 65. They do more heat damage to the target. And the cost and weight are about the same. They all come in two ammo quantities, 100 and 200 as well so no weird low ammo version. The one

large drawback to the lasers is simple the amount of heat they generate when fired. It is very important on most chassis to perform a temp check before firing since firing can cause massive overheating damage to an OKE. Any of the flying types and chassis that fire double shots are very susceptible to this. Each chassis has a different red zone for temp but not firing over 65 is a good place to start from. The manual says thicker armor makes overheating more pronounced but from testing I did not find it to be a large factor. If you want to build with high continuous fire rates try adding a cooling system, it will help keep temps lower so you can fire faster. The more powerful the laser the more heat it generates so be aware of that as well when you upgrade.

Sub-weapons

There are 5 different types of sub weapons: rockets, missiles, large missiles, floating mines, and scatter mines. Each chassis has specific types and quantities it can use and ammo amount is chassis dependent, not ammo dependent like main weapon ammo. Ammo for these are much smaller so care must be taken when programming to try and not waste these. They are generally much more powerful than main weapon attacks but are much fewer as well so be careful. Unlike main ammo, every type of ammo will have different properties and effects that may not be listed in game so they will be explained here.

Rockets

Of the 3 projectile sub weapons these have the highest risk reward curve. They do not seek and only travel in a straight line at a target. It is not listed in game but there are two types of rockets, shotgun style and burst style, 3 of the shotgun type and 2 of the burst.

The first 3 rockets available are of the shotgun type. These travel forward and at 35m break apart into smaller explosives that spread outward on a cone. Upon impact with either an OKE or the ground they explode into a large fireball. It is deceptive but the game does not have splash damage and only impacts of the projectile deal damage, but the explosions produce a tremendous amount of heat and being in that area can be enough to cause heat damage. The effective range of these rockets are weird, before 35m they do very little damage because they have not opened up yet so the game counts it as one weak projectile, after 35m and up to 100m the spread isn't large enough yet and will usually have multiple hits, causing a lot of damage. From 100 to about 140m the spread and the trajectory usually causes a large fireball that does a lot of heat to anything in it, that may still cause damage. Like regular shotguns the game means each projectile has the listed firepower and there is a sweet spot of 70-80m that can cause all of the rocket to hit a target. As a result the shotgun type rocket under certain conditions are the highest damage output weapons in the game by a large margin. In testing I've recorded 56% damage from one snake rocket on the test OKE, higher than the most powerful of large missile by a couple of percent. Overall these are my second favorite sub weapons to use. Unlike missiles they cannot be jammed and either the spread

jumping to avoid will still usually land a hit. A volley of rockets that land can kill almost anything.

The second type of rocket is the burst type. At a certain range these explode into a starburst pattern throwing projectiles in every direction. The last two rockets developed are of this type and they work very well with flying enemies but don't do much damage against ground enemies. The projectiles are not concentrated like the shotgun type and they give off very little in terms of heat damage. They have higher listed firepower numbers but actual damage in game is much less. Not worth it over the shotgun types except under specific circumstances.

Missiles -

Missiles usually have the same ammo capacity as rockets so they can be seen as interchangeable in a certain sense. Some chassis carrying up to 24, which is a large amount of firepower to deal with. Unlike rockets, missiles only do damage if they contact an enemy but they have internal guidance systems to help with that. Each missile has a range stat, but all of them are short or medium. The range roughly indicates at which distance the missile stops tracking and just fly off straight. Short range is good up to around 120, medium to around 140-150m. After that they fly off in whatever direction last pointed. Not every missile has the same level of guidance but I would say most are fair. The Wildcat specifically is much better at guidance and the Avenger is so below that you should avoid it all together. You can fire these in volleys up to 5 and I usually will do 2-4 because if the first shot knocks an enemy prone then the others will hit. Or all of them will miss, it's a gamble. The average guidance is only fair on most of these and are sort of easily avoided. They are also susceptible to the radar jammer and can be lost to those frequently as well. Overall I'd say missiles are good, easy to use if you make sure to use them in their effective range. Powerful but nothing over the top.

Large Missiles -

So very similar to regular missiles, but they do massive damage, 2.5x at least to a similar level missile, their guidance is phenomenal on even the worst one so missing is very hard to do, they all have medium or long as a range meaning you can fire almost across the whole map early on. The research progression on every other item in the game is very linear for firepower, cost, ect... but the 4th most powerful large missile, the Gorgon, is available to research at the start of the game. The starting large missile, Shogun, has a firepower of 150 and the Gorgon has 192, making it more powerful than the next 4 large missiles to develop. With the first jammer not available to research until turn 60 there is no defence for a long time. The Gorgon missile alone makes large missiles the easiest and most over powered attack for most of the game. It's the 5th most powerful, but the Leviathan and Medusa share the same firepower stats and they are only 2 points higher. It's an egregious amount of damage early game. The only thing keeping you from abusing it at the start is

simply of the first two available chassis only the prowler can carry large missiles, and only one on them. The first chassis available to research (Livewire) can carry 4. The only disadvantage is you cannot carry as many. That's really it. When compared the missiles these win almost every time. The manual even says at one point if the game seems hard just load up on large missiles and fire all of them. At some point in medium and hard the enemies start firing volleys of these at you and you realize how much trouble you are in so be prepared for that. A volley of 4 can easily kill an OKE. These are the main reason that choosing between the jammer and the cooling unit is hard to do.

Scatter Mines

Here's the deal, all the mines in the game suck (except one). Each mine has completely different effects, damage, spread, tactics. But they are almost never more useful than any of the main 3 projectile sub weapons. Scatter mines specifically sit on the ground and will not detonate until you are within a certain proximity, usually stepping on them. Again the game does not have splash damage so each mine shoots projectiles but if they don't hit then nothing happens. So even though the mines boast the highest firepower ratings in the game the most assuredly do the least damage of all. There is one exception but I'll explain that one when we get to it.

Tactically is the largest problem with mines. There isn't a good way to use them effectively. Each combat starts with units cluster together on opposite sides of the battlefield so to get the enemy to step on the mines you lay is very hard to dom you either have to lay them and retreat, which is hard because the battlefield simple isn't that large. Or you have to move forward laying as you go and hope the enemies try to circle around you and walk to them, which is also unlikely since most combat is face to face. It's very hard to walk past units without engaging in a fight. Coupled with mines are only withal to the unit that laid them which means friendly fire is now on the table and that nuts allies are almost always closer to the mines than the enemy. Most chassis fire them forward as well so it sort of blocks off forward motion for the team. The kouger lays them from behind and makes more sense as it moves forward. The two flying types that can lay scatter mines aren't available until the very end of the game making it unlikely you will ever use then single player. They are so tactically hard to use effectively and reliably for the small amount of damage they to it's just a waste. When I write ai I don't even put mine radar chips in because they are such a small actual threat it's not worth the chip space, and the enemy ai uses a lot of kougers and lays mines down every combat, still not that big of an issue. Anyways hears the details on each scatter mine:

- Tightrope fire shot upward in a tight shotgun like blast, very hard to do full damage
- Zonde jump up in the air above the OKE and then fire a shotgun like blast downward. Better at hitting than tightrope.

- Voodoo claims to have 265 firepower (highest in the game) but I hit a test oke with 5 in a row and he took 10% total, making it one laser shot basically. Just the worst attack in the game.
- Zeus stay on the ground, fire a very wide shotgun burst, more outward than upward and actually does reasonable damage.
- Deathospheres. Awwww yea, the only good mine, I might even say great. The trick with these is they roll toward the enemy when they are in a certain proximity, around 30m. You csn avoid them once they lock on, the have guidance, jammer does nothing, and they hit like trains. They don't throw projectiles, just a single pointed hit, dumping all of that sweet 210 firepower strait into the enemy. It's one of the most powerful attacks in the game. The only downside is how late it is in the R&D cycle meaning you won't get to use it much. These bring dropped overhead with an Iron Death is fantastic. It's so late in fact that you don't see them in single player until late maps on hard campaign. It's a shame really.

Floating mines

These are even less tactically useful than the scatter mines because they float above the battlefield. You don't even get flying types until mid way through the game and one 1 until the very end on hard campaign. And of the 5 types only one seems to be triggered by ground OKE, meaning unless you are flying they do nothing. For some reason. The enemy units love deploying them and they are available at the start but why? The only upside I can see is the constant floating up and down give the battlefield a cool atmosphere.

- Macabre doesn't seem to trigger with a ground oke, so who gives a shit
- Anthaless will trigger and fires a shotgun blast downward but they have to walk right underneath.
- Steelfang does not trigger with ground type.
- Kraken does not trigger on ground type, but can hit if it fires out and hits a guy directly in front of you before it floats up. Dumb.
- Empire also does not trigger with ground type, so overall, these are tactically useless.

CPU

The hardware of the CPU is pretty strait forward. You have a NC (chipset size) stat and speed stat. The weight, production and fund cost are negligible so use whatever it faster and large enough to accommodate you're AI design. The size is misleading because it the total size of the chipset, not the usable chip count. Every chipset is square and the outside square of chips on the chipset are the return path and you cannot edit them, so the editable chips on a chipset is one square size smaller than the listed size. A 100 (10^2) has 64 (8^2) editable chips and so forth. There are only three speeds, slow, medium, and fast. Faster is better of course because it allows the OKE to process more information faster. A large but slow chip can take so long to

cycle through that radar checks for incoming missiles for instance might get missed and you take unnecessary damage.

I was able to determine the actual time to process one chip in ms at each speed. The chipset processes all chips the same speed.

- Low = 24ms
- Medium = 12ms
- Fast = 8ms

So you can see medium takes $\frac{1}{2}$ the time that slow does, and fast takes $\frac{1}{3}$ the time. Not really important but neat nonetheless.

Development wise there is an early jump to a fast chip if you can figure out how to utilize a 100nc design the game starts you with. The first chip available to develop is the A-RI661P1 chip, and it's a fast, so if you can fit your whole program into the tiny chip it's the best early game upgrade that will carry you all the way to the end. If you need a larger footprint to develop your AI that's ok too. Only the starting chipset are slow and speed is the biggest determine factor in how reactive your design is but even the medium chips are twice as fast and should suffice.

Fuel tank

This determines how much fuel the unit carries into combat. It's the least important piece of hardware and usually I just put the largest fuel tank on that will fit the weight and process goal I have, its also the first thing I remove when I need to fit those goals. Some chassis use a lot of fuel, flying types, and running the jammers use a lot if fuel, but I rarely have fuel capacity issues. Also every fuel tank as available at the start of a campaign so there isn't even any R&D to worry about.

Armor

An okes resilience to damage comes from its chassis durability and it's armors defence factor. Durability is basically hit points and defence factor is a damage reduction, although I do not know the formula. Most of the armors are available at the start of a campaign but there are some very heavy armor available to develop as you get deeper in. For the most part the engine output will determine how much armor you can carry since it is so heavy. By far the heaviest single piece of hardware on an OKE. As a result you might have upgraded every other part of a design and have plenty of extra weight to spare but not enough for the next armor so you wait on engine development a lot. Basically always put the thickest armor on you can, not much of a strategy really.

Optional

This is a hardware slot not required to be filled in order to complete a design, every other slot must be selected. There are only 3 types of optional equipment: cooling units, missile jammers, and armor repair units. Choosing

between these will be difficult so weigh the options and access the enemy carefully. The jammer might be more useful if the enemies are using a lot of large missiles or the cooling unit might be better to increase your laser fire rate. These are all good choices so play around with strategy. None of these are available at the beginning of a campaign so all of them must be developed and there are 3 levels of each.

Missile jammer - a special optional equipment item that can be deployed to try and prevent missiles from locking onto an OKE. Large and regular missiles lock onto targets and seek them out if they are in range and these can be used to cause the missiles to redirect away. There are 3 different ones available and each one is better than the last. These use fuel when they operate and the better versions can use a lot of fuel so be careful about how you use them. These do nothing to stop any other type of attack.

Cooling unit - used to increase the rate an OKE cools itself, these are very important due to how much of an issue overheating is in the game. Passive units.

Armor repair - this repairs armor on an OKE between turns.

Paint pattern -

You can change the paint on each chassis to one of 5 predefined patterns. Some look great, but most are a wash of horrible colors and patterns. It's an impressive array of awful designs so enjoy! It is a shame the cpu only chooses the default color every time, it would be nice to fight against some purple tanks once in a while.

=====

9. Software

A. Overview

The OKE in the game are controlled by the cpu and the program on the chip. During the Hardware design you choose the cpu that will be used for the OKE before you can actually program. The two main properties of the chips are the capacity (NC number) and the speed.

The cpu capacity determines how much you can program onto the CPU. It's how many chips are on the cpu. Each cpu is square so equal number of rows and columns of chips. The outside square of the chip is always the return flow for the chip so they cannot be edited during the

design. So a 100nc chip is 10 rows by 10 columns (square root of 100) minus the outside square of chips ($10-2*10-2$) so 64 modifiable chips. The larger the cpu capacity the more complex programs you can design. The first cpu in the game is 100nc (8x8) and the largest is 256nc (14x14).

The speed determines how fast the processor can complete a command. Each chip on the cpu takes the same amount of time to complete. So a faster cpu can complete more instructions in a given amount of time. Faster chip, faster response time. There are only three levels of speed: slow, medium, and fast.

All of the cpu's in the game operate the same way, the logic flow starts at the upper left chip corner, drops into the upper left chip, and if it gets to the lower right chip it then starts back at the top. It operates as the cpu logic flow. The logic flow moves from chip to chip based on the direction of flow arrows, executes that chip, then moves to the next directed chip. Some chips have conditional modifiers that allow it to change the flow directions. This serves as the bases of logic. If you check something and it's true then the flow follows the red arrow, otherwise it will follow the green arrow. This forms the bases of the OKE programming language. The conditional modifiers are preprogrammed types so it's pretty restrictive but it is very visual and is simple to use. Albeit not very accurate as you will see in the chip breakdowns.

B. Chip Explanation

To create a program you must place chips on the cpu and link arrows together so they create a path through the chip. The green arrow indicates the direction the process moves after the chip is executed and red arrows are alternative paths traveled if a condition is met. There are 40 different types of chips you can use to create your program and each one has special properties.

1. Action Chips

These are chips that the OKE has to execute physically. Not all chassis can perform

all the different actions so they might not be available when you are editing. The

OKE can only perform one action at a time and there is a priority list explained below.

a. Jumping chips

Only 2 and 4 legged chassis have the ability to jump and it's used primarily as a defense mechanism. It's a very fast way to move around or avoid projectiles or obstacles. The 2-legged oke can jump much faster than the 4 legged type but the 4-legged are more stable when they get hit while jumping so they can recover much faster when they fall down. There are 6 types of jumps: forward, backward, right, left, strait up, and ducking. The chips have no options other so implementing them is very simple. Jumping strait up and ducking are faster than directional jumping and are very well suited to closer range projectile avoidance. Ducking is better at avoiding lasers and missiles since they tend to stay at the same head height and vertical jumping is better at avoiding rockets, assault, and shotgun ammo since they fall to the ground over distance. In general use directional jumping to avoid incoming projectiles when things are at a medium range and vertical/ ducking when a projectile is detected at a very close range.

b. Attack Chips

There are two types of attacks available to all chassis and an additional grapple option available to 2 and 4 legged types. These all auto aim against enemies at all times and only will shoot allies if they are in between itself and the enemy.

1. Main weapon fire

This chip only has a direction and an arc but it actually performs a lot of automatic actions. It only fires one time each time you call it, but certain chassis will fire multiple shots per call (kouger for example). The chip will also automatically turn the oke to fire the weapon at an enemy as well. The 2 legged chassis can only fire in the front so they will turn to attack but the other chassis usually have turret style tops that allow them to fire in 360 degrees without changing the forward direction of the oke. So tanks can fire to the left or right while moving back and forth to give them a certain amount of avoidance if you want but the prowler will have to turn all the way to shoot. There is no range listed on the chip to edit but it has a preset range of 200m at all times. It will not lock onto anything further than that. The chip sweeps for enemies in the arc range with a bias of enemies closer to the center line. So if there are two enemies in the same sweep range but one is directly ahead and one is off to the right slightly then it will aim at the more center one. It's an automatic process and sometimes it will change the chosen target back and forth but it's very hard to work around since it's not really under your

control. While the chip does a lot of automatic processes it will not check if an enemy is in range of the equipped main weapon, if there is ammo available, if the oke is overheating, or if there is anything between the enemy and itself (obstacles, friends, ect..). You must program those checks in yourself to avoid issues because it will attempt to fire with no ammo or if it's causing itself to overheat and die. Firing lasers generate a lot of heat but shotguns and assault round should be fine under most condition. Roughly, the lasers have a range of 200m but loses accuracy after 170m, the assault cannon are good to 170m but really fall off after 140m and become very inaccurate, and shotguns are maxed out at 140m but 70 to 100m is the sweet spot. Feel free to adjust for your design for sure. The laser and assault also loose a small amount of damage at the extreme of their range (1% usually) and the shotguns damage is based on how many of the pellets hit the enemy. I have noticed no accuracy difference between types of ammo or chassis.

This chip is robust enough that it can be it's own complete program. For fun, build a laser based unit and put only this chip at the top right and point the arrow to the left so it's the only chip on the cpu and it can hold it's own against the starter OKEs. It won't do anything but fire constantly and turn to the enemy but that's enough to kill some very weak designs.

The one thing that the chip will do that is frustrating but is unavoidable is it will auto aim to a currently dying enemy but it will not fire at it. As a result it will not turn to fire at another enemy even if it is currently a threat and it won't fire at the slowly dying unit and finish it faster. It's a weird quirk that will sometimes cause you to get shot or simple lose time while being locked into a limbo state. I wish I could tell you how to avoid it but I don't think there is a way.

2. Sub Weapon Fire.

Similar to the MWF chip, there is direction and arc for aiming but there is also a quantity to fire, 1 to 5 rounds. There are 5 different types of sub-weapons, rockets, missiles, large missiles, scatter mines, and floating mines. The targeting system is the same as the MWF chip as well so all those notes apply when not using the mines. The mines don't aim but will fire out of the chassis in whatever direction the mine layer is pointing. Some chassis fire them out of the front or out of the weapon turret but some will fire them from behind so be aware of how the chassis you are programming uses the mine layer. Once mines are laid they cannot be triggered by the unit that laid them but they will go off if an ally runs into them. Also if there is a unit close to the mine layer when they shoot out they will explode and damage the other unit so be careful when laying them out to not shoot close allies. For an in depth explanation of each type of mine see the mine section.

The 3 projectile types of sub-weapons aim using the same system as the MWF chip but will repeat firing up to 5 times. Each chassis can only fire one at a time and it will check if it needs to change actions between each fire in a volley but it will not change targets if you use

a volley attack. If you fire only one sub-weapon at a time then it can freely choose different targets if they are in the sweep range for each instance. It tends to be more effective to fire multiple missiles at once but not as useful for rockets, but feel free to play around with different strategies. Again it uses an set range of 200m but different missiles will indicate they work at different ranges. This means they will only lock on and seek in a certain range and will move strait after that range. Short range is roughly 100 to 110m, medium is up to 140-150m, and long is up to 200m. Rockets actually do not seek but break apart and either turn into a shotgun type or a star burst style of projectiles. One thing to note is that there is no splash damage in the game so while the rockets typically create a large amount of explosions it doesn't do any damage unless the individual projectiles hit the enemy (or ally). The explosions will create an OKE to get hot if it's in the area so it might cause a OKE to overheat and take damage but not direct damage will occur.

Again this chip will not look for enemy distance, obstacles, allies, temperature, ammo remaining, ect.. so you must check for these things yourself. I would say that since there is a very limited amount of sub-weapon ammo compared to main weapon to be more proactive about when to actually fire these lest you waste the precious ammo on avoidable misses. All missiles also have internal guidance systems so different missiles will seek differently, details about this is in the missile descriptions.

3. Grapple

The 2-legged and 4-legged chassis also have an extra attack option called grapple. The chip has no options at all and is all automatic in nature. If an enemy is within 30-35m and in a direction the oke is facing it will attempt to do a jumping lunge with with a short range arm attack that does a crushing amount of damage. It's by and away the most powerful attack in the game if you can land it but the conditions are very tricky to get it to fire. If the chip is called but the conditions it uses determine if it can attack are not met it will do action. It will will not dry fire like the other two attack chips. The 2 legged chassis will only attack in the forward direction but the 4-legged type can attack in any direction if it thinks it can. The Maniticore chassis specifically is designed to do extra damage with this attack. The one thing to be away of with this chip is it will do a similar lock situation when an enemy is dying but still in range. The issue with this is you will be in range when the enemy oke explodes and you can take a lot of damage from this, so it helps to put a walk backward chip right after this one so it will automatically walk backward if it is not trying to attack and hopefully will get out of range of the dying OKE.

c. Movement Chips

These are the standard movement used by all OKEs. All chassis types can use forward/ back, and turn left/ right, the legged chassis can also use side stepping, and the flying type can use altitude up and down. These chips don't have any options and are pretty basic. You can only perform one type of movement at a type with the exception of flying type with altitude adjustments. The forward direction is always the front direction of the lower chassis and not the turret or the direction of the weapon systems. None of these chips have any sort of feedback so if you walk forward into an obstacle it will not stop unless you command it to do something different. It's up to you to develop the movement and avoidance code for your specific design.

d. self destruct

This is a special action chip that will cause the OKE to self destruct and explode. When an oke explodes it creates a shower of flying parts that can damage close units. Once this command is issued it cannot be turned off so only let this be commanded if you really mean it. Tactically I cannot think of a scenario where this more fore effective than just firing remaining weapons or attempting to grapple but it's an option.

e. Jammer

This is a special chip that can be used if a missile jammer is used in the optional hardware slot. It activates the jammer and gives a percentage chance to cause a locked on missile or large missile to deflect to a random direction. There are 3 levels of jammer and each one has a better chance of deflection. This action costs fuel with better jammers costing more fuel. It's worth noting these use a lot of fuel and should be only used if you have checked if a missile is in a certain range instead of using it any time a projectile is in range or you may run out of fuel rather quickly.

2. Radar Chips

All the radar chips operate the same but search for different object. They search an area centered on itself at a sweeping angle and distance away and if anything matching the type is in the area then it returns true and diverges the chip flow down the red arrow. You can select the center angle, the sweep angle, and the range. The range is 5m to 300m but be aware the combat arenas can be larger than 300m so it might cause problems in the logic if you are not careful. The center angle can be set 360 degrees and 0 degrees is the front of the oke at all times. The sweep range is 11 to 354 degrees, which the clever person will realize is slightly less than 360 degrees. So if you want to see if there is anything in say 50m on any direction you would need to make two checks that overlap to cover that small sliver of 6 degrees that cannot be

checked by one chip but honestly is never is an issue so don't worry about it. Unlike the weapon fire chips, there is no distinction of how close the searched object is to the center angle, just simply if it's in the sweep. The range also always starts at the origin so you can't search for a distance band with one chip, you have to use multiple chips. It also won't tell you how many it detects just that at least one is detected. You will also notice it's not line of site detection just blanket area detection. So you see if there is an enemy and then fire at it there might be an ally or an obstacle in between you and the enemy and you shoot it instead. These sorts of checks will fill the main logic of your design.

a. Enemy

This searches for any alive enemy, it will not detect currently dying oke units though. Use this for finding enemies for either attacking, avoiding, setting engagement distance, determining best tactics for a given enemy location scenario, movement directions, ect... Your teams highest priority is to eliminate the enemy so most of your choices will be based around finding and shooting the enemy as quick as possible.

b. Ally

This detects friendly units excluding yourself of course. Used effectively for avoiding friendly fire, setting up movement tactics, avoiding jumping into allies during evasive jumping, and building firing and pack tactics. One of the largest blind spots of the starter OKE designs is their complete lack of ally awareness.

c. Projectile

This detects any of the enemies main and sub weapons, any of the scatter pellets from mines, and any parts of okes that fly off when they explode. It does not include detect mines themselves even if they are floating. It also will not detect your own or your allies projectiles, but it will detect the projectiles of the mines your allies laid. The one thing to be aware of is this only tells you if a projectile is in the sweep area, not if it's actually heading toward you. So if a laser is aimed at an ally but detected you might react for no reason. This is going to be the chip that will be used to avoid incoming attacks. Learning how to avoid enemy fire under the various conditions will be one of the largest factors in a strong program.

d. Missile

Similar to the projectile radar chip but only returns true if it detects an enemy missile or large missile. It does not detect rockets, mines, lasers, shotgun pellets, or assault rounds. This is going to be

used usually for using the missile jammer chip. You can usually just assume the incoming assault is a missile and always use the jammer in sequence with the rest of your oke's avoidance system, but since the jammer uses fuel to operate (and a lot of fuel depending on the strength of the unit) it might be better to only to use it when you detect missiles specifically.

e. Obstacle

Obstacles are going to cover any of the static objects you cannot move through on the battlefield and also the edge of the map. There is some weird fuzzy logic here about how well defined the edges of objects are when it comes to matching up to what you can detect and what you can, say, shoot through. Objects on the field are different heights, the edges are sloped, ect.. But the radar only checks if it is in the swept area in 2 dimensions and not 3. It will also not see objects sometimes that are in the swept area and you can spend a lot of time shooting rocks you might assume you checked for. It is an issue that causes issues but I am unaware of any way around them really. Using these chips effectively is going to be one of the largest difficulties in the program design. There are a lot of random objects on the battlefield that need to be avoided while moving, avoiding incoming fire, or to not be shot at while trying to hit enemies. It will also help you avoid getting stuck in the corners of the map and other various aspects of moving around the battlefield. The test fields (field 2 does not have any objects, just map edges) have a small handful of regularly placed objects but the battlefields of the maps are filled with randomly placed and various sized objects. On easy there will be a few but on medium and hard campaigns the amount increases substantially. Learning to navigate these issues will be one of the biggest factors in being successful in the harder campaigns.

f. Mine

This simply detects mines in the swept area. It doesn't know the difference between scatter (on the ground) and floating mines and just returns true if it detects either. Flying OKes won't trigger most scatter mines and all other chassis will only trigger some of the floating mines. So attempting to avoid all mines may not be very useful since they might not even trigger. For the most part the mines are not very dangerous since they don't actually do radius damage but only do damage if the projectiles they shoot hit your oke. Honestly I don't usually use any mine detection in programs since it can complicate the design and use a lot of space on the chipset while not offering that much protection. It should be noted that the enemies in the campaign's use a lot of mines in it's tactics so you might find this chip very important, but I usually skip it.

3. Status Chips

This class of chip are used to check internal systems stats. The logic in them is a bit strange and counter intuitive so let me explain. Each stat is compared to a set value with a conditional check. The slider does not go to 0, which is a very important number I've heard. There are three types of conditional operators; equal, greater or equal to, and less or equal to. The text in the menus is misleading since it uses the > and < symbol which usually indicates greater or lesser than but both are also equal as well. Combined with not being able to select 0 on the number slider means you have to use the chips less intuitively than you might expect.

a. Ammo

Coming in main and sub weapon checks, use these to check if you have ammo before you try to fire. Since both of the fire routines will dry fire (shoot with no ammo) you need to make sure you have ammo or the oke will simply do nothing. Because of the logic of the chip, you have to select 1 on the slider and then the > operator and then point the red arrow to the fire chip. If you didn't know otherwise you might select 1 and the < and diverge if it has no ammo. This would be wrong and will leave you with 1 round that could have been fired. Not a big deal for the main weapon but the sub weapon has much less ammo, so be careful of the hidden rule when using these chips.

b. Fuel

This checks to see if you simply have fuel. If the oke runs out of fuel it stops but can still fire weapons. It's not really worth checking for honestly. Running out of fuel is pretty rare unless you are spamming the missile jammer.

c. Damage

Simply checks the oke damage percentage. I guess if you wanted you could adjust the aggressiveness based on current damage, like dumping all sub weapon ammo remaining, or making a run forward to use the self-destruct action. Not super important but nice to have as an option.

d. Temp

This is the one subsystem that should be checked regularly. Each chassis has a dangerous red zone temp that will cause damage to itself. Many different things will cause the oke to heat up, mostly laser fire or taking damage. Weapon firing should wait to fire if the temps get close to the damage range or simply firing the weapons will cause damage. If there is no check for this it is hard to have a unit kill itself from overheating. Some units are worse about overheating than others and some weapons cause more

heat than others as well. Lasers generate the most heat and firing volleys of missiles also cause a lot so be sure to check current temp before using those chips. If you are having problems with heat try using a chiller in the optional slot of the chassis hardware. Laser rate of fire can be reduced quite a bit if you have to wait for the temps to drop every shot so using a chiller is a good way to increase fire rate. Another thing to note about absorbing heat is explosions on the field can cause a lot of heat even if it doesn't do direct damage. So a rocket that misses with each projectile can still cause heat damage from the explosions they cause when they hit the ground around you.

e. Ally

Simply checks to see how many ally units are still alive. Useful for making decisions based on how outnumbered you might be, or down on firepower, or changing flanking tactics. More situational than other checks, but still has it's uses.

f. Timer

Every round has a max time length. All combats during the campaigns are 150 seconds long but VS. Mode combats could be longer. When the time is up and both team still has units alive it ends in a draw. If you want to change tactics based on how long the match has lasted this is the chip to use.

4. Special Chips

Mostly having to do with flow execution and changing action commands.

a. Neutral

Does nothing other than move the flow to the next chip. Used for connecting sections of code together.

b. Stop

This will cause any current actions that are being executed to stop before the flow moves forward. It can be used to cancel any actions except jumps/ducks that have already started. Useful for when you need to quickly change the priority of the okes actions in case of emergencies.

c. Wait

This will wait on the chip until the current action finishes executing. The flow will continue once the action is done. Useful for any situation when you don't want dropped action inputs, or lost cycles,

or miscounting during loops, ect... This is very useful to counteract the action priority systems and making sure inputs are executed.

d. Random

A simple random diverge for inserting randomness into your program pathing. There are two sliders letting you choose the randomness as defined by a fraction ($1/2$ equals 50% for example). It's important to put some randomness in your programs. If, for example, you have a team all using the same program and they are all programmed to jump to the right when avoiding forward fire, after some time all the units will move to the edge of the field together and are very open to being cornered and losing. This is very useful for things like choosing how often to fire sub weapons or jumping left or right. It can also be used to exit out of loops. The red arrow direction is when the random is true (diverge $2/3$ of the time for example).

5. Memory Chips

These chips work as the only way to record information internally and to communicate to other units. These are very useful and more difficult to use but can increase the level of program complexity quite a bit.

a. Radio Chips

These consist of 2 chips: the first starts sending a constant signal out of a certain color across one of 5 channels, and the other checks to see what color a particular channel is currently. These are the only way to send signals back and forth between units. Each channel is default black so if you want to tell your other units you are laying mines for example you can set channel 1 to red, and other units can know they should help defend if they notice that channel 1 turns red. The thing to note is that the channel is changed for all units, any unit can change the color of the channel and then read it back, it's not changed for just one unit.

For example, if you have a tank and two 2-legged units on a team and the tank has a routine that changes channel 2 to blue, either of the other units can read the channel is blue. Then either of the 2-legged can change the channel to red to tell the tank it has been received. Any unit can change any channel to any color and then it stays that color until it is changed.

You can create much more complex tactics using these but the extra code will take up a lot of chips on the cpu so be that it might not be feasible with the smaller chips.

b. Counter Chips

These 2 chips allow you to do some basic arithmetic and some simple conditional checks. The Counter chip allow you to choose one of 5 different variables (A through E), perform either add, subtract, multiply, divide, or store to the variable, and then choose a number to modify it with using a slider (with all the sliders you cannot use 0 for some reason but goes to 100). All 5 variables start at 0 at the beginning of each combat. So if you choose variable A, +, and 1, each time the chip is activated the variable A is increased by 1. Essentially the operation is performed on the variable and then stored back to that variable. You can also use the arrow, or store command, to make the number whatever you want. You might have noticed it does not seem to support copying one variable to another position and you would be right. Not natively at least but a clever person can figure it out if it's really required for some reason.

The counter comparison chip will diverge when a set condition is met. You choose which variable, a number between 1 and 100, and if the number is greater than or equal to, less than or equal to, or equal to. If the condition is true then it diverges. Again notice you cannot simply choose 0 and you cannot choose only greater than or lesser than. You also cannot choose not equal to, or any other standard operants to might be used to. You also cannot compare two different variables.

Because of the strange restrictions it 0 ends up being a hard number to work with. You can't store 0 to any of the variables and you can't test if a number is only 0. You can test if a number is 1 or 0 by choosing 1 and then the lesser than or equal to. You can subtract 1 from one to get to 0 but if you don't have a way to check then usually you will go very far into the negatives (which is allowed for some reason). To give you an idea of the odd limitations here here is an example. I may want to jump backward if I have jumped left or right more than 4 times. I then choose variable A to keep track of how many times I jump and reset it once I've jumped backward. To do this I check to see if A is > 5 (remember the chip symbol is mislabeled and actually means >=, greater than or equal to). If the condition is not met then then it adds 1 to A and jumps either left or right, If it diverges then 1 is stored in A and the oke jumps backward and the cycle starts again. The odd thing to note here is the first time this routine is used it will jump left/right 5 times instead of 4 because the counter always starts at 0. This was just an example of how the quirks of the system might cause strange issues. If you wanted you could change the logic to diverge if A=4 and then subtract 4 from A when you jump back.

These chips are some of the more useful chips you can employ in your design and can be used in man clever ways. These allow you to make active choices in how you respond to given situations compared to simple reacting to the current situation the other chips provide. They can also be used to do interesting flow path adjustments to allow smaller, more efficient code. Really these are some of the more flexible chips you can use. I will give some examples later in some ways to use these that are less obvious.

C. Logic flow overview

Effectively, there is two separate actions going on at all times. There is the chip flow and the current action stack. The chip flow is simply the logic flow as it moved through the chip. The action stack is whatever the oke is doing at any given time.

The chip flow is the order and path the program is executed. The flow always starts at the top left chip on the board (the one in the purple barrier) It then drops down into the top left editable chip and then simply follows the arrow path through the chip. At any point if the flow points toward the outer purple chip ring it flows through a path of neutral chips to the bottom right chip. When the flow reaches that chip it starts at the top and continues the loop. Simple as that. You can create loops inside the cpu as well, but be careful to make sure they have a way to exit the loop. If you try to put chips on the cpu that are either not connected to anything or don't point toward another chip the editor will show an error and not let you exit the builder screen. The flow has to be continuous at all times with no disconnected sections of code or chips that do not flow to other chips.

You will also notice there is no way to jump around the cpu space. The flow must move from one space to the next and has no way to jump around. The arrow directions can only point in the standard up, down, left, and right, no diagonals. These restrictions cause the main flow design limitations (later sequels to the game address these issues but this first game is fairly rudimentary in it's implementation). As you start to take your plans and try to implement them onto the chip you will realize the difficulty involved. I will go into some examples of ways to shrink your code in a later section.

The cpu flow takes the same amount of time to execute each chip on the cpu and the time is based on the speed of the chipset. There is Slow, Medium, and Fast and it's determined by the chipset chosen in the hardware design phase. The catch is the flow never stops (except when the Wait chip is used, which I will get into shortly) even though the oke actions take various lengths of time to execute. This means the chip can easily command the oke to do many different actions at once which it clearly cannot execute. This may result in a lot of dropped action commands unless you are aware of how the system determines how it executes these actions.

To decrease the reaction time you can use a faster speed chipset or to use less chips in the flow path. With a slower chipset or a very large or complicated design it might not register an incoming projectile before it hits you. You can compensate for that by increasing the range of the projectile checks but the side effect will be a lot of false positives. That is because a projectiles may be in the swept are you are checking but are not actually traveling toward you.

Chip Action Priority List

The action stack is what the OKE is doing at any given moment. Each OKE can only perform one action at a time (you can't move and fire for example) but as the cpu instructions tell the oke to perform different actions it follows a priority list. If a specific action is of a higher priority then that action instruction is dropped and the new one replaces it. Each action only lasts a certain amount of time and once it is performed, unless there is a new command, the oke will go neutral and do nothing. So if you want to keep walking you must keep instructing the oke to walk, or it will stop after a short distance. It's important to understand the priority list because it is easy to arrange things in a way that causes commands to be dropped. You can also take advantage of the priority be arranging chips in a way that allows you to make smaller routines. I'll give some examples later. Here is the action priority list:

Highest Priority ----> Lowest Priority

1. Self Destruct - This is the highest priority chip and will always execute if commanded.
2. Grapple - this chip will only activate if there is an enemy within its range otherwise will do nothing. You can litter any neutral paths on your chip with this if you want just in case you can use it since it's such a powerful attack since it won't do anything if it can't land.
3. Directional Jumps - forward, backward, left, right , vertical. If the oke is already performing a jump and another jump command is given, it will be ignored until the first just is finished. You can put multiple jumps in a single path and usually only the first one will be executed.
4. Duck
5. Main and sub weapon firing - these are equal priority so whichever one is commanded first will be performed. If the ammo for the command runs out it will not change it's priority so if you want to have main weapon fire if sub ammo is out then you must program is to not command the sub weapon to fire. If you use multiple rounds with the sub weapon the wait command will wait until all of them are fired, even if it is out of ammo.
6. Movement Chips - walk forward, backward, turn left/right, shuffle left/right all have the same priority. These will cancel each other unlike the jump commands. So if you command to move forward and then turn left, it will turn left.

Special Actions

Ascend/Descend - the flying type can also move upward and downward while performing other movement actions and also main/sub weapon firing.

It will prioritize ascending over descending if both are in sequence. Like the other chassis, they still cannot move in any other direction while performing any other action (which is a shame really, if the flying type could perform moving strafe runs they would overpower every other oke in the game, owell).

Stop Chip - this will cancel any current action, will not cancel any of the jumps, but can cancel a duck command if commanded fast enough. Note, this will also cancel the grapple command (but the oke will probably move forward a little bit before it cancels its)

Wait Chip - this will stop the flow of the cpu and wait until the current action finishes. This is the only way to sync up the chip flow to the current action. There will be times when you have to use these chips but usually you can let the priority list take control of the most important action is. If you are trying to count how many time the oke performs an action it and you need it to do it accurately it will required to use this chip in the loop, otherwise you might count multiple times as the chip loops through the same path but the oke only performs the action once. Also useful for making sure higher priority chips do not replace lower priority ones you want to finish executing.

The game prioritizes jumping>attacking>moving so basically defending>attacking>searching and movement. For the most part this is going to match your design philosophy but if it doesn't you will have to work around it using the stop and wait chips to keep from overriding your commands.

D. General Strategies

The manual tells you that a good program accounts for four different elements: attack, evade, search for enemies, and movement. I will alter this list a small amount and say a good program will have attack, defence, search for enemies, and situational awareness, and tactics. It is not required to have all of these elements but it can help create a more robust design and using it as a checklist will give you an idea if you are creating a program with an obvious blindspot. Different chassis will need more or less attention paid to each element

1. Attack

The language of Carnage Heart is kill thy enemy so it's pretty obvious that should be your main goal when programming. Each oke has the ability to use main and sub weapons, and some can also use a melee grapple attack. There are a variety of different kinds of weapons to use and ways to use them so developing winning tactics will hinge on how you use the weapons most effectively.

Understanding when you should fire is going to be the hardest part of programming this part of the program.

Main weapons consist of shotguns, assault rounds, and laser bolts and will be the bulk of your offense. While the subweapons offer a bit more variety in their actions and use.

Learning how to use each type of weapon effectively is paramount to success.

Defend

Most chassis cannot handle taking a lot of direct hits for very long before falling so it's imperative to develop a system to attempt to avoid as many attacks as possible. Different chassis will make this task easier or harder but suffice to say, simply attempt to move out of the way of incoming attacks and try not to step on mines. 2 legged types are the most nimble of all and can easily avoid a projectile by simply jumping to the side. Four Legged types can do similar but are not as nimble or can jump as far so they are usually within the danger zone of a shotgun blast or shotgun type rocket. Ducking and vertical jumping are also great at avoiding very close incoming projectiles. Ducking better at lasers and missiles and jumping good for rockets and shotguns since they tend to have a falling trajectory.

Tanks are very limited to basically rolling forward and backward so are at a large disadvantage when attacked head on. The main key in tank avoidance is to keep enemies to your flanks and always be moving. It sounds simple but in practice is very difficult.

Flying types can simply just move up and down or strafe to avoid incoming shots. They are naturally better at avoiding shotgun, assault, and shotgun type rockets due to the trajectory effect of the projectiles.

Including a missile jammer can be a great way to try and avoid those high powered and maneuverable projectiles but they are not perfect. Each version has a higher percent chance to jam than the last but also uses much more fuel. When programming try to only activate it when missiles are detected, or risk running out of fuel on the battlefield.

The best way to avoid damage is to try and keep the enemy at a far range. Lasers and long range missiles are great at attacking from a safe distance but staying at those ranges is hard and accuracy is poor.

Defence should generally be your top priority but with cases can be made the best defense is a strong offense.

Search for Enemies

If you are not engaged in a firefight go find one. Don't sit idly by and wait for the enemy to come to you. The range of the radar is 300m but the battlefield can be larger than that so be aware of that if your design just stops moving sometimes and doesn't look try to find anything, they might just be out of range. It's a simple aspect but it must be there or your design might just stand around and do nothing.

Situational awareness

Knowing where obstacles, your allies, your enemies, the edge of the map, and mines are all help determine the best tactics to use at any point. Shooting allies in the back, jumping into a rock and failing to avoid a missile, and jumping into a corner and getting surrounded are all examples of simple ways that failing to account for the surroundings causes a lot of preventable damage. You will likely add a lot of checks to help determine what the best action is at any time. Even just walking across a obstacle filled map become impossible if you do not have the correct logic in place to deal with multiple rocks at one time.

Some easy example are simply checking if there is an obstacle in the direction you are going to jump before you do. Check to see if there are allies in front of your line of fire. Checking to see if there are obstacles before you fire those expensive large missiles into them. Not walking into a mine. There will be lots of these and learning what checks are important and which ones are not will help guide how your design operates.

Tactics

On a more abstract level you need to try and give your team battlefield tactics they can use to overpower a stronger enemy. Will you try to spread a single team out to trap enemies in a crossfire? Will you use multiple range weapons to engage at different ranges? Will you add a flying type to a team to and use it for suppressive fire? These concepts and any number of other ideas are going to be up to you to decide and then figure out how to implement via code, hardware configuration, and team building.

These concepts will be the hardest to implement and may require the use of advanced programming techniques and concepts. If you cover the previous parts of the code then you will automatically have a strategy but the application of a more concrete approach can really deliver the difference between winning 75% of the time to winning 100% of the time.

A list of randomly ordered programming tips

Examples of Designs

Chassis: any 2 legged design

Main Weapon: laser

Sub Weapon: missile or large missile

Chipset: any 100nc chip

Optional: jammer

