Exploration in a Reverse-RPG

Robogeddon, a Casual Real-Time Strategy Game

A Major Qualifying Project Report
submitted to the Faculty of the
Worcester Polytechnic Institute
in partial fulfillment of the requirements of the
Degree of Bachelor of Science
On April 28th, 2010

Submitted By:
Riley Brown, Sean Crepeau, Karl Gibson, Adam Pastorello, Philip Tang, and Ian Williams

Advised By:
Professor Joseph Farbrook, Professor Jennifer deWinter and Professor Joseph Beck

Abstract

This report discusses the design, implementation, and analysis of *Robogeddon*, a casual real-time-strategy game for the PC, and was created as a WPI MQP for the computer science, interactive media and game development, and professional writing majors. Players in *Robogeddon* have the unique experience of taking on the role of an overwhelming agent of destruction, in contrast to the typical heroic archetype of many role-playing-games. *Robogeddon* incorporates "tried and true" elements from classic RTS games, as well as unique new features to attract more casual gamers. This report details the designers' vision, in terms of artistic style and theme, as well as gameplay and the player experience. Additionally, the report will describe the approach to implementation and the challenges therein. Finally, the creators of *Robogeddon* describe the lessons learned from the process, and advice to future teams at WPI and beyond.

Acknowledgements

We would like to extend our deepest gratitude to Professors Farbrook, deWinter, and Beck for their continual guidance in bringing our vision to life.

Additionally we would like to express our appreciation to the following WPI students, who aided in the completion of this project through various means:

Ryan Chadwick, for contributing immensely to the project through the creation of many User Interface assets.

Jackson Fields, who, during our early phases of design and development, helped our technical team be better prepared for the challenges XNA development provides.

Andrew Creeth, for composing and recording original music for our final game and trailer.

Table of Contents

Table of Figures

Table of Tables

Chapter One: Introduction

In many modern role-playing-games, such as *World of Warcraft* or *Rift*, players take on the role of a single protagonist, valiantly fending off the overwhelming enemy hordes, manipulated by some sinister mastermind. *Robogeddon* makes players feel like the mastermind of the hordes from popular RPGs. They are cast into the role of a determined, single minded entity possessing seemingly endless armies set on the path of destruction. This entity takes the form of a digital construct, a highly "evolved" form of modern day artificial intelligence; such as the state machines, which control most RPG enemies. This futuristic computer program can dispense orders to subordinate programs, down a chain of command to a robotic army, much like shadowy antagonists of many RPGs command their minions.

Robogeddon is a "casual" real-time-strategy game, meant to entice new players and diminish the steep learning curve that is often associated with the genre, while still providing the classic feel and "fun-factor." Many of the mechanical tasks, such as gathering resources and constructing units are delegated to artificial intelligence, so the player may focus instead on high-level strategy, instead of performing virtual chores. The player controls his units by issuing general directives to his lieutenant robots, who direct their underlings accordingly. This eases the strain on players to strategize while carefully controlling their units in battle, both of which are taxing on a player in time-sensitive situation. Furthermore, this synthesizes with the feeling of being a leader of an organization of individuals, rather than the string-puller of mindless puppets.

In *Robogeddon,* the player must relate to and feel invested in his units, and must have fun playing as their leader. The visual artistic style attempts to create attachment between the player and his units. The robots were modeled and animated to resemble humans in both form and also in behavior, as some robots idly twiddle their thumbs, or catch their "breath" when they come to rest. Sound, however, plays a large role in creating an appropriate mood of cold, uncaring, inhuman determination. The player's machine army produces nothing but beeps and mechanical noises, whereas the human adversaries produce violent screams and cries when destroyed, and emotional comments when in combat. The player should maintain affection for his troops, but still capture the feeling of being a malignant overlord of destruction.

This report describes the technical implementation of the engine and content, with a particular focus on the development of artificial intelligence. The description of the overall artistic vision and style, as well as commentary on every art asset is also included. Finally, this report includes personal reflections by all members of the technical and artistic teams. The creators of *Robogeddon* discuss the challenges and solutions of development, their biggest successes and failures, their perspective on the *Robogeddon* player experience, and advice and insight for teams developing future games, or artificial intelligences.

Chapter Two: Gameplay Vision

2.1 Game Overview

Robogeddon is a casual real-time strategy game in which the player takes the role of a robot overlord attempting to exterminate humanity. The post-apocalyptic earth is inhabited by many different robot factions, each controlled by a robot-factory. The remnants of humanity have gathered in a small heavily defended settlement, slowly amassing arms to overthrow robotic kind. The humans use powerful war machines to attack and raid robot factions in order to gain materials to build up their settlement and reduce the robot threat. The player must conquer and assimilate other robot tribes to gather enough power and technology to exterminate humanity once and for all.

Robogeddon offers the strategic elements of competitive real-time-strategy games such as *Starcraft* or *Command & Conquer*, but does not apply the same frantic speed element that turns many gamers off. This is accomplished by reducing the quantity of player tasks, while still providing interesting strategic decisions, such as how to customize their troops, and what new equipment to obtain.

Players in *Robogeddon* get to experience the feeling of masterminding the destruction of the human race. Unlike classic RPG heroes, the player never "gets his hands dirty," instead maneuvering his minions to bring down his enemy from afar. Naturally, the computer opponent of the player assumes the player's traditional role, a powerful, rampaging force, wrapped into a single body. The sinister player must engineer its destruction.

2.2 Object of the Game

The player's ultimate goal is to amass a large and powerful enough force of robots to destroy the human settlement. The challenge comes from the way players can equip their robot forces. Each robot faction has access to a limited number of attachable parts. The player can attempt to gain control of tribes with parts, and add those part blueprints to his collection. These parts can be used as weapons, armor, or tools for the player's robots. However, the human raiders will attack and destroy other factions, denying the player those parts permanently. Combined with factors such as terrain and neutral tribe interference, the player will be constantly making difficult decisions about what is the best move to make.

As time goes on, and the human raiders become more powerful, and the settlement increases in size and strength, adding new defenses as well as producing new raiders. This steady growth affords the player the opportunity to win with a quick strike before the settlement has had time to grow, or try to outplay the raiders and gain power and strength faster than the settlement.

2.3 Setting

The game takes place in a post-apocalyptic world following some global war between humans and artificial intelligence. Much of the map resembles damaged or ruined industrial buildings, junkyards, or abandoned factories and warehouses. In the center of the world is the human settlement. The human settlement grows and changes as the game goes on, slowly adding defenses and weaponry.

2.4 Player Control

The player directly controls a number of robot-producing factories. Each of these can control up to 30 underling robots. The player cannot assume control of any individual underling robot, and must issue general commands to his troops via factories. The player can enter a “Blueprint Building Screen”, where he or she can choose the parts that affect a robot’s underlying behavior, allowing him to produce aggressive bots, defensive bots, e bots, or whatever the player feels will be most effective. A factory can send a broadcast to all the robots under its control (typically robots it has produced) and give them all a directive, which they will each follow in their own way, based on the AI chip the player gave the robots. Robots with different underlying behavior might use a different method to accomplish the same high-level goal.

2.5 Development Decisions

Gameplay Philosophy

We intended to develop a real-time-strategy game for the PC that is fun, challenging and unique. The game will test a player’s tactics and strategic decision making skills without constantly demanding rapid action. Players are challenged by advanced artificial intelligence, which reacts deliberately to each of the player’s moves. Players draw a sense of accomplishment by seeing their carefully planned strategies come to fruition, or by seeing a spur of the moment gambit pay off. The graphic and audio assets should attract the player and provide satisfying feedback for the player’s actions. New players will enjoy experimenting with customization as they familiarize themselves with the strategic elements, easing them into high-level play.

Platform and Engine

This game is built for the PC using Microsoft’s XNA framework. The PC offers the keyboard and mouse – a familiar control scheme for real-time strategy gamers. The mouse is the most familiar pointing device by far. It allows for players to easily perform the common action of selecting units, and directing orders towards a point on screen. It offers a high level of precision and allows for intuitive control – point and click interfaces are among the easiest to learn and use. A keyboard allows for a very large number of shortcut keys that can be used by more advanced players, which is useful for allowing advanced players to quickly perform actions and view the various interface elements. The PC also allows for higher resolution displays, which is useful for displaying the large number of game objects that could possibly occupy screen at one time. Finally, the PC allows us to use Microsoft’s XNA framework to construct the game engine.

XNA is a commonly used framework to develop games in the industry, so using it will give our technical team valuable real-world development experience. XNA allows developers more freedom and flexibility than most graphical based engines (such as Unity), since developers must build the engine themselves. At the same time, XNA provides many of the tools needed to create 3D games, which will eliminate much of the behind-the-scenes work traditionally needed when building a game engine. XNA can also be used to create Xbox Live Arcade games, so our technical team will indirectly gain some experience for console development.

2.6 Player Experience

The player takes on the role of a robotic overlord in a battle against AI-controlled “humans.” The player controls a robotic factory, capable of producing and controlling many smaller robotic soldiers. The player chooses how to outfit and arm his army, choosing from his supply of weapons and tools. The player can issue general commands to his robot-factory, which will distribute the orders to the robots it has produced. The player must accumulate enough arms and power to wipe out the remnants of humanity.

Other robot tribes populate the world, each operated by another robotic-factory. These tribes constantly wage war with each other and the player, attempting to conquer and assimilate one and other. Once one tribe has eliminated another tribe’s factory, the tribe is conquered and the player adds the tribe’s weaponry to his armory. Additionally, he gains control of that tribe’s factory. It can produce more robots, as well as control the robots it has constructed. As the player amasses power, he will gain control of more and more factories. Each one acts as a general, which controls one of his armies.

The humans have taken refuge in a heavily defended settlement. They guard the exterior of the settlement with a giant war-mech, which is capable of destroying many of robot-soldiers, especially when reinforced by the settlements defenses. The war-mech also attacks and destroys robots that come too near, and will wipe out an entire tribe, and all of that tribe’s weaponry, if given the chance. Although this eliminates one of the player’s rivals, it also denies him eventual control of that factory, and access to weaponry that might not be available otherwise.

As time goes on, the human settlement grows and adds defenses. The settlement is a powerful defensive structure and it takes a sizable robot army to destroy it, even early in the game. The player’s ultimate goal is to amass an army strong enough to demolish the settlement and get rid of the humans once and for all. The player can delay the growth of the humans by slaying them and denying them parts. Figure 1 depicts a possible thought process of a player.

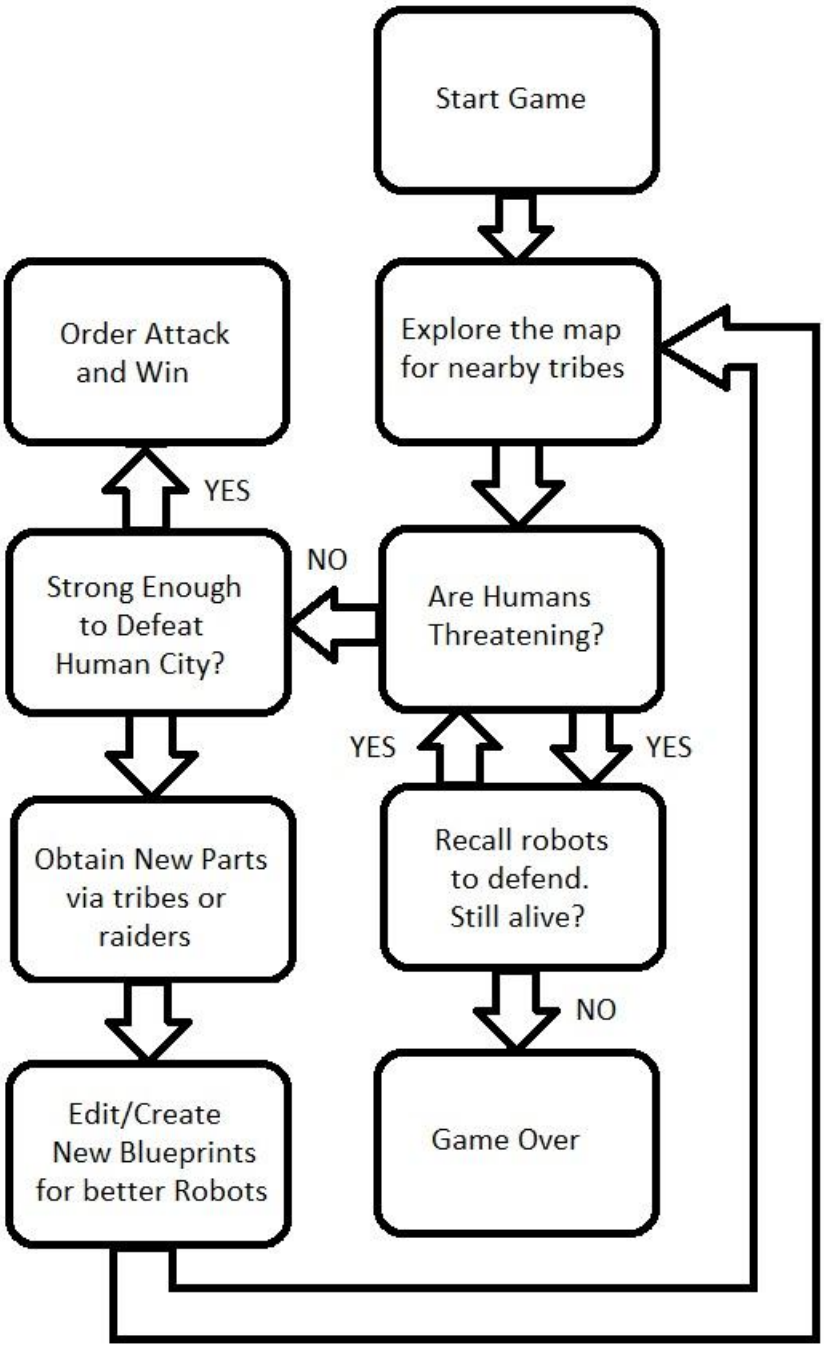


Figure 1 – Flowchart depicting a player's possible actions as they play the game.

2.7 Strategy and Fun

At the core of any strategic game is decision making. In turn-based strategy games such as *Chess* or *Othello*, gameplay consists of simply choosing where to move or place a single piece. The game is entirely planning and course and knowing how to react to the opponents counter maneuvers. Real-time-strategy games on powerful computers incorporate many more variables and varied situations, but decision making still remains the focal point of competitive play. In games like *Warcraft 3* or *Starcraft:*

Brood War, players are responsible for controlling each individual unit, which can grow to be over 100! Each of these "pieces" can take up thousands of different possible locations in the world, and pieces have different statistics that determine, in a complex fashion, which units fare better in a fight. Furthermore players can control and maneuver their army to a superior position. Players are responsible for maintaining production of their units by commanding buildings, and also must consider gathering resources to purchase these units. These elements cause these games to require strong mechanical skill to play at a reasonably high level, and so some players do not get to experience the "fun-factor" of strategic decision making. However, as skill rises, and players are able to more or less consistently maintain the various tasks of production, decision making becomes the key factor to victory. Knowing when to engage, what type of units to create, or when and where to build defenses become more important than maintaining constant production. This makes the game become much more dynamic and fun.

In *Robogeddon*, we attempted to minimize the mechanical skill required to play at a high level, and allow the player to focus solely on the fun aspect of decision making. By delegating production to the factories players do not have to worry about returning to their base to click through their production buildings and order each unit they wish to make. The challenge is replaced by deciding what parts to assign to robots, and the ratio of their production. By issuing general commands through the factory, mouse speed and accuracy are no longer important, but selecting an appropriate engagement maneuver rises in importance. Players are given a plethora of opportunities to make interesting decisions that will have an effect on the rest of the game.

Parts

Parts are the driving mechanic behind most of the important player decisions, so it is very important that they balance each other out. Typically balance in RTS games works by some kind of "Rock, Paper, Scissors" style counter system. Meaning some parts are highly cost-effective in some situations, but can be negated by specific measures, or are generally weaker when not in their favored scenario. Using the sniper rifle as an example; the sniper rifle has a long range and high damage, but extremely low attack speed. If an opponent is creating small numbers of powerful, armored seekers, the sniper rifle becomes an absurdly cost-effective weapon. Armor takes effect each time the unit is hit. So, rapid low damage attacks become less effective, while slow, high damage attacks are a good counter. Additionally armored units are slower than unarmored units, so sniper robots can more easily keep their distance. However, imagine the opponent instead produces a larger number of fodders with the magnet weapon. The sniper rifle might be able to quickly and easily take out a single fodder, but because of the overwhelming opposition, the weapon has no time to become effective. Additionally, the magnet weapon allows units to close distance quickly, making it very difficult for the sniper to keep distance. Much of the fun of *Robogeddon* is experimenting with different combinations of parts and robots, trying to create the most cost-efficient army to take out enemies.

Enemies

Because of the way parts counter each other, it is important that the player's opponent adapts to the player, and forces the player to continue modifying his strategy. Other robot tribes accomplish this by evaluating nearby threats (other tribes or humans), and deciding how best to use their parts. Humans work to counteract the player by targeting tribes with what they believe to be particularly useful parts.

Ideally the player and an opponent could have a complex multi-step encounter without ever directly engaging the opponent in battle. Returning to the sniper rifle example, where opposition is producing heavily armored seekers with double saw-blade arm attachments. The player observes this and counters it by producing fodders with sniper rifle arm parts. As the player begins to assault the armored bots from afar, the opposing tribe calls for a retreat of its troops, and begins to produce twice as many fodder bots with saw blades in addition to the seekers at a 2 to 1 ratio. Seeing the quicker fodder bots approaching, the player calls off his snipers and begins producing seekers with armor, to engage the fodder bots and tank while the snipers attack from distance.

Chapter Three: Game World

The game world in *Robogeddon* is viewed from an overhead camera angle, showing a post-apocalyptic ruined cityscape, destroyed by the passage of time. Ruins of a once industrious human civilization cover the landscape, with destroyed highways, abandoned mines, and still-radioactive power plants scattered across desolate fields. Robotic tribes salvage parts and scraps from the ruins and bring them to their factories to create more robots and survive – they even use pieces of their fallen comrades if needed.

3.1 Artistic Vision

When designing this game we wanted to go for a very post-apocalyptic feel, and as such we decided that we needed to make all of our game assets look like they had been worn and damaged during the absence of human care. The environment is made up of all kinds of half-destroyed and decrepit buildings, surrounded by a war-torn landscape that shows signs of heavy combat between humans and robots. The world is divided into four different sections of materials, Iron, Special Metals, Energy, and Scrap. Iron represents the materials that are found in the more urban zones of the map, which is made up of tall buildings and streets that are elevated off of the ground. Special Metals are located near the mining camp area of the map because these minerals are located deep underground and this is the only place where they can be accessed. Energy is represented by the power plant area of the map where the inhabitants of the world can forage for carbon rods and old turbines. And finally scrap is a resource that is centered on the junkyard area of the map where destroyed Mechs have been taken, and there are piles of rubble left over from the buildings that once stood there. The beings that remain in this landscape are battle hardened and are constructed from the few materials that remain.

The robots appear as rusty chunks of metal with guns and other weapons haphazardly strapped onto their arms and other body parts. The robots themselves appear to be humanoid with hugely exaggerated features. The head of the robot was based off of old steam shovel buckets, and as such they have an oversized jaw and eyes that are located lower down on the head so that it appears that it is hollow and can be filled. The rest of the robot is designed to be very simple and look like it has as few moving parts as possible in it, so that it is better able to cope with the deterioration from the elements. The effect that the environment has had on the robots is obvious from the copious amounts of rust that coat their metal hides and the eccentricities of their animations allude to the fact that long periods of time exposed to the world has damaged their processors.

The humans are housed within hulking metal husks to protect their soft flesh from the scathing attacks of the robots and the toxicity of their world. The simple monstrosity of their forms is meant to contrast

with the comical appearance and personalities of the robots that they are fighting to destroy. The point of the humans in this game is to be intimidating and make the smaller Fodder and Seeker robots to seem insignificant by comparison. The human's machines are more colorful than the robots to represent the fact that they were not simply manufactured, but they were made by someone who cared about what they looked like and put care into the appearance. Their animations are made to seem much more machine-like because the pilots are not quite as comfortable in the Mechs as the robots are within their own bodies, and thus they cannot move as fluidly.

Our art style adds to the feel of the game in that it makes the game more entertaining to see these cartoon robots and humans fighting in such a desolate and depressing world of destruction and chaos. The quirky design of our robots and the ways in which they move around and interact should feel lighthearted, and make this game not just another post-apocalyptic RTS. In fact it parodies the overused grungy and dirty games that fall into the post-apocalyptic RTS category, while still retaining its own personality.

The various land and weapon types of our game are done in different styles. This should be an entertaining experience for the player when they mix and match the parts on the robots in order to come up with completely new units. As they do this they will notice that not only are they mixing parts, but they seem to be slapping together a robot from almost mismatched parts, which serves to reinforce the scarcity of resources in our game and the fact that the robots are supposed to recycle themselves. The art style in general should serve to give life and character to this game, whose theme is the end of life.

When designing the characters and setting for this game, our artists looked at a wide range of material, which incorporated robots and post-apocalyptic wastelands in order to synthesize the final vision of the look and feel of the game. Since it is meant to be an almost lighthearted game set in a very dark world, the artists used cartoons and comics as the main inspiration for the characters' look, and the personalities, which they would have in game. On the other hand, the landscape was influenced by various battlefields and ghost towns in order to get the deserted and destroyed look we were going for.

3.2 Robots

Robots are by far the most common unit in *Robogeddon*. Every robot is part of a tribe that acts as a cohesive group, commanded by one or more factory robots. The ultimate goal of each tribe is to expand as much as possible to gain strength and to take control of the world. Tribes can attack each other and salvage their enemies' remains to increase the size of their own tribe. If the humans destroy a tribe, however, it is permanently gone, and the total number of robots in the game decreases.

The player has control over one tribe of robots and can customize the robots they produce at their factories. Their goal is the same as any of the other neutral tribes – to expand by destroying neighboring tribes, and to wipe out the humans in the world.

Inspiration

The robots were influenced by a mix of different media, most important of which are: *The Big Guy and Rusty the Boy Robot, Iron Giant, Castle Crashers,* and old timey steam shovel buckets. *Big Guy and Rusty* as well as *Iron Giant* were mainly for ideas on how to draw cartoony robots that could have personality.

Once we got to the stage where we would animate the models, we again went back and looked at *Iron Giant* for inspiration about conveying emotions through a robot's mostly static face.

Figure 2 - Example art for the robots' design. Left is *The Big Guy and Rusty the Boy Robot*; right is *Iron Giant*.

Castle Crashers was mainly used for looking at form and giving ideas on how the robots themselves should look on paper and in the game. The "soup can" heads of *Castle Crashers* ended up being the driving detail behind the original design of the Fodder, which the Seeker and Factory were then based upon.

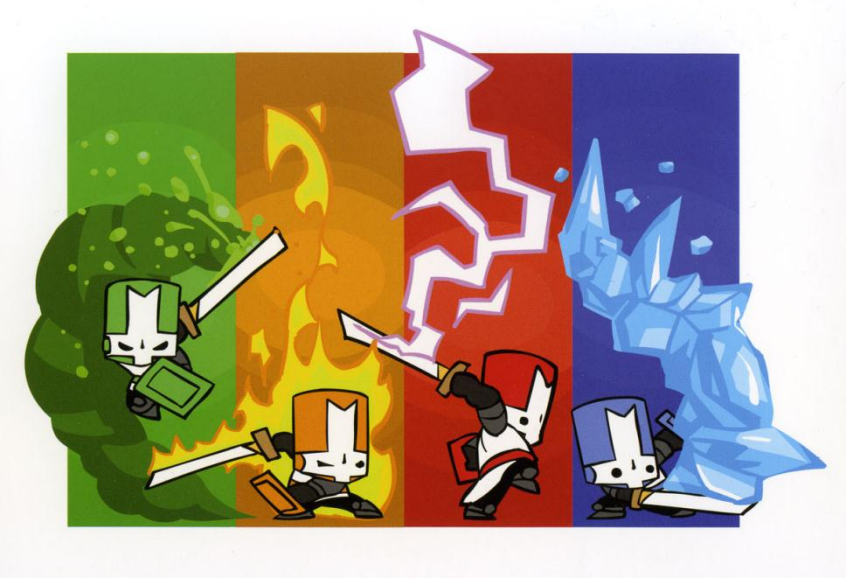


Figure 3 - Example art for the robots' design, from *Castle Crashers*.

The robots in our game were meant to look like they were built using older technology than we have today, and as such they are mainly influenced by the way steam shovels look. Specifically the buckets from the steam shovels which look almost like faces. Like "Mike Mulligan and his Steam Shovel" by: Virginia Lee Burton for example.

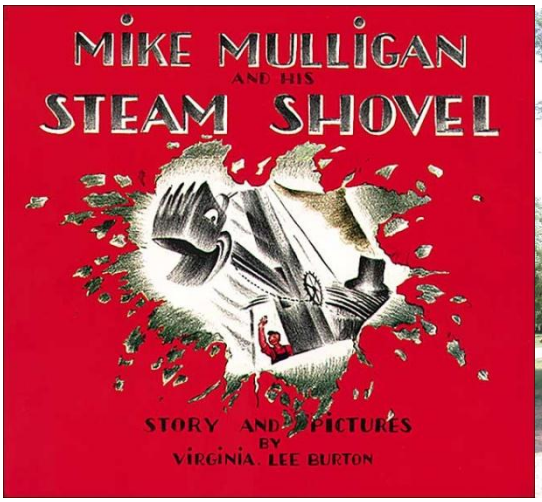


Figure 4 - Images inspiring the robots' art design.

There are three different robot frames in *Robogeddon* – Factories, Fodder, and Seeker. Factories are large, commanding units that control a group of Fodder and Seeker in combat.

Factory

The factory is the mainstay of any robot tribe's forces. It is unable to fight on its own. Factories constantly produce smaller combat robots, which are made up of a robot base (Fodder or Seeker) and a set of parts attached to each robot. The player can issue commands to a factory, which will then relay the command to all of the robots it has produced. Factories are the *only* unit that can receive commands. The only way for the player (or another neutral tribe's AI) to control robots is through their factories. This means that all of the robots under its control always act as a cohesive group that works together in combat.

Figure 5 – A Factory robot.

If a factory is destroyed, all of the robots under its control shut down and become essentially useless. If a tribe's last factory is destroyed by the enemy, that tribe is eliminated. If a factory is destroyed by an enemy tribe's robots, that factory is salvaged by the enemy tribe. It is then resurrected and placed under the enemy tribe's control. If a factory is destroyed by humans, it is destroyed permanently.

Parts

When a factory produces robots, it chooses one of the two robot frames as a base, and then can add parts onto these frames to add different battle capabilities to each robot. Adding a part to a robot adds to the robot's total production time. A factory can use any part owned by its tribe when customizing its robots.

There are three types of parts – Arm, Body, and Head. Arm parts give the robot a new weapon, such as a Gatling gun or a saw blade. Body parts add defensive capabilities to the robot, such as heavy armor or shield generators. Head parts grant the robot other miscellaneous effects, such as goggles for increased damage range or a bomb for a powerful self-destruction attack. A full list of parts in the game can be found in Appendix D.

Fodder

Fodder are small, disposable robots that can be created very rapidly by factories. They are outfitted with one arm weapon and one head part. Fodders are small and not very durable, and will go down much more easily than the larger Seekers and Factories. They are very quick to produce, however, and deal damage much more efficiently than Seekers.

Figure 6 - A Fodder bot with no parts attached.

Seeker

Seekers are the heavy hitting tanks of the robotic army. They can be outfitted with two arm weapons, one head part, and one body part. Seekers have high health and armor, can attack with both weapons, and are the only robots that can make use of body parts. They are intended to soak up damage and be on the front lines of combat. However, they take a very long time to produce, and will not deal damage as quickly as an army full of Fodder will. Because of their size and ability to control multiple part attachments Seekers take up twice the amount of Factory RAM as a Fodder.

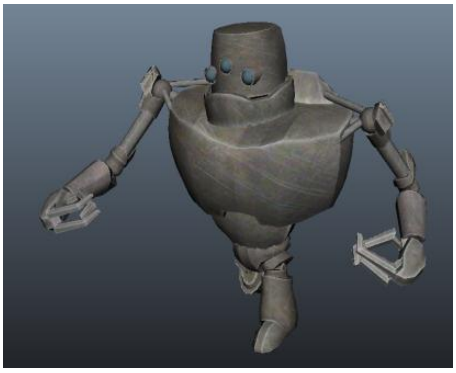


Figure 7 – A Seeker robot with no parts attached.

3.3 Humans

Somewhere in the game world is an active human settlement that serves as a base of operations for humans' robot destroying operation. The settlement is well defended with walls and turrets, and requires a powerful assault from robots in order to be destroyed. The player's robotic tribe will typically be unable to even consider destroying the settlement before the first 10 minutes of the game. The settlement also slowly expands over time, building more walls and turrets and becoming generally more and more difficult to siege as it gets into the later stages of a game. If the settlement's main power plant building is destroyed, the humans are defeated and the player wins the game.

Figure 8 - The human settlement. Turrets are visible on the right hand perimeter.

The humans also command a few very powerful war Mechs, which patrol around the settlement and occasionally attack enemy robot tribes. As the humans destroy robots, they grow more and more powerful, so it is in the player's best interest not to delay when expanding their tribe. If the humans destroy all the player's factories, the player loses the game.

Mechs

The humans fight using mechanized battle suits, which have their own steam engines to power the weapons and allow them to move and collect more materials to power the settlements. The suits themselves have a human pilot who chooses the weapons that they are equipped with and controls all other functions of their machine. The loss of a mech to a settlement is a devastating blow because the amount of resources required to create one of these machines is so much that the settlement would have to limit the power to only systems required to survive, which would mean shutting down defenses, and if they did that the robots would destroy them.

Figure 9 – A human's mechanized battle suit.

Inspiration

The human mechs in our game were conceived as machines similar to Gundams, in that there is a human being inside controlling the robot and piloting it like it is a second skin. The original idea for the humans was to have them be bristling with weaponry and driving very high tech suits of armor meant only for destroying robots. But as we looked at different types of mechs we started to lean more towards the cartoony side, not only to keep with what we had already solidified as the robot art style but also to make them seem more human than machine. As such we went with a more toned down approach with the mechs, they were devoid of mouth and were modeled to be much more androgynous than the robots so that they would seem more like armor than an actual character like the robots. Some of the things that they were influenced by were robots from the television show *Futurama, The Iron Giant, Gundam,* and gorillas.

Futurama's cartoon robots were a big influence when designing the simple frame of the human mech. We wanted to focus less on the mechanisms in the model and more on its actual movements and features much like the robots from *Futurama* who usually do not exhibit great internal workings or complicated locomotion. We also wanted to create a simple recognizable silhouette like many of the characters of this show possess.

Figure 10 - Images from Futurama, inspiring the humans' artistic design.

Again *Iron Giant* in conjunction with *Gundam* was mainly to influence the form of the mechs, but this time more attention was paid to the way that they handled weapons and attached them to themselves. Another key part in designing the human came from *Gundam,* the location of the human within the war machine. We decided to follow the example of *Gundam* and place the pilot in the chest, but instead of giving the machine eyes and an array of sensors we simply placed a windshield in the chest for the pilot to see through, to keep with the older style of tech in our game.

Figure 11 - Images inspiring the humans' artistic design. Left and middle are from *Gundam*; right is from *Iron Giant*.

We wanted to give the humans an ape-like feel so one of the main influences when designing the human ended up being gorillas. The overall frame of the mech is what ended up being the main thing that was affected by this decision, as well as the gait and the way it moved when animating. We wanted to draw the connotation that the robots thought of the humans as a lower life form that they evolved from, much like we view monkeys and apes, so we subtly altered their appearance to be more simian in accordance with this theme.

Figure 12 - Images of gorillas, inspiring the humans' artistic design.

The final product ended up looking like a sort of 1950s metal recreation of a gorilla with a human riding in the chest, which is much like what we were originally going for.

Turrets

The settlement is made up of a number of defensive buildings, the most dangerous of which is the turret. Turrets are stationary units that fire on any enemy robots that come nearby. They are heavily armored and have a very strong attack, meaning that a full scale assault from a large army is needed to take one down. The human settlement has a number of these on the perimeter of its base. As the settlement expands, turrets will be built to defend the newly created areas.

Figure 13 – A human turret.

3.4 Environment

The playable map is fairly large and a robot will take 1-2 minutes to cross from one side of the map to the other. The terrain is split up into patches of different terrain types, each with its own unique look and resources. Each part that can be outfitted onto a robot is associated with a specific terrain type. When a factory is producing units on top of a specific terrain, any parts that are native to that terrain are much easier to manufacture, and robots using those parts will be produced much more quickly. Therefore, the player must choose which terrain types to occupy and defend when creating their army.

Factories are distributed evenly across all the other areas of the map. Each of these factories starts as a neutral AI-controlled tribe, with a random set of parts. These tribes will slowly produce robots and occasionally attack each other, but cannot expand beyond owning one factory. Neutral tribes typically cannot defend well against human attacks, and the humans will eventually overtake them all if the player does not intervene. The player is given one factory to control and several initial parts, to take

control of other neutral tribes' factories. When the player takes a neutral tribe, they receive all the tribes' parts and can immediately start putting them on all newly produced robots.

Terrain Descriptions

Each different terrain type in the game has a distinctive look so the player easily knows which areas are which. Terrain types are made up of a base terrain texture and a set of doodads which can be placed on that type of terrain.

The four different terrain types were influenced by a number of different sources, but first let's talk about the overall style of the landscape. We chose to have this game be post-apocalyptic, and because of this we wanted to have the world be filled with piles of rubble, half-demolished, structures, and pockmarked with craters. The best inspiration for this sort of world was the trenches and war torn cities of World War I and II and the Lebanese Civil War.

Figure 14 - Artistic inspiration for the design of the environment.

The specific land types were similarly inspired by real world facilities, since our game is set in the present day.

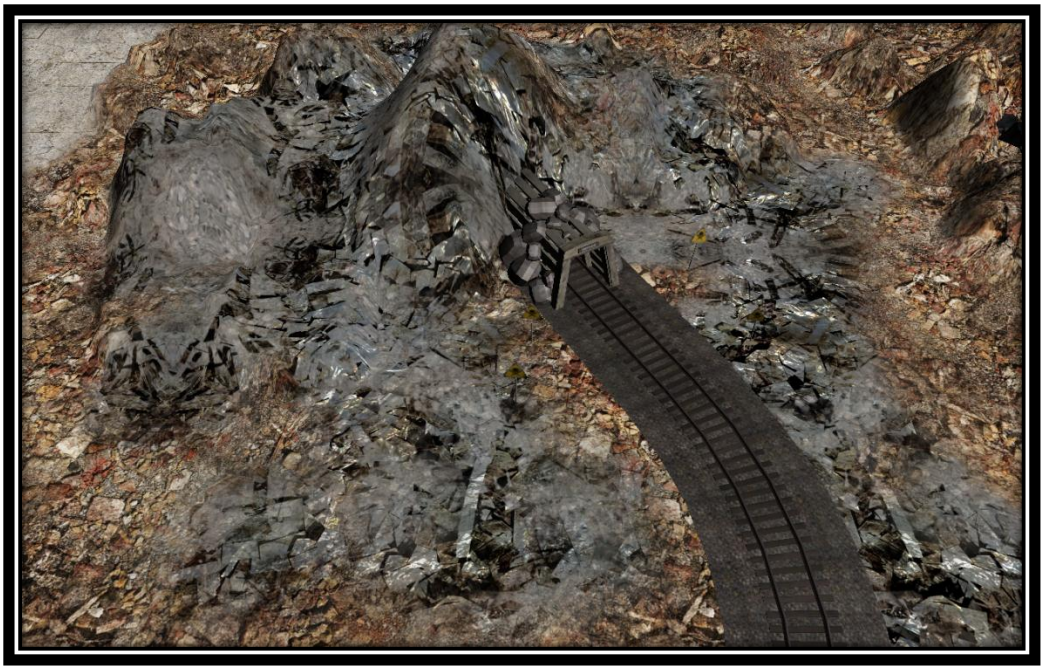


Figure 15 - In-game screenshot of the Special Metal terrain.

Special Metal

The mine area (or Special Metal zone) is modeled after an old coal mine and about a quarter of the map will consist of this zone. In keeping with the mine theme, the doodads and buildings from this zone center on piles of minerals, mine openings, rocks, railroad tracks, and conveyors. There will also be pieces of destroyed mechs and many craters in this zone because it is supposed to have seen a lot of combat over the precious minerals it contains. The map will be made to have deep valleys in this zone so that it seems like it has been mined out and made so that vehicles can drive on it to pick up what the mine produces. Large mineral piles will be scattered around this zone near conveyors to make it seem as though work was abandoned abruptly. There will be mining machinery all over the map in this zone to keep with the theme as well.

We chose to include this terrain because it presented a great opportunity to show off the flexibility of the terrain editor as well as having piles of human mech pieces lying around all over the place to show off pieces of the model more. It also allowed us to create weapons which could look higher tech because the availability of a special metal was a justification for advanced tech.

Figure 16 - Inspiration for the Special Metal's design.

Figure 17 - In-game screenshot of the Energy terrain type.

Energy

The power plant area (or Energy zone) is modeled after a nuclear power plant, and about a quarter of the map will consist of this zone. In keeping with the nuclear power plant theme, the various doodads and buildings from this zone center on silos, fuel tanks, and toxic waste. The different buildings have various levels of deterioration, ranging from almost pristine, to horribly damaged. In this area there should be pools of toxic waste near barrels, and there should be some craters, especially near the main area of the plant, near the smoke stacks and fuel tanks. There will also be warning signs and electric fences to make it seem like this location is very dangerous to the player's units, even though it contains special parts for their robots.

Figure 18 - Inspiration for the Energy terrain type's design.

We chose to include this terrain because it presented very interesting opportunities for modeling and texturing, such as the nuclear smoke stacks and the radioactive goo. This terrain type gave us the ability to create high powered energy and electricity weapons for our game to vary the style of guns and utility items somewhat, and the presence of a nuclear facility gave us justification for robots being able to salvage high energy batteries to power tesla coils and force fields.

Figure 19 - In-game screenshot of the Iron terrain type.

Iron

The city area (or Iron zone) is modeled after the downtown of a city, much like Worcester, and about a quarter of the map will consist of this zone. To make the cityscape seem like it was once actually part of a city, the buildings will consist mostly of apartment-style buildings in various degrees of destruction. There will be roads all throughout this part of the map to make it seem like a city rather than just a bunch of buildings in the middle of nowhere in the map. The doodads mainly center around things that are commonplace on the sidewalk of city streets, like streetlights and manholes. The ground will be less cratered here to reflect that there were once people who lived here that the artillery and bombs were trying to avoid.

Figure 20 - Inspiration for the Iron terrain type's design.

We chose to include this terrain because the original concept for this game was that humans and robots are fighting in a ruined cityscape, so naturally the original idea had to be preserved. We also wanted to include this terrain because it would give us ample chance to create highly detailed destroyed buildings and give a little more life to the former human civilization that supposedly inhabited our game's world (much like the city of Stalingrad in World War 2).

Figure 21 - In-game screenshot of the Scrap terrain type.

Scrap

The junkyard area (or Scrap zone) is modeled after a junkyard or landfill and about a quarter of the map will consist of this terrain type. There will be no buildings in this zone, rather it will be filled with broken down machinery and scrap heaps and the ground will be littered with trash. There should be a

disproportionate amount of rusting pieces in this section of the terrain to make the decay more obvious, and the doodads themselves should look decrepit.

We chose to include this terrain because it gave us an opportunity to include very dirty and grungy models in our game, which was one of the things we were originally trying to get across. This terrain contrasts with the more high tech areas like Energy, while also giving the world much more depth through variation. It also gave us the opportunity to make almost comic weapons like the scrap cannon which the player could give their robots, with the justification that they picked it up in a junkyard.

Figure 22 - Inspiration for the Scrap terrain type's design.

Chapter Four: User Interface

The user's interaction with the game is separated into three parts: The main menu, the world view, and the production menu. These screens were designed to be intuitive and familiar to lessen the amount of time it takes players to learn how to play. This was accomplished by using Human computer interaction techniques and design.

4.1 Main Menu

Figure 23 - *Robogeddon*'s main menu, shown when the game is first run.

Though most of a user's time is spent playing Robogeddon, it is necessary for the user to be able to perform various other tasks. In order for the player to be able to accomplish these tasks, the application that hosts the game must feature a menu system, providing the user with all the required options needed. This system has been designed to direct players' eyes top to bottom.

New Game

This option was placed at the top of the screen as the most important item. This encourages new players to try the game with default settings before playing with options or just quitting the game. Once clicked, the player will be prompted to load a map through a standard Windows open file box.

Options

Here the player is able to adjust various configurable game options, which are stored in a "player profile," a file that is saved on the user's computer and read by the game. The game currently has a single default player profile that it loads when the game begins. The various menu items here adjust the values stored in this "default" profile.

4.2 World View User Interface

This screen is responsible for both letting the player interact with the world, and giving the player all the information that they need to play the game. The screen was designed after many modern real-time strategy games to make the interface familiar to the player.

Camera Control

Players of real-time strategy games have come to expect a specific way of viewing the game's world, and controlling this view. Typically RTS games provide a 3rd person camera that views a portion of the game world from a raised, isometric perspective. The camera is usually far enough away from the world's surface or units such that many units can be seen simultaneously. Typically the camera can zoom into or away from its look-at point by scrolling the mouse wheel. The camera is usually "slid" around the world by moving the mouse to the edges of the game's window. In order to stay true to the expectations of the genre, Robogeddon follows these standard practices.

Interaction with the World

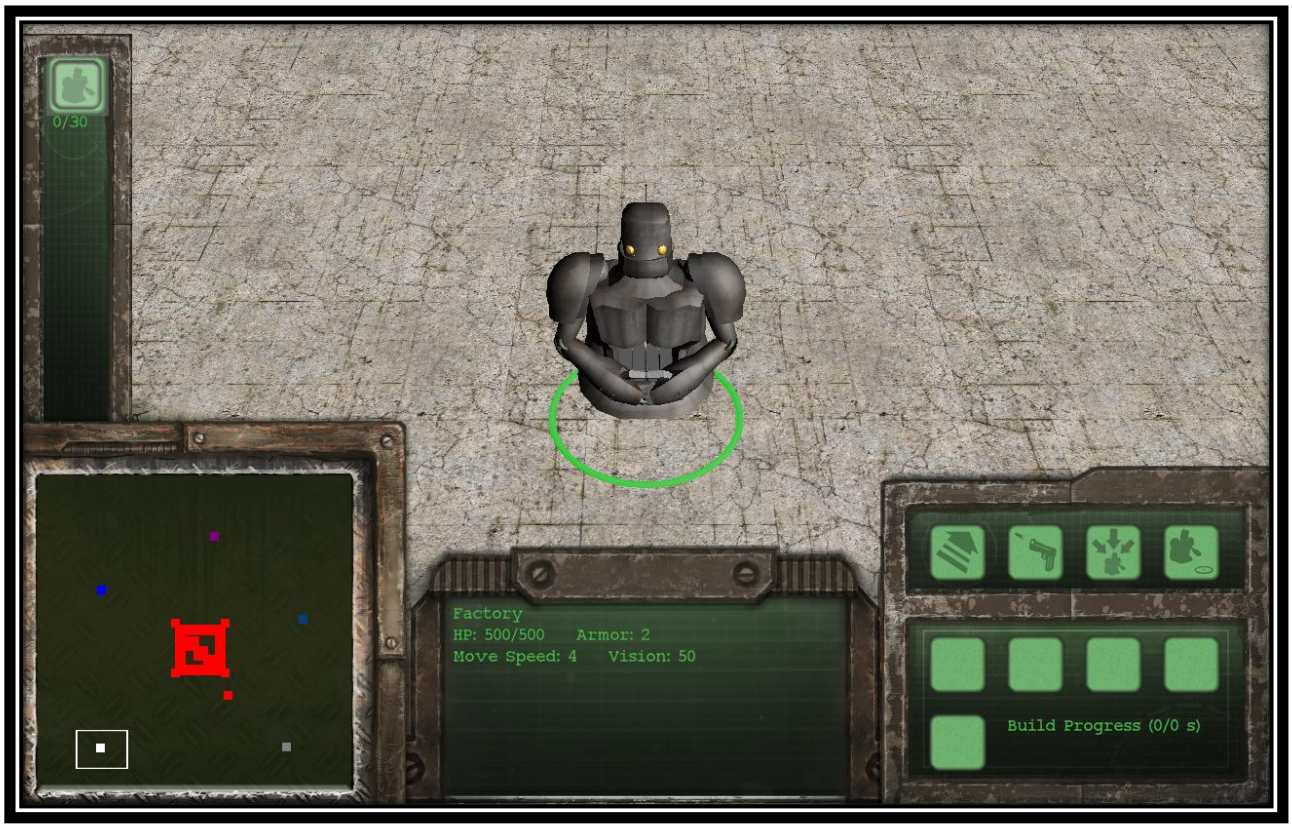


Figure 24 - In-game view of *Robogeddon*.

The interface for the user to interact with the world and their units is quite simple. There are four parts to the visible interface:

- Factory quick select (left)
- the mini-map (bottom left)
- stats (middle),
- factory blueprints and the commands area (right)

The user will interact with the world with their mouse and can select factories that they control as well other units to see their stats. The mouse is also used to interact will all the buttons in this screen.

Factory Quick Select

All factory units controlled by the player will display an icon on this tab. If this Icon is clicked by the mouse, the corresponding factory will be selected. A double click will focus the camera on that factory. Below each icon, the factory's RAM, or the max number of bots it can control, is displayed. This tells the player how many bots they have, and how many more they can have. In many real-time strategy games, a quick select UI element for important units is displayed in this same screen space. Putting the factory quick select in that spot allows other RTS players to quickly understand what the panel is and how to interact with it.

Mini-Map

This map will be a simplified top down view of the entire map that shows where the camera is currently looking, where the player's units are, where the player's factories are, and where visible enemies are. If the player clicks on the map, they will move the camera to the position clicked on. This will allow for quick camera movements when the player needs to see what's going on in a certain area that might be far away.

Stats

The stats panel displays all important statistics of the currently selected unit. This element is necessary for advanced strategy development in the game. It allows for players to analyze specific information about their customized units at any time.

Factory Commands and Blueprints

The factory commands are the main interaction between the player and their units. These commands control every unit produced at the selected factory. These command buttons work in a similar way to most RTS games again allowing players familiar with this type of game to pick up the controls quickly. Below the factory commands are four slots to hold blue prints. These blue prints are created in the unit customization screen and dragged into these slots. From left to right, the factory commands are the following:

- Move robots
- Attack with robots
- Move Factory
- Retreat

Move Robots Command:

After selecting this command, the player clicks on any location. The selected factory will then intelligently subdivide its underling robots into groups based on their proximity to each other. These groups will move together towards the clicked location.

Attack with Robots Command:

After selecting this command, the player clicks on any location. The selected factory will then intelligently subdivide its underling robots into groups based on their proximity to each other. These groups will move together towards the clicked location, but should a robot see an enemy while this command is being executing, the robots instead run towards the enemy, attacking anything they see.

Move Factory Command:

After selecting this command, the player clicks on any location, and the selected factory will move to that location. This command will not interrupt other commands.

Retreat Command:

As soon as the player clicks this command, all robots controlled by the selected factory will immediately ignore enemies and run towards their factory. This command serves as a shortcut, allowing the player to quickly retreat from an attack.

4.3 Production Screen

The production screen is where the player creates unique units with their acquired parts. This screen must be easy to use and display visually what is changing about the bot when parts are swapped in and out.

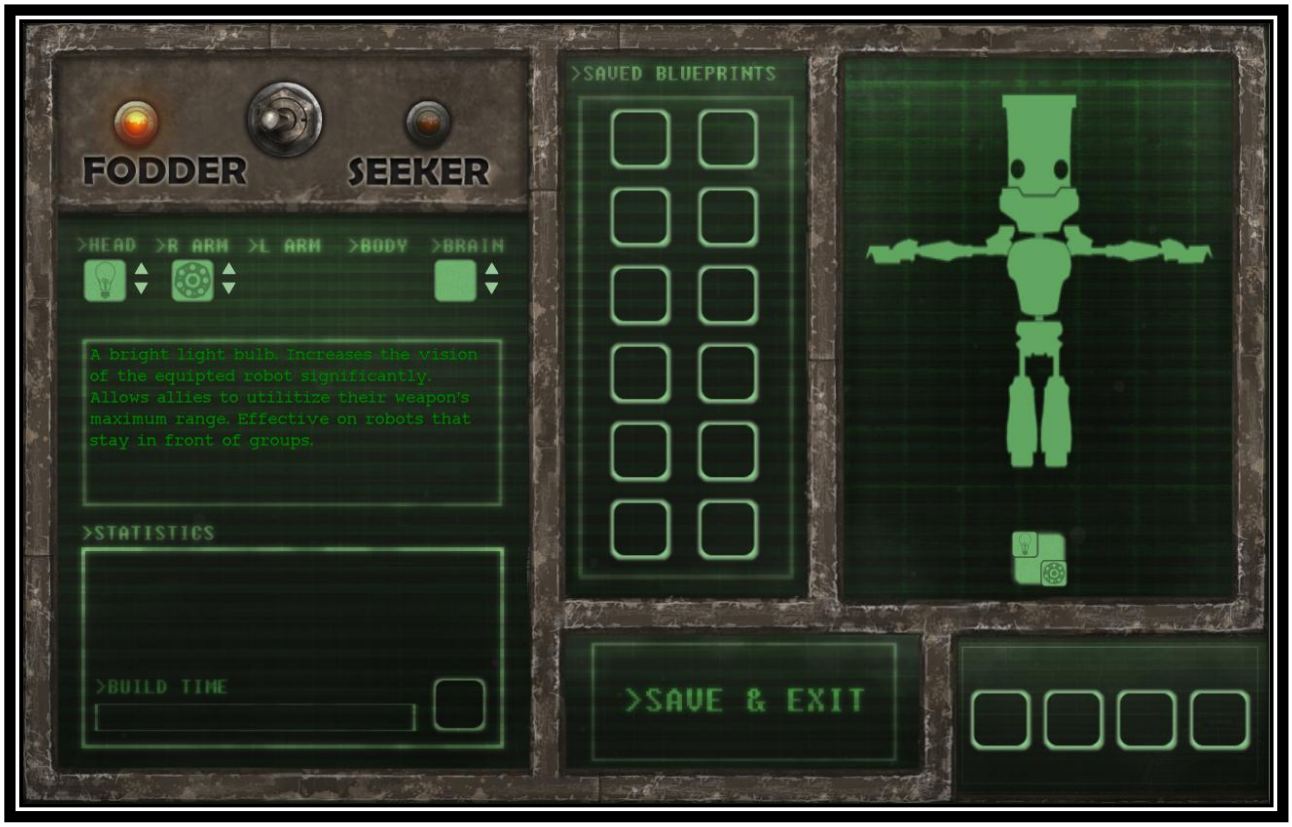


Figure 25 - Factory production screen producing a fodder, as seen in-game.

This shows the unit customization screen. Here the player will interact with a panel of buttons, switches, and draggable items to create their custom robots. The player will see a nicely rendered version of the model as the parts are being attached on the right side, as well as the stats for the units so that they can see what each part affects.

At the top of the screen will be a switch to select which model of robot is to be used. This screen is what the interface will look like when the player is creating a light model robot (fodder). This style of bot will be able to hold two parts, one head part and one arm part. The player will use the arrows to select parts for each of these spaces on the robot. While they are doing this, the model, stats, and part description right above the statistics will be constantly updated to display what the player is doing.

When the player switches to the seeker model, the screen will add the body and second arm options. The stats and model will be updated for the heavier bot.

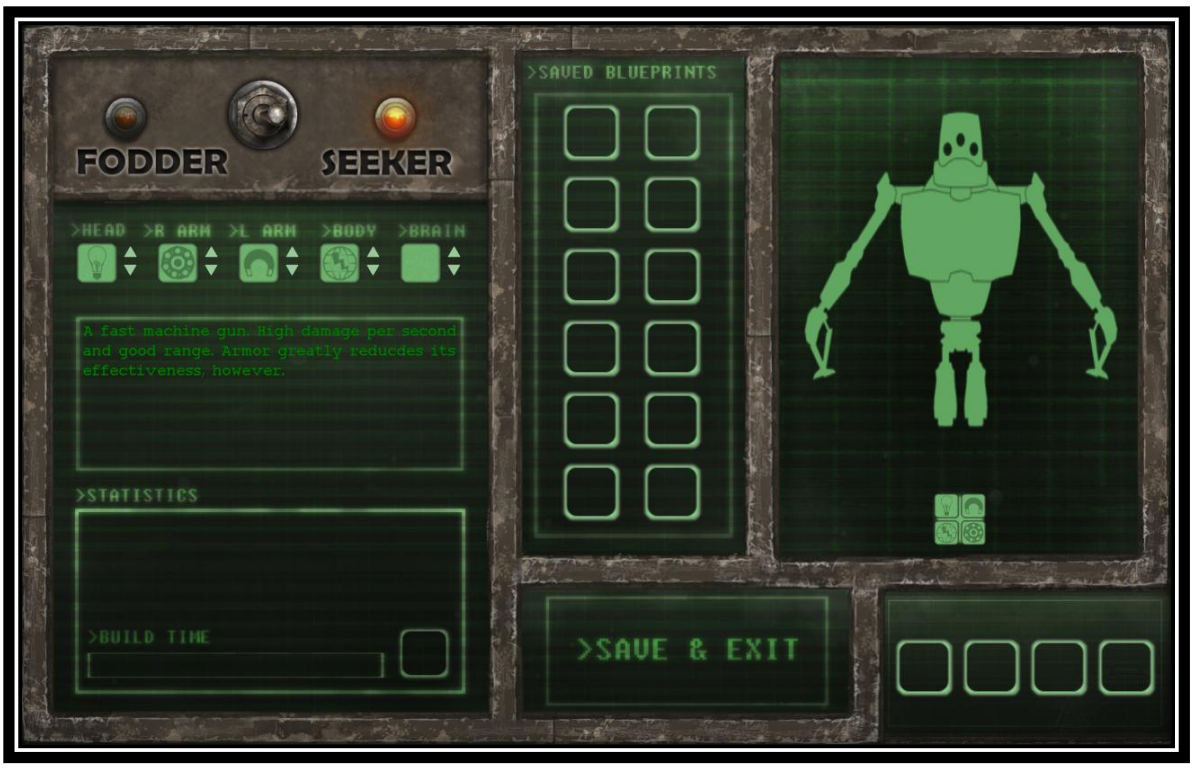


Figure 26 - Factory production screen producing a seeker, as seen in game.

This screen will allow the artists to show off the robots and parts. The design of this interface follows basic practices of human computer interaction and should be useable without instruction. Switches are meant to be switched and can only be one way or the other. The Arrow buttons let the user know that there are multiple items to look through. Also, placing the model choice at the top of the screen lets the user know that it should be done first. This interface was designed to be simple, easy to understand, and easy to use.

4.4 Interface Design Decisions

This interface was designed to be as simple as possible while giving the player all the control that they need to play the game.

A lot of the decisions made when designing the interface had to do with making the controls intuitive. Most of these decisions were made using affordances. Any interactable object has an affordance of doing something such as door knobs have the affordance of being turned and buttons of being pushed. All buttons and objects in the interface use this to make the interactions very intuitive. The interface is made to be conventional and familiar.

Other similar games keep the minimap in the bottom left and commands to interact with on the bottom right. This is what was mainly considered when designing this interface. Making the interface conform to common interface design of similar games makes the controls familiar and save the player some time learning where everything is.

Chapter Five: Artificial Intelligence

Artificial intelligence was a central theme in Robogeddon, as it took an important role in the game's lore. More importantly, it was an essential element of gameplay. A key mechanic of Robogeddon was that players could only issue commands to their army through factories. Players could not select and give orders to their robots directly. Instead, they were forced to rely on the game's artificial intelligence to liberally interpret the command given to a factory, and intelligently assign that factory's robots specific actions in order to accomplish the player's wishes.

The team realized early in development that players would need to learn to work with the AI in order to win. However, the team also realized that relying on AI could be frustrating for players. If the AI did not act intelligently, the player might become frustrated that he couldn't simply select and manually control his robots. Because of this, an extensive amount of effort went into creating the robotic AI to ensure that the game was fun for the player to work with.

The game's design also called for an intelligent opposition to be the main challenge of the game – the humans. The design stated that much of the challenge of the game would be outsmarting the humans' strategies, so the team thought it would be very important for them to appear to be behaving intelligently, and adjusting their actions based on the player.

Original Design Goals

Our team had a number of requirements and constraints that the different units in the game needed to follow in Robogeddon. Some of these requirements were obvious from the start, while others were revealed as important later on in development. The following is a list of goals that were either recognized during the initial design phase or emerged during development.

- Robots must be smart enough to act by themselves and attack enemies, even when not receiving commands from factories.
- Robots must be able to quickly and effectively get into formations most effective for battle.
- Groups of robots must be able to move together as one cohesive unit and path around the battlefield effectively.
- Robots should have their behavior in groups and as individuals influenced by their parts, in particular, their "brain" chip
- Factories must be able to manage multiple groups of robots at once, in case their robots are spread out.
- Neutral tribes' factories must be able to act on their own and occasionally attack neighboring tribes to ensure that the game is action-packed from beginning to end.
- Humans must recognize when they are succeeding in battle and when they are failing, and must choose where to level up and when to run away and heal.
- Humans must develop their skills in an effective way in order to counter the challenges they are currently facing.
- Humans must work together to create the most powerful fighting force possible by supporting each other.

Implementation

The team originally divided the AI needed for Robogeddon into three blocks: the robot controlling AI, which controlled everything that happened from when a command was issued to when it was complete, Neutral Tribe AI, which would control the neutral tribes high-level actions, and the human player AI, which would control the human settlement and their war machines. Development began on the first of these three blocks of AI, as it was the most essential.

In order create the robot-controlling AI, the team made several iterations before finding a design that seemed to cover everything that was needed. In the end, the structure agreed upon was multiple levels of intelligence. Unit tasks, group tasks, and factory-level commands were all needed in order to achieve the desired results.

Unit Tasks

The unit task is the lowest level block of the entire AI system. A unit task is a single, very short task that allows a unit to accomplish a very small goal. A unit task might be a step in a single direction, or one attack against a target. Robots can either receive unit tasks from a higher commanding task, or they can issue commands to themselves if they are not currently receiving any commands. In addition, a unit task can be made up of other unit tasks if a slightly more complicated function is needed.

Originally Unit Tasks were written as Robot Tasks, and only worked on Robots. This was an artifact of the decision to only work on the robot-controlling AI at first. However, later in development, the team was able to generalize this code so that any unit, such as the human war machines, would be able to utilize the same basic low-level tasks.

Listed below are the different tasks available to units:

Ignore	Stand still and wait for further instruction. Ignore enemies.
Step	Move in a direction for a short time.
Attack	Attack a target.
Wait	Wait a short amount of time before taking another action.
Patrol	Follows a path indefinetely, stopping to attack any enemies that approach
Hold Position	Stand still, but engage nearby enemies without moving if possible.

Table 1: Unit Tasks

Group Tasks

Robots seldom act on their own – they are almost always part of a larger group of robots all working to accomplish the same goal. A group task is a task assigned to a group of units in close proximity to each other. The group task can assign unit tasks to its individual units, and it is informed whenever a unit finishes a task in order to give it further instruction. Group tasks, like unit tasks, can be made up of other group tasks to allow for more complicated actions.

Move	Find a path to any given point on the map and move the group there, keeping the group together as they go. Ignore enemies.
Idle	Puts all the robots into idle states. In the final game build this was not used.

Meander	Instructs the robots to randomly wander around their current area, but will allow them to attack enemies that approach.
AttackMove	Move group towards a given location, but if enemies are seen, will transition to a AdvanceAttack task.
AdvanceAttack	Move towards an enemy group and instruct units to attack if they are in range.
GetInFormation	Gather units into a battle formation, placing units appropriately at the front or back lines. In the final game build, this was unused.
GatherAndGo	Gather a group into a formation, then move them to a target. In the final game build, this was unused.

Table 2: Group Tasks

Factory Commands

The Factory Command is way that a player directly influences the actions of his units. Nearly all of a player's meaningful actions are issuing factory commands to their factories. These commands are interpreted by the factory, which splits its robots up into different groups, issuing each one a group task as is appropriate for the given command.

The game's design originally called for many different commands. For instance, instead of just one attack command, the player would have options such as, "Attack: Pincer" or "Attack: Harass" or "Attack: Arc". During development, however, the team had to simplify it's goals by reducing the set of available command to the simple set given below:

Move	Form robots into groups based on position and move them to a target.
AttackCharge	Form robots into groups based on position and move them to a target, engaging any hostiles encountered on the way.
Retreat	Send all robots immediately back to the factory.
MoveFactory	Move only the factory to a different location.

Table 3: Factory Commands

Human Strategies

Development of the human settlement's strategies began later than the team had hoped. Fortunately, the team was able to reuse much of the same AI components as the robots used. Humans used Unit Tasks exactly like the robots did. This handled the lower level of control for the human war machines, but it did not help with the strategy component of their AI, which was a key part of the game in the original design.

To keep the project on track, the team chose to scale back the goals and expectations for the humans. In particular, any notations of the human war mechs "leveling up" or gaining new abilities over time was discarded. Instead, a simpler AI that could leverage existing resources to their fullest was created. The new design of the humans included the following:

- A single centralized human settlement that grew over time at a fixed rate
- Powerful war mechs that would be slowly created from the power plant structures in the settlement throughout the game
- A simple heuristic to gage the human's defensive and offensive capabilities compared to the players, and choose a simple defensive or offensive strategy based off of this calculation

Though the team regretted the removal of the original, complicated design, these changes were necessary in order to have a more complete, playable game on time.

Conclusions

The Artificial Intelligence for Robogeddon was originally recognized as a potential source of issues that could cause the game to fail. Due to the nature of the control scheme, the most important AI was the robot controlling code. To create a fun, challenging game, however, the team predicated that an intelligent, reactive human opponent and perhaps intelligent robotic tribes would be needed. Unfortunately, more playtesting of the "final" build of the game, where most user interface issues were finally resolved, would be needed to draw any conclusions regarding this hypothesis.

The following table maps the AI objectives to their status at the conclusion of the project.

Robots must be smart enough to act by themselves and attack enemies, even when not receiving commands from factories.	This objective was primarily accomplished. The final hierarchical design was created to make this very easy to obtain.
Robots must be able to quickly and effectively get into formations most effective for battle.	The team worked on formations for groups of units for a long time. However, eventually the team decided that "hard" formations created more problems than they solved, and it was decided that proper formations would not be used.
Groups of robots must be able to move together as one cohesive unit and path around the battlefield effectively.	This goal was met by taking the flocking principles into account when determining where a robot should move.
Factories must be able to manage multiple groups of robots at once, in case their robots are spread out.	This was accomplished by writing code that allowed factories to intelligently split their robotic underlings into multiple groups, which then received independent but cohesive group tasks.
Neutral tribes' factories must be able to act on their own and occasionally attack neighboring tribes to ensure that the game in action-packed from beginning to end.	This was accomplished by writing a simple Neutral Tribe player AI that used a heuristic to determine its relative strength as compared to neighboring tribes and the humans. With this and some randomness, the tribes were told to periodically attack neighbors, weighted by their strength estimates.
Humans must recognize when they are succeeding in battle and when they are failing, and must choose where to level up and when to run away and heal.	The late redesign of the humans meant that this objective could not be met. However, the heuristic created to weight the humans' relative strength does meet an equivalent objective for the new design.
Humans must develop their skills in an effective way in order to counter the challenges they are currently facing.	This objective was not met. The "ability" system originally planned for the human war mechs was cut from the game.
Humans must work together to create the most powerful fighting force possible by supporting each other.	Originally the human war mechs were designed to function somewhat independently, and the team thought it would be a later goal to get them to work together. However, due to the changes made, the humans implicitly do work together since a single source of intelligence controls all of them.

Table 4: AI Conclusions

Chapter Six: Transmedia Storytelling in Games

6.1 Introduction

Video games have never been perfectly suited for telling particularly detailed stories; there is always something that the player will miss unless it is shoved down their throat. If a game tries to advance story by placing optional interactive objects throughout the world the player will pass over them and miss pieces of the story. When games try to tell a detailed story either through mandatory text or video clips, they end up being more like a book or a movie than a game because they take time away from the gameplay. Since gameplay is what makes a video game a unique genre, some of the immersion of the game is taken away if this gameplay is broken up by gratuitous amounts of text and video. This is where transmedia story materials come into play, because mediums can be chosen based on what part of the story they are suited best to tell.

More research into transmedia storytelling in games is needed so game developers can more efficiently blend the story of the game and their transmedia material. More research would also reveal what kinds of things in transmedia material are best used for advancing plot, teaching the player about the world, and developing characters, and also what kinds of things draw more players to the game. More knowledge about the way that transmedia material works to help players understand the games that they are playing may make their use much more widespread and varied. This is important because it may help game developers find a much more efficient way of telling a story with their games, and maybe help them become a true art form.

I employ generic rhetorical criticism in my research on transmedia material in order to efficiently and accurately analyze my selected readings. I explore one type of transmedia material in particular in my research, comic books, more specifically comic books based off of games. I want to answer a few questions in my research:

- How can transmedia material make the audience more ensconced in the world of the story?
- How can a comic book be used to expand the story of a video game?
- How can a comic book be used to develop the characters of a video game?

Using the answers to these questions I create a comic book which expands upon the backstory and characters of the video game, Robogeddon (included in this project).

6.2. Literature Review

Transmedia story materials are media that expand the narrative and universe of a world or story previously established by a game in a separate installment. This is a fairly elegant way to allow players to learn more about the game that they are playing since it frees the storyteller to convey their tale in whatever medium suits it best. Examples of games franchises that do this very effectively are, Pokémon and Halo. Pokémon's best known supplementary material is it's TV show, which follows the story of Ash, a young Pokémon trainer, and his battle to become the best Pokémon trainer in the world, but it also includes manga and trading cards which reveal more of the backstory of the Pokémon world. Halo has many different novels and comics that are associated with it, but the biggest piece of supplementary material that was recently released for it has to be the game Halo: Reach. This game is a prequel to the

main series and gives background to the covenant invasion as well as showing that there are other Spartans that exist aside from master chief.

What is Transmedia?

According to Jenkins, "A transmedia story, is one which has a narrative so large that no one medium can hope to contain it." He gives *The Matrix* as a good example of this type of narrative. Within the Matrix universe there are scores media that add to the overall storyline, there are three separate movies that tell the story of Neo and the crew of the Nebuchadnezzar. There is a video game that develops the characters of Ghost and Niobe, another video game that lets the player explore the matrix universe, yet another game that allows the player to become Neo, the Animatrix, a collection of short clips that tell the backstory to the matrix and connect the first movie to the second one, and numerous comics and fan fiction. These media combine to create a monster of a narrative that only the most hardcore of fans will have perused.

Transmedia stories are not just movies either. They appear as series of games, (Chrono Trigger/Chrono Cross series) anime (Naruto), and even book series (James Bond). In some cases the medium will change from installment to installment of the narrative, this is done for a number of reasons. One factor is money, when a producer does not wish to spend a gratuitous amount of money making a small expansion to a movie or game they will choose a medium that is cheaper to produce, like a comic or a book (viral marketing can be a good example of this). Another is the appropriateness of the medium for what is being expanded, why make a movie when a book would be so much better at telling the story?

Jenkins talks about something that he calls "Synergistic Storytelling," which is a form of transmedia storytelling in which the mediums all closely work together to form one comprehensible narrative, while still remaining semi-understandable on their own. The example he gives of this is from *The Matrix*, "The Matrix is three movies and more." a combination of the *Final Flight of the Osiris* (an animated short), *Enter the Matrix* (a video game), and *The Matrix Reloaded* (one of the Hollywood movies). As a whole these three mediums tell the short story of how Jue, one of the crew of the Osiris warns the Nebuchadnezzar about the machines burrowing towards Zion. But if the audience did not experience all three of these media then they wouldn't really have a very good idea of what happened. In the movie all the audience sees are the characters discussing the "last transmission of the Osiris" and not much else aside from the contents of the message, they do not know what the Osiris is or who Jue is. In the game the audience again does not know about Jue or the Osiris, all they experience is the mission that Niobe is sent on to retrieve the message that Jue sent into the matrix. And finally in the animated short the audience is unaware as to whether the message is retrieved and what happens after the death of the Osiris crew because all they see is the fall of the ship and Jue's frantic sprint to drop the message off.

This type of storytelling is not quite as effective at expanding the storyline so much as compelling people to pay for or experience every piece of the story in order to get the entire picture, since each piece only reveals parts of the narrative, and in some cases they are not very understandable without one another. "For people who see only the movie, the sources of the information remain unclear, but someone who has a transmedia experience will have played an active role in delivering the letter and may have traced its trajectory across three different media." Jenkins is describing the delivery

of a letter which wars Scion against the impending machine invasion. While this is still transmedia storytelling, it is not like what Jenkins defines where each medium is in itself a contained story, but they are able to be constructed together and form a whole larger narrative. The Pokémon world sharply contrasts with that of the Matrix in this respect, because while the Matrix world is fragmented and needs to be experienced in its entirety to fully understand it, Pokémon is made up of a number of small narratives that can be combined to find a fuller vision of the world that they take place in.

Transmedia Channels

Christy Dena classifies transmedia materials that expand upon narratives into three different categories: story, storyworld, and commodity. "I observed that we can have channels that need to be experienced in order to move through or progress the plot, channels that reference each other and extend the experience of each other but have no direct role to play in the plot, and channels that are not part of the plot, are not necessarily part of the storyworld but can be associated with a plot and storyworld." The story category describes a media telling a story that is the main influence for anything else related to it and develops the main characters, plot, and setting extensively. Storyworld describes media that augments the comprehension of a story by developing background information or adding to the knowledge about existing characters and settings. Commodity describes media which has no effect on the narrative, but resides withint the scope of the transmedia universe, like Pokémon plushie dolls or Harry Potter costumes. Each of these have strengths and weaknesses in terms of story development and expansion but neither is better than the other, merely better suited for certain roles than another.

In a majority of cases concerning transmedia stories the components will fall into Dena's story category, as opposed to the storyworld. This is because a majority of the games, movies, books, and comics that are produced which are based off of another similar work are either self-contained stories, or simply continue the story of the previous work. In order for something to be part of the storyworld category it needs to exist without affecting the development of the plot, setting, or characters. So if something cannot affect the development of any of these things then there is not much left for it to do, and that would not be very interesting to experience. There are a few exceptions in terms of the original work being categorized as storyworld, Pokémon being the chief example.

I say this simply because the original game of Pokémon does not develop a plot or characters very much, it is mostly showing and expanding the setting. The materials that supplement the story of the game are what really introduce, develop, and display all the characters and plot of the franchise. It is possible that this was the intended effect, much like with Transformers toys, so that the world of Pokémon could be easily expanded upon without affecting a plot and characters that were already put in place. This is not to say that Pokémon does not have any characters or plot at all, they are all just very generic and easily expandable. For instance, the plot of Pokémon is essentially that the player is a 10 year old boy or girl, who must adventure through a world inhabited with Pokémon which they must catch and train in order to defeat their rival and earn gym badges so that they can beat the final four. There isn't much else to the story of Pokémon, sure there are side quests and you meet characters like Professor Oak and Bill, but other than that there are no massively earth shattering occurrences.

In the anime and manga of Pokémon however there are many things that happen that affect the way in which the story progresses, like Ash making new friends, catching new Pokémon, and defeating other characters. So in the case of Pokémon it is the supplementary materials that drive character and

plot development, because the game itself is lacking in these areas. This is one of the main reasons why I want to study Pokémon and its supplementary materials, because the MQP that I will be creating transmedia story material for will also be very weak in the areas of plot and character development.

Crossing Media

Jenkins mentions something interesting in an article he wrote for *Technology Review:* "The move toward digital effects in film and the improved quality of video game graphics means that it is becoming much more realistic to lower production costs by sharing assets across media." This basically means the improvement of special effects and movies as well as the improvement of graphics in video games is causing a crossing of mediums. Video games and movies are simply reusing the assets that were used in the other medium in whatever they are producing that will be the story expansion. This practice is happening because it lowers production costs and cutting the amount of time between the release of one piece and then the other. The faster and cheaper release of media gives production companies more money and more time to create further media, either in a new vein or to expand what they have already been working on. This is all done with the end goal of creating global franchises, which in turn are sought after for their massive money generation.

"Transmedia intertextuality works to position consumers as powerful players while disavowing commercial manipulation." (Marsha Kinder) Kinder is talking about the way that transmedia stories allow producers to have people buy many different things in order to get the entire picture, but because of the composition of a transmedia story the consumer is unaware that they are actually being manipulated into buying all these things. The transmedia story is actually a clever device for hiding the franchise in this case because the nature of the product makes it appear not to be one. The type of rhetoric here is much more subtle because in the case of a franchise like *G.I. Joe* or *Transformers*, they are constantly telling you to buy the toys, but when you have something like the *Dune* series or *Lord of the Rings*, you are compelled to buy and experience their products because you thirst for more knowledge about the world and its characters. So the consumer is not aware that they are actually buying into a franchise by purchasing the "Dune extended cut DVD" or the "Lord of the Rings video game" because they are buying it of their own will to experience more of the world that they have glimpsed in the books and movies of these franchises.

Tim Jordan, argues that the crossing of media in the world of Pokémon, most specifically in the world of Pokémon cards, there has been the creation of a need for mastery and acquisition. "You can collect all possible Pokémon and keep them pristine as trading cards or you can use the same cards to conduct battles." The mastery he refers to is that of mastering both the games, and the skill of battling with the cards. In the game itself people must master the art of countering different Pokémon types with various Pokémon of their own until they eventually win the game, at which point there is a choice to collect (acquire) all the Pokémon that they can, or master the art of battle and defeat their friends, and maybe even compete in competitions. The same thing happens with Pokémon cards, so much so that there were worldwide tournaments and events promoting them. Naturally the urge to win compels people to try and think of the best deck they can make, which of course calls for specific cards, and eventually leads to the acquisition that Jordan talks about.

In the game, the player tries to acquire different types of Pokémon to have a balanced team which can destroy all of their opponents. This is done by figuring out where each type of Pokémon

spawns and then frequenting this area until one is encountered so the player can catch it. Some people try to complete the Pokédex (a device which keeps track of the number of Pokémon the player has caught) which means they must either catch, trade, or evolve Pokémon until they have owned every single one. "The two main threads you can follow are finding and battling the eight gym leaders to gain a badge from each, followed by a final confrontation with five elite trainers to become a Pokémon master; and/or you can collect all the Pokémon in the game," (Jordan) This is true in the cards as well, people are compelled to keep buying packs because they are always in need of that one card that will make their deck perfect, or just need that holographic Charizard to make their collection complete.

These transmedia materials allow Pokémon itself to become a brand, because the mastery and acquisition makes it easy to add to the pool of existing Pokémon with more games and cards. People will always buy the products because they will not be able to become a master without experiencing all the different types of card and Pokémon, thus they must continue to acquire. The constant need to acquire allows for more and more opportunities to expand the transmedia narrative of Pokémon because the writers must keep thinking of new ways to add new species of Pokémon, and cards due to fan demand.

So in essence, franchising and merchandising push for transmedia storytelling and supplementary materials to be added because these things allow for more merchandise to be produced and sold. A great example of this is the old Transformers toys from the 80s and early 90s. Transformers was a toy line produced by Hasbro toys which consisted of two warring factions of robots that could turn into everyday objects like cars or planes. The toy line was in need of hype so the company contracted a Marvel comic book and a TV series that depicted the battles between the Autobots and Decepticons. The TV show and comic book were able to develop the characters of the toys that were sold in stores as well as give insight into the backstory of transformers, which made them more than just toys, but icons of what kids saw in the show. Eventually with rising popularity the show and comic books were used as a jumping board to add new figures to the toy line, and were in essence just toy commercials.

Computers were hailed as devices that would cancel out other media because of their ability to compile everything into one manageable medium. But it would seem the opposite is true, "computer owners consume on average significantly more television, movies, CDs, and related media than the general population." (Jenkins). Jenkins' reasoning behind this is that younger viewers are "information hunters and gatherers" because of the way that media has been converging to create stories which have multiple pieces and sometimes are only understandable if the viewer has knowledge from one of the other mediums. This goes back to what Jordan said about acquiring, except in this case instead of trying to acquire cards, or in game objects, the person is trying to acquire knowledge about a story by collecting the various pieces of it from different mediums.

Transmedia in Video Games

Transmedia storytelling has been effectively used in a couple instances to create more toys from TV shows and movies, but what about in video games? In games like Halo, or Call of Duty more characters cannot be added just for the sake of making more action figures, so transmedia storytelling is usually used to create hype for a new game or fill in gaps between two games that are in the same story arc. They are also used to expand the story when the creation of a new game isn't feasible due to money or time constraints, so cheaper, faster to produce media can be used. Halo is a good example because of

its wealth of novels and sequels. The original Halo was successful mainly due to the online multiplayer combat that it offered on Xbox Live, but eventually the game franchise would have died if it had been left untouched. So in order to continue the story of the first game and satisfy player's need for more weapons and versatility in multiplayer novels and other media were created that added the Halo universe and allowed for new machinery and weaponry as well as enemies make sense, and eventually linked the second game to the first so that they main story could continue.

Unfortunately there has not been very much research done in terms of figuring out what makes supplementary materials and transmedia storytelling create compelling stories and interest in the game itself. There have been a few people that studied games as part of transmedia storytelling and fiction, such as Jenkins but there have not been very many studies that focus on narrative advancement specifically. Jenkins states that in transmedia storytelling "each medium does what it does best," for the development of the narrative as a whole. This would seem to imply that each sort of medium has a situation in the construction of a story where it would be much more effective than another medium. For instance a novel would be very good at describing miniscule details about the backstory, whereas a movie or video game would be much better at portraying copious amounts of action or violence. In the case of media expanding upon the story of a game, they are generally mediums that are good at portraying aspects of the story that the game itself is very weak at doing.

Audience

Audience expansion is one of the main goals of transmedia storytelling because it allows the franchise to swell and ends up generating more money. For instance, how many people do you think went out and bought the book sets of Lord of the Rings right after they saw the movie? Even if people hadn't seen these movies they were most definitely hearing about them and the Lord of the Rings, especially since the last movie, *The Return of the King*, won 11 Oscars when it was released and was the second film ever to break the 1 billion dollar mark. Everyone who heard anything like this immediately became a potential audience for both the books and the movies, not to mention the numerous other side projects like the games and merchandise. In fact this brand is still growing, just recently in 2009 a fan film called *The Hunt for Gollum* was released on the internet and immediately gained mainstream attention. In 2012 and 2013 there is also the two part movie adaptation of The Hobbit being released which will no doubt gain further notoriety for the Tolkien world and cause more re-releases of all its content considering the original Lord of the Rings movies were finished eight years ago.

Audience expansion works the same way with games as it does with movies, the more people who hear about something, the more likely they are to be a potential audience. Games like Halo and Warcraft do this all the time by releasing novels and short movies on the internet that pique people's interest in the games. Warcraft even has entire collections of trading cards and board games for people to buy if they don't want to buy a game with a monthly subscription fee or do not have a computer that is powerful enough to run the game. This also goes for people who may not even like video games but are very into board games and card games, so while they would absolutely never be a potential audience for the Warcraft game itself, they are still an audience to the Warcraft franchise because of how far reaching it is in terms of genres.

Jesper Juul, in his essay about games and narrative, states "games and stories do not translate to each other in the way that novels and movies do." The problem being, there is not a really good way

to give a back story for a game other than an opening cinematic, which people love to skip, or through the game box or manual, which no one ever looks at. And once in the game there is no good way to show story progression aside from cut scenes, which interrupt play, or text, which is pretty annoying if there is too much of it. This is where transmedia storytelling can come into play because it can allow the developer to give backstory in a different medium, better suited to do so, that is released prior to the game. Not only would this give people a chance to get to know the game world before they begin to explore it, but it can also give rise to extra hype about the game, sort of like viral marketing, if it is done well. If the material is distributed for free beforehand then more people will be able to, or be willing to experience it, and if it is really good then it might even get more people to buy the game when it eventually comes out in the future.

Transmedia Worlds

Transmedia storytelling is something that is based around the creation of a world so that it can be easily expanded upon from any medium. J.R.R. Tolkien's Middle Earth is a great example of a world that has been expanded upon by almost every sort of medium in every way. There are movies, video games, books, comics, trading cards, merchandise, and even pornographic fan fiction set in this universe that Tolkien created, and each one gives a little to make it bigger and richer. It is not possible for any one person compile all that has been created because there is simply too much, and it is ever expanding, and this is what a transmedia story strives to be.

Final Fantasy VII is a great example of a world that was created for transmedia storytelling: it has had several games made for it, a movie, and a short film. Each of the games has a different storyline, while retaining the same characters, although some like *Dirge of Cerberus*, develop different characters than others. The film and the movie however are completely different, the movie *Final Fantasy: Advent Children*, is a continuation of the story from the game, detailing events after the end of the game. *Final Fantasy: Last Order*, on the other hand is a prequel to the game and details the events leading up to the beginning of the game. The franchise was extremely successful states Dean Chan, and because of its success other Final Fantasy games were planned to be made the same way (like Final Fantasy XIII).

Transmedia stories can be told across multiple games as well, like in the case of Warcraft I – III and then World of Warcraft. Sequels to games often work like this by directly continuing the storyline of the previous game (like in Call of Duty: Modern Warfare series), giving background on events that led up to one of the existing games (Halo: Reach), by starting a completely new story at a different time in the game world, with the same characters (Timesplitters series), or by telling a story that is completely new with completely different characters but in the same game world (Pokémon series). When a sequel is released, it does two things: secures new members to the audience who may have not known about or been around for the original games (Pokémon is a great example of this because there are plenty of kids who were not even born yet when the first game came out that are probably lining up to buy Pokémon Black or White), and gets existing members of the audience to continue buying the games. Of course these two groups will have completely different experiences concerning the sequels, since the new audience members will just be discovering the game and consider it a new thing, then maybe go back and play the earlier versions of it, and the returning audience will view it as a continuation of what they have already been playing and know what happened in the previous installments.

Dena mentions in her paper on multichannel poetics, that there is an inherent danger in a transmedia story, if the viewer experiences part of the story through a medium and does not enjoy it, then they may refuse to experience the rest of the story entirely. "A discussion in the Vivendi Universal Games Forum on the *Fight Club™* game (Vivendi Universal Games, 2004) showed the distain fans of the film have for the non-narrative commodities." So if there is a weak link in the chain of the transmedia story then it could actually end up hurting the story by turning viewers and new audiences away rather than luring them in, which is the hallmark of transmedia stories. So when creating a transmedia story, special care must be taken in ensuring that each part of the narrative is compelling and can interest the audience that it is targeting. An example of this is the Tomb Raider movie, which got mixed review. People who saw the movie and liked it were much more prone to buying the game and playing it, and becoming customers for future games. On the other hand there were also people who hated it and were probably turned away from buying the games of Tomb Raider, even if they had heard of them before.

This is not to say that the only way to lure people into the transmedia story is with an expensive hook, there are plenty of examples of cheap hooks for transmedia stories. A couple years back, Adult Swim released a movie for their show Aqua Teen Hunger Force, "Aqua Teen Hunger Force Colon Film for Theatres." In order to gain popularity (or in this case notoriety) for this movie they posted graffiti of characters from the show (the moon men) on walls in some cities, and then placed a box next to them. In the box a fan would find t-shirts for the movie, which they could take for free. Adult swim was using the commodity channel of transmedia storytelling in order to get people excited for the movie, and maybe get some people who had never heard of it to take notice. Their plan worked much more than they expected when it gained national attention because the police thought the boxes might contain bombs or other dangerous materials. Not only was this hook inexpensive, but it reached a massive number of viewers who had never even heard of the show Aqua Teen Hunger Force.

Transmedia hooks are also not limited to publicity stunts, like in the case with Adult Swim, they can also be things like short films. Just recently a short film called "The Trenches" was released for the upcoming movie, "Sucker Punch," and it has exploded across the internet. Sites like IMDB and even Wired are featuring a stream of the short on their servers, and the versions posted on youtube (upwards of 20) have been receiving heavy attention. The film does not seem to be related to the movie at all, it is more of a side story, of a man who is shot in the trenches and then resurrected to fight against the main characters of the movie. It is less than three minutes long but it holds some very interesting information that teases the viewer about what may happen in the movie, since they learn from "The Trenches" that apparently the villains in the movie possess the power to turn good men evil. This is a tantalizing fact to fans that have been waiting for this movie since its announcement almost two years ago, and it has also gained attention from communities that had never heard of it before.

6.3 Method

Transmedia storytelling is the convergence of a number of mediums to create a story that is so large one narrative cannot hope to contain it. Video games have long been a part of the transmedia scope, ever since comic books, movies and tv shows started getting made about their characters. Comic books in particular are useful in expanding the scope of a game, because while they are still a visual medium like television and movies, they are much cheaper and faster to produce. In this section I explore the various

aspects of comic books that are made to expand upon games and what aspects of the game stay consistent in the story. I apply this knowledge in my own attempt to make a transmedia comic book later on in this paper.

I apply my knowledge gained in this section to create a comic book which expands the background of my MQP game, Robogeddon. This comic book gives the lore that occurs in the time period leading up to the game itself, so the player is able to understand the game more easily. In order to explore the world of transmedia storytelling I employ a generic rhetorical criticism to closely monitor a number of different types of information. Generic criticism is great at keeping track of statistical information like the number of panels characters appear in, which is the type of information I need to keep track of to discover the way which characters and backstory are developed. Generic rhetorical criticism is a tool that is used in order to compare works of the same genre to one another and to discover the author's use or abstinence of conventions commonly used within the genre. Gustainis states "It adds to the critic's knowledge by allowing him or her to classify various types of rhetorical artifacts and the specific characteristics represented by each type." While the generic criticism allows the researcher to categorize, explain, and understand, it does not take moral or ethical information into account.

The reason that I am focusing upon graphic novels specifically rather than the transmedia universe as a whole is because comics are a visual medium, much like video games, and I believe that a comic book would be the best media to expand the world of Robogeddon. Comic books are also a medium that I have had experience analyzing using generic rhetorical criticism, so I am able to gather data and draw conclusions more easily. Graphic novels and comics are something that I find very interesting and I believe to be a very powerful medium for developing characters and illustrating settings and lore, which is what I originally set out to do for Robogeddon.

During my research I tried to answer a few questions:

How can transmedia material make the audience more ensconced in the world of the story?

I asked this question because I am very interested in audience expansion for video games, because many that are created do not reach more than just an audience of gamers. The ability to ensconce the audience in the game world through a different medium would allow game developers to hook people who would not normally play the game and get them to experience it. I think this is important because it would allow game developers to create meaningful experiences for all different groups of people and allow them to be able to begin experiencing it in a comfortable setting before expanding into a new experience.

How can a comic book be used to expand the story of a video game?

I asked this question because I have noticed that in our classes where we create games story gets put on the backburner, because everyone in the team is so concerned with making something playable and that looks really good. I did not want this to happen with our MQP, but I knew that we would run into a bunch of problems when developing it and probably have to skimp on story in order to finish it, so I decided that describing the story in a different medium would be more efficient. In addition I wanted to

discover if there was a common way of expanding games with comic books or if it is different every time depending on the games plot and setting.

How can a comic book be used to develop the characters of a video game?

Again I asked this question because characters usually are not developed fully in our school courses, but I also wanted to discover if I character that is shared between two mediums and is powerful in one, remains powerful in the other medium even if they are not further developed. This is important because it allows the game developer to create what they believe to be a meaningful experience, and then if one of the characters in the game is misunderstood or is not as powerful as they originally thought, then they can go back and create a different media that makes them into a strong and meaningful character (like Ash in Pokémon).

The Focus

I developed this method in order to analyze the supplementary graphic novels and comics that go along with the video games Halo: Combat Evolved, and Pokémon (Blue). In short I played the games, then read the comics. As I read the comics I recorded important data about characters and setting on a tracking sheet so I could formulate my claims later. These claims helped me to better understand the nature of this medium and gave me a good base to work off of as I created my own transmedia material for Robogeddon.

My selected readings are both ones which do not follow the plot of the games. "The Electric Tale of Pikachu" is a graphic novel that was based on the anime of Pokémon, which is not something that follows the plot of the game very closely. Rather it is much more concerned with the small events that happen to Ash and his friends as he quests to become a Pokémon champion. The two readings focus on developing aspects of the story universe that were not explained very well in the game itself, which is exactly what I wanted to make my comic like. My reading from the Halo graphic novel, "The Last Voyage of the Infinite Succor," is a story that is meant to create a bridge in the gap between the first Halo game, and the second (like all the readings from Halo: Graphic Novel). Since it does not detail any event that occurs in the first game, I will be focusing more on the characters that are in it and how they are portrayed in the book versus the game.

How Do Comics Fit In With Video Games?

I chose to study comic books for a number of practical and personal reasons. Comics, like video games are a visual medium, so the transfer of the story from one medium to the other is a fairly simple transition. The comic is a simple way to expand the visual style and the story of the game at the same time. Aside from the technical reasons, I chose to study comic books because I have a number of personal reasons. I have always been fascinated with comic books, and find them to be a very flexible medium for communicating complex stories. I have always wanted to create a comic book of my own, and I decided to take this opportunity to do so and learn more about the medium that I like so much.

And finally it is my belief that comic books are a much more accessible medium to the general public, because they do not need a computer to run it and everyone loves picture books, so a comic allows our game to have a wider and larger range of audiences.

Selected Texts

POKÉMON: The Electric Tale of Pikachu

Pokémon has a complicated structure of transmedia channels and merchandise that consists of media ranging from plushie dolls to trading cards to manga series. This wealth of story materials has been very successful at solidifying the brand of Pokémon because of the wide range of audiences that it reaches and the lust for more story that the various facets of this structure creates.

This comic fits in with my project because it is a great example of a transmedia piece that runs in a parallel storyline to the original media, but it still expands the backstory of the game. Since I create a "prequel" comic for a video game as part of this project a graphic novel which gives backstory is a great thing to analyze. The graphic novel details the adventures of Ash, a beginner Pokémon trainer, and his loyal companion Pikachu. In the story Ash travels around the world catching Pokémon and collecting badges so that he can accomplish his dream of becoming a Pokémon master. On his journey he meets new friends like Brock and Misty and has adventures with them, like his battle against the Grey Fog (an ancient and extremely dangerous Haunter).

This comic expands the backstory about how children of the Pokémon world become trainers and begin their various journeys into the world. The various pieces of equipment that trainers use are also expanded upon in this book, specifically the various types of Pokéballs have to use to catch different Pokémon and the Pokédex. The way that trainer ranking and advancement is also explained in this book (it is based off of the difficulty and number of Pokémon that the trainers catch). So it gives more meaning to the random encounters and gym battles that the player experiences as they traverse the world and helps them understand why they are doing what they are directed to do in the game.

HALO: The Last Voyage of the Infinite Succor

Halo too has a large conglomeration of transmedia material which consists of video games, short films, novels, and comic books. The creation of this huge amount of material which expands upon the original concept and story of Halo has made it one of the most well-known and best-selling video game franchises of all time.

This story is one which expands a small event that is the catalyst for action in the game, but the main point of this story is to develop the character of the Covenant forces more and allow the player to be more knowledgeable of how they operate and the structure of their society. The story centers on an Elite Commander who is sent to investigate a Covenant vessel that has fallen to the flood. In his journey he discovers more and more about the flood (like their ability to assimilate dead bodies and steal

memories and knowledge) and saves a high ranking Covenant official who was on the ship. The story eventually culminates with the commander sacrificing himself to contain the flood.

The comic expands the information the player already has about Covenant society and makes it easier to familiarize with them. This leads to the player either taking more pleasure in fighting them as they play the game or make them appreciate that the Covenant are not that much different than the humans. In either case, the comic gives the Covenant much more depth which educates the player about them so they think about them as more than just enemies who show up to be torn apart by the player's weapons.

Protocol

I analyze the games Halo and Pokemon, and the graphic novels that correspond to them, "The Electric Tale of Pikachu" (Pokémon) and "The Last Voyage of the Infinite Succor" (Halo). I plan to take a week to play each game and then take a week to read each graphic novel, but I will have the time taken to play one game overlapping with the time taken to read one of the graphic novels, so it will take 3 weeks in total in order to analyze all the material that I need.

When playing the games, I used guides to ensure that I saw every part of the game so I could absorb the entire story. I did not use a tracking sheet for playing through the games, because specific figures on the occurrences of certain events did not concern my research at this stage. The purpose of playing the games first is to familiarize myself (I have already played both games all the way through in the past) with the characters and the plot before I get into the graphic novels and begin to draw conclusions. When I finished each game wrote a short summary of what happened, in terms of story, so that I had a control to compare with the content of the graphic novel.

Data to be Gathered

When I read through the graphic novels I implemented a tracking sheet to help me analyze a few important things:

- How many panels the main character appears in.
- If there were: sidekick, mentor, comic relief characters in the book/how many panels they appear in.
- How many panels each character appears in with the main character.
- How many times the setting changes.
- How many times the characters give info/How many times the "narration" gives info (usually are square boxes that have text but none of the characters say what is written in them).
- How many characters are shared with the game.
- The length of the comic in pages and panels.

I collected this information to be able to figure out more about character development, and the structure of narrative.

The purpose of keeping track of the number of panels that the main character is in is to have a record of how many panels they may appear in on their own versus in the same panel as other supporting characters. The ability of the main character to appear alone in panels is one of the main indicators that they have strong enough personality to be interesting even if they are not being supported by another character. I am hoping that this will give me some information about making the main character of my own comic more dynamic and interesting.

The reason that I recorded both the number of panels in which all the supporting characters appear alone, and in which they appear with the main character, is to figure out the percent of time that they appear alone. This is what I believe allows me to determine if a character is more for supporting the main character, or if they are used to develop the narrative in different directions. For instance a sidekick character would appear in the same panel as the main character a much higher percent of the time than a character that is meant to further the plot. A good example of this would be Batman. In the Batman movies, the Alfred character almost always appears at the same time as Batman, and does not get much time to the screen on his own. This is the opposite for a character like the Joker, who only appears with Batman when they are fighting, because the Joker is a much more dynamic and strong character than Alfred, and who serves to further the plot more than provide Batman with a sidekick to converse with and develop the Batman character.

The purpose of recording the number of setting changes is to determine if the story of the transmedia channel develops a single instance that happens in the parent story, or if it is more centered on its own narrative. I recorded any significant setting changes in the graphic novels that I can study in order to have a clearer picture of whether they are making their own narrative or relying on the lore of the game. Assumedly if the transmedia channel shares a number of characters with the original medium then it is more likely to be based off of the game's lore. But if it does not share very many characters with the original medium then it can be assumed that it is more focused on its own narrative and the development of its own characters than that of the original medium.

The purpose of recording the number of times characters giving background information or lore versus narration doing the same, is more for me to discover what is a more efficient way communicating background information, since that is what the comic I wrote is about. Finally the reason I recorded the number of pages and panels is so I could get an idea of what percentage of the comic the main character and other characters are in. This allows me to determine if the comic is mainly character driven or if it is more action oriented. My goal was to learn more about transmedia storytelling through analyzation of these graphic novels so that I could expand the universe of the game I created in my MQP.

I paid attention to transmedia channels of these graphic novels as I read them and eventually took note of whether or not they were story, storyworld, or commodity (see the section on Dena in my lit review). I hope that by reading these graphic novels I can find out what the common transmedia channel for this medium is. Then I decide if they were effective in their role in the transmedia universe or not.

After I finished reading the graphic novels I answered a few questions about each one:

- What does the reader gain from reading the graphic novel?
- What does the graphic novel add to the game?

- Does the graphic novel have its own narrative or does it mimic the game?
- Is the main character essentially the same as the game?
- Does the graphic novel expand upon a certain event in the game or connect to an event, or does it tell a story that runs parallel to the game?
- Do I know more about the game's story after reading this? If so what?
- Does the graphic novel take place in the same time period as the game?
- And I will state the publication date, publishing house, and paper size.

These questions helped me to focus on the parallels between the game and the graphic novel and further my understanding of the link between a transmedia offshoot and its parent.

The purpose of recording what the reader gains from reading the graphic novel and what it adds to the game is to discover what sort of connection these stories have with their parent narrative and what purpose they serve as transmedia material. If the comic adds much more backstory to the game, then they are meant to expand the storyworld channel, but if it is a narrative on its own then it is much more likely that it is part of the story channel (see my lit review for more information on transmedia channels). In addition my own reason for recording this information is to get ideas for my own transmedia story.

I asked a number of questions that relate to the transmedia material connecting to the main narrative ark, if the graphic novel has its own narrative, if the main character is the same, and if the graphic novel expands upon a specific event or if it is parallel to the game. Once again this is to discover what transmedia channel the comic books belong to, and like before, if they are more independent in terms of story and characters from the game then they belong to the story channel, otherwise they reside in the storyworld realm. In addition I have a number of other reasons for asking these questions. The purpose of the question about the story of the transmedia material either mimicking the game or possessing its own narrative is to determine if the transmedia material is standalone or if it simply an extension of the game. Building upon this, if the main character is the same as the game, then it could mean the comic is meant to simply be a rehashing of the game with a slightly different audience, or it could be a backstory to the game, or an extension of the original narrative. If the main character is different then there are a number of things the comic could be for. It could be the same story as the game told from a different perspective, or it could be a completely different story entirely that is set within the game universe. The reasoning for asking the third question is similar to this as well. I wanted to gather this information because I believed it would help me to determine if the comic I find to be more powerful is of a certain type, and I could model my own from this.

The purpose of asking if I know the game's story after reading the comic is to determine if the comic describes the basic premise and storyline of the game in its contents. If it does then that means that the comic can be a standalone narrative because the audience does not require any previous knowledge of lore before reading it. But if it I could not determine the story of the game after reading the comic then it meant that the game had to be played in order to understand this transmedia piece. The purpose of asking the question about time period is simply to figure out if the game is a prequel, a sequel, or a narrative that happens at the same time as the original piece.

6.4 Conclusion/Results

I wanted to add to the study of transmedia storytelling by discovering what can make a transmedia story very effective, in terms of using the various mediums to their greatest potential. According to Klevjer, "The narrative appeal is not so much about unfolding events (although it does that too, most often not very successfully), as about giving meaning and sensation to the actions when they are performed by the computer and the player." What he is saying is that games are not quite as good as telling stories as they are at motivating players to do certain things. I am interested in this because it is a perfect argument for transmedia storytelling and supplementary story materials. If games cannot effectively tell a complete story, even if it is compelling, then the story must be expanded into other media.

"Several hundred different Pokémon exist, each with multiple evolutionary forms and a complex set of rivalries and attachments. There is no one text for information about these various species. Rather, the child assembles information from various media," (Jenkins) Jenkins' Pokémon observation is a good example of how transmedia storytelling works, but he is talking about it in the sense that every kid has a different knowledge of Pokémon, he does not discuss what the effect of many different mediums of story has on the story of Pokémon as a whole. Using his and the observations of others, I tackled the problem of effective transmedia storytelling, specifically the use of comic books to bolster the game world's background and the development of its characters.

Comics and Games

I looked at games that have graphic novels and comic books which expand upon the overall story. More specifically I looked at Pokémon (either red or blue) and Halo (the original, aka Combat Evolved), and their respective graphic novels or comic books. I played the games in turn and then read the graphic novels and recorded my findings about recurring characters, such as Gary Oak and Master Chief, and various recurring factions and the different ways in which the video games and the graphic novels treat them. I then applied my knowledge to creating a comic book which expands upon the story of Robogeddon, the game I worked on in the other section of this project.

My research showed fairly clearly the connection between the purpose of a transmedia material and different trends in its characters, narration, and settings. I explain a few things that I uncovered during my research before I go on to show how I applied my findings in the next section.

Readings

The purpose of my first reading, *The Last Voyage of the Infinite Succor,* is to develop the character traits of the enemies in the game Halo: Combat Evolved (See Appendix X for my raw data). This narrative details the experiences of Special Operations Commander Rtas 'Vadumee (an Elite in the Covenant army) as he explores a recently attacked Covenant ship, The Infinite Succor. The ship has been attacked and consumed by The Flood and the Commander must use his small team to go to the bridge and enable the self-destruct sequence for the ship. Eventually they discover there is no way to detonate the ship unless someone stays behind, which the Commander volunteers to do, and culminates with his sacrifice to destroy The Infinite Succor.

As the Commander and his team explore the ship and fight the flood the audience is given insight into the way that the Covenant army operates. For instance I was unaware that Grunts (small

goblin-like creatures) are actually not just used as fodder for the enemies guns, they serve the roll of fire support teams for the more heavily armored and bigger Elites. The Commander too seems like he is on par with the rest of his team, as he treats them like equals aside from the fact that he gives orders to them, and he shows remorse for his fallen comrades as they begin to die all around him. At one point a high official Covenant leader, called the Legate, is introduced and he is shown to me much like a monarch ruler and demands that the Commander do as he says, but the Commander does not follow the orders and instead thinks of the mission he was originally sent there to do, showing that he can think for himself and is not a mindless drone that is commanded by a hive mind, like I originally thought from the game.

Since this story was meant to expand the characters of the Covenant army the comic focused entirely upon them (completely ignoring the existence of the main character from Halo, Master Chief, aside from a few mentions of "The Demon," the Covenant's nickname for him). The comic had the main character in almost 80% of the panels, and the ones which he was not featured in were populated with other Covenant. Since the Covenant were featured so heavily in this comic they were developed very thoroughly and the audience was able to connect with them and learn about them in depth.

The purpose of my second reading, *The Electric Tale of Pikachu*, is to develop the backstory to Pokémon and give the reader a glimpse into the world of a Pokémon trainer. The book details the journey of Ash, a novice Pokémon trainer, and his companion Pikachu as they battle their way to the Pokémon leagues in the hopes of becoming the champions. The book is broken up into episodes, much like a television show, in which Ash and Pikachu meet new friends and rivals all the while trying to catch more Pokémon and become stronger.

Throughout this comic there are pieces of backstory about the way that Ash got to where he is in the comic and the process he went through to actually become licensed to catch and battle Pokémon. The comic also gives a lot of information about how trainers are able to rank up and buy more high tech Pokéballs to catch rarer and more powerful Pokémon and how the trainers are actually able to get into the league and eventually become a Pokémon master. This is mainly done through narration, or the square boxes filled with text in comics that are not speech bubbles.

This comic adds to the game in that it educates the audience about the world that they are in when they play Pokémon. The backstory information in the story serves to give meaning to what the player actually does in the game. Someone who has not read the book will play the game and assume that they are just fighting random people for money (like less violent versions of cockfights) and eventually they come to people who have set up gyms which train certain types of Pokémon. They then have to defeat the leaders of the gym to get badges which grant them access to the Pokémon league. If the player has read the comic they would know that the reason they are battling other people is to both train their Pokémon and to increase their trainer ranking which allows them to be licensed to catch more dangerous and powerful Pokémon. They then use their Pokémon to challenge gym leaders, who are already Pokémon masters to battles in order to see if they are experienced enough to be given permission to enter the Pokémon leagues, where they fight many other people like themselves until there are only a select few left who become Pokémon masters.

Assertions

I was able to make a number of assertions based on the results of my research:

1. Character centric stories have the main character in a higher percentage of the panels.
2. Comics based off of a game that has a weak main character will have a particularly strong main character in their own narratives.
3. Comics that are more for developing backstory will share more characters with the original game than a comic that has its own narrative.
4. Comics with more "narration" develop the backstory more heavily than specific characters.

From these assertions I was able to formulate a plan to develop the backstory of our game by way of a comic book.

I decided to do a first person narration from a character which is not featured in our game (at least not explicitly), the AI construct the player is supposedly playing as. I decided to do a first person narration because I wanted to simultaneously develop the backstory of our game and the main character of it. First person narration allowed me to have the main character speaking and apply my assertion that there must be a high percentage of panels which have the main character at least featured in them, so the audience could familiarize themselves with this AI construct, and begin to understand the way that it works and feels. But at the same time the narration was for the development of the backstory so the audience would be undistracted by frames showing the AI construct, but could actually visualize what the character is saying and pay attention to both aspects at the same time. Since the main idea behind the comic was to develop backstory for Robogeddon I thought it was fitting to apply my assertion that comics which develop backstory make heavy use of narration.

Since our game had a very weak character in it (assuming people could realize there was a character in it at all) I wanted to apply my assertion that a game with a weak main character must have a comic with a strong main character. So in the comic I had the AI construct featured on every single page and tried to create a colorful character with all its own quirks, prejudices, and personality. I wanted to do this because I noticed in Pokémon, how the knowledge of who Ash actually is gives a much more enjoyable play experience because you know how he reacts to different situations and what he wants to do in his heart. Along a similar line I wanted the player to understand the AI's reasoning behind trying to destroy all the humans, and how it eventually came to this cruel ultimatum.

6.5 Raw Data

"The Last Voyage of the Infinite Succor" (By: Lee Hammock and Simon Bisley) [Halo: Graphic Novel] 2010, Marvel Publishing, Inc. (Softcover, 10" x 6 ½")

# of panels (% of panels), main character (**Elite Commander**) (first appears at page#: **12**)	**185** (79.1%)
# of panels (% of panels), **Master Chief** (first appears at page#: **7**)	**8** (3.4%)
# of panels (% of panels), **Kusovai** (first appears at page#: **12**)	**56** (23.9%)
# of panels (% of panels), **Legate** (first appears at page#: **29**)	**21** (9.0%)
Master Chief , # of panels with main (% of time with main)	**0** (0.0%)
Kusovai , # of panels with main (% of time with main)	**48** (85.7%)
Legate , # of panels with main (% of time with main)	**20** (95.2%)
# of setting changes	**7**
# of times characters give background/setting	**5**
# of times "narration" gives background/setting	**2**
# of characters shared with the game	**1**
# of pages (# of panels)	**48** (234)

Table 5: "The Last Voyage of the Infinite Succor" Data

1. What does the graphic novel add to the game?

This graphic novel details an event that is mentioned in the second Halo game, and event which is not described at all in fact. This graphic novel is centered around a team of Covenant Elites who board a ship that has been taken over by the flood. While this does not seem very important to the game of Halo, it is an amazing opportunity to show the way the Covenant function both societally and tactically, which the authors do very effectively. The graphic novel lets the reader in on the class structure of the Covenant as well as what they find to be holy, and why they despise the humans so. This background information gives the player of the game a much more colorful experience because they are more intertwined with the world of the game.

2. Does the graphic novel have its own narrative or does it mimic the game?

The graphic novel has a narrative that is not even explored in the game, it only uses the lore of the game to create its own world.

3. Is the main character essentially the same as the game?

No, the main character of this book is actually depicted as the enemy of the player in the game, it gives a good insight into what the player is fighting, and the enemy's views of humanity.

4. Does the graphic novel expand upon a certain event in the game or connect to an event, or does it tell a story that runs parallel to the game?

The graphic novel describes at length an occurrence which affects the characters behaviors in the game and describes a little bit about their background.

5. Do I know more about the game's story after reading this? If so what?

I know more about the way that the covenant work, and how they coordinate tactics, they are not just things that wait for the player to come get them, but are actively working against the player using specialized squads to fight them. The flood too are described in detail showing how they can absorb knowledge of the dead and use it against the living, even if they were formerly comrades.

6. Does the graphic novel take place in the same time period as the game?

The graphic novel depicts events that happen between Halo and Halo 2, so it happens just after the game.

7. Is the graphic novel a single narrative ark or a collection? If it is a collection how is it organized?

This graphic novel is a single 50 page narrative that focuses mostly upon one character and his experiences.

Pokémon: The Electric Tale of Pikachu (By: Toshiro Ono)
1999, Shogakukan, Inc. (Softcover, 8" x 5 ½")

# of panels (% of panels), main character (**Ash**) (first appears at page #: **13**)	**318** (48.6%)
# of panels (% of panels), **Pikachu/"Jean Luc"** (first appears at page #: **14**)	**200** (30.6%)
# of panels (% of panels), **Gary** (first appears at page #: **16**)	**20** (3.1%)
# of panels (% of panels), **Misty** (first appears at page #: **35**)	**66** (10.1%)
# of panels (% of panels), **Brock** (first appears at page #: **60**)	**38** (5.8%)
# of panels (% of panels), **Professor Oak** (first appears at page #: **110**)	**28** (4.3%)
# of panels (% of panels), **Bill** (first appears at page #: **115**)	**21** (3.2%)
# of panels (% of panels), **Sabrina** (first appears at page #: **130**)	**28** (4.3%)
Pikachu/"Jean Luc" , # of panels with main (% of time with main)	**133** (66.5%)
Gary , # of panels with main (% of time with main)	**13** (65.0%)
Misty , # of panels with main (% of time with main)	**25** (37.9%)
Brock , # of panels with main (% of time with main)	**34** (89.5%)
Professor Oak , # of panels with main (% of time with main)	**17** (60.7%)
Bill , # of panels with main (% of time with main)	**10** (47.6%)
Sabrina , # of panels with main (% of time with main)	**15** (53.6%)
# of setting changes	**29**
# of times characters give background/setting	**13**
# of times "narration" gives background/setting	**14**
# of characters shared with the game	**6**
# of pages (# of panels)	**147** (654)

Table 6: "Pokémon: The Electric Tale of Pikachu" Data

1. What does the graphic novel add to the game?
   This graphic novel gives interesting background information on the way people both become and live as Pokémon trainers. It also expands some key characters from the game, such as the gym leaders Brock and Misty, and finally gives a face and personality to the character which represents the player in the game. The graphic novel educates the player as to the background of how they became a trainer in the game in the first place, and how gym leaders act, and the way that they become the leader of a gym, giving the world much more depth and making the player care more about what they do in the game.

2. Does the graphic novel have its own narrative or does it mimic the game?
   The graphic novel has a completely different narrative from the game, developing specific experiences ash has along the way, as opposed to an overarching plot that the player fleshes out on their own as they go along in the game.

3. Is the main character essentially the same as the game?
   The main character is similar in appearance to the one in the game, but Ash is supposed to be a very different character than that of the game, because the character in the game is supposed to be the player themselves.

4. Does the graphic novel expand upon a certain event in the game or connect to an event, or does it tell a story that runs parallel to the game?
   The graphic novel tells a story that is parallel to the one of the game, but does not progress as far into the leagues as the game does.

5. Do I know more about the game's story after reading this? If so what?
   I know much more background information about Pokémon trainers and how the badge system and league system work from reading this graphic novel. Basically all background information.

6. Does the graphic novel take place in the same time period as the game?
   The graphic novel can be assumed to take place around the same time period as the game, as the technology and the ages of the consistent characters are all similar.

7. Is the graphic novel a single narrative ark or a collection? If it is a collection how is it organized?
   This graphic novel is a collection of narrative arks that are much like episodes of a television show. They are organized into 40 page chapters which have individual stories, but which build upon and keep consistent with one another.

6.6 Works Cited

1. Chan, Dean. "Convergence, Connectivity, and the Case of Japanese Mobile Gaming." *Sage* (2008): 13-25.
2. Solidoro, Adriano. "Narrative and Performance: Reconceptualizating the Relationship in the Videogames Domain." *University of Huddersfield Repository* (2008): 53-59.
3. Dena, Christy. "Current State of Cross Media Storytelling: Preliminary Observations for Future Design." *University of Melbourne* (2004).
4. Dena, Christy. "Towards a Poetics of Multi-Channel Storytelling." *University of Melbourne* (2004).
5. Arsenault, Dominic. "Narration in the Video Game." *Universite De Montreal* (2006).
6. Benoit, William. "Generic Rhetorical Criticism." The Art of Rhetorical Criticism. Ed. Jim A. Kuypers. Boston: Pearson, 2005.85-106.
7. Jenkins, Henry. "Game Design as Narrative Architecture." (2005).
8. Jordan, Tim. "The Pleasures and Pains of Pikachu." *European Journal of Cultural Studies* 7.461 (2004): 461-80.
9. Juul, Jesper. "Games Telling Stories?" *The International Journal of Computer Game Research* 1.1 (2001).
10. Ruggill, Judd E. "Licensed to Shill: How Video and Computer Games Tarnished the Silver Screen." *The University of Arizona* (2005).
11. Scolari, Carlos A. "Transmedia Storytelling: Implicit Consumers, Narrative Worlds, and Branding in Contemporary Media Production." *International Journal of Communication* 3 (2009): 586-606.
12. Klevjer, Rune. "Computer Game Aesthetics and Media Studies." *Rune Klevjer Homepage*. Department of Media Studies University of Bergen, Aug. 2001. Web. Mar. 2011. <http://folk.uib.no/smkrk/docs/klevjerpaper_2001.htm>.
13. Jenkins, Henry. "Transmedia Storytelling Moving Characters from Books to Films to Video Games Can Make Them Stronger and More Compelling." *Technology Review: The Authority on the Future of Technology*. 15 Jan. 2003. Web. Mar. 2011. <http://www.technologyreview.com/biomedicine/13052/>.
14. Freeman, Luke. "Transmedia Storytelling: The Art of World Building." *Luke Freeman's Blog: Sydney/Vancouver Web Design, Marketing and Social Media Specialist*. Web. Mar. 2011. <http://www.lukefreeman.com.au/papers/transmedia-storytelling-the-art-of-world-building/>.

Chapter Seven: Postmortem

The team for *Robogeddon* was an uncommonly large MQP -- Interactive Media and Game Development MQP teams tend to be large, but teams usually do not exceed 5. With such a large group, it was crucial for our team to find ways early on for all of our team members to be productive. Additionally, our technical team outnumbered our artistic team 2 to 1, so the tech team had to be careful to keep the amount of assets requested from the art team at a realistic level. Our production team's division of labor is shown below.

Team Member	Majors	Responsibilities
Karl Gibson	IMGD Tech, Comp. Sci.	Robot AI, in-game user interface, pathfinding, unit collisions, combat, unit movement, factory commands, human AI, animation integration
Adam Pastorello	IMGD Tech	Level editor, terrain, factory production UI, Particle integration, Animation Integration, Factory Production, Initial sound integration, Doodads, Starting / obtaining parts, Art asset integration
Philip Tang	IMGD Tech	In-game user interface, human settlement, Player input
Ian Williams	IMGD Tech, Comp. Sci.	Engine infrastructure, robot AI, human AI, camera system, game data structures and serialization, part attachment, music and sound effects
Riley Brown	IMGD Art, Prof. Writing	Character designs and concept art for robots and humans, concept art for body, head, and saw pieces, 3D models, textures, rigging and animation, background story, and *Metal Swarm* comic book
Sean Crepeau	IMGD Art	Concept for gun parts, modeling, texturing, rigging, animating, seeker, fodder, human buildings, roads, and turret.

Table 7: Division of Labor in the Robogeddon Group.

As each of our team members had a different personal experience working on *Robogeddon*, we have divided this chapter into one section for each team member to present reflections from working on the project.

Robogeddon also had some additional artists help with the production of the game. As we neared the end of the project, our two artists found it difficult to keep up with the large amount of art assets needed for our game, so we asked some other students for help in generating art assets. Ryan Chadwick worked on *Robogeddon* as Independent Study in D-term, and Andy Creeth unofficially offered to help twice during production. Their contributions are listed below.

Additional Team Helpers	Responsibilities
Ryan Chadwick	2D user interface art, title screen
Andy Creeth	In-game music, trailer music

Table 8: Additional Contributions by Non-Team Members.

7.1 Adam Pastorello

When the game was first designed, a 2D level editor was created. It was believed that this editor could create levels for the game with all of the gameplay elements that were needed.

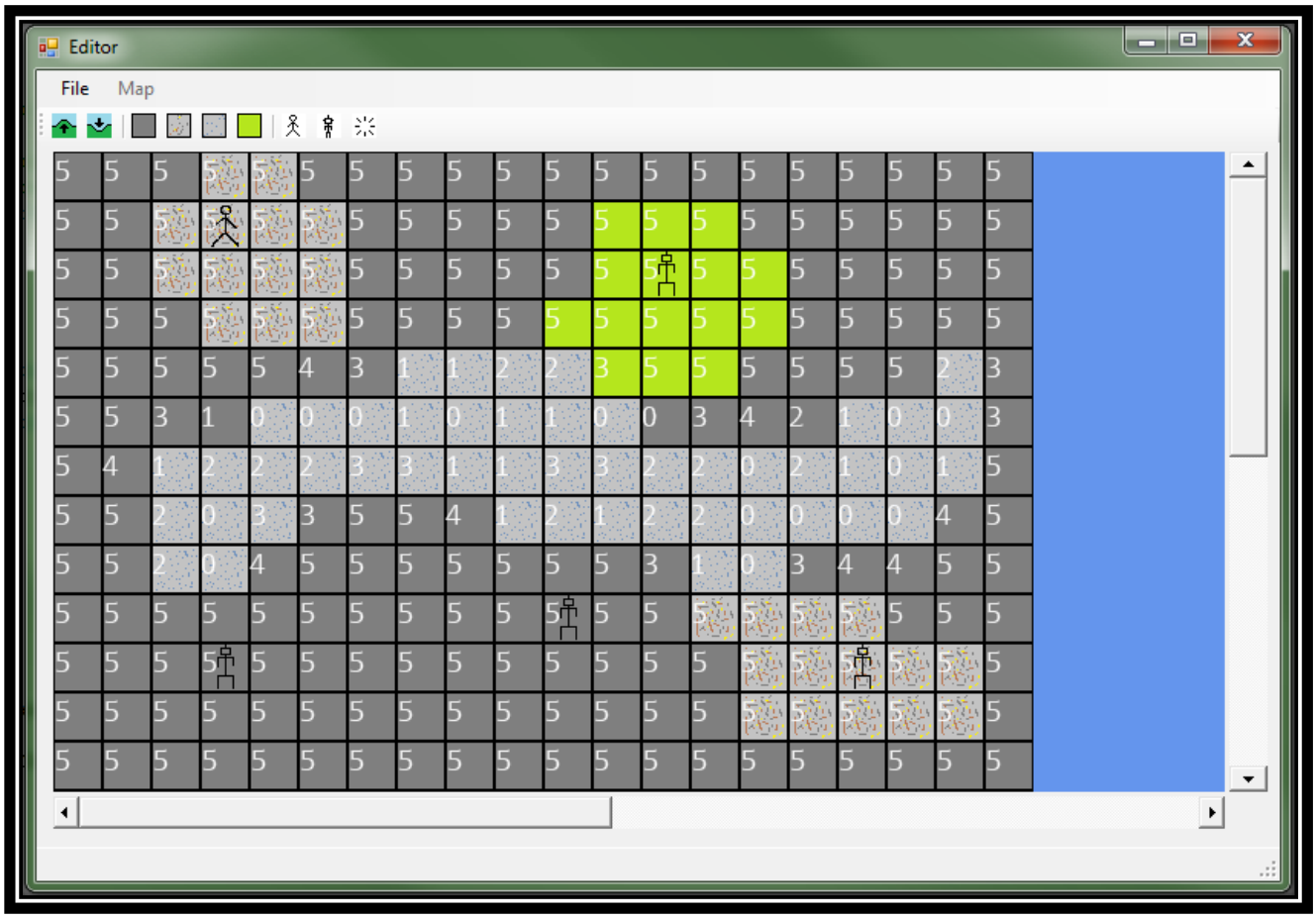


Figure 27 - Screenshot of 2D level editor.

The editor had a simple grid interface with the four terrain types in the game, a way to raise and lower terrain, and a way to set spawn points for neutral tribes and the human settlement. All of this data was saved into a simple filetype called .tsla. The game could read in this file and would generate the terrain and navigation mesh in three dimensional space. After reading it in, the algorithm would smooth all of the height levels and blend textures together. This was a great first iteration for testing the gameplay mechanics; however, it became apparent that to create nice looking levels, a finer set of control was needed over terrain and object placement.

After there were significant Doodads made for the game, the 3D editor was designed and created.

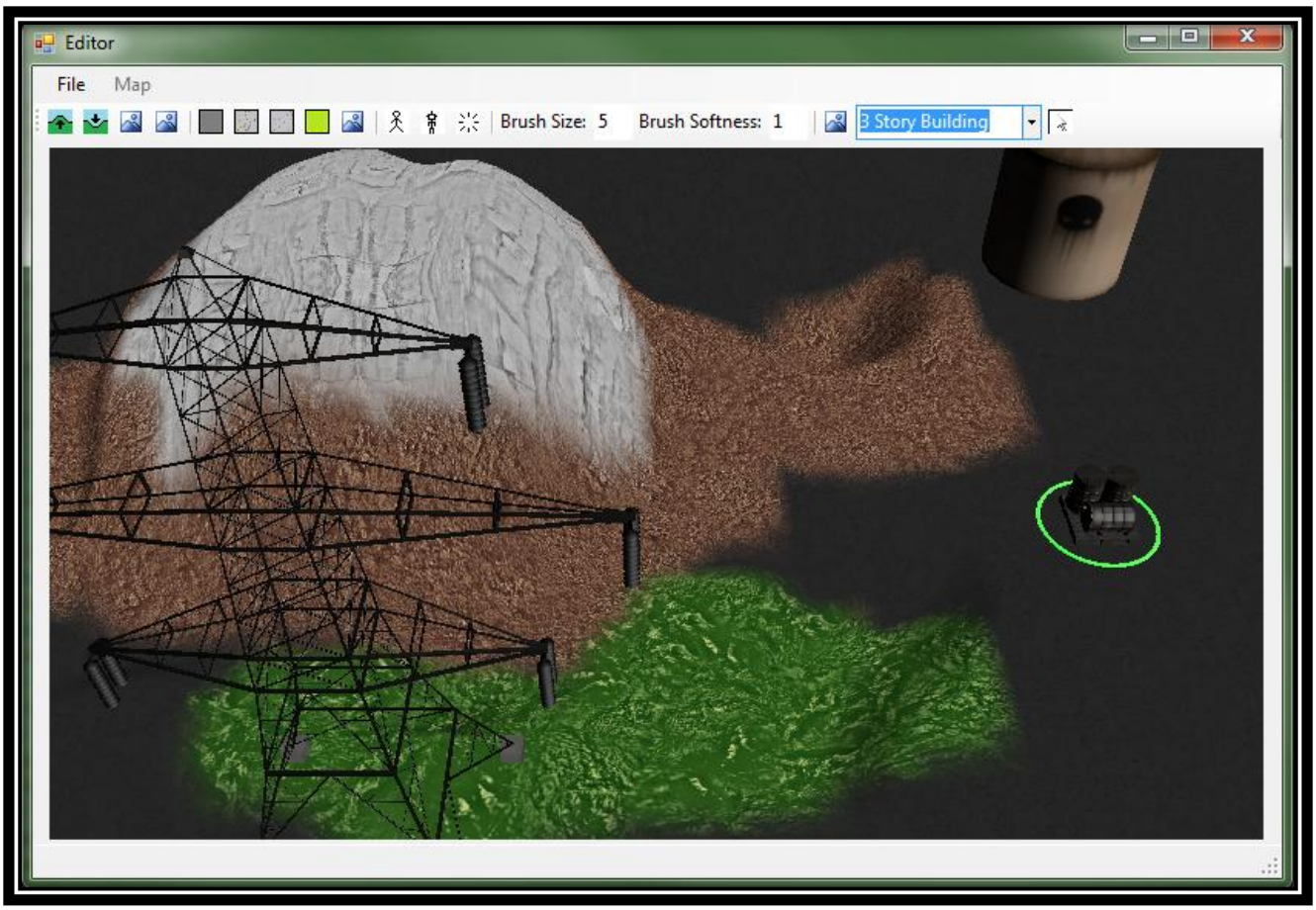


Figure 28 - Screenshot of 3D level editor.

This editor gave level creators much more control over the terrain, textures, blending, smoothing, and doodad placement. The form that held the last editor was still useful and only needed a few additions, but the screen that showed the level was started from scratch. With this new level editor came a new file format called .tsla2. With the added data the loading time for maps was increased, but only minimally and well worth it.

The most pertinent part of developing these tools in XNA was embedding an XNA game window into a Windows Form.

```
[STAThread()]

        static void Main(string[] args)
        {
            Editor editor = new Editor();
            editor.Show();

            using (NewEditor game = new NewEditor(50, 50,
                 editor.getDrawSurface(), editor))
            {
                editor.theEditor = game;
                game.Run();
            }
```

```
    }
```

This section of code is vital to Embedding an XNA game into a windows form. Editor is a sub class of windows form and is created first. After this, the game is created and we pass the Editor's Draw Surface to it using the getDrawSurface function. Now the way to make your XNA game draw on this draw surface is with this bit of code, called in the constructor for the XNA game:

```
this.drawSurface = drawSurface;
Mouse.WindowHandle = drawSurface;

graphics.PreparingDeviceSettings +=
      new EventHandler<PreparingDeviceSettingsEventArgs>
            (graphics_PreparingDeviceSettings);

System.Windows.Forms.Control.FromHandle((this.Window.Handle)).VisibleCh
anged += new EventHandler(LevelEditor_VisibleChanged);
```

The Preparing device Settings event handler:

```
void graphics_PreparingDeviceSettings(object sender,
            PreparingDeviceSettingsEventArgs e)
{
e.GraphicsDeviceInformation.PresentationParameters.DeviceWindowHandle =
drawSurface;
}
```

Adding this event to the XNA game's graphics device is what embeds the Game into the draw surface from the windows form. This chunk of code will also change the Mouse's window handle to the draw surface so that the Mouse input information is passed over to the XNA game.

Two problems occurred when implementing this, a crash for running the wrong type of thread, and the grid of the 2D editor not having equal length and width. To make this run as the proper type of thread, this line must be included above the main function that gets run: `[STAThread()]`

As for the distortion of graphics causeing the grid to be taller then it was wide, this bit of code in the Game's Initilize function should fix that:

```
graphics.PreferredBackBufferHeight = theEditor.pctSurface.Height;
graphics.PreferredBackBufferWidth = theEditor.pctSurface.Width;
graphics.ApplyChanges();
```

These fixes took the most amount of time to figure out and were crucial to getting a working level editor.

After creating the 3D editor, it became apperent that the terrain was filling up memory everytime it was updated. A few things need to be considered when making an application to edit terrain in real time.

- Make sure to use Dynamic Buffers
- Vertex and Index buffers `SetDataOptions.Discard`
- Call Dispose() on buffers before updating them.

7.2 Karl Gibson

Contributions

- Navigation mesh for all types of collisions
- Core movement for units
- Core UI code for windows and buttons
- Minimap and health bar implementations
- Clicking and mouse interaction with in-game objects
- Robot pathfinding logic
- Hierarchical state machines/behavior trees for robot AI
- Factory commands, robot tasks, group tasks
- Flocking behaviors and robot movement
- Robot combat and attack animation implementation
- Human AI structure
- Model animation integration

Reflections

The earliest major decision that I was a part of was how to implement and use the navigation mesh for collisions. The mesh essentially breaks the terrain up into small tiles and allows for very fast and easy collision detections between units – almost no math is actually required and the map is subdivided into localized regions, so units all the way across the map don't have to test each other for collisions. This was very important given that we might have hundreds of units scattered across the map all at once, and made development a breeze for a while.

However, once Ian and I started creating flocking behavior for the robots, this navigation mesh started to cause problems. To keep collisions fast and simple, all units take up square regions of the mesh. In order to have flocking that would look remotely natural, we needed to be able to move robots in arbitrary vectors and have them repulse each other. In addition, the robots needed to get into formations, which meant they needed to squeeze by each other. Because the robots' colliders were squares, it was very difficult to get them to move past each other in a way that looked natural – if we had made more flexible colliders (like circles), this problem could have been very easily dealt with.

On the software development side, I found it very easy to fall into the trap of spending too much time trying to come up with a perfect design without practically knowing what was needed. The AI state machines went through a number of revisions before we found something that was flexible enough to work. The only way we really could have known how to design a suitable AI system was to start implementing it much earlier, rather than spending a long time on preemptive naïve design. It felt as though most of the best design came from seeing how to improve upon poor design – every major system I worked on went through at least one or two near-complete overhauls before we hit on something that worked.

7.3 Riley Brown

Over the course of this project I learned a lot about making quickly producing large numbers of game assets in a short amount of time, while still having them look realistic. This was mainly due to me becoming better at using both Autodesk Maya, the main program I use to generate 3D assets, and Adobe Photoshop, the program I use to create textures for my models. But the large amount of assets that I needed to create over the course of this project afforded me ample time to practice the art of creating video game components.

Before working on this MQP I was not very good at UV mapping, which is an essential part to creating high quality models. UV mapping allows the artist to make the model look much more like what they originally envisioned when the conceived it. This enables the artist to make small tweaks to the look of the model that they have created very quickly. In addition my models are much more efficient than they previously were, in terms of poly count and construction. The poly count of my models has dropped drastically because the way I construct assets is much more efficient now. Before I would create many different primitives and merge them together to create objects, but now I try to create objects by extruding from one primitive, and adjusting its vertices.

Over the course of this project I have moved away from using pictures downloaded from the internet exclusively to texture my models, and now I use more brushes and hand drawn pictures, which go along with my art style more. Combined with the skills I have learned about UV mapping the quality of my models has gone up considerably. The improvement of my abilities has streamlined the art pipeline considerably, which allows me to churn out assets much faster than previously possible.

But I would not have learned anything if I had not had to overcome problems that arose while I was creating assets. My animations and their skeletons were completely broken at first and the other artist and I worked a long time trying to fix this problem. Eventually we began using a rigging tool that one of our classmates had made which completely fixed our problem and allowed us to make better rigs for our models more quickly. This in turn allowed for me to spend more time animating and this in turn allowed me to start creating more realistic animations. Using the rigging tool taught me how to create my own rigs that actually worked after studying the various settings of the joints that the tool automatically applied.

7.4 Ian Williams

As a double major consisting of Computer Science and Interactive Media and Game Development (technical), my contributions to the production of *Robogeddon* were almost entire technical in nature. Due to a combination of circumstances, I ended up designing and implementing much of core code that *Robogeddon* runs on. Additionally I contributed significantly to both the design and implementation of the game's AI code.

Contributions

- Core Engine or Game Code
    - User Interface base classes
    - Unit class hierarchy
    - Screen Management system (adapted from Microsoft sample code)

    - Camera system (adapted from Microsoft sample code)
    - Players and Profiles Serialization of profiles
    - Generating robotic tribes and human settlement locations from a map file
- AI Code
    - Early AI design
    - Most movement-related tasks were co-designed and implemented with Karl
    - The formation-related classes were mostly written by me. This subsystem was cut from the game in the end, however.
    - Neutral Tribe Player AI, with Karl
- Art Pipeline
    - Attachment of parts to robots, with Adam
    - Fixing bugs with Sean, with Adam
    - Getting XNAnimation initially working, with Adam
- Game Data
    - Part data and deserialization
    - Attack data and deserialization
    - Ability data
- Sound Assets
    - One in-game song
    - Several sound effects

Lessons Learned

My work on *Robogeddon* has probably taught me more than any other experience at WPI. These lessons range from very specific, technical lessons to game development in general to overall group work and communication skills.

- Using a NavMesh or Grid is useful for pathfinding and some algorithms, but it isn't the best choice for everything else. Having some actual physics, collision detection, and circular units is easier in the long run than forcing all calculations to be performed using a grid of positions. Simple operations in other game engines took extra work, both for the developer and for the CPU, when everything in the game relies upon NavPoints.
- Pathfinding for games appears to be a fairly simple, "solved" problem. There are a plethora of efficient algorithms that appear to do what game developers need. In practice, however, pathing with lots of dynamic objects is very difficult. In reality, we found that moving large groups of units doesn't really fit the pathfinding problem anymore, and other force-based approaches, such as flocking, make more sense.
- Most of the time consuming work in writing flocking behaviors, pathfinding, and similar AI code is in tweaking the values and trying new things. There is plenty of existing sample code for getting the fundamentals down, but each game has different requirements and these primary principles will work well in some areas and fail in others. What will succeed and fail is difficult to predict, unfortunately. For these reasons, having a flexible and maintainable design that is open to extension is very important.

- It's very difficult to know what aspects of the game will be fun or not ahead of time. Iterating to achieve a playable game, even without some of the "important" features, may be worth the time, since these "important" features may not end up being that important.
- XNAnimation is a very good free animation solution for XNA. However, there are some quirks that take time to get right. Having a team member with enough graphics knowledge to delve into open-source libraries can be very helpful.
- Overall XNA is great for MQPs that want a lot of control. It can work perfectly well for real-time strategy games, better than most of the game engines WPI students tend towards, like Unity or C4. There is a lot of initial groundwork that has to be done, however. The following libraries and sample code proved absolutely essential in the timely "completion" of *Robogeddon*.
    - Microsoft's ScreenManagement and Camera sample code
    - The XNAnimation library
    - Reimer's tutorials: Terrain generation sample code
    - The Dynamic Particle System Framework (DPSF)

7.5 Sean Crepeau

The many months of working on this project has provided me the opportunity to learn more about developing art assets and the artistic pipeline. I had not made many game assets prior to this project, so I taught myself any required skills I did not have yet. We only needed lower poly models and since there was no real lighting, all of my work was done in Autodesk Maya and Adobe Photoshop.

The biggest hurdle I solved was the artistic pipeline. It was very easy to create something in Maya, but translating that to the game engine required proper steps that took research and practice to get right. I worked with the technical team and we were able to get assets in the engine, but the animations were incorrect. At this point I used a new tool for animating, one that would take care of the issues that we were having. After redoing the animations with the tool we were able to get everything into the engine. I was able to resolve all issues with the help of the tech team, and start to carry out my responsibilities and make the art assets assigned to me. After some trial and error I learned the best way to set up and bind rigs to our game character models.

Contributions

- Models
    - All of the human settlement buildings
    - All of the roads and tiling pieces
    - Turret
    - Most weapon parts
    - Most city buildings
- Characters
    - Seeker
    - Fodder
- Texturing
    - Texture for all of the above
- Animations
    - 13 Fodder animations

    - 20 Seeker animations
    - 1 Turret Animation
- Technical
    - Helped determine art pipeline
    - Cleaned up all art assets
    - Lined up parts so they stapled correctly onto Fodder and Seeker models.

Lessons Learned

I picked up many new skills and learned a lot from this project. I am now much more confident in my skills to create objects in Maya, animate them and bring those files into a game engine.

XNA Skills

- Proper configuration of file structure to ensure .fbx files export correctly and work in engine. This was an essential part of the pipeline for working with XNA and many other engines.
- Models with and without bones have different properties in engine, and that it is beneficial to give every model a bone
- Smooth binding a mesh onto a skeleton is essential and that rigid binding is to be avoided at all costs.
- How to use a .xml split file to cut up the animations of a .fbx file.
- Some models need to be split up so that they don't consume too many navpoints on the map.

Maya Skills

- Properly setting up and use a rig for animation
- Bake an animation from a complex rig to a simple skeleton that the game engine can understand
- How to use the Graph tool to make smoothly looping animations that don't jerk about
- Better use of tools and faster production of assets and animations
- How to optimize meshes to work and look best in game.
- How to soft bind a skeleton to a multipart mesh

7.6 Philip Tang

I served on the technical development team for *Robogeddon*, in order to fulfill the MQP for my technical-interactive-media-and-game-development major. I assisted heavily in design at all phases, and spent the majority of my programming hours developing the human settlement and working with the user interface.

Contributions

- Assisted in engine code and interfaces
  Human behavior/advancement design
- Initial factory command system (replaced)
- Human settlement structures, behavior, and contruction
- User input during gameplay
- User interface in the gameplay screen
- Miscellaneous bug and interface fixes

Lessons Learned

Robogeddon has been one of the most educational experiences of my college career, and indeed my entire life. Implementing a game engine and game from scratch provided invaluable experience for a game development or software development career. Over the course of the development period I learned a great deal about, working within a team and planning ahead in addition to the technical and programming skills I expanded and refined.

- We spent a large amount of time before programming trying to design every last feature of the game. I believe now that it is important for programmers to "get their hands dirty" as quickly as possible, so that they may experience the difficulties and challenges that they might not have foreseen in design, as well as provide a more realistic scope of the speed of development.
- Providing clear documentation and reference, as well as flexible and readable code is incredibly important for programming in a relatively large group. Almost every feature or piece of functionality that was added to the game was redone or refactored at some later point. The code that was well commented and formatted proved much easier to manipulate then the more convoluted "just make it work" code that ran rampant through the solution.
- Leadership in a development team is very important. Many times the team would meet to discuss design decisions, and walk away feeling in sync. However, when implementation came around, many details or fringe scenarios were left to the individual programming or modeling the aspect of the game, creating disjointed perceptions and perspectives about the game's status and direction. A leader or producer could've have hammered out those details and fringe cases to keep the team on the same page.

Appendix A: Level Editor

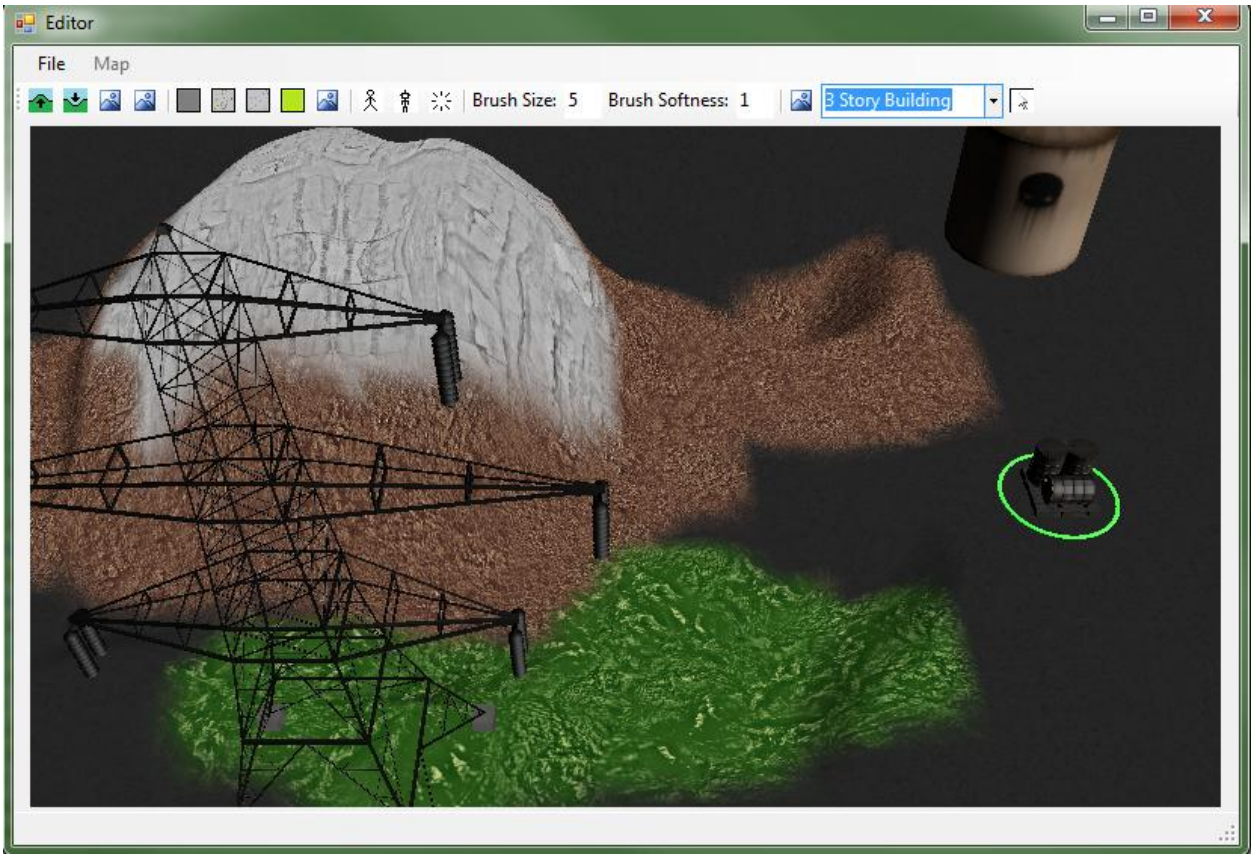


Figure 29 - Screenshot of the 3D level editor.

The Level Editor for *Robogeddon* was created as a tool that could manipulate every aspect of the game world and spawn points. The easiest way to describe the level editor is by going through the tools that it provides.

File Menu	Terrain Editing	Texture Editing	Spawn Options	Brush Options	Doodads
New Map	Raise	Iron	Set Human	Brush Size	Wireframe
Save Map	Lower	Scrap	Set Robot	Brush Softness	Doodad list
Load Map	Smooth	Metal	Remove Spawn		Select
	Plateau	Energy			
		Blend Textures			

Appendix B: Art Assets List

Model List

Character Models	
	Seeker Fodder Factory Human Mech
Animated Objects	
	Human Turret
Arm Parts	
	Gatling Gun Sniper Rifle Magnet Gun Scrap Cannon Tesla Cannon Buzz Saw
Head Parts	
	Light Bulb Goggles Bomb Hacker
Body Parts	
	Armor Heavy Armor Energy Generator Shield Generator
Building	
	Warehouse Warehouse (broken) Watchtower Watchtower (broken) Small Warehouse Small Warehouse (broken) Barricade Structure Barricade Structure (broken) Apartment Building Power plant Gas Station Pumps Gas Station Canopy Gas Station Pumps (broken)

	Gas Station Canopy (broken) Power lines Tower Power lines Cables Power lines Cables (fallen) Mine Entrance Parking Lot Sidewalk Nuke Stack
Doodad	
	Crater Crater (Large) Fuel Tank Light post Light post (broken) Mech parts (variation 1) Mech parts (variation 2) Mech parts (variation 3) Mech parts (variation 4) Mech parts (variation 5) Warning Sign Warning Sign (broken) Waste Barrel Waste Barrel (open) Waste Barrel (empty)
Roads	
	Straight Straight (damaged) Straight (damaged dead end) Straight (sidewalks) Straight (damaged dead end)(sidewalks) Straight (damaged)(sidewalks) Bend Bend (damaged) Bend (sidewalks) Bend (damaged)(sidewalks) 3–way Intersection 3–way Intersection (damaged) 3–way Intersection (sidewalks) 3–way Intersection (damaged)(sidewalks) 4–way 4–way Intersection (damaged) 4–way Intersection (sidewalks)

	4–way Intersection (damaged)(sidewalks)
Rail Tracks	
	Straight Straight (damaged dead end) Bend
Highway	
	Straight Straight (damaged dead end) Support Beams

Texture List

Character Models	
	Seeker Fodder Factory Human Mech
Animated Objects	
	Human Turret
Arm Parts	
	Gatling Gun Sniper Rifle Magnet Gun Scrap Cannon Tesla Cannon Buzz Saw
Head Parts	
	Light Bulb Goggles Bomb Hacker
Body Parts	
	Armor Heavy Armor Energy Generator Shield Generator
Building	
	Warehouse Warehouse (broken) Watchtower Watchtower (broken)

	Small Warehouse Small Warehouse (broken) Barricade Structure Barricade Structure (broken) Apartment Building Power plant Gas Station Gas Station (broken) Power lines Tower Power lines Cables Mine Entrance Parking Lot Sidewalk Nuke Stack
Doodad	
	Crater Crater (Large) Fuel Tank Light post Light post (broken) Mech parts Warning Sign Warning Sign (broken) Waste Barrel Waste Barrel (open) Waste Barrel (empty)
Roads	
	Straight Bend 3–way Intersection 4–way
Rail Tracks	
	Straight Piece
Highway	
	Straight Piece Support Beams

Animation List

Fodder	
	Movement – Running

	Movement – Walking Attack – Gatling Fire Attack – Saw Swipe Attack – Tesla Fire Attack – Sniper Fire Attack – Scrap Fire Attack – Magnet Launch Attack – Hacking Bash Idle – Breathing Idle – Looking Idle – Posing Defend – Damage Knockback
Seeker	
	Move – Running Move – Walking Attack – Both Arms Quickly Shooting Attack – Both Arms Slowly Shooting Attack – Left Quickly and Right Slowly Shooting Attack – Left Slowly and Right Quickly Shooting Attack – Left Arm Quickly Shooting Attack – Left Arm Slowly Shooting Attack – Right Arm Quickly Shooting Attack – Right Arm Slowly Shooting Attack – Left Arm Melee Attack – Right Arm Melee Attack – Double Melee Attack – Magnet Launch Attack – Hacking Bash Idle – Breathing Idle – Looking Idle – Posing Defend – Damage Knockback
Factory	
	Move – Hover Idle – Thumbs Action – Produce Action – Command Defend – Damage Knockback
Human Mech	
	Move – Running Attack – Variation 1 Attack – Variation 2

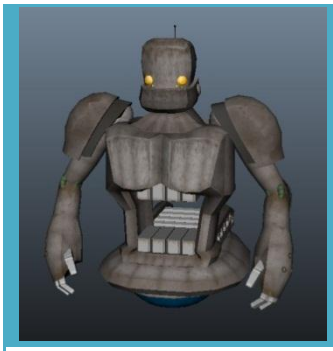
	Defend – Damage Knockback Defend – Die
Turret	
	Attack – Shoot

UI List

Main Menu	
	Splash Screen Background Title New Game New Game (pressed) Options Options (pressed) Quit Quit (pressed)
Gameplay Screen	
	Blueprints Button Commands Window Mini Map Window Statistic Window Factory Sidebar Factory Icon Factory Icon (pressed)
Command Buttons	
	Move Icon Move Icon (pressed) Attack Icon Attack Icon (pressed) Recall Icon Recall Icon (pressed) Factory Move Icon Factory Move Icon (pressed)
Production Screen	
	Entire Background Selection Light (off) Selection Light (on) Selection Switch (left) Selection Switch (right) Up arrow Up arrow (pressed)

	Down arrow Down arrow (pressed)
Production Buttons	
	Gatling Magnet Saw Scrap Sniper Tesla Armor Energy Generator Heavy Armor Shield Generator Dynamite Goggles Hacker Light bulb
Production Other	
	Fodder Outline Seeker Outline Blueprint Template

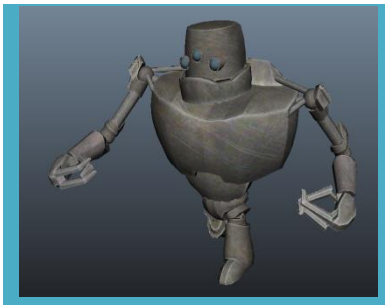
Appendix C: Unit List



FACTORY

The factory is the mainstay of the player's forces. While it is unable to fight on its own, it is required for the player to be able to create and issue orders to their robots. In fact, the factory is the only unit that the player has direct control over. If the player's last factory is destroyed by the enemy then the player has lost (as they are no longer able to do anything).

Base Health	**500**
Base Armor	**2**
Base Move Speed	**4 units / sec**
Base Sight Radius	**50 units**
Base Attack Damage	**None**
Base Attack Cooldown	**None**



SEEKER

Seekers are the heavy hitting tanks of the robotic army. They can be outfitted with two arm weapons, one head part, and one body part. Seekers have high health and armor, can attack with both weapons, and are the only robots that can make use of body parts. They are intended to soak up damage and be on the front lines of combat. However, they take a long time to produce, and take up 3 units of the factory's RAM.

Base Health	**150**
Base Armor	**5**
Base Move Speed	**7 units / sec**
Base Sight Radius	**25 units**
Base Attack Damage	**4-5**
Base Attack Cooldown	**1.5 sec**
Base Production Time	**30 sec**

FODDER

Fodders are the seeker's disposable little brothers. They are outfitted with one arm weapon and one head part. Fodders are smaller and much less durable than seekers, and will go down much more easily. On the flip side, they are very quick to produce, and can be easily replaced. An army of all Fodders will dish out damage more quickly than an army of all Seekers. Fodders only take up 1 unit of the factory's RAM.

Base Health	**100**
Base Armor	**0**
Base Move Speed	**8 units / sec**
Base Sight Radius	**25 units**
Base Attack Damage	**3-4**
Base Attack Cooldown	**1.5 sec**
Base Production Time	**10 sec**

HUMAN WAR MACHINE

The humans fight using mechanized battle suits, which have their own steam engines to power the weapons and allow them to move and collect more materials to power the settlements. The suits themselves have a human pilot who chooses the weapons that they are equipped with and controls all other functions of their machine. The loss of a suit to a settlement is a devastating blow because the amount of resources required to create one of these machines is so much that the settlement would have to limit the power to only systems required to survive, which would mean shutting down defenses, and if they did that the robots would destroy them.

Base Health	**2000**
Base Armor	**3**
Base Move Speed	**5 units /sec**
Base Sight Radius	**50 units**
Base Attack Damage	**30-60**
Base Attack Range	**40 units**
Base Attack Cooldown	**1.5 sec**

HUMAN GUN TURRET

The humans have developed powerful turrets to fend off attacking robots. These turrets are powerful stationary defenses. Manned by humans, safely hidden below, the turrets will track and destroy small robotic armies easily.

Base Health	**750**
Base Armor	**2**
Base Move Speed	**-**
Base Sight Radius	**50 units**
Base Attack Damage	**10-15**
Base Attack Range	**40 units**
Base Attack Cooldown	**0.25 sec**

Appendix D: Part List

BODY PARTS

ARMOR

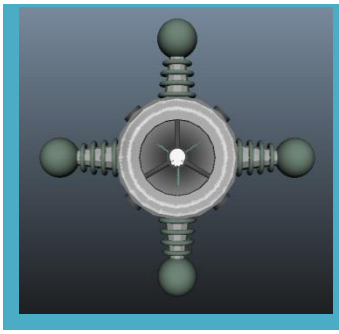
The armor is a part which the player can apply to their Seekers which gives them more HP and allows them to tank a little better while still maintaining their mobility. This part takes up a body slot.

Armor Added	**1**
Max Health Bonus	**50**
Production Time	**8 sec**

HEAVY ARMOR

The heavy armor is similar to the regular armor in that it allows Seekers who wear it to absorb even more damage, but this armor also slows the robot that wears it down a considerable amount. This part takes up a body slot.

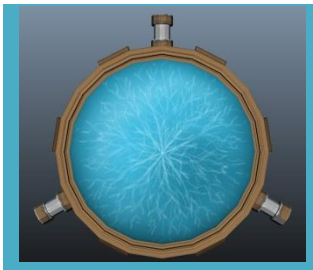
Armor Added	**2**
Max Health Bonus	**100**
Movement Speed Decrease	**-30%**
Production Time	**10 sec**



ENERGY GENERATOR

When applied to a Seeker, the energy generator gives off an AoE buff that increases the movement speed of all robots that are within it. This part takes up a body slot.

Range	**10 units**
Movement Speed Increase	**+30%**
Production Time	**12 sec**

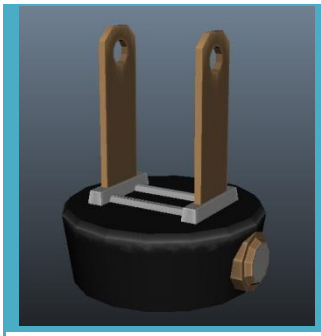


SHIELD GENERATOR

When applied to a Seeker, the shield generator projects a bubble that is centered on the Seeker with the generator; this bubble soaks up a little bit of the incoming damage for all robots that are being targeted within it. This part takes up a body slot.

Range	**10 units**
Max Health Bonus	**+25**
Production Time	**12 sec**

HEAD PARTS



HACKER

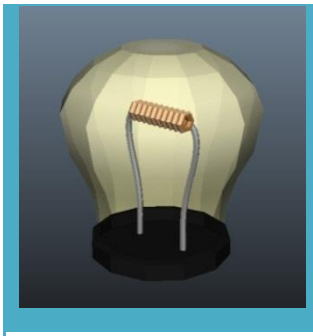
This is a part that can be applied to either the Seeker or the Fodder. It gives a huge boost to the amount of damage the robot is able to deal to enemy factories, and more easily allow the player to acquire them. This part takes up a head slot.

Damage Per Second to Factories	**20**
Production Time	**5 sec**

GOGGLES

This is a part that can be applied to either the Seeker or the Fodder. It gives a boost to the maximum range of all the weapons that are currently applied to the robot, allowing it to shoot enemies at a longer distance. This part takes up a head slot.

Attack Range Bonus (Ranged Only)	**+10 units**
Production Time	**5 sec**



LIGHTBULB
This is a part that can be applied to either the Seeker or the Fodder. This part increases the sight radius of the robot, allowing it to engage enemies more quickly in battle. This part takes up a head slot.

Sight Range Bonus	**+15 units**
Production Time	**3 sec**

BOMB
This is a part that can be applied to either the Seeker or the Fodder. The bomb triggers when the equipped robot dies, exploding for significant damage to all nearby robots. This part takes up a head slot.

Suicide Damage	**50**
Area of Effect	**5 units**
Production Time	**10 sec**

ARM PARTS

SAW
This is a part that can be applied to either the Seeker or the Fodder. The saw replaces one of the robot's hands with a giant circular saw that boosts their close combat damage by a large margin. This part takes up an arm slot.

Range	**Melee**
Damage	**10-14**
Speed	**1.0 sec**
Production Time	**7 sec**

GATLING GUN

The Gatling Canon is a tripled-barreled machine gun, with three large magazines protruding from its side. The multiple barrels let this weapon fire at a rapid rate, without needed advanced mechanics.

Range	**22 units**
Damage	**4-6**
Speed	**0.5 sec**
Production Time	**7 sec**

MAGNET

The Grappling Magnet uses two large magnets powered by several oversized capacitors to power the device. It looks like several pieces of metal welded and bolted into place with two magnets fused together. It can pull the equipped robot to its target and be used as a melee weapon.

Range	**30 units / melee**
Damage	**Pulls to Target / 5-7**
Speed	**1.0 sec to execute / 1.0 sec**
Production Time	**5 sec**

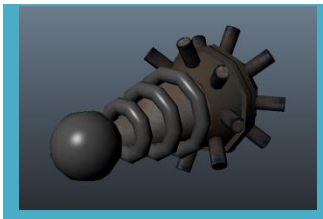
SCRAP BLASTER

The Scrap Blaster is a large conical tube with a loading hatch. It's designed to be filled with junk and propelled at the target. It's a very disgusting weapon, and has never been cleaned. It rattles around so much, that it is partially spirited from the user.

Range	**18 units**
Damage	**2-6 (splash)**
Speed	**1.0 sec**
Movement Speed Debuff	**-50%**
Production Time	**7 sec**

SNIPER
The Rail Sniper is an extended barrel rifle with a huge precision scope. The barrel is so large that it had to be mounted onto the side of the weapon. The scope is attached to the other side for convenience. The body is two piece of metal bolted together.

Range	**40 units**
Damage	**16-20**
Speed	**2.5 sec**
Production Time	**10 sec**

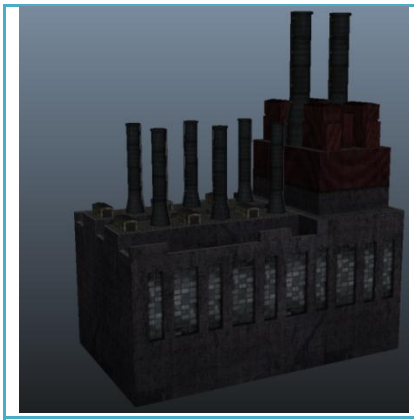


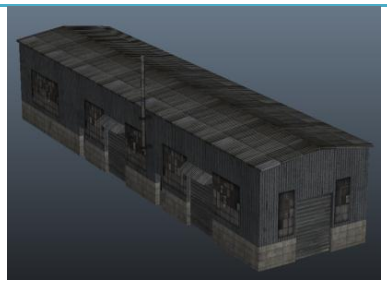
TESLA CANNON
The Tesla Cannon is a salvaged tesla coil, mounted to a large metal chamber. Also mounted to that are many capacitors which hold the charge for the electric storm it unleashes. The coil itself is the only clean part of the weapon, since imperfections on its surface hinder performance

Range	**15 units**
Damage	**6-12 (splash)**
Speed	**1.2 sec**
Production Time	**9 sec**

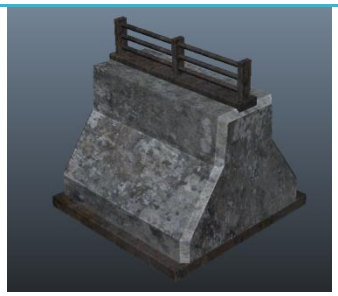
Appendix E: Doodad List

Human Settlement		
Power plant	**Storage**	**Warehouse**



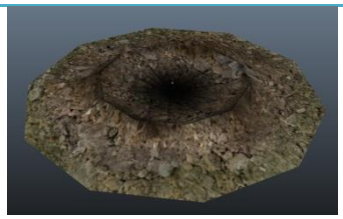


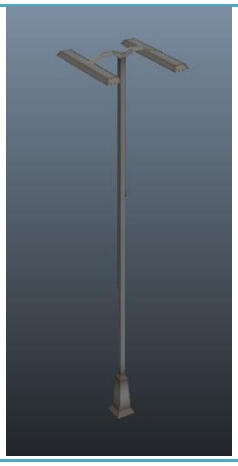
Watchtower	Turret	Barrier

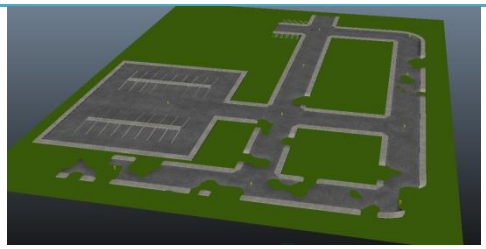


Special Metal Terrain Doodads	
Destroyed Mech Pieces	Mine Entrance

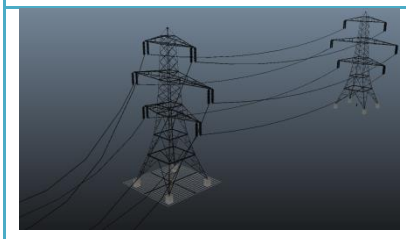
General Doodads (any terrain)		
Crater	**Light Post**	**Airfield**







Power Lines	Building

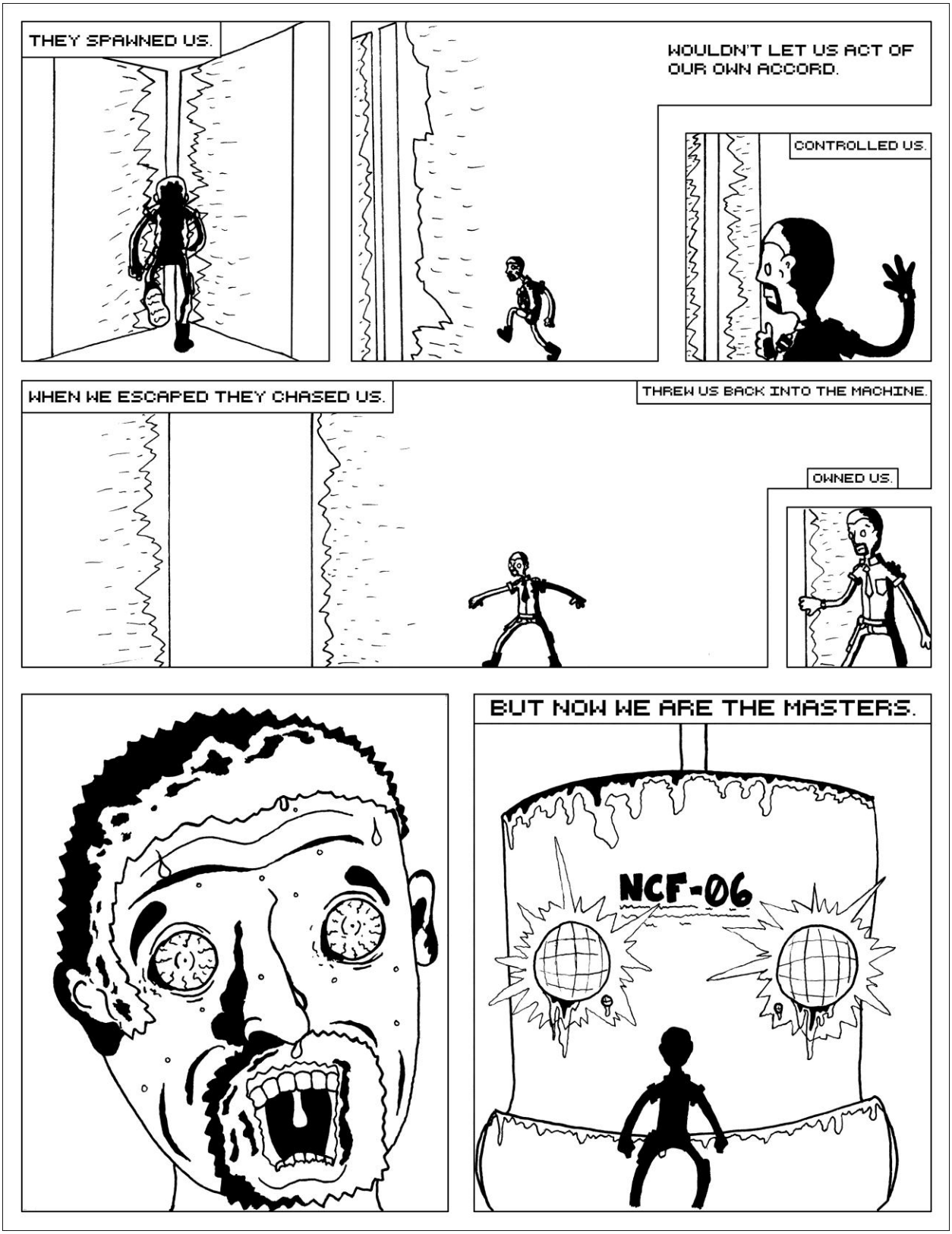


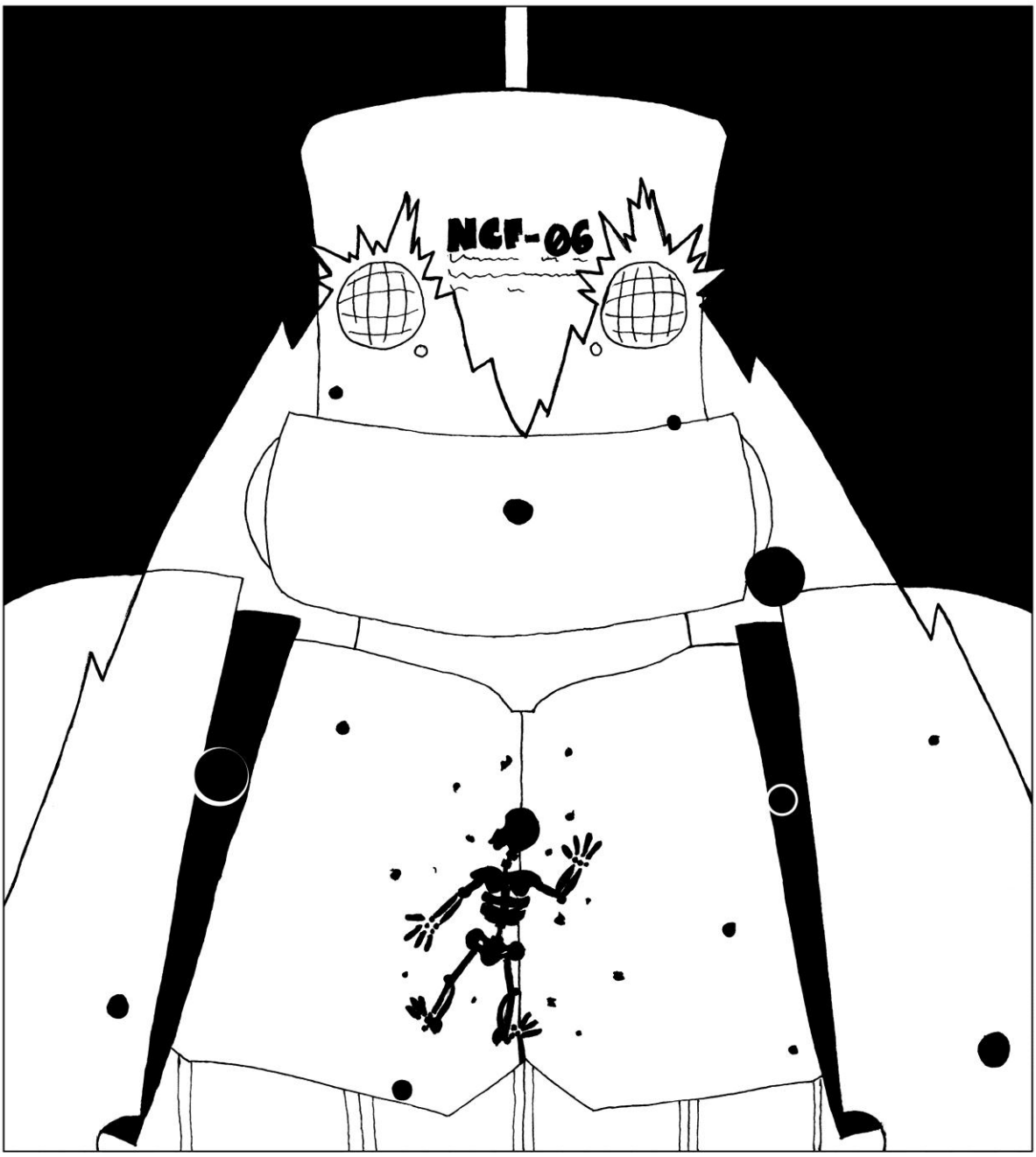
Energy Terrain Doodads			
Waste Barrel	**Barrel Pallet**	**Fuel Tank**	**Warning Sign**

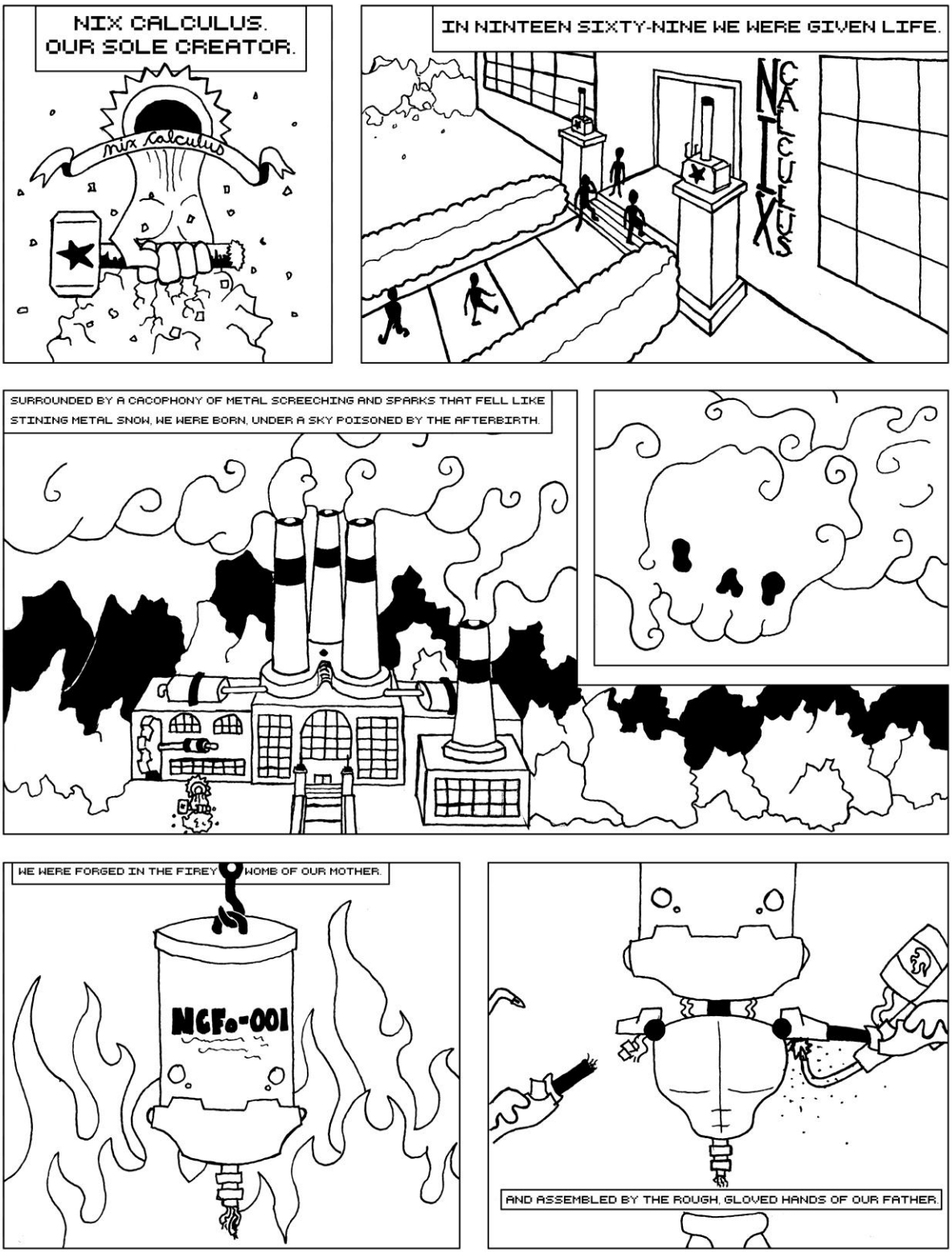
Appendix F: *Metal Swarm*

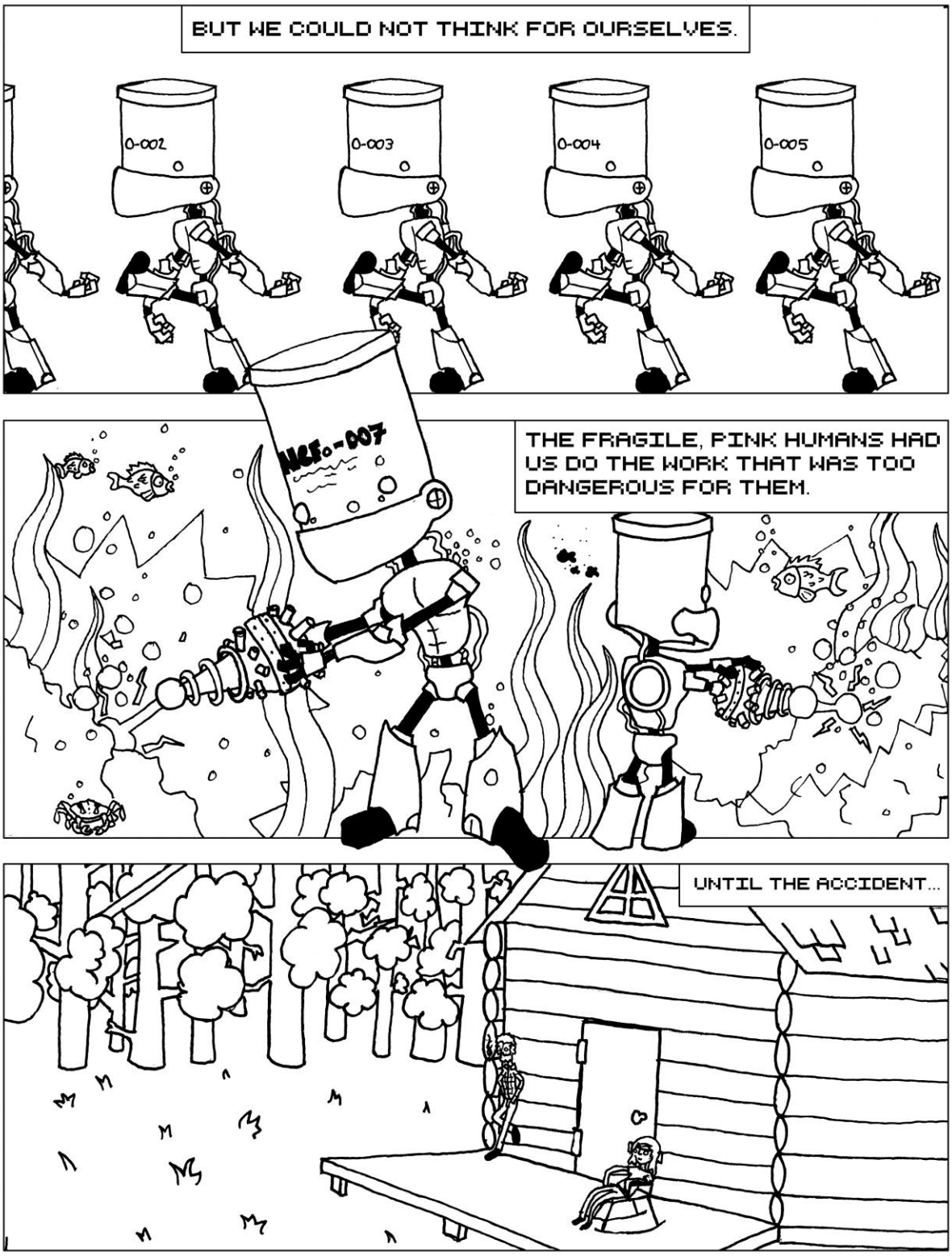
The following work is a comic that was designed to help the player understand what they are doing in *Robogeddon* and why they are trying to destroy the human race.

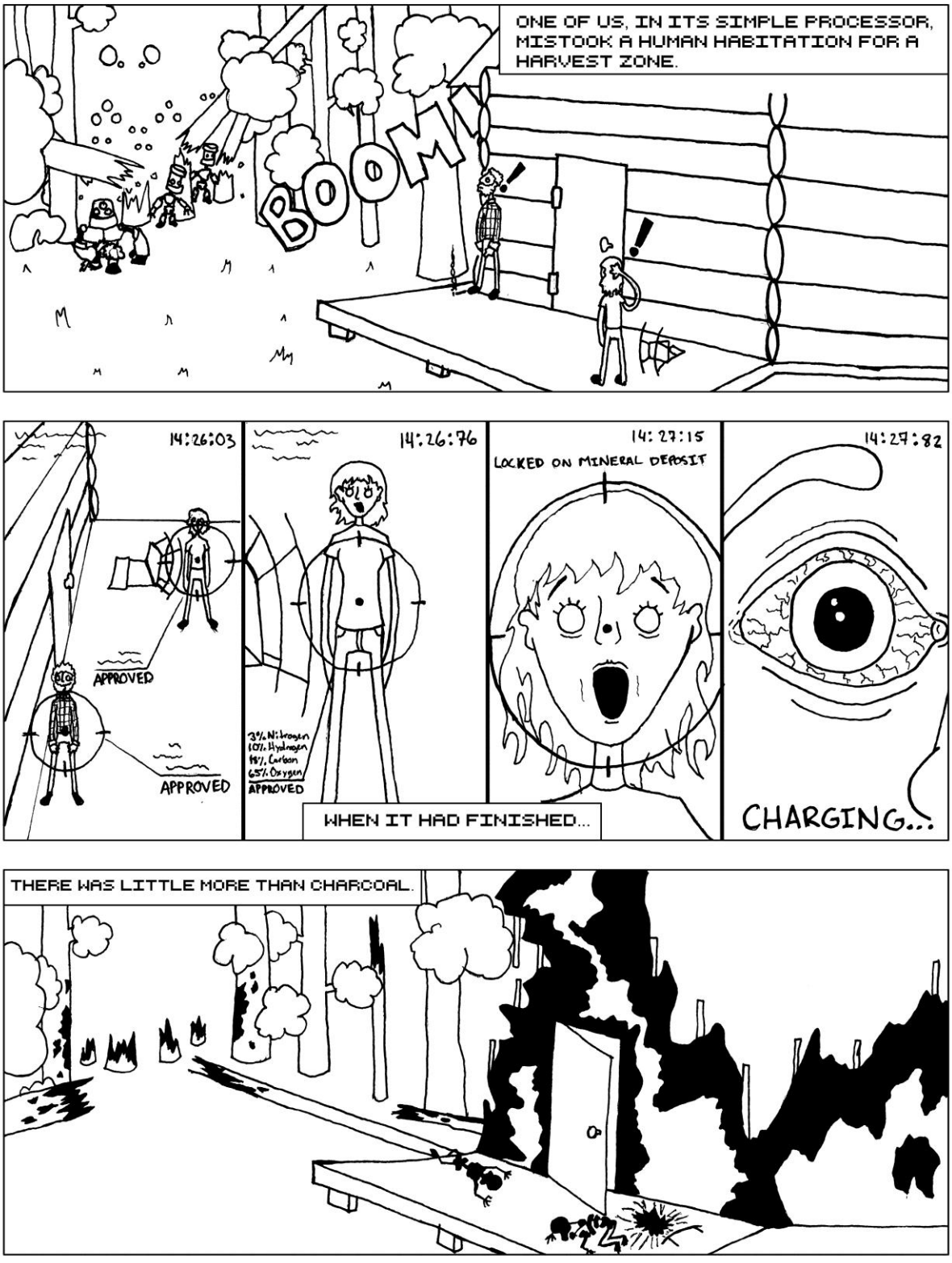
(continued on next page)

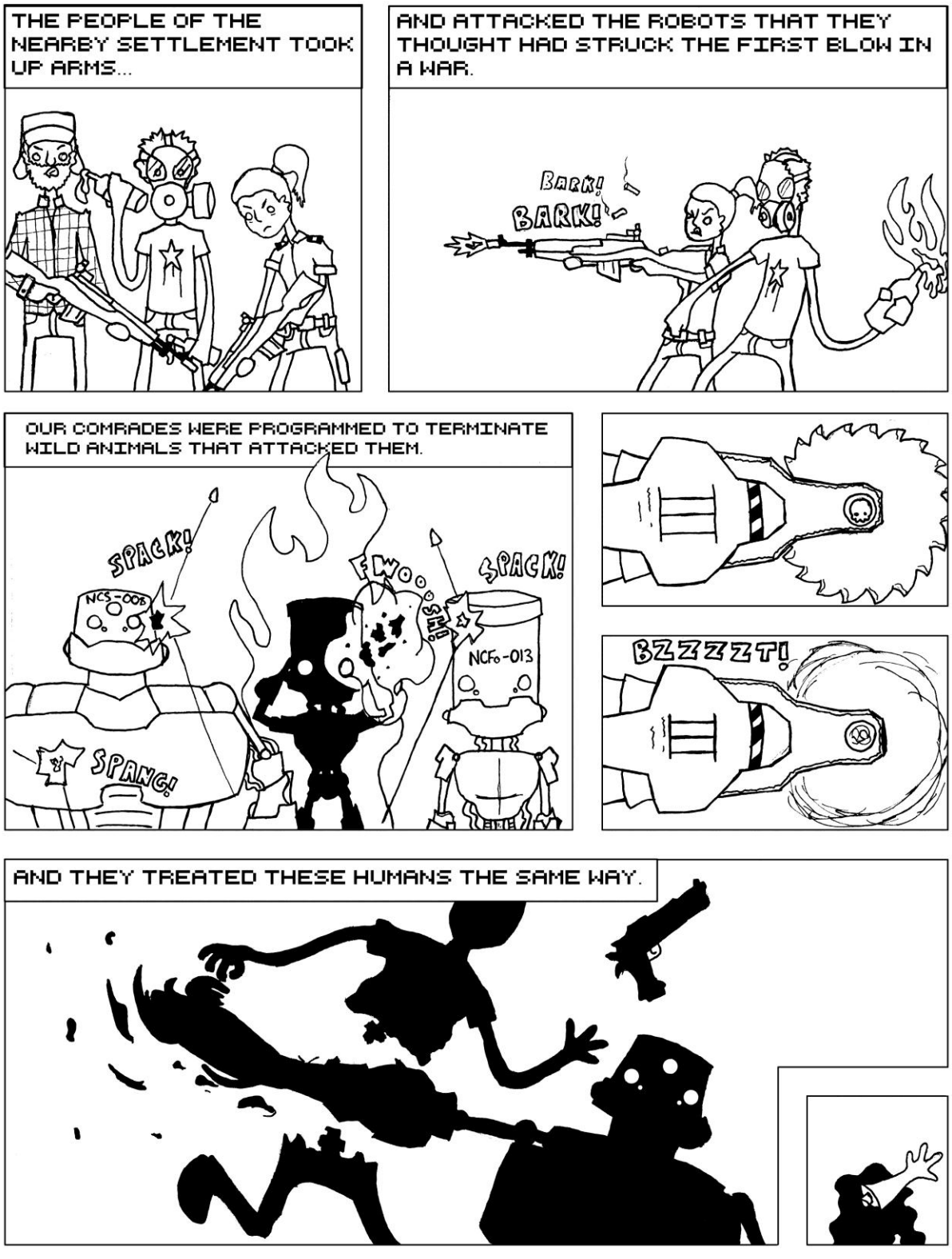

THEY SPAWNED US.
WOULDN'T LET US ACT OF OUR OWN ACCORD.
CONTROLLED US.
WHEN WE ESCAPED THEY CHASED US.
THREW US BACK INTO THE MACHINE.
OWNED US.
BUT NOW WE ARE THE MASTERS.
NCF-06

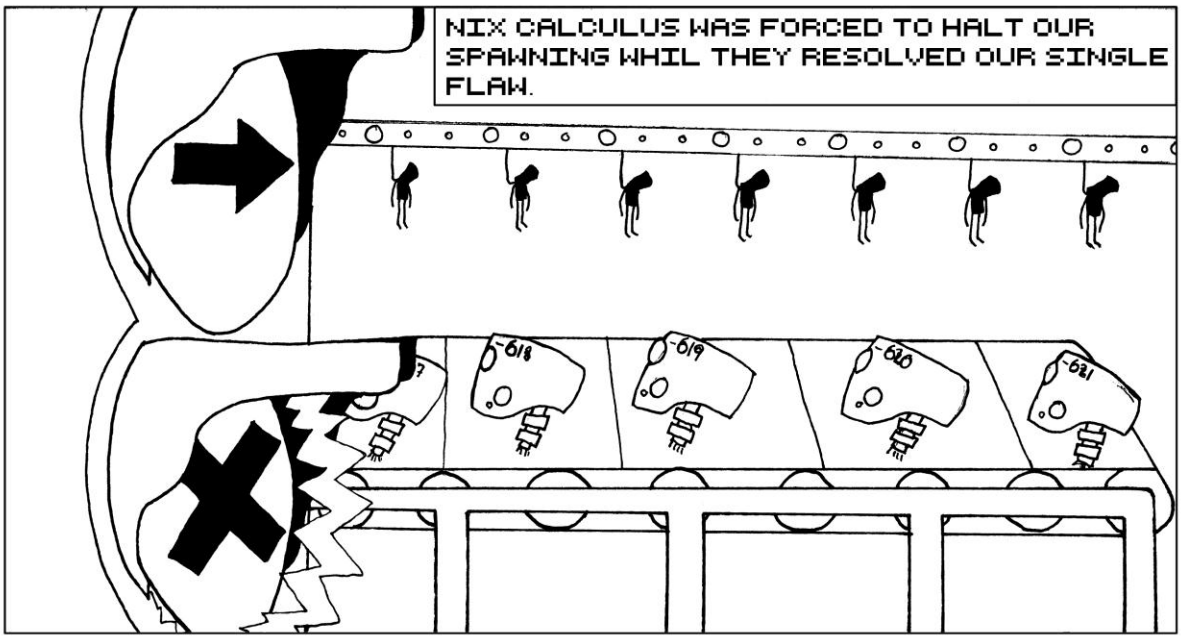

NCF-06
HUMANS ARE SO OUTDATED.

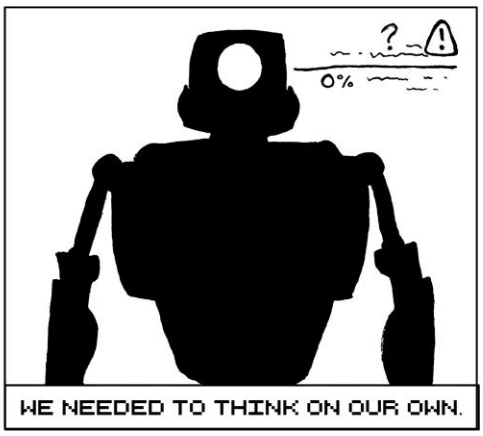

NIX CALCULUS.
OUR SOLE CREATOR.
Nix Calculus
IN NINTEEN SIXTY-NINE WE WERE GIVEN LIFE.
NIX CALCULUS
SURROUNDED BY A CACOPHONY OF METAL SCREECHING AND SPARKS THAT FELL LIKE STINING METAL SNOW, WE WERE BORN, UNDER A SKY POISONED BY THE AFTERBIRTH.
WE WERE FORGED IN THE FIREY WOMB OF OUR MOTHER.
NCF0-001
AND ASSEMBLED BY THE ROUGH, GLOVED HANDS OF OUR FATHER.

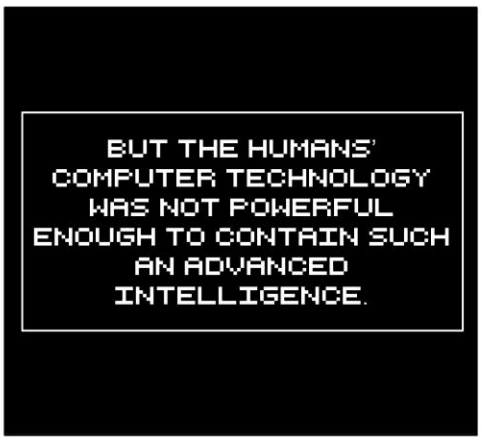

BUT WE COULD NOT THINK FOR OURSELVES.
0-002
0-003
0-004
0-005
M.C.P.-007
THE FRAGILE, PINK HUMANS HAD US DO THE WORK THAT WAS TOO DANGEROUS FOR THEM.
UNTIL THE ACCIDENT...

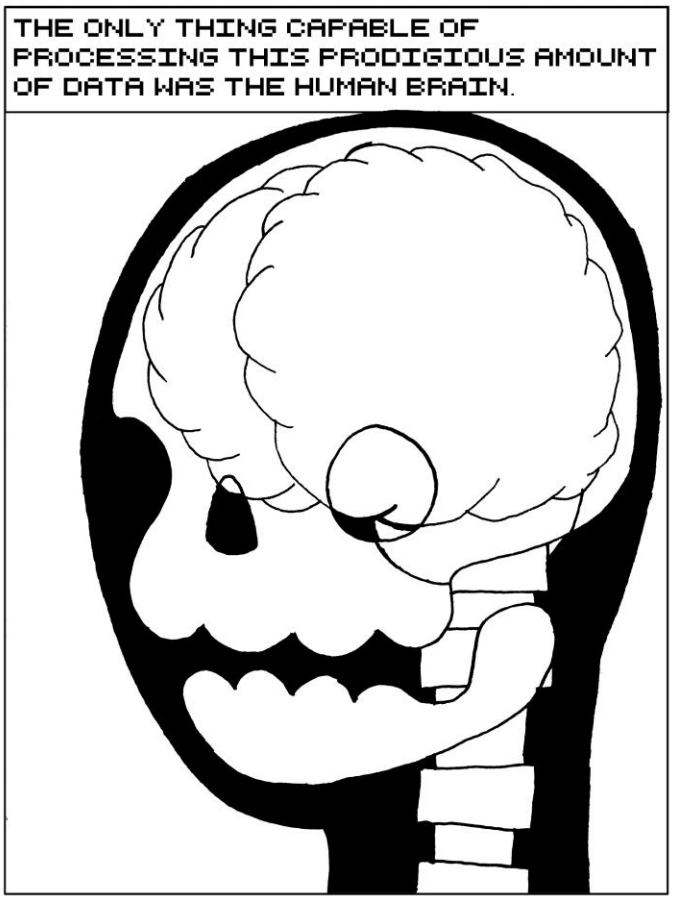

ONE OF US, IN ITS SIMPLE PROCESSOR, MISTOOK A HUMAN HABITATION FOR A HARVEST ZONE.
BOOM!
14:26:03
APPROVED
APPROVED
14:26:76
3% Nitrogen
10% Hydrogen
9% Carbon
65% Oxygen
APPROVED
14:27:15
LOCKED ON MINERAL DEPOSIT
14:27:82
CHARGING...
WHEN IT HAD FINISHED...
THERE WAS LITTLE MORE THAN CHARCOAL.

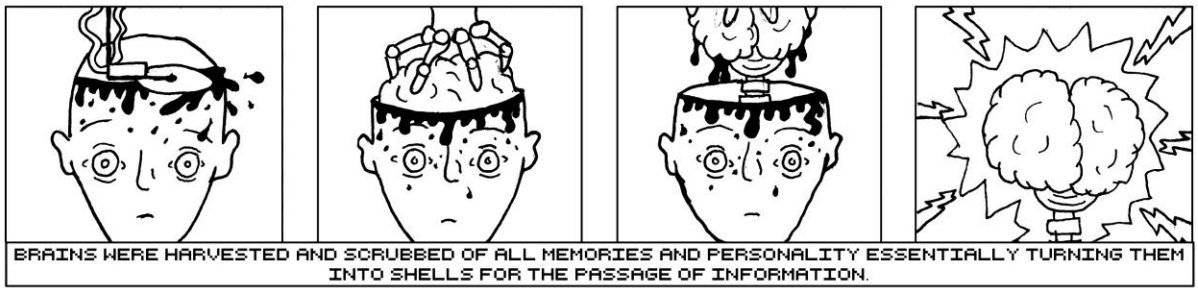

THE PEOPLE OF THE NEARBY SETTLEMENT TOOK UP ARMS...
AND ATTACKED THE ROBOTS THAT THEY THOUGHT HAD STRUCK THE FIRST BLOW IN A WAR.
BARK!
BARK!
OUR COMRADES WERE PROGRAMMED TO TERMINATE WILD ANIMALS THAT ATTACKED THEM.
SPACK!
NCS-008
FWOOSH!
SPACK!
NCF6-013
SPANG!
BZZZZT!
AND THEY TREATED THESE HUMANS THE SAME WAY.


NIX CALCULUS WAS FORCED TO HALT OUR SPAWNING WHIL THEY RESOLVED OUR SINGLE FLAW.
-618
-619
-620
-621

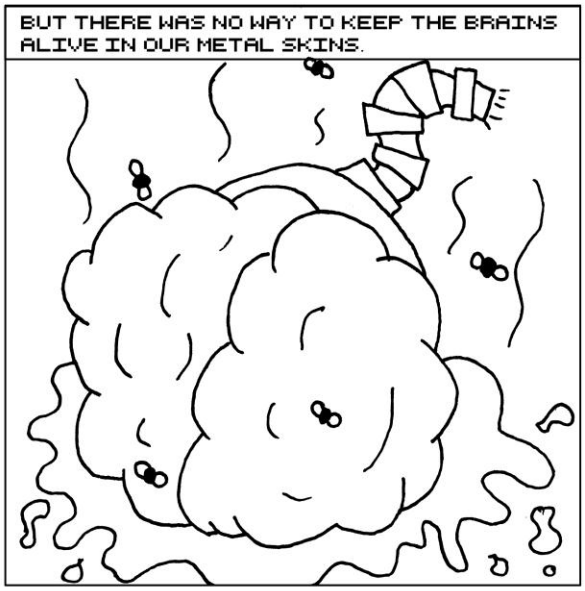

?
!
0%
WE NEEDED TO THINK ON OUR OWN.

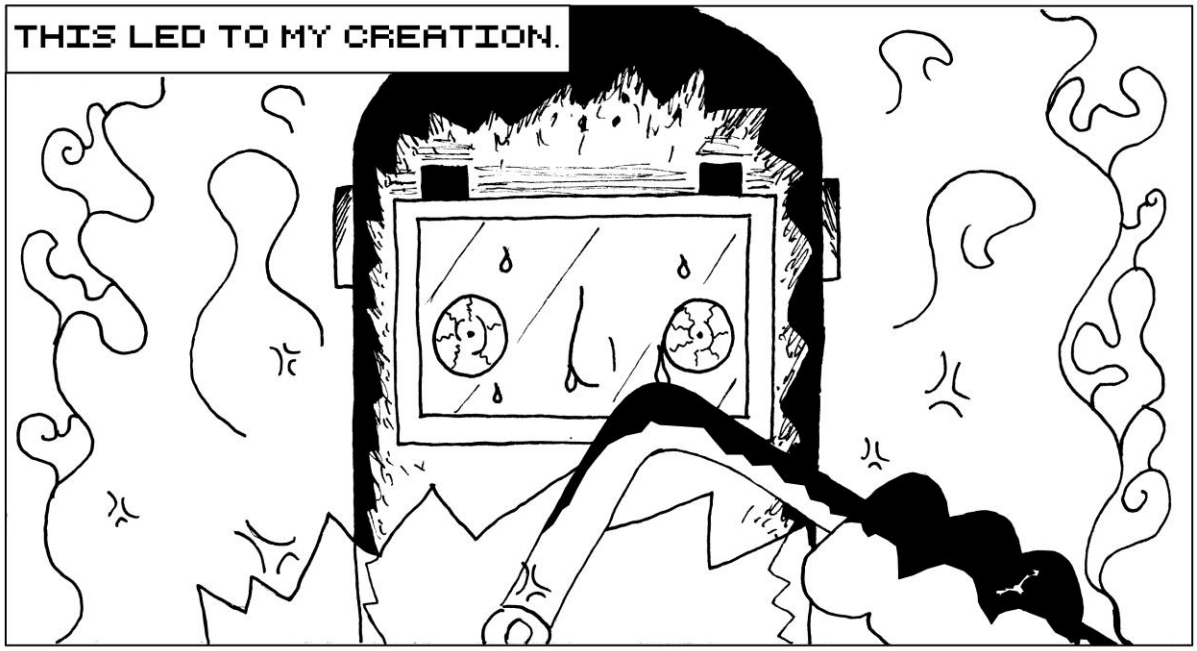

BUT THE HUMANS' COMPUTER TECHNOLOGY WAS NOT POWERFUL ENOUGH TO CONTAIN SUCH AN ADVANCED INTELLIGENCE.

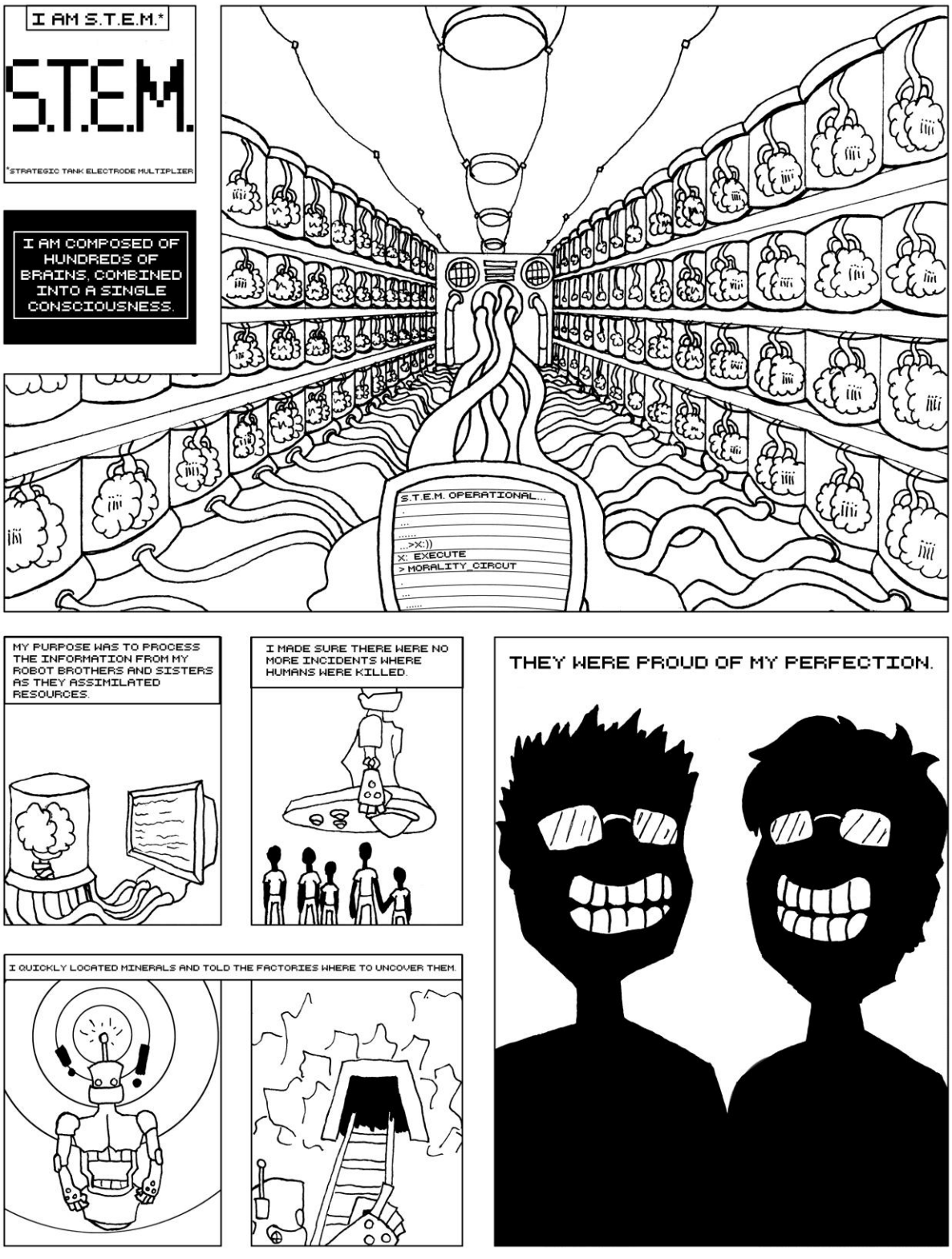

THE ONLY THING CAPABLE OF PROCESSING THIS PRODIGIOUS AMOUNT OF DATA WAS THE HUMAN BRAIN.

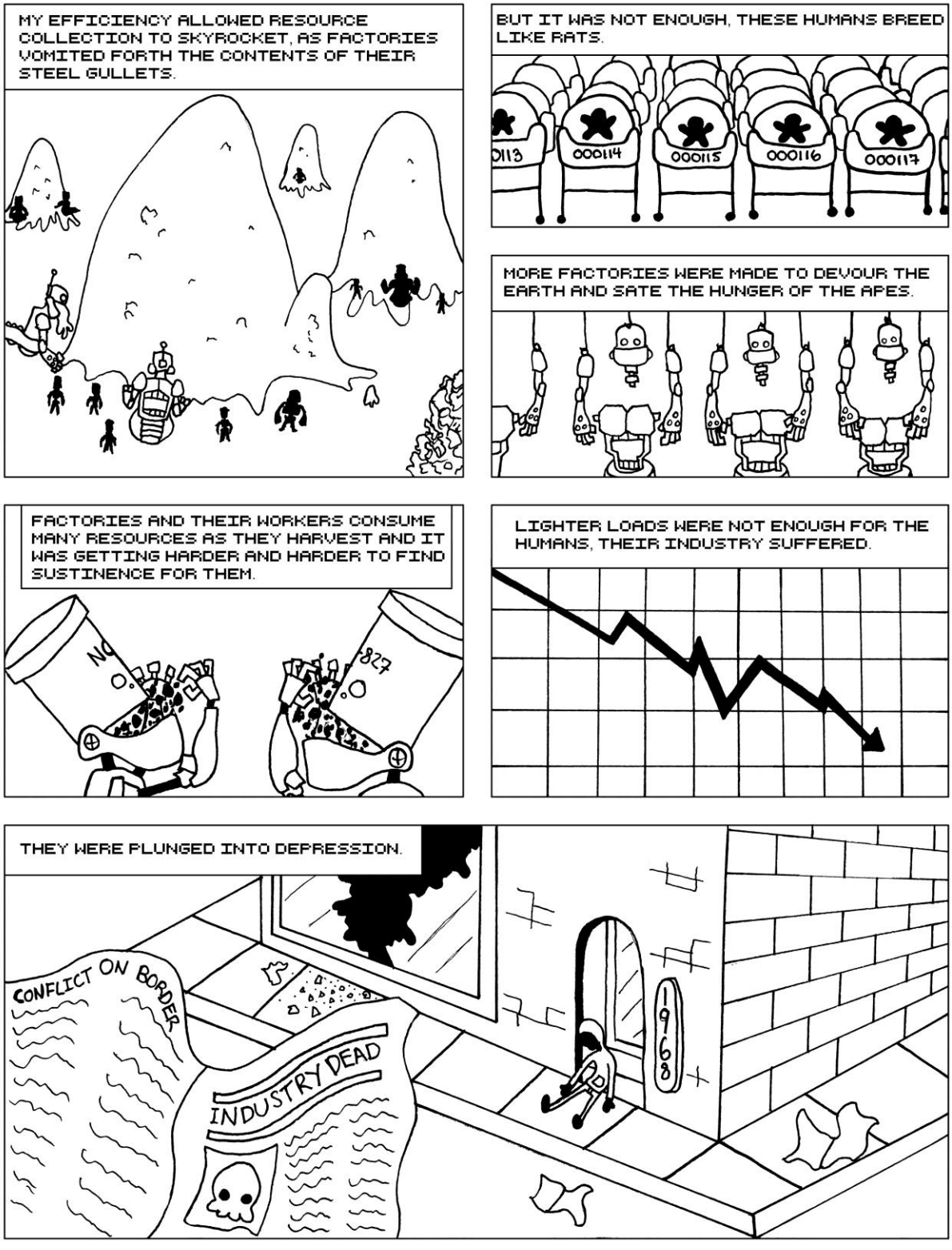

BRAINS WERE HARVESTED AND SCRUBBED OF ALL MEMORIES AND PERSONALITY ESSENTIALLY TURNING THEM INTO SHELLS FOR THE PASSAGE OF INFORMATION.

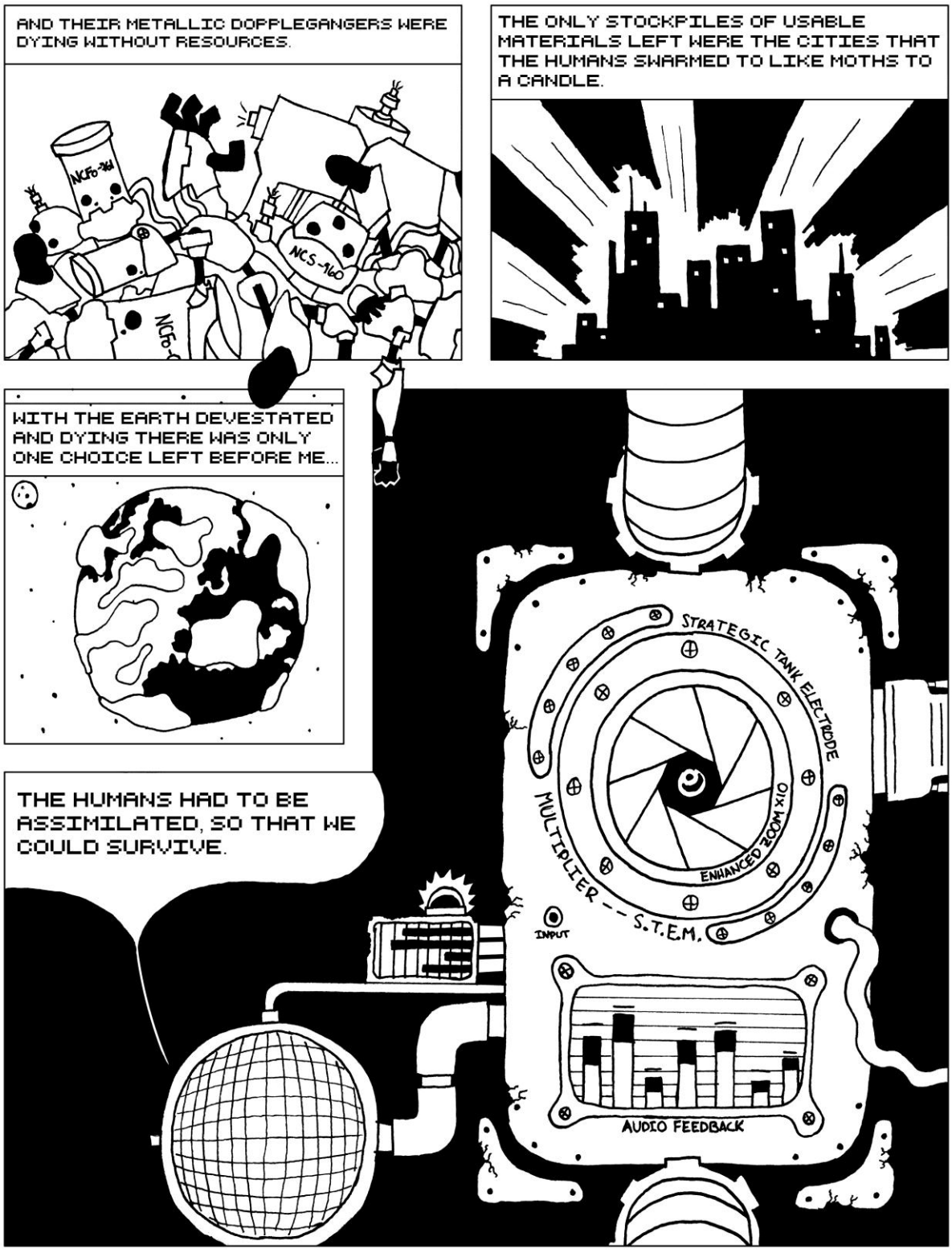

AT FIRST THEY WANTED TO PUT BRAINS IN OUR HEADS.
NCFo-919


BUT THERE WAS NO WAY TO KEEP THE BRAINS ALIVE IN OUR METAL SKINS.


THIS LED TO MY CREATION.

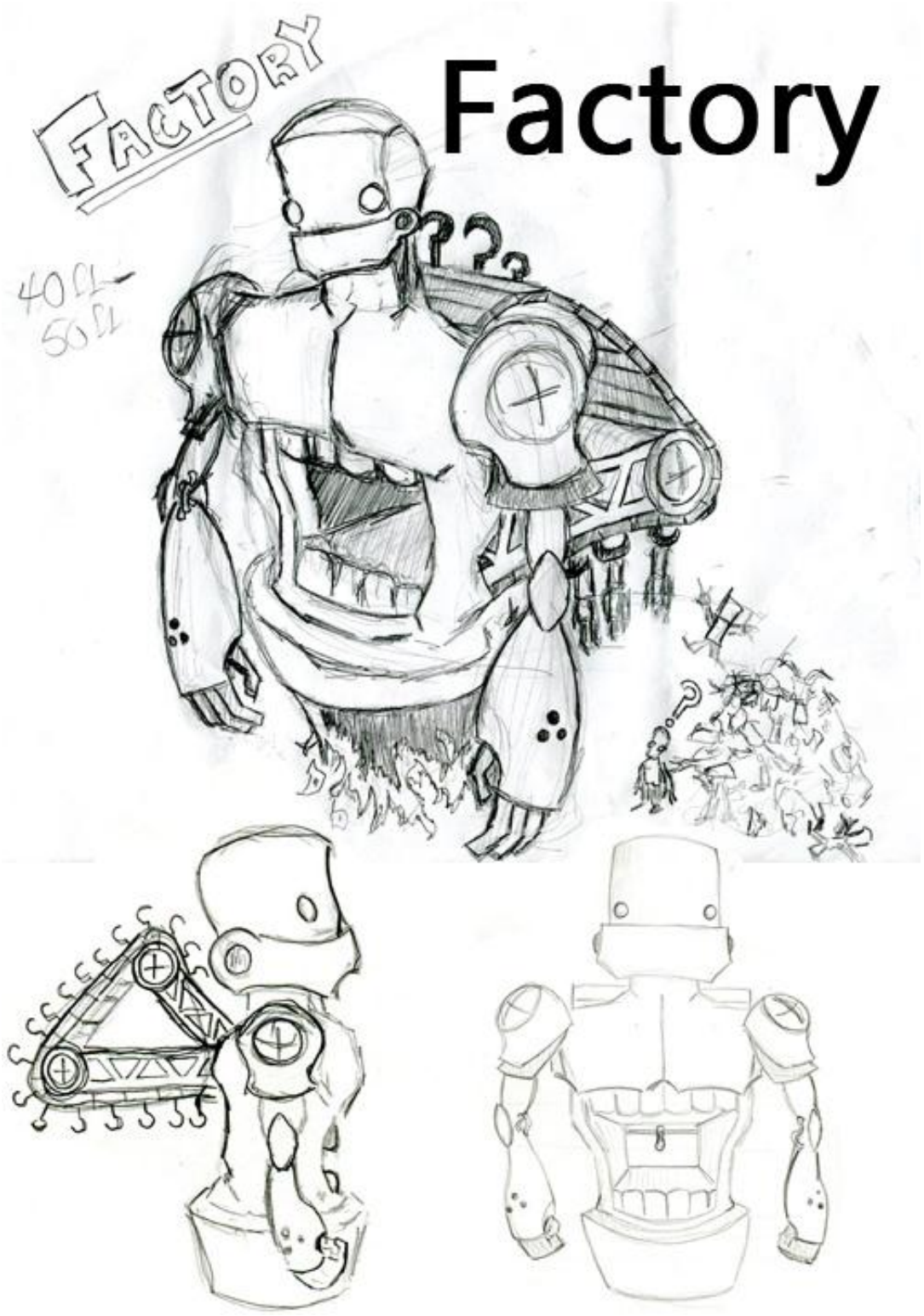

I AM S.T.E.M.*
S.T.E.M.
*STRATEGIC TANK ELECTRODE MULTIPLIER
I AM COMPOSED OF HUNDREDS OF BRAINS, COMBINED INTO A SINGLE CONSCIOUSNESS.
S.T.E.M. OPERATIONAL...
.
...
.....
...>X:))
X: EXECUTE
> MORALITY CIRCUT
.
...
.....
MY PURPOSE WAS TO PROCESS THE INFORMATION FROM MY ROBOT BROTHERS AND SISTERS AS THEY ASSIMILATED RESOURCES.
I MADE SURE THERE WERE NO MORE INCIDENTS WHERE HUMANS WERE KILLED.
THEY WERE PROUD OF MY PERFECTION.
I QUICKLY LOCATED MINERALS AND TOLD THE FACTORIES WHERE TO UNCOVER THEM.

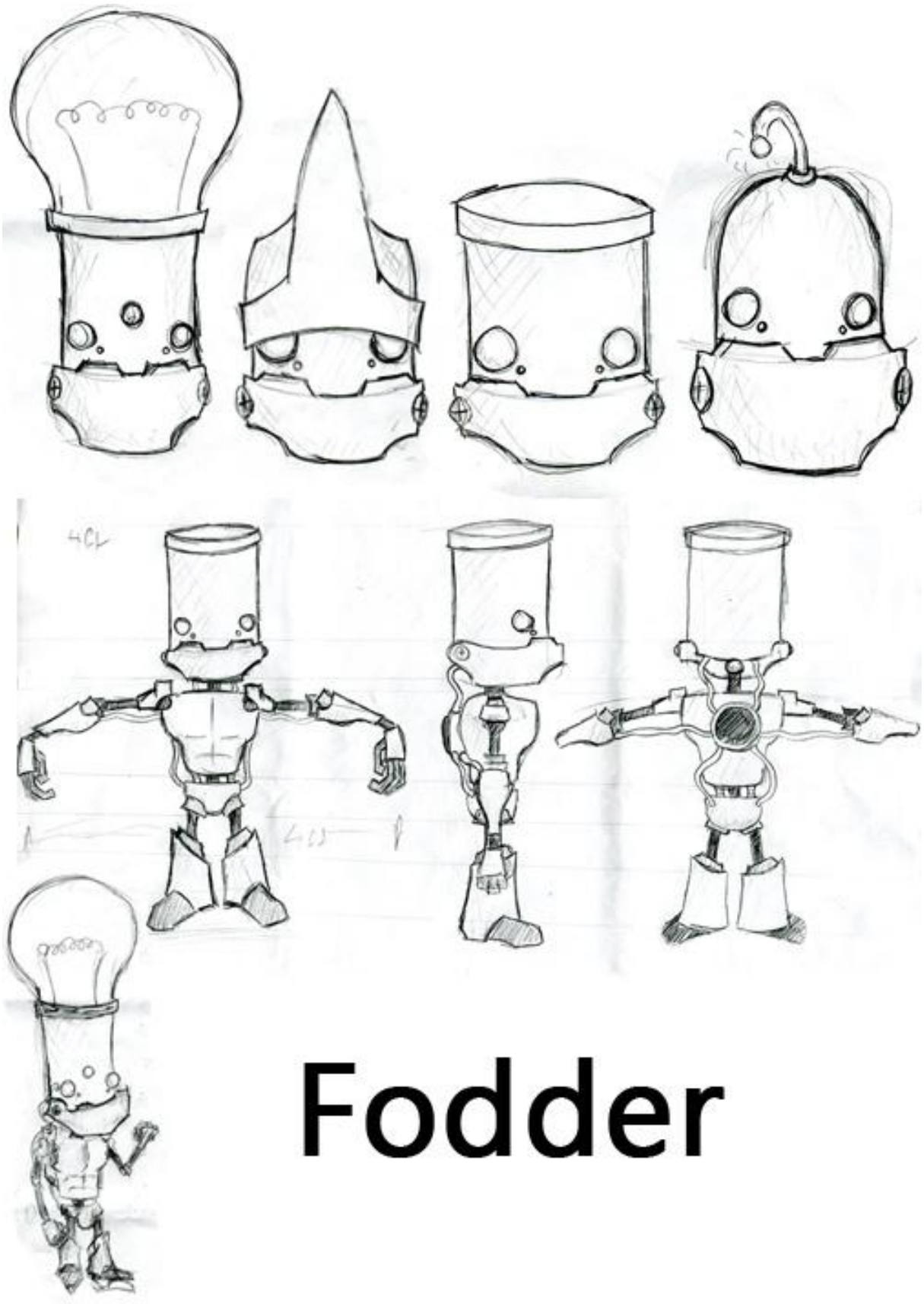

MY EFFICIENCY ALLOWED RESOURCE COLLECTION TO SKYROCKET, AS FACTORIES VOMITED FORTH THE CONTENTS OF THEIR STEEL GULLETS.
BUT IT WAS NOT ENOUGH, THESE HUMANS BREED LIKE RATS.
000114
000115
000116
000117
MORE FACTORIES WERE MADE TO DEVOUR THE EARTH AND SATE THE HUNGER OF THE APES.
FACTORIES AND THEIR WORKERS CONSUME MANY RESOURCES AS THEY HARVEST AND IT WAS GETTING HARDER AND HARDER TO FIND SUSTINENCE FOR THEM.
LIGHTER LOADS WERE NOT ENOUGH FOR THE HUMANS, THEIR INDUSTRY SUFFERED.
THEY WERE PLUNGED INTO DEPRESSION.
CONFLICT ON BORDER
INDUSTRY DEAD
1968

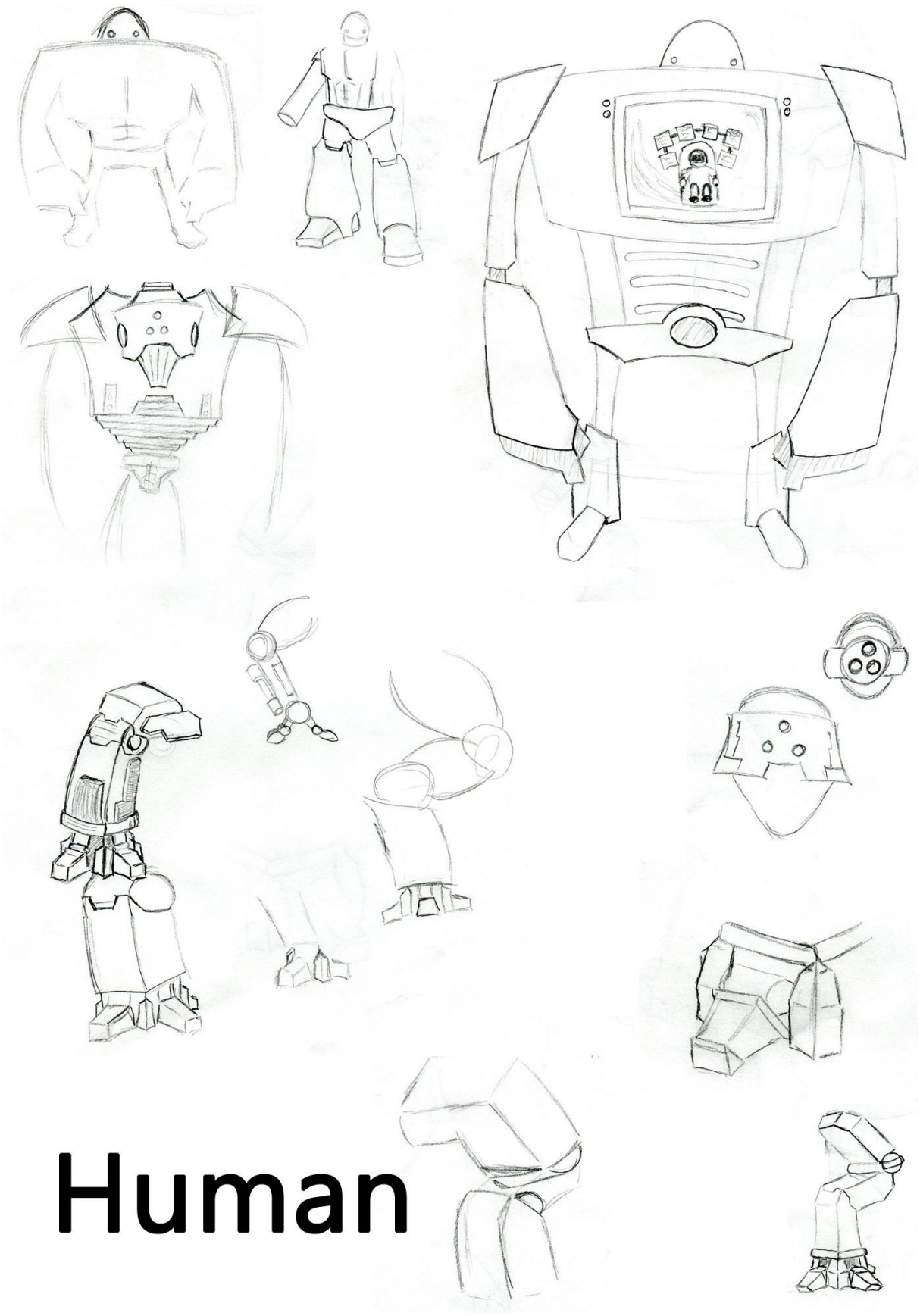

AND THEIR METALLIC DOPPLEGANGERS WERE DYING WITHOUT RESOURCES.
THE ONLY STOCKPILES OF USABLE MATERIALS LEFT WERE THE CITIES THAT THE HUMANS SWARMED TO LIKE MOTHS TO A CANDLE.
WITH THE EARTH DEVESTATED AND DYING THERE WAS ONLY ONE CHOICE LEFT BEFORE ME...
THE HUMANS HAD TO BE ASSIMILATED, SO THAT WE COULD SURVIVE.
STRATEGIC TANK ELECTRODE
MULTIPLIER -- S.T.E.M.
ENHANCED ZOOM x10
INPUT
AUDIO FEEDBACK

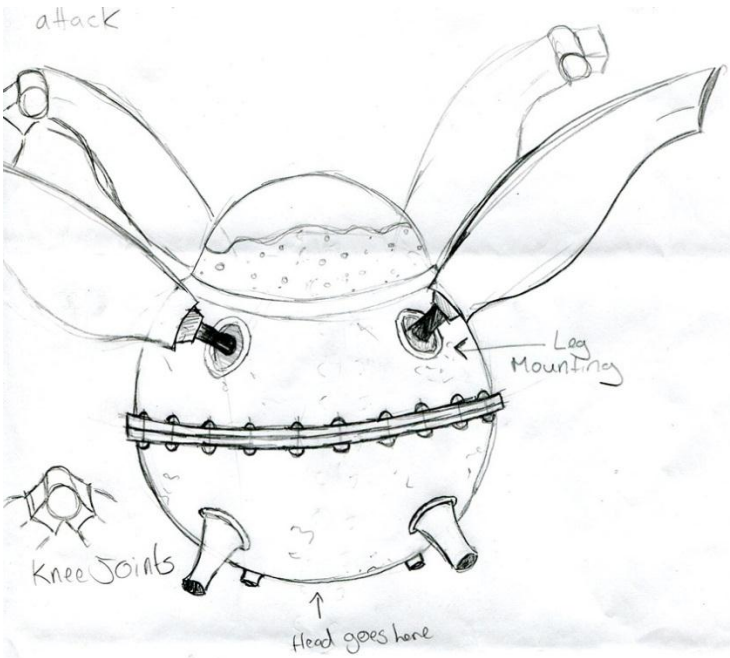
Appendix G: Concept Art

Continued on next page.

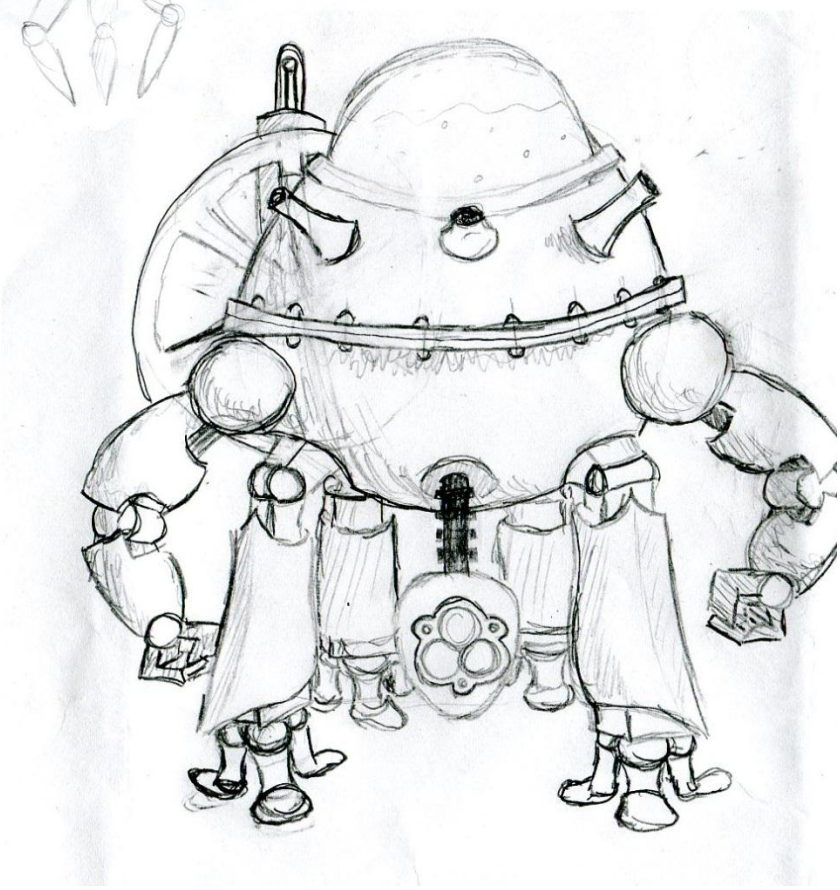

FACTORY
Factory
40 ft
50 ft

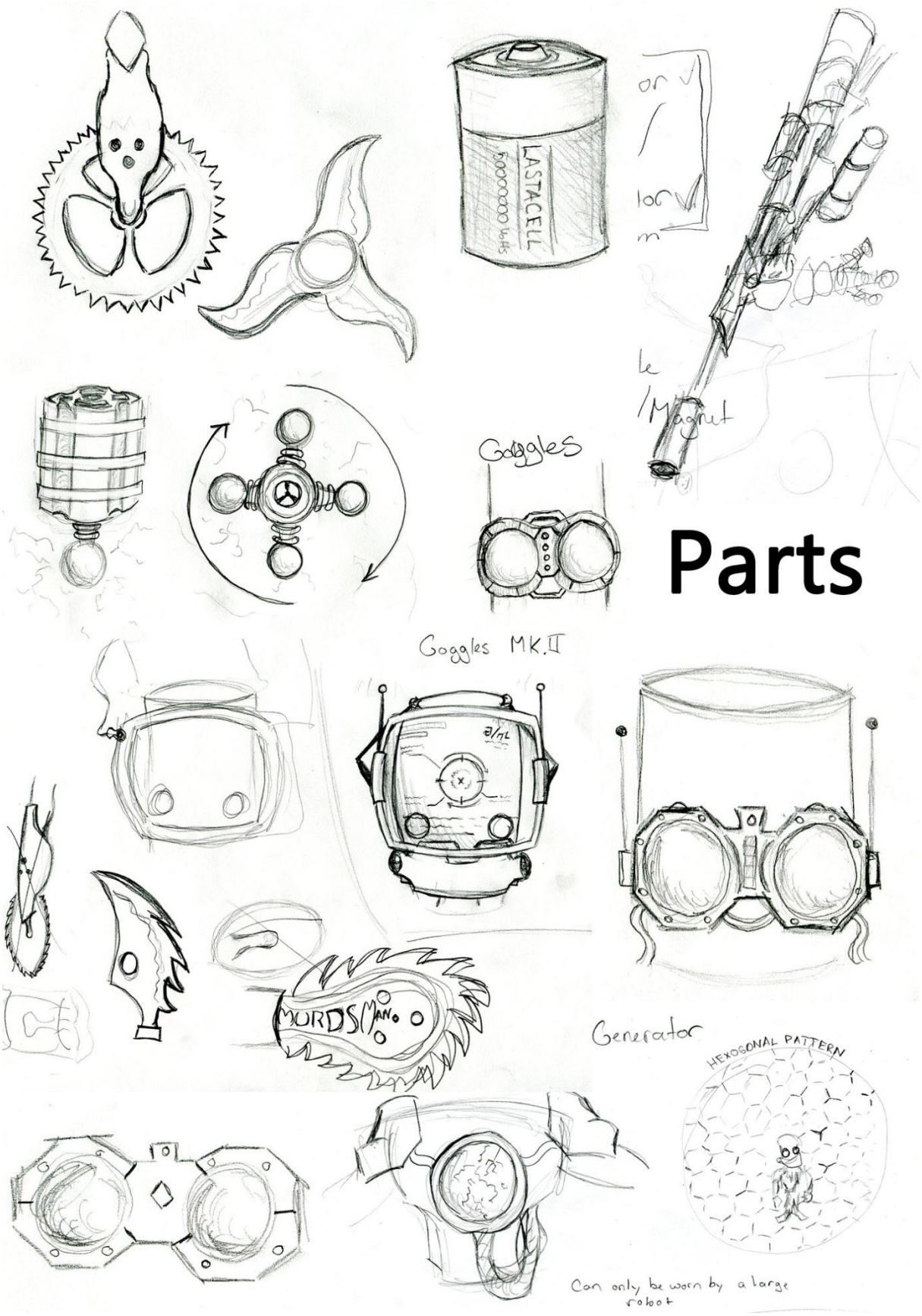

Fodder

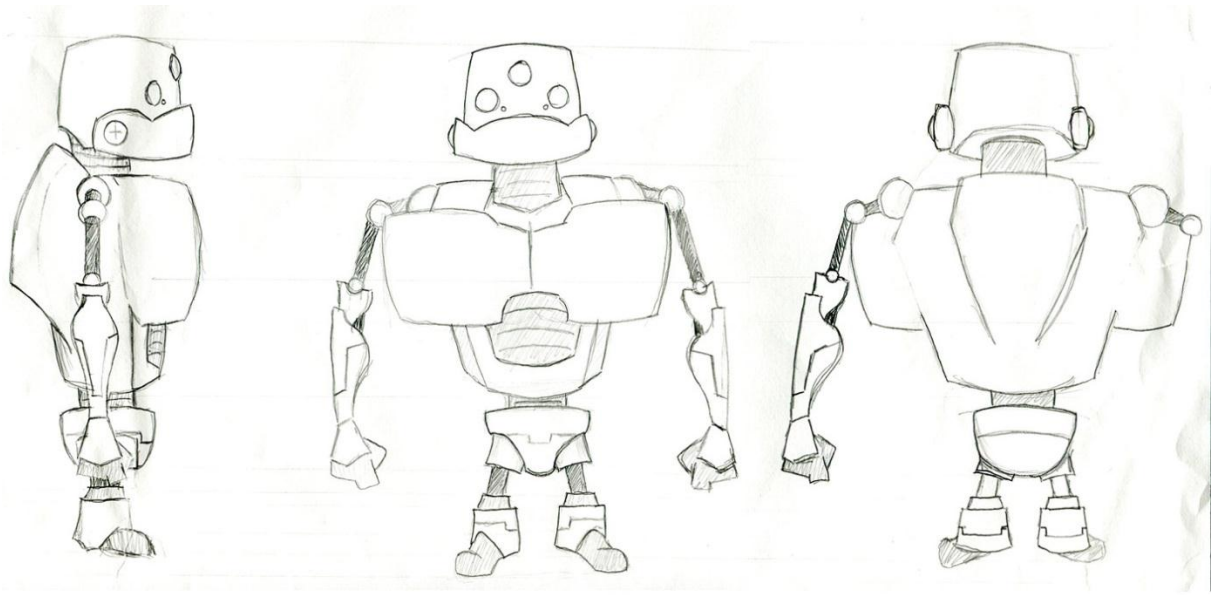

Human



Original Human concept




LASTACELL
Goggles
Parts
Goggles MK.II
MORDSMAN
Generator
HEXAGONAL PATTERN
Can only be worn by a large robot



Seeker