

Razor Statement

Robots with a down home flare

Objective

JunkBots is designed as a social game for a mobile platform or Facebook. It is a two player game where each player programs a junkyard fighting robot. After programming the robot, the two robots fight until one of them is defeated. Players will be able to change the robot programming at predetermined intervals as long as both robots continue to function. The physical appearance of the game is that of a fighting game but it functions like a classical RPG.

Tone

This game should reflect a good ol’ boy country-like mentality. Consider the personality of the Engineer from Team Fortress 2. He is an engineering savant but with a down home flare.



Setting

The game is set in a very rural area. Here there is an auto junkyard run by two brothers, known as the Junkmen. Both are incredibly learned. But instead of pursuing the engineering sciences as a career, they opt to tinker all day every day in the junkyard. They enjoy every part of their laidback lifestyle. One of their favorite pastimes is to build robots out of the junkyard stock and pit them against one another. The winner wins a cache of Moonshine. The younger brother always wants to beat the older and the older brother can never stand losing to the younger. At the end of the day, it’s just a good ol’ fashioned sibling rivalry.

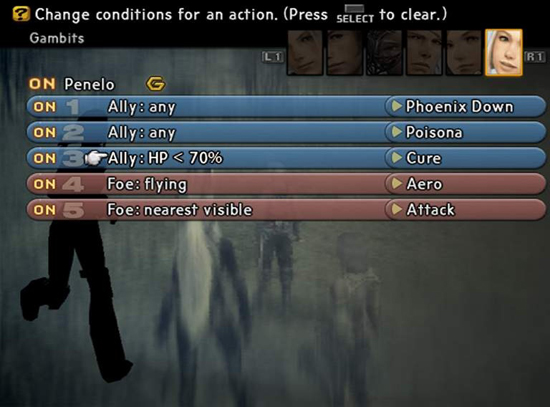
The JunkBots

The junkyard battlebots are assembled out of the automotive surplus from the junkyard, so they can have very unusual sizes and shapes. The carefree style of the brothers is evident in their construction of these JunkBots. Players will be able to construct their own JunkBots under the guise of one of the two brothers. They will be able to choose from various car and truck parts that all have their own individual stats. Large bulky parts will result in more durability for the JunkBot. Lighter parts will result in the JunkBot having better fuel efficiency. Fuel is consumed upon executing special commands. The lighter the bot, the less fuel it consumes.

Programmin’

Players have to program their robots, so that they can fight autonomously. This programming will be done through an intuitive set of code that is managed by a user interface. A clear example of this can be seen in Final Fantasy XII’s gambit system. This system allowed the player to set up a primitive AI framework for their action-rpg characters to operate within.



One of the simplest commands will be “Attack.” This would execute a basic physical attack, like a punch or kick, depending on the JunkBot’s structure, and requires no fuel. “Attack” is the default command that every JunkBot executes if no other commands are able to be performed. Other commands will consume fuel. A “Power Attack” is a simple example of a fuel consumption move. The JunkBot will detach its left arm with its right arm and broadside the enemy with it for extra damage. A “Power Kick” is the leg equivalent where the JunkBot does a flying side kick. Such commands would be written like; *Power Attack: Opponent: Durability < 30%* and *Power Kick: Opponent: Durability < 10%*. These would translate to; Power Attack when opponent’s durability is less than 30% of max and Power Kick when opponent’s durability is less than 10% of max respectively.

There are going to be numerous types of attacks, and special commands. Some may simply damage the opponent, some may cause delayed reactions, and some may buff the player or debuff the enemy. All of these will have to be balanced, but the results will create a lot of interesting emergent strategies.

The Junk

All JunkBots must have a Head, a Torso, a waist, two Legs, and two Arms. Each of these part categories can be fit with several different kinds of Junk equipment. The durability, fuel cost, and abilities of the JunkBot are determined by the bot’s equipment. JunkBot’s also can be equipped with peripheral and back mounted equipment.

The Head is a purely cosmetic piece as far as the gameplay is concerned. It has no bearing on the JunkBot’s durability, fuel cost or abilities. Heads are customized with some fun items like a corn pipe, a hillbilly hat, a beard made of tangled electrical wires or a combination of them. The basic model for the head resembles the face of the Junkman that the JunkBot represents. The player can customize these heads outside of the game.

The Torso houses the fuel tank, battery, engine and all the other necessary motor parts that the JunkBot needs to have to be able to function. Various forms of defense equipment are attached to the torso to give the JunkBot durability. A reinforced frame could be fitted, which would dramatically increase durability at the cost of the JunkBot being heavier and less fuel efficient. A truck hood, on the other hand, would provide much less durability, but the lower weight would allow for lower fuel costs.

The Waist, like the head is a purely cosmetic piece. It does not have any bearing on the durability, fuel costs or abilities of the JunkBot. However, it is an attachment point for the waist peripheral equipment. Peripherals will be further explained below.

The Legs can support a few different types of equipment. Both legs could be fit with shocks to reduce recoil and damage from attacks. Alternatively, they could be fit with hydraulics to boost the leg attack power at the cost of higher fuel requirements. Legs can also be devoid of equipment for speed and low fuel cost.

The Arms can support either offensive or defensive equipment. Unlike the legs, each arm can wield its own piece of equipment. An example of an offensive weapon is a shotgun. A defensive weapon like a car door can be wielded as a shield to reduce incoming damage. A JunkBot can also be unarmed. This will make the bot weigh less and cost 0 units of fuel to attack. It will not however do as much damage as a proper weapon.

Peripheral equipment can be mounted either on the shoulders or the waist of the JunkBot depending on the type of equipment. Most of this equipment is utility based. Utility based equipment provides some type of boost to the JunkBot or has a non-damaging side-effect. An external fuel tank could be equipped to the waist of the JunkBot to increase the total amount of fuel available for use. An example of a shoulder based peripheral would be a set of high beam lights. These can be used in combination with another move to obscure the animation and effects of that move from other players.

Back mounted equipment fits on the back of the JunkBot connected to the Torso. These pieces of equipment either provide extra attack power for the JunkBot or boost the JunkBot’s abilities. A gun rack is an example of an offensive piece of back mounted equipment. The JunkBot would be able to reach back, grab a gun or guns, shoot the other JunkBot and put them back. An additional engine could be mounted on the back instead to increase the speed of all of the JunkBot’s moves.

Constructing the JunkBots

Both Junkmen take turns picking parts until both JunkBot’s are fully built. All of the available parts are in a junk pile in the center of the field. The first player chooses whichever piece of equipment that they want from the pile. The second player then picks two pieces of equipment from the pile. They keep alternating picks, two at a time, until player 2 picks the final piece. Both players should then each have a torso piece, two arm pieces, a leg piece, a back mount, a shoulder peripheral, and a waist peripheral.



The purpose of this turn based selection process is to give each player an opportunity to pick equipment to counter the opponent if they choose to do so. Alternatively, they could just pick the parts that they want. In any case, strategy begins here. Players will be given 30 seconds to make their picks. If time expires, parts will be automatically chosen for them. As the player highlights each part, their stats and abilities will be displayed. Once all the parts have been chosen the battle will begin. The unselected parts will remain the junk pile for use in later rounds.

Gentlemen’s Rules

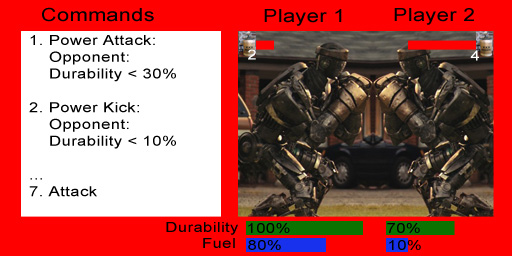
Once both players have constructed their JunkBots they will have 30 seconds to set up two custom commands. Once both players have submitted their commands or 30 seconds have passed, the robots begin to fight. At set intervals during the fight, each player will get to either add a single new command or alter a single existing command. A maximum of 6 commands are allowed per robot in addition to the default attack command. Neither player can see the other’s commands. The goal is to try to figure out how the other player has programmed their robot and then combat its weaknesses. If a JunkBot’s durability reaches 0, it falls apart and the other player is declared the winner of that round. If both JunkBots fall apart at the same time, then the game is declared a draw. If both JunkBots are still functioning after a set amount of time, the one with the most durability is declared the winner.

Moonshine

What is the Deep South without Moonshine? Moonshine serves as a currency in the game. As the JunkBot executes commands, the Moonshine gauge fills up. The stronger the move, the more Moonshine is generated. Moonshine is spent as a cost to fix the JunkBots, via programmed “Repair” commands. As far as the Junkmen are concerned, the Moonshine is contributed to a pot and the winner of the competition gets the Moonshine. Moonshine can also be spent at the end of a round in order to exchange or replace a piece of equipment. It costs 1 jug of Moonshine per part. The only exception is that the loser of a round gets to select 1 part to exchange or replace for free.

Sample User Interface

Below is a sample of the user interface as the player would see it during the game. The code entered by the player is displayed on the left. On the right, the player’s JunkBot battles the other JunkBot. The durability and fuel of each JunkBot is displayed to the player. In the upper corners of the fighting window, there is a Moonshine gauge for each JunkBot.



Victory!

JunkBots is intended to be a quick and fun strategy based game. The programming mechanics of the JunkBots will provide an interesting and emergent pvp experience. The player will always be attempting to come up with a new and better tactic to use against his opponent. Players will also be consistently trying to predict and learn their opponent’s moves and styles of play.

C:\dev\Students\CH9\pwilkes\PIE 1\Game_Design_Documents\GDD8_Mechanics\mechanical_description.jpg

Artistic Style: Overall

The artistic style of the game should be semi-cartoony. More particularly it should be cell-shaded, like the artistic style of the Borderlands series. With this style, objects and characters are still realistic, but have a black outline like cartoons do. In Borderlands, this artistic style matches the humorous backdrop of the game and its characters. *JunkBots* has a similar backdrop, as the game is based in reality, but with humorous and semi-realistic robots made out of junk.





Artistic Style: The Junkmen

The Junkmen are not the main focus of the game. Their robots are. However, they do need their own styles and personalities. They are brothers, but not twins. One is a few years older than the other. They are stereotypical country boys. Items and behaviors characteristic of the Deep South should influence their designs. Some examples are; overalls, straw hats, beards, corn pipes, chewing tobacco, guns, banjos, and of course moonshine. Consider Scooter, the mechanic from Borderlands, and his personality. These characters should appear laid back and greasy from their constant mechanic work.





Artistic Style: The JunkBot

Junk that can be used to build JunkBots should be more clearly outlined than the junk in the background in order to distinguish it. The face of the JunkBot should be a simplified version of the face of the Junkman that built it. Only a mechanical version of the facial features should be designed as the player will be able to dress up this head with hats, beards etc. before the game starts. This will be the only distinguishing feature of the JunkBots other than the direction they are facing. Each JunkBot will have identical arms(shoulder to hand), legs(waist to foot) and identical waists and torsos. Additional equipment will simply be painted over the basic framework, or will replace it entirely. Either way, all surplus junk equipment should be designed according to the framework of the JunkBots. Peripheral equipment will be attached to the shoulders and/or the waist of the JunkBots, but these pieces of equipment will not be part of the initial framework. The same applies to any back mounted equipment. Each piece of equipment is detailed below.

Artistic Style: The Junkyard

The game takes place in a junkyard, and this junkyard serves as a static backdrop for the entirety of the game. Junk should be piled up in huge mounds. Most of the junk is comprised of wrecked vehicles and general auto parts. Tires, doors, truck beds, cars up on blocks, trailers, and partially assembled JunkBots should cover the landscape. The Junkyard should also be replete with JunkBot specific parts like arms, legs and weapons to show that constant tinkering is going on.



Artistic Style: Animations and Particle Effects

*JunkBots* is essentially a fighting game, so there will be a significant amount of attack animations and particle effects. A basic arm attack would be a straight punch to the torso of the other JunkBot. A basic kick attack would be a straight kick to the waist area of the other JunkBot. An arm based power attack would have the JunkBot pull off one arm with the other and use the detached arm as a club to hit the other JunkBot, finally re-attaching the arm. A leg based power kick would have the JunkBot jump at the opponent a deliver a side-kick to the torso. Equipment based moves, like guns would have the JunkBot raise the gun and fire it at the Torso of the other JunkBot. Particle effects would also be needed for the shot, and the other JunkBot would need some type of reaction and recoil from that attack. These pieces of equipment will be further defined below.

User Interface: Start Screen

When the game begins, the player should be shown a main menu with a splash screen of the Junkmen building their JunkBots as well as the title of the game. There should be a start button to start the game, an Instructions button that leads to an instruction page, and a JunkBot customization button. The customization button is the only one that would really need to be explained. The player can dress up the robot head with hats, beards, overalls, and other country themed items to give their JunkBots a little more character. This customized head should then be implemented in the game as the JunkBot’s head.

User Interface: Part Selection

The first screen to be displayed is the part selection screen, where players will select the parts to construct their JunkBots. The interface can be very similar to the image below. The Junkyard will appear in the background and the usable parts will be in a pile in the center of the screen. The available parts should be semi-randomized. There should be enough of each type of part to fully equip both JunkBots. There should also be extra parts left over. Duplicate parts can also be put in the pile. There could be 4 or 5 different arm mounted shotguns as opposed to just 1 for example. Part of the fun of the game is strategizing based on the parts available.

Each player’s JunkBot will be shown as an empty wireframe with the exception of the head and waist, which will already be filled in. The player should be able to roll over or click on a part in order to read its stats and abilities. An arm mounted shotgun for example would show the name of the part, how much fuel is consumed to use it, how much weight it adds to the JunkBot and how much basic damage it does. The part could then be dragged over to the JunkBot wireframe, or simply a submit button could be used to permanently select that part. Once a part is selected, that part appears over the wireframe of the JunkBot, showing both players, what part was picked and also what parts the player still needs. The part that was picked is also removed from the junk pile in the center of the screen.

Adding parts to the JunkBot will change its stats, so the stats of the JunkBot should be clearly displayed on this screen as well. Parts can be selected in any order. In other words, you can select arm parts without having selected a torso first. The player does not have to fully equip their JunkBot. If they want to fight without a piece of Torso armor, for example, they should be able to do so. “Unarmed” should be a selectable option on screen for each part in this case. This will allow for more strategic combinations. Once both of the JunkBots have been constructed the game will proceed to the fighting screen.



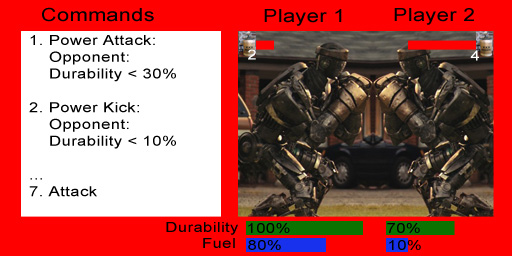
User Interface: Part Selection: Rules

Before any parts are picked, each player must choose a straw out of a straw hat. The player who gets the short straw gets to determine whether or not they would like to pick first. Once this is determined the player has 30 seconds to choose a single part from the pile. If they do not select a part in 30 seconds, then a part will be chosen for them. The other player then gets 30 seconds to pick any two parts from the pile. If the selections are not submitted by the end of 30 seconds then, parts will be chosen for them. When a part is selected for the player, it is chosen at random according to the spaces on their JunkBot that have yet to be filled.

Players will continue to alternate picks in this fashion until the second player only has one more part to pick. Both JunkBots will then consist of seven parts a piece, namely; a torso, a back mount, two arm pieces, a waist peripheral, a shoulder peripheral, and a leg mount. All parts are optional, as the player can specifically choose to not equip a special part. In that case, a default frame for that part of the JunkBot will be displayed i.e. unarmed arm.

User Interface: Battle Screen

Once the JunkBots have been built, they are shown squaring off against one another. The Durability, Fuel value, and Moonshine Gauges are displayed around the JunkBots in the fighting window. The other portion of the window is a special user interface where the player can set the programming for their JunkBot. Options for creating these commands should be simple and intuitive. Drop down menus would suffice for this.



To enter a command the player would have to fill in a few fields and then hit a submit button to add it to the command list. The first field is the name of the ability. This drop down list would be populated with all of the abilities that the JunkBot can use based on the equipment that had been selected. So, “Shotgun” would only show up in the list if the JunkBot was equipped with an Arm Mounted Shotgun. The second field is the Target of the command. If an offensive command, such as the shotgun was entered into the first field, then only “Opponent” would be selectable. Some abilities affect the player’s own JunkBot, so “Self” would be the alternative selection for non-offensive pieces of equipment. The third field is a special condition to execute the move. This condition refers to a numerical value in regards to the Target field of the command. In the following example, “Power Attack: Opponent: Durability < 30%” would have the JunkBot execute a Power Attack when the Opponent’s Durability is less than 30% of its maximum. Other conditions can refer to the fuel or moonshine of either JunkBot depending on the Target of the command.

User Interface: Battle Screen: Combat Rules

The winner of the round is the last JunkBot standing. Players begin by setting up two initial commands in addition to the default attack command. They have 30 seconds to do so. The robots will then begin to fight based on these commands. The speed values of each JunkBot will ultimately determine how fast the JunkBot executes its commands. Speed values are inversely proportional to the weight of the JunkBot and some moves may take longer to execute than others. Valid commands are executed in the order that they appear in the command list. The speed ultimately dictates how often the command list is re-evaluated after the previous evaluation is complete. A higher speed will result in more passes through the command list. Every 30 seconds both players will have the opportunity to either add a new command to their list or alter a single existing command. This is not required, but it is an opportunity to vary strategy. The sooner that this command is submitted, the sooner it has a chance to be used. A maximum of six commands can be added to the list in addition to the default attack command. This default attack command does the same 15 damage as a basic unarmed attack.

Neither player will be able to see the command list of the other player. If they pay attention they can make educated guesses about their opponent’s strategy based on the JunkBot animations and stats at a given time. The loser of the round is the first JunkBot to reach 0% Durability. If both JunkBots kill each other, the round is considered a draw and neither player gets a point. The JunkBot with the most health at the end of 4 minutes is declared the winner of the round.

The loser of a round is then briefly shown the junk pile from before. There may also be new parts in the pile that were not there before. The loser is able to exchange or replace a single part of their JunkBot for use in the next round. This does not cost anything of the player.

The Durability and Fuel gauges of the JunkBots are completely restored between rounds. The first player to win two rounds wins the game. The victor is then shown a victory screen of their Junkman enjoying his cache of Moonshine. The loser is then shown a defeat screen of their Junkman feverishly assembling a new JunkBot to re-challenge his brother. Replay buttons are then made available to each player. Once both are hit, then both players can assemble new JunkBots and begin again.

User Interface: Battle Screen: Moonshine

Moonshine functions as a form of currency for each player. Whenever a command is executed, moonshine is generated. This should be displayed as a progress meter or gauge like the Durability and Fuel Gauges. Once the gauge fills up all the way, one jug of moonshine is attributed to the player, and the progress bar is reset to 0. The gauge would measure progress from 0 to 100. A basic attack would add 5 points to the gauge whereas a power attack would add 10 to the gauge. In Battle, moonshine can be included as a command in order to repair the JunkBot. The repair command would consume 1 jug of moonshine and repair 50% durability to the JunkBot. Such a command would appear as; Repair:Self: Durability < 20%. This would only execute if there was at least one jug of moonshine available and only if the JunkBot’s durability is less than 20% of its maximum. A maximum of 5 jugs of moonshine can be held at a time. The Moonshine gauge should also carry over from round to round.

Between rounds, the loser is able to exchange or replace a single part of their JunkBot for free for the next round. Both players can also spend their moonshine here. Both players can exchange or replace a part at the cost of 1 jug of moonshine. Say both players have 2 jugs of moonshine at the end of round 1 and player 2 loses round 1. Player 2 gets a part for free and then can spend his 2 moonshine to get 2 new parts. Player 1 then opts to spend 1 jug of moonshine to get a new part and saves the other for a potential repair command in the next round.

Durability of the JunkBot

The healt and weight values of the JunkBot are directly proportional to each other, so durability will account for both of these variables. The default durability of each JunkBot is listed below. Note that an unequipped JunkBot has no shoulder peripheral, waist peripheral, or back mounted equipment. These basic durability values are added together with the durability of the parts. This will determine the overall durability of the JunkBot. Each JunkBot contains 100 units of fuel as well.

|  |  |
| --- | --- |
|  | Durability |
| Parts | Base |
| Head | 5 |
| Torso | 35 |
| Arm 1 | 15 |
| Arm 2 | 15 |
| Shoulder Peripheral | 0 |
| Waist | 5 |
| Legs | 25 |
| Waist Peripheral | 0 |
| Back | 0 |
|  | 100 |

Speed of the JunkBot

The speed of the JunkBot is inversely proportional to the durability of the JunkBot. This speed value represents the amount of time that passes between each evaluation of the player’s command list. Once all statements in the command list have been processed, the game waits for a pre-determined amount of seconds until it re-evaluates the command list again, based on the current variables. Given the default durability of a JunkBot without any equipment, its speed will be approximately 2 seconds. This is set as the default benchmark speed.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Durability | Speed |  |
| Parts | Base |  |  |
| Head | 5 | 0.833333 |  |
| Torso | 35 | 5.833333 |  |
| Arm 1 | 15 | 2.5 |  |
| Arm 2 | 15 | 2.5 |  |
| Shoulder Peripheral | 0 | 0 |  |
| Waist | 5 | 0.833333 |  |
| Legs | 25 | 4.166667 |  |
| Waist Peripheral | 0 | 0 |  |
| Back | 0 | 0 |  |
|  | 100 | 16.66667 | 2 seconds |
|  |  | 8.333333 | 1 second |

The speed value of 16.66667 is determined by dividing the total durability value of the JunkBot by 6. Dividing the speed value by 2 gives the approximate speed value of the JunkBot based on a 1 second timeframe. Below is another example of a JunkBot that is fully fit with all of the heaviest parts currently in the game.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Heaviest JunkBot |  |  |  |  |  |
| Parts | Durability | Base | Total | Speed |  |
| Head | 0 | 5 | 5 | 0.833333 |  |
| Reinforced Frame | 175 | 35 | 210 | 35 |  |
| Tailgate | 75 | 15 | 90 | 15 |  |
| Shotgun | 45 | 15 | 60 | 10 |  |
| High Beams | 10 | 0 | 10 | 1.666667 |  |
| Waist | 0 | 5 | 5 | 0.833333 |  |
| Hydraulics | 75 | 25 | 100 | 16.66667 |  |
| Gas Can | 10 | 0 | 10 | 1.666667 |  |
| Truck Box | 75 | 0 | 75 | 12.5 |  |
|  | 465 | 100 | 565 | 94.16667 |  |
|  |  |  |  |  |  |
|  |  | Re-Evaluation Time: |  | 11.3 | Seconds |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  | Re-Evaluation Time: | | Durability / 6 = Speed | | |
|  |  |  | Speed / 8.33333 = re-eval in seconds | | |

The Durability column displays the durability values of each individual part. The Base column is the base durability of a JunkBot. These are added together to form the Total column. Each of the values in the Total column is divided by 6 and added together to get the approximate speed value of the JunkBot with the selected parts. This final value of 94.16667 is then divided by 8.33333 to get the Re-Evaluation time in seconds of the JunkBot. In the above example, this heavy JunkBot will have its command list re-evaluated every 11.3 seconds compared to the 2 seconds of the unequipped JunkBot. The slow strong power of the heavy JunkBot will balance out the fast but weak power of the unequipped JunkBot.

Junk Parts: Defensive

Every part that can be equipped to the JunkBot has a durability value associated with it. Defensive parts are equipped solely to increase the amount of durability the JunkBot has. Defensive parts come in two varieties. Torso parts give the most durability and are equipped to the Torso section of the JunkBot. There are also shields that can be equipped to the JunkBot’s arms in order to give more durability. That said, most of the offensive weaponry are equipped to the JunkBot’s arms, so there is a tradeoff there. The Table below shows the amount of durability that each part adds to the total durability of the JunkBot. “Unfit” and “Unarmed” are the default values of the JunkBot without special equipment attached to them. It is also worthy to note that only 1 Torso piece may be equipped, but 2 Arm pieces can be equipped. For example, if 2 arm shields are available, the player may opt to pick up both a Tailgate and a Car Door. Each of these parts should be self-explanatory.

|  |  |  |
| --- | --- | --- |
| Description | Parts | Durability |
| Torso | Truck Hood | 140 |
| Torso | Grille | 105 |
| Torso | Reinforced Frame | 175 |
| Torso | Reinforced Bumpers | 70 |
| Torso | Unfit | 35 |
| Arm – Shield | Car Door | 60 |
| Arm – Shield | Wheel With Tire | 45 |
| Arm – Shield | Tailgate | 75 |
| Arm | Unarmed | 15 |
| Arm | Unarmed | 15 |

Junk Parts: Offensive

The Offensive junk parts are a little more complicated. They each have durability values, damage values, fuel costs, and moonshine growth values. These are represented in the table below. Durability is the amount of durability that is added to the JunkBot when the associated parts are equipped. As before “Unfit” and “Unarmed” represent the default values of the JunkBot without any piece of equipment. The Fuel column shows the amount of fuel that the attack consumes upon use. The stronger the attack, the more fuel required to use it. Notice that the “Unfit” and “Unarmed” attacks cost 0 units of fuel to use. The Damage column shows the amount of damage that is done by attacking with that weapon. Finally the Moonshine column shows the amount of Moonshine earned upon executing each of these attacks. Three attachment areas of the JunkBot are represented here, namely Leg, Arm, and Back.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Description | Parts | Durability | Fuel | Damage | Moonshine |
| Leg | Hydraulics | 75 | 0 | 45 | 5 |
| Leg | Shocks | 50 | 0 | 30 | 5 |
| Leg | Unfit | 25 | 0 | 15 | 5 |
| Arm | Shotgun | 45 | 10 | 45 | 10 |
| Arm | Jumper Cable Prong | 30 | 5 | 30 | 5 |
| Arm | Axel Lance | 45 | 10 | 45 | 10 |
| Arm | Unarmed | 15 | 0 | 15 | 5 |
| Arm | Unarmed | 15 | 0 | 15 | 5 |
| Arm | Hunting Rifle | 45 | 10 | 45 | 10 |
| Back | Gun Rack | 60 | 20 | 60 | 20 |
| Back | Truck Box | 75 | 25 | 75 | 25 |

The Legs of the JunkBot are considered to work as one unit as opposed to the arms that can be equipped independently. The Legs can be fit with weak Shocks or heavy Hydraulics. The damage done by an unarmed kick attack is directly proportional to the damage done by an unarmed punch attack. Equipping Shocks or Hydraulics to the legs of the JunkBot will result in the JunkBot having increased durability and damage output but with less speed. The Moonshine gained from a kick attack is 5 out of 100. Legs can also be programmed to have a power kick which would earn 10 Moonshine instead of 5 and deal an additional 15 points of damage. However, this power kick would cost 5 units of fuel to execute. Unarmed Arms can also be programmed with a power attack as well, and it works just like the legs.

The Arms of the JunkBot can each be fit with its own piece of equipment. This accounts for the 2 “Unarmed” listings. The damage dealt by each of these weapons is directly proportional to the durability that they offer. Likewise the Fuel cost and Moonshine gains are directly proportional to each other. Keep in mind that there are other pieces of arm equipment, like the shields, that do not have offensive capabilities. The Shotgun and the Hunting Rifle are guns that the JunkBot holds and fires. The Jumper Cable Prong is essentially an arm with electrical prongs that deal additional damage upon contact with the opponent. That said, it does more damage than an unarmed attack, but less than the guns. The Axel Lance rivals the guns in power. It is a large car axel that is used to jab the enemy.

The Back Mounted equipment represented here is the Gun Rack and Truck Box. The Truck Box is heavier and is essentially a large tool box that could be found in the back of any truck. The Gun Rack is a rack of guns of different kinds. The JunkBot will grab a gun from the Gun Rack, shoot the opponent, and put it back. For the Truck Box, the JunkBot will grab a handful of tools and throw them at the opponent. Compared to the other offensive pieces of equipment, these are the most powerful, require the most fuel, and offer the greatest durability and Moonshine growth. That said, there are other more utilitarian options for the player to pick from.

Junk Parts: Utility

There are several different kinds of utility parts. These parts can be equipped to the Arms, to the waist, to the shoulders, and to the back of the JunkBot. Often these provide passive boosts to the JunkBot. Others are designed to work with or against other parts in order to allow opponents to counter each other’s strategies during part selection. Each of these will be explained below based on their equipment sections.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Description | Parts | Durability | Fuel | Damage | Moonshine |
| Arm | Banjo | 30 | 5 | 0 | 5 |
| Arm | Tire Iron | 30 | 5 | 0 | 5 |
| Arm | Grapple Arm | 20 | 25 | 0 | 10 |

The three pieces of Arm equipment in the table above are single use pieces. After they are used, the JunkBot’s arm reverts to the default “unarmed” state. The Banjo and Tire Iron both serve the same purpose. They are used to knock off an opponent’s shoulder or waist peripheral. This will remove it from the battle. The tradeoff here is that both players will lose pieces of equipment. Peripherals are mostly passive, and will be explained more below, so this is a purely strategic move of the player. Five units of fuel are consumed in order to use them and they provide small amounts of durability. Much like an unarmed attack, their use generates 5 Moonshine. Unlike the unarmed attack, neither of these deals any damage to the other opponent. That said, removing a peripheral from the opponent will reduce the opponent’s overall durability. Likewise, once the Banjo or Tire Iron is used, the player’s own durability will decrease by 15 as the arm reverts back to its unarmed state.

The Grapple Arm works in a different way but accomplishes a similar goal. It costs a significant amount of fuel to use, but allows the player to steal a piece of arm equipment from the opponent, ultimately replacing the Grapple Arm with that piece. This could be used to obtain an arm shield, like a Tailgate, from the opponent to increase the player’s durability and reduce the durability of the opponent. Alternatively, the player could use the Grapple Arm to take a Shotgun from the opponent in order to increase his damage output and reduce his opponents. Or, if the player is clever, they could take a Banjo or even another Grapple Arm from the opponent. When a piece of equipment is removed from a JunkBot, the associated command should be removed from that JunkBot’s command list as well.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Description | Parts | Durability | Fuel | Damage | Moonshine |
| Waist | Exhaust Pipes | 10 | 15 | 0 | 15 |
| Waist | Fans | 5 | 15 | 0 | 15 |
| Waist | Extra Fuel Tank | 15 | 0 | 0 | 0 |
| Waist | Gas Can | 10 | 0 | 0 | 0 |

With the Waist peripherals listed above we see two new types of equipment functions. Note that none of these do direct damage to the opponent and only the first two cost fuel to use and generate Moonshine upon use. The Exhaust Pipes are used to obscure the player’s action, by engulfing the player’s JunkBot in thick black smoke. In the player’s command list, the Exhaust Pipe command and the command to be obscured are two separate commands but are associated together. The Exhaust Pipe command is created as a sub-command of the other command and should be indented to show that relationship. For example it could be written as:

2. Shotgun: Opponent: Durability < 45%

3. Exhaust Pipes: Self: Fuel > 30%

4. …

In this example, the Exhaust Pipes command would only execute if the JunkBot’s remaining fuel was greater than 30% and only if the opponent’s durability was less than 45%. If the JunkBot’s remaining fuel was less than 30%, the Shotgun command would still be processed, but the Exhaust Pipes command would not be processed. Going back to an earlier example; if the shotgun was stolen by a Grapple arm, then both of these commands would be removed from the command list. If the Exhaust Pipes were knocked off, only the Exhaust Pipes command would be removed. The Fans are simply a direct counter to the effects of the Exhaust Pipes. This command is set up by a player to blow the smoke away from the opponent as soon as the Exhaust Pipes are triggered.

The Extra Fuel Tank and the Gas Can are used only to boost the fuel of the JunkBot. Essentially they are the same. The Extra Fuel Tank boosts the JunkBot’s maximum fuel by 50. The Gas Can is a command that can be set to refill the JunkBot’s fuel by 50. After which, the Gas Can is removed. The Fuel Tank provides more durability, but the JunkBot’s fuel drops by 50 if the tank is knocked off the JunkBot by a Banjo or Tire Iron. Basically this means that if a JunkBot has 110 units of fuel remaining with an extra fuel tank, but then the fuel gets knocked off, then the JunkBot will only have 60 units of fuel remaining. The Gas Can provides less durability and is disposed of after use, but the fuel value of the JunkBot will never diminish due to equipment being knocked off.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Description | Parts | Durability | Fuel | Damage | Moonshine |
| Shoulder | High Beams | 10 | 15 | 0 | 15 |
| Shoulder | Mirrors | 5 | 15 | 0 | 15 |
| Shoulder | Antenna | 5 | 0 | 0 | 0 |
| Shoulder | Fusebox | 5 | 0 | 0 | 0 |

Shoulder Peripherals work exactly like the waist peripherals do, except they are located on the Shoulders of the JunkBot. The High Beams, like the Exhaust Pipes, are used to obscure the player’s command, but with a bright light instead of smoke. The idea is that the bright light blinds the opponent. The Mirrors, like the Fans are setup to immediately counter the opponent’s use of the High Beams. The light from the High Beams is reflected away from the player allowing them to still see the opponent. The fuel costs, durability, and Moonshine growth values are identical to their waist peripheral counterparts.

The Antenna is used to boost the overall speed of the JunkBot. As far as the fiction is concerned, the Antenna allows for faster and more accurate communication between the Junkman’s command list and the JunkBot. When this part is equipped, the JunkBot’s speed value is reduced by 5 making the re-evaluation time faster. The table below illustrates the change in a JunkBot’s speed with the Antenna equipped.

|  |  |  |
| --- | --- | --- |
|  | Normal | Boosted |
| Total Durability | 200 | 200 |
| Speed Value | 33.33333 | 33.33333 |
| Speed Change | 0 | -5 |
| Re-Evaluation Time | 4.000002 | 3.400001 |

The Fusebox is a unique piece of equipment that counters a piece of Back Mounted Equipment called the Battery. The Battery, seen below, adds boosted attack power to the entirety of the JunkBot. The idea is that electricity flows around the outside of the JunkBot and does additional damage upon contact with the opponent. With the Battery equipped, all types of attacks are boosted by 10. If the opponent equips a Fusebox, then any additional attack power from the Battery is negated or simply reduced by 10.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Description | Parts | Durability | Fuel | Damage | Moonshine |
| Back | Battery | 15 | 0 | 0 | 0 |
| Back | Engine | 25 | 0 | 0 | 0 |
| Back | Wench/Tow Hook | 20 | 25 | 0 | 10 |
| Back | Small Still | 10 | 0 | 0 | 0 |

An external Engine provides a speed boost to the JunkBot much like the Antenna does. However, the speed boost that it gives is larger, because it is heavier. The external engine reduces the speed value of the JunkBot by 10 making the re-evaluation time faster. This is calculated just like the Antenna.

The Wench/Tow Hook is another interesting, yet cost heavy, piece of utility equipment. It is used to restore a piece of knocked off equipment to the JunkBot. It can only be used once and is removed after being used. This is the only way to reclaim removed peripherals, but it also takes up the valuable back mount. Like many of the other utility based pieces of equipment, it is often chosen to counter the opponent. Using it costs 35 units of fuel in order to recover a part and generate 10 Moonshine. It is added to the command list like any other command and recovers the most recent part that was removed. It will do nothing if no parts have been removed.

The Small Still is a unique piece of back mounted equipment that serves to boost the amount of Moonshine gained from each command. With this equipped, each command fills an additional 5 points of the Moonshine gauge. This is real useful for players that want to be able to repair their JunkBots more often or for those who want to save their Moonshine in order to exchange it for new parts at the end of a round.

Combat Example Stats

The following is an example of how two JunkBots fight one another. It will also show how much of a difference that the programming can make. Consider the following 2 JunkBots and their stats. Each will be detailed and explained.

The first JunkBot is designed with low durability and a high speed value. It also sports the light weight speed boost antenna and the attack boosting battery.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| JunkBot 1 |  |  |  |  |  |
| Parts | Durability | Base | Total | Speed | Bonus |
| Head | 0 | 5 | 5 | 0.833333 |  |
| Torso-Unfit | 0 | 35 | 35 | 5.833333 |  |
| Arm-Unfit | 0 | 15 | 15 | 2.5 |  |
| Arm-Unfit | 0 | 15 | 15 | 2.5 |  |
| Antenna | 5 | 0 | 5 | 0.833333 | Speed - 5 |
| Waist | 0 | 5 | 5 | 0.833333 |  |
| Legs-Unfit | 0 | 25 | 25 | 4.166667 |  |
| Waist-Unfit | 0 | 0 | 0 | 0 |  |
| Battery | 15 | 0 | 15 | 2.5 | Atk. + 10 |
|  | 20 | 100 | 120 | 20 |  |
|  |  |  |  |  |  |
|  |  | Antenna Boost: |  | -5 |  |
|  |  | New Speed Value: |  | 15 |  |
|  |  |  |  |  |  |
|  |  | Re-Evaluation Time: |  | 1.800001 | Seconds |

The second JunkBot is equipped with all of the heaviest equipment. It also has a fuel refill.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| JunkBot 2 |  |  |  |  |  |
| Parts | Durability | Base | Total | Speed | Bonus |
| Head | 0 | 5 | 5 | 0.833333 |  |
| Reinforced Frame | 175 | 35 | 210 | 35 |  |
| Tailgate | 75 | 15 | 90 | 15 |  |
| Shotgun | 45 | 15 | 60 | 10 |  |
| High Beams | 10 | 0 | 10 | 1.666667 |  |
| Waist | 0 | 5 | 5 | 0.833333 |  |
| Hydraulics | 75 | 25 | 100 | 16.66667 |  |
| Gas Can | 10 | 0 | 10 | 1.666667 | Fuel Refil |
| Truck Box | 75 | 0 | 75 | 12.5 |  |
|  | 465 | 100 | 565 | 94.16667 |  |
|  |  |  |  |  |  |
|  |  | Re-Evaluation Time: |  | 11.3 | Seconds |

Assuming that JunkBot 1 uses only a single Unarmed Attack and JunkBot 2 uses only a single Truck Box attack, JunkBot 1 will attack 6 times as much as JunkBot 2(11.3/1.8 = 6.28). JunkBot 1 also gets an attack boost. This is represented in the Base Stats Table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Base Stats |  |  |  |  |
| JunkBots | Ratio | Dmg | Bonus | Damage |
| JunkBot 1 | 6 | 15 | 10 | 25 |
| JunkBot 2 | 1 | 75 | 0 | 75 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Iteration 1 |  |  |  |  |  |
| JunkBots | Durability | Fuel | Dmg | Fuel | Moon |
| JunkBot 1 | 120 | 100 | 150 | 0 | 30 |
| JunkBot 2 | 565 | 100 | 75 | 25 | 25 |

The Iteration tables are representative of JunkBot 2 attacking once and JunkBot 1 attacking six times. The Iteration 1 table shows the basic durability and fuel of each JunkBot. The Dmg column shows the damage output based on the speed ratio where JunkBot 2 gets to attack once. The Fuel consumption is listed as is the Moonshine generated. The same formulas are used in Iterations 2 and 3 below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Iteration 2 |  |  |  |  |  |
| JunkBots | Durability | Fuel | Dmg x 2 | Fuel | Moon |
| JunkBot 1 | 120 | 100 | 300 | 0 | 60 |
| JunkBot 2 | 565 | 100 | 150 | 50 | 50 |
|  |  |  |  |  |  |
| Iteration 3 |  |  |  |  |  |
| JunkBots | Durability | Fuel | Dmg x 3 | Fuel | Moon |
| JunkBot 1 | 120 | 100 | 450 | 0 | 90 |
| JunkBot 2 | 565 | 100 | 225 | 75 | 75 |

` JunkBot 1 will be destroyed during Iteration 2. JunkBot 2 would still survive after Iteration 3. So what if JunkBot 1 executes 2 commands, namely 2 unarmed attacks. The tables below show those results.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Base Stats |  |  |  |  |  |
| JunkBots | Ratio | Dmg | Bonus | Damage |  |
| JunkBot 1 | 6 | 30 | 20 | 50 |  |
| JunkBot 2 | 1 | 75 | 0 | 75 |  |
|  |  |  |  |  |  |
| Iteration 1 |  |  |  |  |  |
| JunkBots | Durability | Fuel | Dmg | Fuel | Moon |
| JunkBot 1 | 120 | 100 | 300 | 0 | 60 |
| JunkBot 2 | 565 | 100 | 75 | 25 | 25 |
|  |  |  |  |  |  |
| Iteration 2 |  |  |  |  |  |
| JunkBots | Durability | Fuel | Dmg x 2 | Fuel | Moon |
| JunkBot 1 | 120 | 100 | 600 | 0 | 120 |
| JunkBot 2 | 565 | 100 | 150 | 50 | 50 |

Both JunkBots would die during Iteration 2 and at roughly the same time exactly. However, JunkBot 1 would have built up enough Moonshine for a repair command by attack 5 out of 6. The repair command could be initiated with attack 5 to heal JunkBot 1 by 50% and ultimately win the round. All that aside, there are many other attack options that JunkBot 2 can utilize. JunkBot 1 could also use basic kicks or power attacks as well. There should not ever be a concrete strategy for victory.