

# Scripting Tutorials 2012

## Armor Abilities

12/17/2012

Kirby\_422

So, say we have far too much time on our hands, and we want to make stuff like those armor abilities in reach. We start just blasting out connections, make a list if you have a bad memory.

### **How shall we monitor players correctly?**

object\_lists shift after all; player0 won't always be player0. We'll simply store each player in their own global. How will we make sure we aren't doubling up storing the same player in every global? giant script to math it. How will we free a global up after a player dies? put it in part of their activation script to check their life.

### **How will we offer them to swap abilities?**

a device control (button) wouldn't sync location, however a vehicle would. Check when they enter a vehicle, kick them out, swap abilities, change vehicle to your previous ability.

### **What if we wanted the client wanted to know which ability was his?**

OS . The seat methods in the Tagging with Scripts tutorial, with the boost example. If you do not want to use OpenSauce, you can always skip this.

Activating would obviously be flashlight, unless you wanted to make an obscure activation via opensauce and animations.

### **What abilities are we going want for this example script?**

Convert shield to health, camo (OSless method), random teleport (why not? lol), suicide explosion (for lawlz), double shield.

### **How are we going to prevent them from being spammed?**

Do we want to use energy limits? make them only able to use them once every minute? This example is going do energy limits, you can do it however you want. Do we want the player to spawn with energy? How much?

**For completed script, see the bottom (Page 11).** I'll only be showing parts here and there throughout this tutorial.

We're using host detection, I'm overlapping an OS command so you can just remove this line if you're using OS without having to change anything else.

```
;; Host detecting. vehicle, host_detect, must spawn naturally (without scripts)
(script static boolean machine_is_host (!= (unit_get_health host_detect) -1))
;;If using OpenSauce, remove this. ^^ There is an OS command with the same name,
;; that the scripts will use instead.
```

Well, first of all, we want to worry about all of those players, making sure we don't have to worry about unreliable object\_lists. As previously mentioned, having a global for each player works. I stated that we would decide this through math. How you may ask; It's no different than  $x=5$ . We check if the item in the object list is equal to any of the globals; if they aren't, then we toss them in a unused global.

```
(global unit p1 none)
(global unit p2 none)
(global unit p3 none)
;;<etc>
(global short number 0)
(script static unit player (unit (list_get (players) number)))

(script continuous assign_players
  (if (not (machine_is_host)) (sleep -1)) ;;Non-hosts do not need to run this.
  (if (not (or ;;Or will return true if any of these are true, so, if the player equals any variable, it'll return true. The NOT however, will only return true when the or returns false; eg, when NONE of the player globals equal the player in the object_list. It must be reset when they die, since the corpse is the player being monitored at that point.
    (= (player) p0) (= (player) p1) (= (player) p2) (= (player) p3)
    (= (player) p4) (= (player) p5) (= (player) p6) (= (player) p7)
    (= (player) p8) (= (player) p9) (= (player) p10) (= (player) p11)
    (= (player) p12) (= (player) p13) (= (player) p14) (= (player) p15)
  )) ;;If the player isn't already hooked
  (if (= p0 none) (set p0 (player)) ;;Find the first available global that is not in use (that holds nothing)
    (if (= p1 none) (set p1 (player))
      (if (= p2 none) (set p2 (player))
        (if (= p3 none) (set p3 (player))
          ;;<ETC>
        ))))
  )
  (if (< number (- player_spawn_count 1)) (set number (+ number 1)) (set number 0))
  ;;Only check players that are in the server (I could check the number alive,
  ;; but I didn't for some reason.)
  )
)
```

So, how that we know whose who, what about knowing whose AA is what, and how much energy they have left? Well, globals.. I don't need to show it to you, but I will anyways.

```
.....
/////
;; Abilities
;; 0 - Shield2Health
;; 1 - Camo
;; 2 - Random Teleport
;; 3 - Suicide
```

```

;; 4 - Double Shield
(global short p0_aa 0)
(global short p1_aa 0)
(global short p2_aa 0)
(global short p3_aa 0)
;;<ETC>
;;;;
;; Energy
;; 0 == max available
;; 360 == empty
;; recharge 1 every 1/3rd a second.
(global short p0_e 0)
(global short p1_e 0)
(global short p2_e 0)
(global short p3_e 0)
;;<ETC>

```

Now that we have this stuff, what are we going to do to recharge energy?

```

(script continuous recharge_energy
  (if (> p0_e 0) (set p0_e (- p0_e 1))) ;;If energy hasn't returned to 0, decrement it.
  (if (> p1_e 0) (set p1_e (- p1_e 1)))
  (if (> p2_e 0) (set p2_e (- p2_e 1)))
  (if (> p3_e 0) (set p3_e (- p3_e 1)))
  (sleep 10) ;;Recharge 3 a second.
)

```

Simple decrementing (Since I decided that 0 was all energy available for use.. I should of done it the other way. The only place your energy is checked is when using AA, so you would just alter AA to be able to run with different energy levels.)

Now, we need to define some abilities. We have a few ways we could do this; we can have each player perform the full actions, or go with globals and static scripts to perform the actions. Since statics look nicer, that is what we'll do.

```

(global unit perform_biped none)
(script static void perform_shield2health
  (if (< (+ (unit_get_health perform_biped) (unit_get_shield perform_biped)) 1)
    ;;If shield + health is less than 100%, then just put all shield into health.
    (unit_set_current_vitality perform_biped (* (+ (unit_get_health perform_biped) (unit_get_shield perform_biped)) 75) 0)
    (unit_set_current_vitality perform_biped 75 (* (- (+ (unit_get_health perform_biped) (unit_get_shield perform_biped)) 1) 75))
    ;;Otherwise, just put part of the shield into health till it's full, and keep the remaining parts as shield.
  )
  (set perform_biped none)
)

```

```
;;Clear the variable. I am using this as bug protection (read how the players use it)
)
```

See bottom for the rest of the AA statics (page 13). One is enough to explain the point.

This is AA 0, converting your shield to health. Before this script is called, the player simply has to toss their player global data, into the `perform_biped` global variable. This static is then used by all players. This particular one, compares the sum of health and shields, and if you do not have enough to fill your health with shield converted, it will just convert all of your shield to health. If you have more than enough shield to fully fill your health, it will leave the remaining as shield. At this point, I might as well talk about the vitality commands. `unit_set_current_vitality` takes health and shield numbers in, as the actual number. A default cyborg has 75 health, and 75 shield, therefore when set it is required to have a health out of 75 set. `unit_get_health` will return health as a fraction. 1 being full, 0 being empty, 0.5 as half. You must multiply the `unit_get_health` by the max possible value to keep the value accurate. And, while on this topic, `unit_set_maximum_vitality` will also set the health to full when changing the maximum value.

Next, we look over the checklist. We still have done nothing about the player actually executing these AA. We had noted that we also wanted to monitor whether the player was alive in this too, because we have nowhere else to free the variable. We could free it elsewhere, but I have enough scripts executing at once already.

```
(script continuous p0_usage
(sleep_until (!= p0 none) 15) ;;Wait until P0 is assigned
(sleep_until (or (unit_get_current_flashlight_state p0) (<= (unit_get_health p0) 0)) 5) ;;Wait until
flashlight, or death.
(if (unit_get_current_flashlight_state p0) (begin ;;If flashlight
(unit_set_desired_flashlight_state p0 0) ;;Turn off flashlight
(sleep_until (= perform_biped none) 1) ;;Wait until the global for performing AA is open
(if (= p0_aa 0) (if (< p0_e 180) (begin (set perform_biped p0) (perform_shield2health) (set p0_e (+
p0_e 180))))
;;If AA is 0 (Shield to Health), and has 180 energy, perform it and use up 180 energy.
(if (= p0_aa 1) (if (< p0_e 240) (begin (set perform_biped p0) (perform_camo) (set p0_e (+ p0_e
120))))
;;If AA is 1 (Camo), and has 120 energy, preform it and use 120 energy.
(if (= p0_aa 2) (if (< p0_e 270) (begin (set perform_biped p0) (perform_random_tele) (set p0_e (+
p0_e 90))))
;;If AA is 2 (Tele), and has 90 energy, perform.
(if (= p0_aa 3) (begin (set perform_biped p0) (perform_suicide))
;;If AA is 3, explode. Who cares how much energy you have if you want to die.
(if (< p0_e 180) (begin (set perform_biped p0) (perform_double_shield) (set p0_e (+ p0_e 180))))
;;if AA is 4 and has 180 energy, double shield.
))))
))
;;Setting the 'preform_biped' in each of them was kind of a waste,
```

```

;; I should of done it at the start, but whatever.
)(begin ;;If death, free p0 variable and such.
  (set p0_e 0) ;;Clear energy
  (set p0_aa 0) ;;Clear AA
  (set p0 none) ;;Free P0 Variable
))
)

```

Since the player will be dead if they are unassigned, as well as they won't be using AA, we simply wait until they are assigned. No need to waste other checks until they're there. Now, we can sit around waiting until they die or turn their flashlight on. If they are dead, skip all this AA stuff, and simply unallocated the globals for a new player. Now, on the other hand, if they hit flashlight, we want to check what AA they have, and if they have the energy for it. We can check it in two forms, whether they have the AA AND have the energy to run it, or check the AA and THEN the ability. I myself like checking AA, then energy, because then I can perform nested if statements; where the else statement checks if another possibility exists. This can also be done with a (cond (<boolean> <result>) [(<boolean> <result>)]), but I like if's, it's my personal preference. Cond would actually be the smarter choice in reality.

```
(cond ((= p0_aa 0) (blah)) ((= p0_aa 1) (blah)) ((= p0_aa 2) (blah)))
```

This would be the formatting of it as a cond statement.

Now, we have enough to use the system fine.. but what are we missing? Of course, it's the all important ability to change abilities. We want to leave our old ability where we picked up our new one, so we want it to sync. You people may want to lug a weapon around, but I sure don't. As I mentioned in my goals at the top, I'm using vehicles.

```

(global vehicle aa_veh NONE) ;;Global used to hold tested vehicle.
(global short aa_veh_type 0) ;;What armor ability does this vehicle represent?
(global vehicle aa_swapper NONE) ;;What are we putting in its place?
(global short swap_type 0) ;;What type of armor ability did this player have?
(global unit rec_player NONE) ;;Lets hold a copy of whomever it is in here.
(script static void check_n_swap_aa
  (if (vehicle_test_seat_list aa_veh "aa-driver" (players)) (begin ;;Check if any players are in the tested
  vehicle.
  (if (vehicle_test_seat_list aa_veh "aa-driver" p0) ;;If its p0, get that players AA, and replace it with this
  vehicles' AA.
    (begin (set swap_type p0_aa) (set p0_aa aa_veh_type) (set rec_player p0))
  (if (vehicle_test_seat_list aa_veh "aa-driver" p1)
    (begin (set swap_type p1_aa) (set p1_aa aa_veh_type) (set rec_player p1))
  (if (vehicle_test_seat_list aa_veh "aa-driver" p2)
    (begin (set swap_type p2_aa) (set p2_aa aa_veh_type) (set rec_player p2))
  (if (vehicle_test_seat_list aa_veh "aa-driver" p3)
    (begin (set swap_type p3_aa) (set p3_aa aa_veh_type) (set rec_player p3))

```

```

;;<etc>
))))
(if (= swap_type 0) ;;If the type the player had was health to shield, find a free health to shield
vehicle to toss where the players at.
  (if (volume_test_object storage_vol aa_health_shield_00)
      (set aa_swapper aa_health_shield_00)
      (if (volume_test_object storage_vol aa_health_shield_01)
          (set aa_swapper aa_health_shield_01)
          (if (volume_test_object storage_vol aa_health_shield_02)
              (set aa_swapper aa_health_shield_02)
              )))
  ))
(if (= swap_type 1)
  (if (volume_test_object storage_vol aa_camo_00) (set aa_swapper aa_camo_00)
      (if (volume_test_object storage_vol aa_camo_01) (set aa_swapper aa_camo_01)
          (if (volume_test_object storage_vol aa_camo_02) (set aa_swapper aa_camo_02)
              )))
  ))
;;<etc>
))
.....
;; Common Route (Swapping locations)
(vehicle_unload aa_veh "") ;;Get the player out
(unit_exit_vehicle rec_player) ;;We really don't want them in that vehicle (This should actually be
more effective than vehicle_unload).
(sleep_until (not (vehicle_test_seat_list aa_veh "aa-driver" (players))) 1) ;;Wait until they're out
(objects_attach aa_veh "" aa_swapper "")) ;;Send the new vehicle to where the old one is.
(objects_detach aa_veh aa_swapper)
(object_teleport aa_veh storage_flag) ;;Send the old one to storage.
;;
.....
))
)

```

But wait, this is a static script. What uses it? where does that aa\_veh come from? Well, it comes from here of course

```

(script continuous swapping_aa
(set aa_veh_type 0) ;;Define the AA type
(set aa_veh aa_health_shield_00) ;;set a vehicle to the variable
(check_n_swap_aa) ;;and check the variable
(set aa_veh aa_health_shield_01) ;;Try a new vehicle
(check_n_swap_aa) ;;and run the check again
(set aa_veh aa_health_shield_02) ;;you get the idea
(check_n_swap_aa)
(set aa_veh_type 1) ;;Try a different AA type
(set aa_veh aa_camo_00) ;;With its vehicles.
(check_n_swap_aa)
(set aa_veh aa_camo_01)

```

```
;;<Moar boar wants MOAR>  
)
```

Just spam your way through all the vehicles constantly. This script will execute all of that earlier script, check\_n\_swap\_aa for each vehicle you sent through this.

## If you are not an OpenSauce User, then this is the end for you. Get the full script at the bottom of this document (Starting at Page 11)

Now, all you sauce users are complaining, "I want the user to know what he has!" Well, since we're using OS, we can track local players! And since I'm lazy, we're using the console to print info to players. You people may be more interested in things such as Post Processing to display the info to users. Now, printing information, this is something halo doesn't normally just do on its own. The game has an internal variable known as developer\_mode, not to be confused with the startup parameter devmode. This global variable controls whether the game gives you debugging information or not. Higher than 4 will print messages, but sapien uses 127 by default, so we'll use that; sapien gives us good amounts of info after all (Although you may get occasional messages like too many bipeds to render, or things falling outside the BSP to their death.. nice information, but not what we're after. Still fun to have on screen though.)

So, we're storing copies for the local player. This is just easier since I'm not caring who the host is. If you pay attention to who the host is, they can fish out their own info, and clients can just all store in p0 since they dont use those globals, but I don't care that much.

```
(script static unit local_player (unit (player_data_get_object (player_local_get) slave_unit))) ;;Get the  
local player  
(global unit loc_player NONE) ;;Store local player (used to determine when they die)  
(global short local_ability 0) ;;Store AA number  
(global short local_energy 0) ;;Store energy  
(global unit current_vehicle NONE) ;;Modified boost script (detecting closest vehicle)  
(global short vehicle_id -1) ;;Modified boost script (detecting vehicle your sitting in)  
  
(script continuous reset_on_death ;;Same stuff thats at the bottom of the p0,p1, etc. clean up on  
death.  
(sleep_until (!= (local_player) loc_player) 10) ;;If the local player doesnt match the hooked body,  
they must be dead.  
(set local_ability 0)
```



```

(set local_energy 0)
(set loc_player (local_player))
)

```

**;;See boosting tutorial.**

```

(script static void identify_vehicle
.....
;;-1 - unset
;;0 - Unknown (most likely a real vehicle)
;;# - INSERT NAME
.....
;(sleep 30) ;;Make sure you have a very short enter animation (1 tick), otherwise you'll need to
uncomment and edit this.
;;(if (and (> (objects_distance_to_object (local_player) current_vehicle) #####) (<
(objects_distance_to_object (local_player) current_vehicle) #####)) (set vehicle_id 1)
;;enter vehicle, and use objects_distance_to_object with dev. Enter a range above and below, that is
unique to this vehicle (eg, dont place the driver seats in the same places for these AA)
;;Uncomment this stuff if inuse.
(set vehicle_id 0)
;;)
) ;;This can be used alongside boosting. This block is basically the boost, except I ignored default
vehicles.
(script continuous select_vehicle
(sleep_until (and (!= (objects_distance_to_object (local_player) (player_data_get_object
(player_local_get) nearest_object)) -1) (= (unit_data_get_real (unit (player_data_get_object
(player_local_get) nearest_object)) driver_power) 0)) 1)
(set current_vehicle (unit (player_data_get_object (player_local_get) nearest_object)))
)
(script continuous vehicle_AA_check
(sleep_until (and (>= (unit_data_get_real current_vehicle driver_power) 0.5) (<
(objects_distance_to_object (local_player) current_vehicle) 1)) 5)
(identify_vehicle)
(if (!= vehicle_id 0) (begin (set local_ability vehicle_id)
;;Add something else in for aa 0, like if its 20. Or, change the unknown vehicle to some other number
(the smarter idea, but I don't really care)
(unit_exit_vehicle (local_player))
(vehicle_unload current_vehicle ""))
))
(sleep_until (not (and (>= (unit_data_get_real current_vehicle driver_power) 1) (<
(objects_distance_to_object (players) current_vehicle) 1)))) 5)
(set vehicle_id -1)
)

```

**;;copy of the energy recharge, just for local player.**

```

(script continuous local_recharge_energy

```

```
(if (> local_energy 0) (set local_energy (- local_energy 1))) ;;If energy hasnt returned to 0, decrement
it.
(sleep 10) ;;Recharge 3 a second.
)
```

Just the energy drain from the top area. The actions are still performed on the host.

```
(script continuous local_usage
(sleep_until (unit_get_current_flashlight_state (local_player)) 1)
(if (= local_ability 0) (if (< local_energy 180) (set local_energy (+ local_energy 180)))
;;If AA is 0 and has 180 energy, use up 180 energy.
(if (= local_ability 1) (if (< local_energy 240) (set local_energy (+ local_energy 120)))
;;If AA is 1 (Camo), and has 120 energy, use 120 energy.
(if (= local_ability 2) (if (< local_energy 270) (set local_energy (+ local_energy 90)))
;;If AA is 2 (Tele), and has 90 energy
(if (= local_ability 4) (if (< local_energy 180) (set local_energy (+ local_energy 180)))
;;if AA is 4 and has 180 energy
))))
(sleep 10)
)
```

```
....
;; Player now has the ability tracked... Up to you decide on what to do with it now.
(script continuous printing_info
(set developer_mode 127) ;;Displaying debug info, and print statements.
(clear) ;;Clears other text from screen. This keeps the appearance constant.
(if (and (= local_ability 0) (< local_energy 180)) (print "ability usable [shield -> health]"))
(if (and (= local_ability 1) (< local_energy 240)) (print "ability usable [camo]"))
(if (and (= local_ability 2) (< local_energy 270)) (print "ability usable [random teleport]"))
(if (= local_ability 3) (print "ability usable [self destruct]"))
(if (and (= local_ability 4) (< local_energy 180)) (print "ability usable [double shields]"))
(print "ability disabled [not enough energy]"))
))))
(print " ") ;;Empty line
(print " ") ;;Empty line
(inspect local_energy) ;;Displays number in white.
(print "-----")
(print "360")
(sleep 15)
)
```



This is what this interface looks like ingame.

# Final Product

(No, I'm not color coding this part. I've spent enough time coloring the examples above. Check there for colors)

This script was written back in august for the most part (i've added like 10 lines since then) but simply not being at home was the only reason this hadn't been released lol.

```
.....  
;; Player hooks.  
(global unit p0 none)  
(global unit p1 none)  
(global unit p2 none)  
(global unit p3 none)  
(global unit p4 none)  
(global unit p5 none)  
(global unit p6 none)  
(global unit p7 none)  
(global unit p8 none)  
(global unit p9 none)  
(global unit p10 none)  
(global unit p11 none)  
(global unit p12 none)  
(global unit p13 none)  
(global unit p14 none)  
(global unit p15 none)  
;;  
.....  
.....  
;; Abilities  
;; 0 - Shield2Health  
;; 1 - Camo  
;; 2 - Random Teleport  
;; 3 - Suicide  
;; 4 - Double Shield  
(global short p0_aa 0)  
(global short p1_aa 0)  
(global short p2_aa 0)  
(global short p3_aa 0)  
(global short p4_aa 0)  
(global short p5_aa 0)  
(global short p6_aa 0)  
(global short p7_aa 0)  
(global short p8_aa 0)  
(global short p9_aa 0)  
(global short p10_aa 0)  
(global short p11_aa 0)  
(global short p12_aa 0)  
(global short p13_aa 0)  
(global short p14_aa 0)  
(global short p15_aa 0)  
;;  
.....  
.....  
;; Energy  
;; 0 == max available  
;; 360 == empty  
;; recharge 1 every 1/3rd a second.
```

```

(global short p0_e 0)
(global short p1_e 0)
(global short p2_e 0)
(global short p3_e 0)
(global short p4_e 0)
(global short p5_e 0)
(global short p6_e 0)
(global short p7_e 0)
(global short p8_e 0)
(global short p9_e 0)
(global short p10_e 0)
(global short p11_e 0)
(global short p12_e 0)
(global short p13_e 0)
(global short p14_e 0)
(global short p15_e 0)
;;
;;;;

(global short number 0)
(script static unit player (unit (list_get (players) 0)))

;; Host detecting. vehicle, host_detect, must spawn naturally (without scripts)
(script static boolean machine_is_host (!= (unit_get_health host_detect) -1))
;;If using OpenSauce, remove this. ^^ There is an OS command with the same name,
;; that the scripts will use instead.

(script continuous assign_players
(if (not (machine_is_host)) (sleep -1)) ;;Non-hosts do not need to run this.
(if (not (or
(= (player) p0) (= (player) p1) (= (player) p2) (= (player) p3)
(= (player) p4) (= (player) p5) (= (player) p6) (= (player) p7)
(= (player) p8) (= (player) p9) (= (player) p10) (= (player) p11)
(= (player) p12) (= (player) p13) (= (player) p14) (= (player) p15)
)) ;;If the player isn't already hooked
(if (= p0 none) (set p0 (player))
(if (= p1 none) (set p1 (player))
(if (= p2 none) (set p2 (player))
(if (= p3 none) (set p3 (player))
(if (= p4 none) (set p4 (player))
(if (= p5 none) (set p5 (player))
(if (= p6 none) (set p6 (player))
(if (= p7 none) (set p7 (player))
(if (= p8 none) (set p8 (player))
(if (= p9 none) (set p9 (player))
(if (= p10 none) (set p10 (player))
(if (= p11 none) (set p11 (player))
(if (= p12 none) (set p12 (player))
(if (= p13 none) (set p13 (player))
(if (= p14 none) (set p14 (player))
(if (= p15 none) (set p15 (player))
))))))))))))))
)
(if (< number (- player_spawn_count 1)) (set number (+ number 1)) (set number 0))
;;Only check players that are in the server (I could check the number alive,
;; but I didnt for some reason.)
)

;;So, to reduce used scripting space, just make them all statics
(global unit perform_biped none)
(script static void perform_shield2health

```

```

(if (< (+ (unit_get_health perform_biped) (unit_get_shield perform_biped)) 1)
;;If shield + health is less than 100%, then just put all shield into health.
(unit_set_current_vitality perform_biped (* (+ (unit_get_health perform_biped) (unit_get_shield perform_biped)) 75) 0)
(unit_set_current_vitality perform_biped 75 (* (- (+ (unit_get_health perform_biped) (unit_get_shield perform_biped)) 1) 75))
;;Otherwise, just put part of the shield into health, and keep the remaining parts as shield.
)
(set perform_biped none)
;;Clear the variable. I am using this as bug protection (read how the players use it)
)
(script static void perform_camo
(object_create_aneu random_camo_powerup) ;;Create the powerup
(objects_attach perform_biped "body" random_camo_powerup "") ;;Move the powerup to the player
(objects_detach perform_biped random_camo_powerup)
(set perform_biped none) ;;Remove hook.
(sleep 10) ;;Wait 1/3rd a second
(object_destroy random_camo_powerup) ;;If it still exists, get rid of it
)
(global short teleport_loc 0)
(script static void perform_random_tele
(set teleport_loc (random_range 0 6)) ;;Generate random number from 0 to 5.
(if (= teleport_loc 0) (object_teleport perform_biped loc_0)
(if (= teleport_loc 1) (object_teleport perform_biped loc_1)
(if (= teleport_loc 2) (object_teleport perform_biped loc_2)
(if (= teleport_loc 3) (object_teleport perform_biped loc_3)
(if (= teleport_loc 4) (object_teleport perform_biped loc_4)
(object_teleport perform_biped loc_5)
))))))
(set perform_biped none)
)
(script static void perform_suicide
(unit_kill perform_biped)
(effect_new_on_object_marker "effects\suicide" perform_biped "body")
;;This really is just wasting space, but still lol.
(set perform_biped none)
)
(script static void perform_double_shield
(unit_set_current_vitality perform_biped (* (unit_get_health perform_biped) 75) (* (unit_get_shield perform_biped) 150))
;;Still wasting space lol.
(set perform_biped none)
)
(script continuous recharge_energy
(if (> p0_e 0) (set p0_e (- p0_e 1))) ;;If energy hasnt returned to 0, decrement it.
(if (> p1_e 0) (set p1_e (- p1_e 1)))
(if (> p2_e 0) (set p2_e (- p2_e 1)))
(if (> p3_e 0) (set p3_e (- p3_e 1)))
(if (> p4_e 0) (set p4_e (- p4_e 1)))
(if (> p5_e 0) (set p5_e (- p5_e 1)))
(if (> p6_e 0) (set p6_e (- p6_e 1)))
(if (> p7_e 0) (set p7_e (- p7_e 1)))
(if (> p8_e 0) (set p8_e (- p8_e 1)))
(if (> p9_e 0) (set p9_e (- p9_e 1)))
(if (> p10_e 0) (set p10_e (- p10_e 1)))
(if (> p11_e 0) (set p11_e (- p11_e 1)))
(if (> p12_e 0) (set p12_e (- p12_e 1)))
(if (> p13_e 0) (set p13_e (- p13_e 1)))
(if (> p14_e 0) (set p14_e (- p14_e 1)))
(if (> p15_e 0) (set p15_e (- p15_e 1)))
(sleep 10) ;;Recharge 3 a second.
)
(script continuous p0_usage
(sleep_until (!= p0 none) 15) ;;Wait until P0 is assigned
(sleep_until (or (unit_get_current_flashlight_state p0) (<= (unit_get_health p0) 0)) 5) ;;Wait until flashlight, or death.
)

```

```

(if (unit_get_current_flashlight_state p0) (begin ;;If flashlight
(unit_set_desired_flashlight_state p0 0) ;;Turn off flashlight
(sleep_until (= perform_biped none) 1) ;;Wait until the global for performing AA is open
(if (= p0_aa 0) (if (< p0_e 180) (begin (set perform_biped p0) (perform_shield2health) (set p0_e (+ p0_e 180))))
;;If AA is 0 (Shield to Health), and has 180 energy, perform it and use up 180 energy.
(if (= p0_aa 1) (if (< p0_e 240) (begin (set perform_biped p0) (perform_camo) (set p0_e (+ p0_e 120))))
;;If AA is 1 (Camo), and has 120 energy, perform it and use 120 energy.
(if (= p0_aa 2) (if (< p0_e 270) (begin (set perform_biped p0) (perform_random_tele) (set p0_e (+ p0_e 90))))
;;If AA is 2 (Tele), and has 90 energy, perform.
(if (= p0_aa 3) (begin (set perform_biped p0) (perform_suicide))
;;If AA is 3, explode.
(if (< p0_e 180) (begin (set perform_biped p0) (perform_double_shield) (set p0_e (+ p0_e 180))))
;;if AA is 4 and has 180 energy, double shield.
))))
;;Setting the 'perform_biped' in each of them was kind of a waste,
;; I should of done it at the start, but whatever.
)(begin ;;If death, free p0 variable and such.
(set p0_e 0) ;;Clear energy
(set p0_aa 0) ;;Clear AA
(set p0 none) ;;Free P0 Variable
))
)

(script continuous p1_usage
(sleep_until (!= p1 none) 15) ;;Wait until p1 is assigned
(sleep_until (or (unit_get_current_flashlight_state p1) (<= (unit_get_health p1) 0)) 5) ;;Wait until flashlight, or death.
(if (unit_get_current_flashlight_state p1) (begin ;;If flashlight
(unit_set_desired_flashlight_state p1 0) ;;Turn off flashlight
(sleep_until (= perform_biped none) 1) ;;Wait until the global for performing AA is open
(if (= p1_aa 0) (if (< p1_e 180) (begin (set perform_biped p1) (perform_shield2health) (set p1_e (+ p1_e 180))))
;;If AA is 0 (Shield to Health), and has 180 energy, perform it and use up 180 energy.
(if (= p1_aa 1) (if (< p1_e 240) (begin (set perform_biped p1) (perform_camo) (set p1_e (+ p1_e 120))))
;;If AA is 1 (Camo), and has 120 energy, perform it and use 120 energy.
(if (= p1_aa 2) (if (< p1_e 270) (begin (set perform_biped p1) (perform_random_tele) (set p1_e (+ p1_e 90))))
;;If AA is 2 (Tele), and has 90 energy, perform.
(if (= p1_aa 3) (begin (set perform_biped p1) (perform_suicide))
;;If AA is 3, explode.
(if (< p1_e 180) (begin (set perform_biped p1) (perform_double_shield) (set p1_e (+ p1_e 180))))
;;if AA is 4 and has 180 energy, double shield.
))))
)(begin ;;If death, free p1 variable and such.
(set p1_e 0) ;;Clear energy
(set p1_aa 0) ;;Clear AA
(set p1 none) ;;Free p1 Variable
))
)

(script continuous p2_usage
(sleep_until (!= p2 none) 15) ;;Wait until p2 is assigned
(sleep_until (or (unit_get_current_flashlight_state p2) (<= (unit_get_health p2) 0)) 5) ;;Wait until flashlight, or death.
(if (unit_get_current_flashlight_state p2) (begin ;;If flashlight
(unit_set_desired_flashlight_state p2 0) ;;Turn off flashlight
(sleep_until (= perform_biped none) 1) ;;Wait until the global for performing AA is open
(if (= p2_aa 0) (if (< p2_e 180) (begin (set perform_biped p2) (perform_shield2health) (set p2_e (+ p2_e 180))))
;;If AA is 0 (Shield to Health), and has 180 energy, perform it and use up 180 energy.
(if (= p2_aa 1) (if (< p2_e 240) (begin (set perform_biped p2) (perform_camo) (set p2_e (+ p2_e 120))))
;;If AA is 1 (Camo), and has 120 energy, perform it and use 120 energy.
(if (= p2_aa 2) (if (< p2_e 270) (begin (set perform_biped p2) (perform_random_tele) (set p2_e (+ p2_e 90))))
;;If AA is 2 (Tele), and has 90 energy, perform.
(if (= p2_aa 3) (begin (set perform_biped p2) (perform_suicide))
;;If AA is 3, explode.
(if (< p2_e 180) (begin (set perform_biped p2) (perform_double_shield) (set p2_e (+ p2_e 180))))
;;if AA is 4 and has 180 energy, double shield.
))))
)

```

```

))))
)(begin ;;If death, free p2 variable and such.
  (set p2_e 0) ;;Clear energy
  (set p2_aa 0) ;;Clear AA
  (set p2 none) ;;Free p2 Variable
)
)
(script continuous p3_usage
  (sleep_until (!= p3 none) 15) ;;Wait until p3 is assigned
  (sleep_until (or (unit_get_current_flashlight_state p3) (<= (unit_get_health p3) 0)) 5) ;;Wait until flashlight, or death.
  (if (unit_get_current_flashlight_state p3) (begin ;;If flashlight
    (unit_set_desired_flashlight_state p3 0) ;;Turn off flashlight
    (sleep_until (= perform_biped none) 1) ;;Wait until the global for performing AA is open
    (if (= p3_aa 0) (if (< p3_e 180) (begin (set perform_biped p3) (perform_shield2health) (set p3_e (+ p3_e 180))))
    ;;If AA is 0 (Shield to Health), and has 180 energy, perform it and use up 180 energy.
    (if (= p3_aa 1) (if (< p3_e 240) (begin (set perform_biped p3) (perform_camo) (set p3_e (+ p3_e 120))))
    ;;If AA is 1 (Camo), and has 120 energy, perform it and use 120 energy.
    (if (= p3_aa 2) (if (< p3_e 270) (begin (set perform_biped p3) (perform_random_tele) (set p3_e (+ p3_e 90))))
    ;;If AA is 2 (Tele), and has 90 energy, perform.
    (if (= p3_aa 3) (begin (set perform_biped p3) (perform_suicide))
    ;;If AA is 3, explode.
    (if (< p3_e 180) (begin (set perform_biped p3) (perform_double_shield) (set p3_e (+ p3_e 180))))
    ;;If AA is 4 and has 180 energy, double shield.
  )))
)(begin ;;If death, free p3 variable and such.
  (set p3_e 0) ;;Clear energy
  (set p3_aa 0) ;;Clear AA
  (set p3 none) ;;Free p3 Variable
)
)
(script continuous p4_usage
  (sleep_until (!= p4 none) 15) ;;Wait until p4 is assigned
  (sleep_until (or (unit_get_current_flashlight_state p4) (<= (unit_get_health p4) 0)) 5) ;;Wait until flashlight, or death.
  (if (unit_get_current_flashlight_state p4) (begin ;;If flashlight
    (unit_set_desired_flashlight_state p4 0) ;;Turn off flashlight
    (sleep_until (= perform_biped none) 1) ;;Wait until the global for performing AA is open
    (if (= p4_aa 0) (if (< p4_e 180) (begin (set perform_biped p4) (perform_shield2health) (set p4_e (+ p4_e 180))))
    ;;If AA is 0 (Shield to Health), and has 180 energy, perform it and use up 180 energy.
    (if (= p4_aa 1) (if (< p4_e 240) (begin (set perform_biped p4) (perform_camo) (set p4_e (+ p4_e 120))))
    ;;If AA is 1 (Camo), and has 120 energy, perform it and use 120 energy.
    (if (= p4_aa 2) (if (< p4_e 270) (begin (set perform_biped p4) (perform_random_tele) (set p4_e (+ p4_e 90))))
    ;;If AA is 2 (Tele), and has 90 energy, perform.
    (if (= p4_aa 3) (begin (set perform_biped p4) (perform_suicide))
    ;;If AA is 3, explode.
    (if (< p4_e 180) (begin (set perform_biped p4) (perform_double_shield) (set p4_e (+ p4_e 180))))
    ;;If AA is 4 and has 180 energy, double shield.
  )))
)(begin ;;If death, free p4 variable and such.
  (set p4_e 0) ;;Clear energy
  (set p4_aa 0) ;;Clear AA
  (set p4 none) ;;Free p4 Variable
)
)
(script continuous p5_usage
  (sleep_until (!= p5 none) 15) ;;Wait until p5 is assigned
  (sleep_until (or (unit_get_current_flashlight_state p5) (<= (unit_get_health p5) 0)) 5) ;;Wait until flashlight, or death.
  (if (unit_get_current_flashlight_state p5) (begin ;;If flashlight
    (unit_set_desired_flashlight_state p5 0) ;;Turn off flashlight
    (sleep_until (= perform_biped none) 1) ;;Wait until the global for performing AA is open
    (if (= p5_aa 0) (if (< p5_e 180) (begin (set perform_biped p5) (perform_shield2health) (set p5_e (+ p5_e 180))))
    ;;If AA is 0 (Shield to Health), and has 180 energy, perform it and use up 180 energy.
    (if (= p5_aa 1) (if (< p5_e 240) (begin (set perform_biped p5) (perform_camo) (set p5_e (+ p5_e 120))))
  )))
)

```



```

;;If AA is 1 (Camo), and has 120 energy, preform it and use 120 energy.
(if (= p5_aa 2) (if (< p5_e 270) (begin (set perform_biped p5) (perform_random_tele) (set p5_e (+ p5_e 90))))))
;;If AA is 2 (Tele), and has 90 energy, perform.
(if (= p5_aa 3) (begin (set perform_biped p5) (perform_suicide)))
;;If AA is 3, explode.
(if (< p5_e 180) (begin (set perform_biped p5) (perform_double_shield) (set p5_e (+ p5_e 180))))
;;if AA is 4 and has 180 energy, double shield.
))))
)(begin ;;If death, free p5 variable and such.
(set p5_e 0) ;;Clear energy
(set p5_aa 0) ;;Clear AA
(set p5 none) ;;Free p5 Variable
))
)
(script continuous p6_usage
(sleep_until (!= p6 none) 15) ;;Wait until p6 is assigned
(sleep_until (or (unit_get_current_flashlight_state p6) (<= (unit_get_health p6) 0)) 5) ;;Wait until flashlight, or death.
(if (unit_get_current_flashlight_state p6) (begin ;;If flashlight
(unit_set_desired_flashlight_state p6 0) ;;Turn off flashlight
(sleep_until (= perform_biped none) 1) ;;Wait until the global for performing AA is open
(if (= p6_aa 0) (if (< p6_e 180) (begin (set perform_biped p6) (perform_shield2health) (set p6_e (+ p6_e 180))))))
;;If AA is 0 (Shield to Health), and has 180 energy, preform it and use up 180 energy.
(if (= p6_aa 1) (if (< p6_e 240) (begin (set perform_biped p6) (perform_camo) (set p6_e (+ p6_e 120))))))
;;If AA is 1 (Camo), and has 120 energy, preform it and use 120 energy.
(if (= p6_aa 2) (if (< p6_e 270) (begin (set perform_biped p6) (perform_random_tele) (set p6_e (+ p6_e 90))))))
;;If AA is 2 (Tele), and has 90 energy, perform.
(if (= p6_aa 3) (begin (set perform_biped p6) (perform_suicide)))
;;If AA is 3, explode.
(if (< p6_e 180) (begin (set perform_biped p6) (perform_double_shield) (set p6_e (+ p6_e 180))))
;;if AA is 4 and has 180 energy, double shield.
))))
)(begin ;;If death, free p6 variable and such.
(set p6_e 0) ;;Clear energy
(set p6_aa 0) ;;Clear AA
(set p6 none) ;;Free p6 Variable
))
)
(script continuous p7_usage
(sleep_until (!= p7 none) 15) ;;Wait until p7 is assigned
(sleep_until (or (unit_get_current_flashlight_state p7) (<= (unit_get_health p7) 0)) 5) ;;Wait until flashlight, or death.
(if (unit_get_current_flashlight_state p7) (begin ;;If flashlight
(unit_set_desired_flashlight_state p7 0) ;;Turn off flashlight
(sleep_until (= perform_biped none) 1) ;;Wait until the global for performing AA is open
(if (= p7_aa 0) (if (< p7_e 180) (begin (set perform_biped p7) (perform_shield2health) (set p7_e (+ p7_e 180))))))
;;If AA is 0 (Shield to Health), and has 180 energy, preform it and use up 180 energy.
(if (= p7_aa 1) (if (< p7_e 240) (begin (set perform_biped p7) (perform_camo) (set p7_e (+ p7_e 120))))))
;;If AA is 1 (Camo), and has 120 energy, preform it and use 120 energy.
(if (= p7_aa 2) (if (< p7_e 270) (begin (set perform_biped p7) (perform_random_tele) (set p7_e (+ p7_e 90))))))
;;If AA is 2 (Tele), and has 90 energy, perform.
(if (= p7_aa 3) (begin (set perform_biped p7) (perform_suicide)))
;;If AA is 3, explode.
(if (< p7_e 180) (begin (set perform_biped p7) (perform_double_shield) (set p7_e (+ p7_e 180))))
;;if AA is 4 and has 180 energy, double shield.
))))
)(begin ;;If death, free p7 variable and such.
(set p7_e 0) ;;Clear energy
(set p7_aa 0) ;;Clear AA
(set p7 none) ;;Free p7 Variable
))
)
(script continuous p8_usage
(sleep_until (!= p8 none) 15) ;;Wait until p8 is assigned

```

```

(sleep_until (or (unit_get_current_flashlight_state p8) (<= (unit_get_health p8) 0)) 5) ;;Wait until flashlight, or death.
(if (unit_get_current_flashlight_state p8) (begin ;;If flashlight
(unit_set_desired_flashlight_state p8 0) ;;Turn off flashlight
(sleep_until (= perform_biped none) 1) ;;Wait until the global for performing AA is open
(if (= p8_aa 0) (if (< p8_e 180) (begin (set perform_biped p8) (perform_shield2health) (set p8_e (+ p8_e 180))))
;;If AA is 0 (Shield to Health), and has 180 energy, preform it and use up 180 energy.
(if (= p8_aa 1) (if (< p8_e 240) (begin (set perform_biped p8) (perform_camo) (set p8_e (+ p8_e 120))))
;;If AA is 1 (Camo), and has 120 energy, preform it and use 120 energy.
(if (= p8_aa 2) (if (< p8_e 270) (begin (set perform_biped p8) (perform_random_tele) (set p8_e (+ p8_e 90))))
;;If AA is 2 (Tele), and has 90 energy, perform.
(if (= p8_aa 3) (begin (set perform_biped p8) (perform_suicide))
;;If AA is 3, explode.
(if (< p8_e 180) (begin (set perform_biped p8) (perform_double_shield) (set p8_e (+ p8_e 180))))
;;if AA is 4 and has 180 energy, double shield.
))))
)(begin ;;If death, free p8 variable and such.
(set p8_e 0) ;;Clear energy
(set p8_aa 0) ;;Clear AA
(set p8 none) ;;Free p8 Variable
))
)
(script continuous p9_usage
(sleep_until (!= p9 none) 15) ;;Wait until p9 is assigned
(sleep_until (or (unit_get_current_flashlight_state p9) (<= (unit_get_health p9) 0)) 5) ;;Wait until flashlight, or death.
(if (unit_get_current_flashlight_state p9) (begin ;;If flashlight
(unit_set_desired_flashlight_state p9 0) ;;Turn off flashlight
(sleep_until (= perform_biped none) 1) ;;Wait until the global for performing AA is open
(if (= p9_aa 0) (if (< p9_e 180) (begin (set perform_biped p9) (perform_shield2health) (set p9_e (+ p9_e 180))))
;;If AA is 0 (Shield to Health), and has 180 energy, preform it and use up 180 energy.
(if (= p9_aa 1) (if (< p9_e 240) (begin (set perform_biped p9) (perform_camo) (set p9_e (+ p9_e 120))))
;;If AA is 1 (Camo), and has 120 energy, preform it and use 120 energy.
(if (= p9_aa 2) (if (< p9_e 270) (begin (set perform_biped p9) (perform_random_tele) (set p9_e (+ p9_e 90))))
;;If AA is 2 (Tele), and has 90 energy, perform.
(if (= p9_aa 3) (begin (set perform_biped p9) (perform_suicide))
;;If AA is 3, explode.
(if (< p9_e 180) (begin (set perform_biped p9) (perform_double_shield) (set p9_e (+ p9_e 180))))
;;if AA is 4 and has 180 energy, double shield.
))))
)(begin ;;If death, free p9 variable and such.
(set p9_e 0) ;;Clear energy
(set p9_aa 0) ;;Clear AA
(set p9 none) ;;Free p9 Variable
))
)
(script continuous p10_usage
(sleep_until (!= p10 none) 15) ;;Wait until p10 is assigned
(sleep_until (or (unit_get_current_flashlight_state p10) (<= (unit_get_health p10) 0)) 5) ;;Wait until flashlight, or death.
(if (unit_get_current_flashlight_state p10) (begin ;;If flashlight
(unit_set_desired_flashlight_state p10 0) ;;Turn off flashlight
(sleep_until (= perform_biped none) 1) ;;Wait until the global for performing AA is open
(if (= p10_aa 0) (if (< p10_e 180) (begin (set perform_biped p10) (perform_shield2health) (set p10_e (+ p10_e 180))))
;;If AA is 0 (Shield to Health), and has 180 energy, preform it and use up 180 energy.
(if (= p10_aa 1) (if (< p10_e 240) (begin (set perform_biped p10) (perform_camo) (set p10_e (+ p10_e 120))))
;;If AA is 1 (Camo), and has 120 energy, preform it and use 120 energy.
(if (= p10_aa 2) (if (< p10_e 270) (begin (set perform_biped p10) (perform_random_tele) (set p10_e (+ p10_e 90))))
;;If AA is 2 (Tele), and has 90 energy, perform.
(if (= p10_aa 3) (begin (set perform_biped p10) (perform_suicide))
;;If AA is 3, explode.
(if (< p10_e 180) (begin (set perform_biped p10) (perform_double_shield) (set p10_e (+ p10_e 180))))
;;if AA is 4 and has 180 energy, double shield.
))))
)(begin ;;If death, free p10 variable and such.

```

```

(set p10_e 0) ;;Clear energy
(set p10_aa 0) ;;Clear AA
(set p10 none) ;;Free p10 Variable
))
)
(script continuous p11_usage
(sleep_until (!= p11 none) 15) ;;Wait until p11 is assigned
(sleep_until (or (unit_get_current_flashlight_state p11) (<= (unit_get_health p11) 0)) 5) ;;Wait until flashlight, or death.
(if (unit_get_current_flashlight_state p11) (begin ;;If flashlight
(unit_set_desired_flashlight_state p11 0) ;;Turn off flashlight
(sleep_until (= perform_biped none) 1) ;;Wait until the global for performing AA is open
(if (= p11_aa 0) (if (< p11_e 180) (begin (set perform_biped p11) (perform_shield2health) (set p11_e (+ p11_e 180))))
;;If AA is 0 (Shield to Health), and has 180 energy, perform it and use up 180 energy.
(if (= p11_aa 1) (if (< p11_e 240) (begin (set perform_biped p11) (perform_camo) (set p11_e (+ p11_e 120))))
;;If AA is 1 (Camo), and has 120 energy, perform it and use 120 energy.
(if (= p11_aa 2) (if (< p11_e 270) (begin (set perform_biped p11) (perform_random_tele) (set p11_e (+ p11_e 90))))
;;If AA is 2 (Tele), and has 90 energy, perform.
(if (= p11_aa 3) (begin (set perform_biped p11) (perform_suicide))
;;If AA is 3, explode.
(if (< p11_e 180) (begin (set perform_biped p11) (perform_double_shield) (set p11_e (+ p11_e 180))))
;;if AA is 4 and has 180 energy, double shield.
))))
)(begin ;;If death, free p11 variable and such.
(set p11_e 0) ;;Clear energy
(set p11_aa 0) ;;Clear AA
(set p11 none) ;;Free p11 Variable
))
)
(script continuous p12_usage
(sleep_until (!= p12 none) 15) ;;Wait until p12 is assigned
(sleep_until (or (unit_get_current_flashlight_state p12) (<= (unit_get_health p12) 0)) 5) ;;Wait until flashlight, or death.
(if (unit_get_current_flashlight_state p12) (begin ;;If flashlight
(unit_set_desired_flashlight_state p12 0) ;;Turn off flashlight
(sleep_until (= perform_biped none) 1) ;;Wait until the global for performing AA is open
(if (= p12_aa 0) (if (< p12_e 180) (begin (set perform_biped p12) (perform_shield2health) (set p12_e (+ p12_e 180))))
;;If AA is 0 (Shield to Health), and has 180 energy, perform it and use up 180 energy.
(if (= p12_aa 1) (if (< p12_e 240) (begin (set perform_biped p12) (perform_camo) (set p12_e (+ p12_e 120))))
;;If AA is 1 (Camo), and has 120 energy, perform it and use 120 energy.
(if (= p12_aa 2) (if (< p12_e 270) (begin (set perform_biped p12) (perform_random_tele) (set p12_e (+ p12_e 90))))
;;If AA is 2 (Tele), and has 90 energy, perform.
(if (= p12_aa 3) (begin (set perform_biped p12) (perform_suicide))
;;If AA is 3, explode.
(if (< p12_e 180) (begin (set perform_biped p12) (perform_double_shield) (set p12_e (+ p12_e 180))))
;;if AA is 4 and has 180 energy, double shield.
))))
)(begin ;;If death, free p12 variable and such.
(set p12_e 0) ;;Clear energy
(set p12_aa 0) ;;Clear AA
(set p12 none) ;;Free p12 Variable
))
)
(script continuous p13_usage
(sleep_until (!= p13 none) 15) ;;Wait until p13 is assigned
(sleep_until (or (unit_get_current_flashlight_state p13) (<= (unit_get_health p13) 0)) 5) ;;Wait until flashlight, or death.
(if (unit_get_current_flashlight_state p13) (begin ;;If flashlight
(unit_set_desired_flashlight_state p13 0) ;;Turn off flashlight
(sleep_until (= perform_biped none) 1) ;;Wait until the global for performing AA is open
(if (= p13_aa 0) (if (< p13_e 180) (begin (set perform_biped p13) (perform_shield2health) (set p13_e (+ p13_e 180))))
;;If AA is 0 (Shield to Health), and has 180 energy, perform it and use up 180 energy.
(if (= p13_aa 1) (if (< p13_e 240) (begin (set perform_biped p13) (perform_camo) (set p13_e (+ p13_e 120))))
;;If AA is 1 (Camo), and has 120 energy, perform it and use 120 energy.
(if (= p13_aa 2) (if (< p13_e 270) (begin (set perform_biped p13) (perform_random_tele) (set p13_e (+ p13_e 90))))

```

```

;;If AA is 2 (Tele), and has 90 energy, perform.
(if (= p13_aa 3) (begin (set perform_biped p13) (perform_suicide))
;;If AA is 3, explode.
(if (< p13_e 180) (begin (set perform_biped p13) (perform_double_shield) (set p13_e (+ p13_e 180))))
;;if AA is 4 and has 180 energy, double shield.
)))))
)(begin ;;If death, free p13 variable and such.
(set p13_e 0) ;;Clear energy
(set p13_aa 0) ;;Clear AA
(set p13 none) ;;Free p13 Variable
))
)
(script continuous p14_usage
(sleep_until (!= p14 none) 15) ;;Wait until p14 is assigned
(sleep_until (or (unit_get_current_flashlight_state p14) (<= (unit_get_health p14) 0)) 5) ;;Wait until flashlight, or death.
(if (unit_get_current_flashlight_state p14) (begin ;;If flashlight
(unit_set_desired_flashlight_state p14 0) ;;Turn off flashlight
(sleep_until (= perform_biped none) 1) ;;Wait until the global for performing AA is open
(if (= p14_aa 0) (if (< p14_e 180) (begin (set perform_biped p14) (perform_shield2health) (set p14_e (+ p14_e 180))))
;;If AA is 0 (Shield to Health), and has 180 energy, perform it and use up 180 energy.
(if (= p14_aa 1) (if (< p14_e 240) (begin (set perform_biped p14) (perform_camo) (set p14_e (+ p14_e 120))))
;;If AA is 1 (Camo), and has 120 energy, perform it and use 120 energy.
(if (= p14_aa 2) (if (< p14_e 270) (begin (set perform_biped p14) (perform_random_tele) (set p14_e (+ p14_e 90))))
;;If AA is 2 (Tele), and has 90 energy, perform.
(if (= p14_aa 3) (begin (set perform_biped p14) (perform_suicide))
;;If AA is 3, explode.
(if (< p14_e 180) (begin (set perform_biped p14) (perform_double_shield) (set p14_e (+ p14_e 180))))
;;if AA is 4 and has 180 energy, double shield.
)))))
)(begin ;;If death, free p14 variable and such.
(set p14_e 0) ;;Clear energy
(set p14_aa 0) ;;Clear AA
(set p14 none) ;;Free p14 Variable
))
)
(script continuous p15_usage
(sleep_until (!= p15 none) 15) ;;Wait until p15 is assigned
(sleep_until (or (unit_get_current_flashlight_state p15) (<= (unit_get_health p15) 0)) 5) ;;Wait until flashlight, or death.
(if (unit_get_current_flashlight_state p15) (begin ;;If flashlight
(unit_set_desired_flashlight_state p15 0) ;;Turn off flashlight
(sleep_until (= perform_biped none) 1) ;;Wait until the global for performing AA is open
(if (= p15_aa 0) (if (< p15_e 180) (begin (set perform_biped p15) (perform_shield2health) (set p15_e (+ p15_e 180))))
;;If AA is 0 (Shield to Health), and has 180 energy, perform it and use up 180 energy.
(if (= p15_aa 1) (if (< p15_e 240) (begin (set perform_biped p15) (perform_camo) (set p15_e (+ p15_e 120))))
;;If AA is 1 (Camo), and has 120 energy, perform it and use 120 energy.
(if (= p15_aa 2) (if (< p15_e 270) (begin (set perform_biped p15) (perform_random_tele) (set p15_e (+ p15_e 90))))
;;If AA is 2 (Tele), and has 90 energy, perform.
(if (= p15_aa 3) (begin (set perform_biped p15) (perform_suicide))
;;If AA is 3, explode.
(if (< p15_e 180) (begin (set perform_biped p15) (perform_double_shield) (set p15_e (+ p15_e 180))))
;;if AA is 4 and has 180 energy, double shield.
)))))
)(begin ;;If death, free p15 variable and such.
(set p15_e 0) ;;Clear energy
(set p15_aa 0) ;;Clear AA
(set p15 none) ;;Free p15 Variable
))
)
(global vehicle aa_veh NONE)
(global short aa_veh_type 0)
(global vehicle aa_swapper NONE)

```

```

(global short swap_type 0)
(global unit rec_player NONE)
(script static void check_n_swap_aa
(if (vehicle_test_seat_list aa_veh "aa-driver" (players)) (begin
(if (vehicle_test_seat_list aa_veh "aa-driver" p0)
(begin (set swap_type p0_aa) (set p0_aa aa_veh_type) (set rec_player p0))
(if (vehicle_test_seat_list aa_veh "aa-driver" p1)
(begin (set swap_type p1_aa) (set p1_aa aa_veh_type) (set rec_player p1))
(if (vehicle_test_seat_list aa_veh "aa-driver" p2)
(begin (set swap_type p2_aa) (set p2_aa aa_veh_type) (set rec_player p2))
(if (vehicle_test_seat_list aa_veh "aa-driver" p3)
(begin (set swap_type p3_aa) (set p3_aa aa_veh_type) (set rec_player p3))
(if (vehicle_test_seat_list aa_veh "aa-driver" p4)
(begin (set swap_type p4_aa) (set p4_aa aa_veh_type) (set rec_player p4))
(if (vehicle_test_seat_list aa_veh "aa-driver" p5)
(begin (set swap_type p5_aa) (set p5_aa aa_veh_type) (set rec_player p5))
(if (vehicle_test_seat_list aa_veh "aa-driver" p6)
(begin (set swap_type p6_aa) (set p6_aa aa_veh_type) (set rec_player p6))
(if (vehicle_test_seat_list aa_veh "aa-driver" p7)
(begin (set swap_type p7_aa) (set p7_aa aa_veh_type) (set rec_player p7))
(if (vehicle_test_seat_list aa_veh "aa-driver" p8)
(begin (set swap_type p8_aa) (set p8_aa aa_veh_type) (set rec_player p8))
(if (vehicle_test_seat_list aa_veh "aa-driver" p9)
(begin (set swap_type p9_aa) (set p9_aa aa_veh_type) (set rec_player p9))
(if (vehicle_test_seat_list aa_veh "aa-driver" p10)
(begin (set swap_type p10_aa) (set p10_aa aa_veh_type) (set rec_player p10))
(if (vehicle_test_seat_list aa_veh "aa-driver" p11)
(begin (set swap_type p11_aa) (set p11_aa aa_veh_type) (set rec_player p11))
(if (vehicle_test_seat_list aa_veh "aa-driver" p12)
(begin (set swap_type p12_aa) (set p12_aa aa_veh_type) (set rec_player p12))
(if (vehicle_test_seat_list aa_veh "aa-driver" p13)
(begin (set swap_type p13_aa) (set p13_aa aa_veh_type) (set rec_player p13))
(if (vehicle_test_seat_list aa_veh "aa-driver" p14)
(begin (set swap_type p14_aa) (set p14_aa aa_veh_type) (set rec_player p14))
(if (vehicle_test_seat_list aa_veh "aa-driver" p15)
(begin (set swap_type p15_aa) (set p15_aa aa_veh_type) (set rec_player p15))
)))))))))))))
(if (= swap_type 0)
(if (volume_test_object storage_vol aa_health_shield_00)
(set aa_swapper aa_health_shield_00)
(if (volume_test_object storage_vol aa_health_shield_01)
(set aa_swapper aa_health_shield_01)
(if (volume_test_object storage_vol aa_health_shield_02)
(set aa_swapper aa_health_shield_02)
)))
))
(if (= swap_type 1)
(if (volume_test_object storage_vol aa_camo_00) (set aa_swapper aa_camo_00)
(if (volume_test_object storage_vol aa_camo_01) (set aa_swapper aa_camo_01)
(if (volume_test_object storage_vol aa_camo_02) (set aa_swapper aa_camo_02)
)))
))
(if (= swap_type 2)
(if (volume_test_object storage_vol aa_tele_00) (set aa_swapper aa_tele_00)
(if (volume_test_object storage_vol aa_tele_01) (set aa_swapper aa_tele_01)
(if (volume_test_object storage_vol aa_tele_02) (set aa_swapper aa_tele_02)
)))
))
(if (= swap_type 3)
(if (volume_test_object storage_vol aa_suicide_00) (set aa_swapper aa_suicide_00)
(if (volume_test_object storage_vol aa_suicide_01) (set aa_swapper aa_suicide_01)
(if (volume_test_object storage_vol aa_suicide_02) (set aa_swapper aa_suicide_02)
)))
))
(if (= swap_type 4)
(if (volume_test_object storage_vol aa_double_shield_00)

```

```

                (set aa_swapper aa_double_shield_00)
            (if (volume_test_object storage_vol aa_double_shield_01)
                (set aa_swapper aa_double_shield_01)
            (if (volume_test_object storage_vol aa_double_shield_02)
                (set aa_swapper aa_double_shield_02)
            )))
        ))))
    ;;;;
    ;; Common Route (Swapping locations)
    (vehicle_unload aa_veh "")
    (sleep_until (not (vehicle_test_seat_list aa_veh "aa-driver" (players))) 15)
    (objects_attach aa_veh "" aa_swapper "")
    (objects_detach aa_veh aa_swapper)
    (object_teleport aa_veh storage_flag)
    ;;
    ;;;;
    ))
)
(script continuous swapping_aa
(set aa_veh_type 0)
(set aa_veh aa_health_shield_00)
(check_n_swap_aa)
(set aa_veh aa_health_shield_01)
(check_n_swap_aa)
(set aa_veh aa_health_shield_02)
(check_n_swap_aa)
(set aa_veh_type 1)
(set aa_veh aa_camo_00)
(check_n_swap_aa)
(set aa_veh aa_camo_01)
(check_n_swap_aa)
(set aa_veh aa_camo_02)
(check_n_swap_aa)
(set aa_veh_type 2)
(set aa_veh aa_tele_00)
(check_n_swap_aa)
(set aa_veh aa_tele_01)
(check_n_swap_aa)
(set aa_veh aa_tele_02)
(check_n_swap_aa)
(set aa_veh_type 3)
(set aa_veh aa_suicide_00)
(check_n_swap_aa)
(set aa_veh aa_suicide_01)
(check_n_swap_aa)
(set aa_veh aa_suicide_02)
(check_n_swap_aa)
(set aa_veh_type 4)
(set aa_veh aa_double_shield_00)
(check_n_swap_aa)
(set aa_veh aa_double_shield_01)
(check_n_swap_aa)
(set aa_veh aa_double_shield_02)
(check_n_swap_aa)
;;<Moar boar wants MOAR>
)

(script startup clients_unneeded
(sleep 10)
(if (not (machine_is_host)) (begin
(sleep -1 recharge_energy)

```

```

(sleep -1 p0_usage)
(sleep -1 p1_usage)
(sleep -1 p2_usage)
(sleep -1 p3_usage)
(sleep -1 p4_usage)
(sleep -1 p5_usage)
(sleep -1 p6_usage)
(sleep -1 p7_usage)
(sleep -1 p8_usage)
(sleep -1 p9_usage)
(sleep -1 p10_usage)
(sleep -1 p11_usage)
(sleep -1 p12_usage)
(sleep -1 p13_usage)
(sleep -1 p14_usage)
(sleep -1 p15_usage)
(sleep -1 swapping_aa)
)
)

```

```

;;;;;;;;;;;;;
;; OpenSauce Only Additions!
(script static unit local_player (unit (player_data_get_object (player_local_get) slave_unit)))
(global unit loc_player NONE)
(global short local_ability 0)
(global short local_energy 0)
(global unit current_vehicle NONE)
(global short vehicle_id -1)

```

```

(script continuous reset_on_death
(sleep_until (!= (local_player) loc_player) 10)
(set local_ability 0)
(set local_energy 0)
(set loc_player (local_player))
)

```

```

(script static void identify_vehicle

```

```

;;;;;;;;;;;;;
;;-1 - unset
;;0 - Unknown (most likely a real vehicle)
;;# - INSERT NAME
;;;;;;;;;;;;;
;;(sleep 30) ;;Make sure you have a very short enter animation (1 tick), otherwise you'll need to uncomment and edit this.
;;(if (and (> (objects_distance_to_object (local_player) current_vehicle) #####) (< (objects_distance_to_object (local_player)
current_vehicle) #####)) (set vehicle_id 1)
;;enter vehicle, and use objects_distance_to_object with dev. Enter a range above and below, that is unique to this vehicle (eg, dont
place the driver seats in the same places for these AA)
;;Uncomment this stuff if inuse.
(set vehicle_id 0)
;;)
) ;;This can be used alongside boosting. This block is basically the boost, except I ignored default vehicles.

```

```

(script continuous select_vehicle
(sleep_until (and (!= (objects_distance_to_object (local_player) (player_data_get_object (player_local_get) nearest_object)) -1) (=
(unit_data_get_real (unit (player_data_get_object (player_local_get) nearest_object)) driver_power) 0)) 1)
(set current_vehicle (unit (player_data_get_object (player_local_get) nearest_object)))
)

```

```

(script continuous vehicle_boost_check
(sleep_until (and (>= (unit_data_get_real current_vehicle driver_power) 0.5) (< (objects_distance_to_object (local_player)
current_vehicle) 1)) 5)
(identify_vehicle)
(sleep 1)

```

```

(if (!= vehicle_id 0) (begin (set local_ability vehicle_id)
;;Add something else in for aa 0, like if its 20
(unit_exit_vehicle (local_player))
(vehicle_unload current_vehicle ""))
)
(sleep_until (not (and (>= (unit_data_get_real current_vehicle driver_power) 1) (< (objects_distance_to_object (players)
current_vehicle) 1)))) 5)
(set vehicle_id -1)
)

(script continuous local_recharge_energy
(if (> local_energy 0) (set local_energy (- local_energy 1))) ;;If energy hasnt returned to 0, decrement it.
(sleep 10) ;;Recharge 3 a second.
)

(script continuous local_usage
(sleep_until (unit_get_current_flashlight_state (local_player)) 1)
(if (= local_ability 0) (if (< local_energy 180) (set local_energy (+ local_energy 180)))
;;If AA is 0 and has 180 energy, use up 180 energy.
(if (= local_ability 1) (if (< local_energy 240) (set local_energy (+ local_energy 120)))
;;If AA is 1 (Camo), and has 120 energy, use 120 energy.
(if (= local_ability 2) (if (< local_energy 270) (set local_energy (+ local_energy 90)))
;;If AA is 2 (Tele), and has 90 energy
(if (= local_ability 4) (if (< local_energy 180) (set local_energy (+ local_energy 180)))
;;if AA is 4 and has 180 energy
))))
(sleep 10)
)

;;;
;; Player now has the ability tracked... Up to you decide on what to do with it now.
(script continuous printing_info
(set developer_mode 127)
(cls)
(if (and (= local_ability 0) (< local_energy 180)) (print "ability usable [shield -> health]")
(if (and (= local_ability 1) (< local_energy 240)) (print "ability usable [camo]")
(if (and (= local_ability 2) (< local_energy 270)) (print "ability usable [random teleport]")
(if (= local_ability 3) (print "ability usable [self destruct]")
(if (and (= local_ability 4) (< local_energy 180)) (print "ability usable [double shields]")
(print "ability disabled [not enough energy]")
))))))
(print " ")
(print " ")
(inspect local_energy)
(print "-----")
(print "360")
(sleep 15)
)

```