ARTIFICE Construct

Public Release

Terminology has been deliberately obfuscated in this document. Certain keywords have been substituted. Full documentation will (most likely) not be made publicly available

1 What is a Construct?

A construct is a network of people and locations. Each person is given several characteristics, including but not limited to height, weight, date of birth, subgroup, residency, and profession.

1.1 Classifying a Person

Every person belongs to a class. There are four classes - **Main, Secondary, Support, Outsider**. We are primarily concerned with the main group, as they have the most groupings and characteristics.

1.1.1 Main Class Sub-Groupings

Within the Main class, each person belongs to one of 4 groups: A, X, S, or H; the H group stands for Hybrid, where they share characteristics of 2 or more of the A, X, or S groups. Each person also has a "Level", which is a letter from A to H or the number 0, determined by the 14-byte string.

Each Main class person also has 14 or 21 numbers associated with it, which we will explore in greater detail later.

1.2 Terminology Used

- 14-string / 21-string: Describes the specific characteristics of a Main class person.
- Level: 0 or A - H.
- Group: A, X, S, or H.
- Mode: The current state of a main class person. Can be one of: **Zone, Chair, Flat, Struct**.

2 Class and Levels

As mentioned above, each main class person has a "level" in [0A-H] and a class out of A, X, S, and H. The following rules exist for creating people:

1. Let X-Y denote a person with level X and class Y. For example, G-S would denote a person with level G, class S.
2. Class A people may only have levels 0 and B.
3. Class S people cannot have level 0.

2.1 A Class

A-class people can be described by a 4 character string, where each character is one of A, B, C, or 0. We call this person **symmetric** if str[0] == str[1] and str[2] == str[3]. For example, "BB00" would be symmetric, while "AA0C" would not be.

2.1.1 Symmetric conversion to 14-string

The first character determines the first 3 characters of the string. We have:

- 0: First 3 characters are 000
- C: First 3 characters are 200
- B: First 3 characters are 400
- A: First 3 characters are 440

The second character determines characters 5 through 8. We have:

- 0: 0000
- C: 4000
- B: 4400
- A: 4442

The rest of the characters are to be filled with 0. For example, "BBAA" would be converted to 4000-44420-000-00.

2.1.2 Asymmetric conversion to 14-string

For an asymmetric A-classer, the 13th character will be 2. Characters 1-3 and 5-8 would be determined by str[0] and str[2]. Then following the 14th character, 7 more characters would be appended, determined by str[1] and str[3].

For example, "BABC" would be converted to 4000-44000-000-20-4404000.

In the event that one side is completely 0 (example "B0B0"), we make the 13th character 1 and stick to the 14-string.

2.2 S Class

The S-Class is a bit more rigid. There are few set archetypes, and a little bit of variation within the archetypes. The available archetypes are:

- T1 through T12 (for example, T8 is valid, but T13 is not).
- C1 through C7
- 1A, 1B, 1C, 2A, or 2B
- CX11, CX12, or CX5
- C1V, C1VCX5, or C1-H

We also assign each person a "seed" number from 0 to 9. Let X.y denote level X with seed y; for example, T8.2 would denote T8 with seed 2.

And the corresponding 14-strings are:

- T8-T12 and T7.0-4: 4442-00000-000-00
- T7.5-9, T6: 4443-00000-000-00
- T2-T5: 4444-00000-000-00
- T1, C7.0-1: 4444-10000-000-00
- C7.2-6: 4444-20000-000-00
- C7.7-9: 4444-20100-000-00
- C6:
  - C6.0: 4444-21101-000-00
  - C6.1: 4444-31101-000-00
  - C6.2: 4444-41101-000-00
  - C6.3: 4444-21202-000-00
  - C6.4-5: 4444-31202-000-00
  - C6.6: 4444-41202-000-00
  - C6.7: 4444-21302-000-00
  - C6.8: 4444-31302-000-00
  - C6.9: 4444-41302-000-00
- C5:
  - C5.0-1: 4444-41312-000-00
  - C5.2-4: 4444-42312-000-00
  - C5.5-7: 4444-41313-000-00

    - C5.8-9: 4444-42313-000-00

- C4, C3.0-3: 4444-44444-000-00
- C3.4-9: 4444-44444-100-00
- C2, C1.0-5: 4444-44444-200-00
- C1.6-8: 4444-44444-300-00
- C1.9: 4444-44444-301-00
- 2B: 4444-12333-000-00
- 2A.0-4: 4444-11344-100-00
- 2A.5-9: 4444-12344-100-00
- 1C.0-4: 4444-44444-012-00
- 1C.5-9: 4444-44444-112-00
- 1B.0: 4444-44444-212-00
- 1B.1-4: 4444-44444-312-00
- 1B.5-9: 4444-44444-412-00
- 1A.0-2: 4444-44444-413-00
- 1A.3-9: 4444-44444-414-00
- CX11.0-7: 4444-44444-400-00
- CX11.8-9: 4444-44444-401-00
- CX12.0-5: 4444-44444-302-00
- CX12.6-9: 4444-44444-402-00
- CX5.0-4: 4444-44444-303-00
- CX5.5-9: 4444-44444-403-00
- C1V.0-4: 4444-44444-210-00
- C1V.5-8: 4444-44444-310-00
- C1V.9: 4444-44444-410-00
- C1VCX5.0-1: 4444-44444-313-00
- C1VCX5.2-9: 4444-44444-413-00
- C1H: 4444-44444-404-00

2.3 X Classes

We can describe an X class person with 7 components. Call these components **G, M, A, H, F, V, X**. These components take on the following values:

- G: 1, 2a, 2b, 3a, 3b, 4a, 4b, 5a, 5b, 5c
- M: 1, 2a, 2b, 3a, 3b, 3c, 4a, 4b, 5
- A: 1, 2, 3, 4a, 4b, 4c, 5a, 5b, 5c
- H: 1, 2, 3, 4, 5
- F: 1, 2, 3-, 3, 4, 5
- V: 0, 1
- X: 0, 1, 2

The values are assigned according to the following rules:

- G to M rules
  - If G is less than or equal to 4a, then MA must be 11.
  - If G is 4b, M must be at most 4b.
  - If G is 5a, M must be at most 3a.
  - If G is 5b, M must be from 3b to 4b.
  - If G is 5c, M must be at least 4a.
- M to A rules
  - If M is 1, A must be 1 or 2.
  - If M is 2a or 2b, A must be 1, 2, or 3.
  - If M is in 3a-3c, A must be in 3-4c.
  - If M is 4a, A must be 4a or 4b.
  - If M is 4b, A must be in 4a-5c.
  - If M is 5, A must be 5c.
- HFVX must be 1100 unless A is 5c.

We define a X-class person as **lower** if GMA is not 5c55c or if their HFV is equal to 110. We define them as **upper** otherwise.

Like the S-class, we assign a seed with values from 0 - 9. We assign values to the 14-string as follows, for **Lower** X-Classes:

- G Values: First 4 numbers
  - 1: 1000

  - 2a: 2000
  - 2b: 3200
  - 3a: 4310
  - 3b: 4420
  - 4a: 4440
  - 4b: 4441
  - 5a: 4442
  - 5b: 4443
  - 5c: 4444
- M Values: Numbers 5 and 6
  - 1: 00
  - 2a.0-4: 10
  - 2a.5-9: 20
  - 2b.0-2: 01
  - 2b.3-6: 11
  - 2b.7-9: 21
  - 3a.0-4: 22
  - 3a.5-9: 23
  - 3b: 40
  - 3c: 42
  - 4a.0-4: 33
  - 4a.5-9: 34
  - 4b: 43
  - 5: 44
- A Values: Numbers 7,8,9
  - 1: 000
  - 2: 001
  - 3: 111
  - 4a: 122
  - 4b: 123
  - 4c: 223
  - 5a: 333
  - 5b: 334
  - 5c: 444

Now, we consider upper X-classes. They have the following archetypes:

- F = 1
  - Type 11: 2D, 2x1D
  - Type 21: 2D, 1D
- F = 2
  - Type 12: 2D, 2x1D
  - Type 22: 2D, 1D
  - Type 32: 1D + 1
  - Type 42: 1D
- F = 3- or 3
  - Type 33: 1D + 1
  - Type 43: 1D
  - Type 53: S
- F = 4
  - Type 44: 1D
  - Type 54: S
- F = 5
  - Type 45: Roll
  - Type 55: H

See upperx.txt for detailed information on all the seeds.

Additionally, note that for upper X-group archetypes, they can be endowed with an 8th descriptor, one of A or C. They can also not have an 8th descriptor at all. Note that the C descriptor is valid only for the types 42, 53, 44, 45, 54, and 55.

2.4 Hybrid Groups

2.4.1 A + S Hybrids

We describe an A + S hybrid with both the A 4 character string and the S descriptor. There are some caveats, though. For the A 4-character string, only the last 2 characters matter; "AA0C" is functionally equivalent to "B00C". For the S descriptor, we want to pick something between T1 to T12 or C5 to C7.

Then we construct the 14/21-string as follows:

- First 4 characters determined by the S descriptor.
- Next 4 characters determined by the maximum of the S and the A descriptors.
- Last x characters should be all 0.

For example, suppose we had a hybrid with descriptors "00AB" and "T3". Then we would construct 21-string 4442-44420-000-20-4444400.

2.4.2 A + X Hybrids

Similarly to above, we use the maximum of the A and the S descriptors. We also stick to lower X-classes rather than upper X-classes, because adding an A descriptor to an upper X class will not fundamentally change the 14 or 21-string.

For example, if we combined a GMA 5b3a4b with "00B0" and seed 3, we would have 21-string 4443-44123-000-20-4442212.

2.4.3 S + X Hybrids / A + S + X Hybrids

These are rare and should probably be avoided. In the event that you do want to use them, however, the same behavior applies with combining maximums. Stick to combining T1-T12/C5-C7 and lower X groups.

2.5 Enhancements

All A groupers, and A + S hybrids (excluding ASX hybrids), are eligible for enhancements. These enhancements change the 14 or 21-byte string, but are only active when the mode is **Chair**.

- First 2 characters:
  - 0: 000 $\rightarrow$ 000 (unchanged)
  - C: 200 $\rightarrow$ 000
  - B: 400 $\rightarrow$ 000
  - A: 440 $\rightarrow$ 310
- Last 2 characters:
  - 0: 0000 $\rightarrow$ 0000 (unchanged)
  - C: 4000 $\rightarrow$ 0000
  - B: 4400 $\rightarrow$ 2000
  - A: 4442 $\rightarrow$ 2400

Certain S and X groupers are also eligible for enhancements. Provided that $H \leq 3$, the enhancement transforms their EF to 24. Obviously this is not beneficial if the existing EF is 21, for example.

3 Key Attributes from Class and Group

I preface this section by stating that will base our calculations off of the enhanced version of any A/S-groupers, unless stated otherwise. We will also refer to the 14 character string as:

$$ABCD - EFGHI - JKL - MN - (A_2B_2C_2D_2E_2F_2G_2)$$

We also use Boolean logic.

3.1 Routine

Each person must go through a "Routine" twice a day. Some combination of the following aspects comprise a routine:

- Unhanced to Enhanced

  $D \leq 1 + \Big((E_1 = 0 + F_1 = 0 + G_1 = 0)||(E_2 = 0 + F_2 = 0 + G_2 = 0)\Big)$

- ChangeU

  $$(E \leq 2||F \leq 1) + G \leq 2 + (G \leq 1||H \leq 1) + I \leq 2$$
  $$\text{If X-Class}: D \leq 1$$

- ChangeL

  $$(E \leq 2||F \leq 1) + (G \leq 1||H \leq 1)$$

- Cat

  $$E \leq 1 + H \leq 2 + I \leq 3$$

- CatBag

  $$(E \leq 2||F \leq 1) + H \leq 2$$

- Support

  $$E \leq 2 + H \leq 1 + J = 0$$

- ChangeD

  $$(E \leq 2||F \leq 1) + (G \leq 1||H \leq 1) + H \leq 2$$

- Clean (using un-enhanced)

  $$\Big(A \leq 1 + B \leq 1 + C \leq 1 + D \leq 1\Big) || \Big((G \leq 1||H \leq 1) + (G \leq 2 + I \leq 2)\Big)$$

- Swing

  $$\Big(A \leq 1 + B \leq 1 + C \leq 1 + D \leq 1\Big) || \Big(F \leq 2 + G \leq 1 + H \leq 2\Big)$$
  $$\text{If X-Class}: D \leq 1$$

- OHP

  $$G \leq 2 + H \leq 3 + I \leq 2$$

- Manual

  $$(E \leq 2||F \leq 1) + G \leq 3 + H \leq 3 + I \leq 3$$

3.2 General Tasks

Each person also has general tasks they must complete, independently of their Routines. We use the enhanced version for calculations here.

When I say $A_x + C_x$, I mean that either $A_1 + C_1$ has to be true or $A_2 + C_2$ has to be true. $A_1 + C_2$ would not apply, as I assert that the x values are equal.

- Driver
  - Type 1:
    $$A_x \leq 1 + C_x \leq 3 + E_n \leq 2 + F_n \leq 2$$
  - Type 2:
    $$E \leq 2 + F \leq 2$$
  - Type 3:
    $$(E_x \leq 2 || F_x \leq 2) + J \leq 1$$
  - Type 4:
    $$J = 0 + L \leq 3$$
  - Type 5 (Not for XA or XC groups):
    $$J = 1 + L \leq 3$$
- Bread
  - If $L \leq 2$:
    $$(E \leq 2 || F \leq 1) + G \leq 3 + H \leq 3 + I \leq 3$$
    $$\text{OR } A = 0 + B = 0 + C = 0 + D \leq 1$$
  - If $L > 2$:
    $$(E \leq 2 || F \leq 1) + H \leq 2$$
- Fire
  $$E \leq 2 + H \leq 2 + I \leq 3$$
- FireMW
  $$(E \leq 2 || F \leq 3) + G \leq 3$$
- Water
  $$(E \leq 2 || F \leq 1) + G \leq 1 + I \leq 1$$

3.3 Chair Type

Recall from the introduction that "Chair" was one of the four modes. Chair is the mode in which people spend the most time. There are 4 types of Chairs: **None, Analog, Power, Shell**. The type is selected according to the following criteria:

- None:
$$B \leq 2 + C \leq 2 + D = 0$$
- Analog:
$$F \leq 2 + G \leq 1 + I \leq 1$$
- Power: All remaining EXCEPT XA and XC.
- Shell: All XA and XC.

4 Calculating Level from Routine and Tasks

We now introduce a powerful tool, the **level**. The decision tree is as follows:

- Is chair type None?
    - Level is 0.
- Does the person meet the criteria for every aspect of Routine?
    - Level is A.
- Out of the following aspects, which aspects does this person meet the criteria for?
- ChangeU, ChangeL, Clean, Swing, Manual, OHP
    -

5 Inputs and Commands

I introduced the concept of "Tech 1" in January 2021. Essentially, every person with the Power or Shell Chair type is endowed with a **hook**. Before we can discuss hooks, however, we must discuss **control schemes**.

5.1 Primary Input Commands

Primary inputs can come from several different sources. Below I lay out all the possible inputs at a tech level of 0, and the criteria for having these inputs available.

- LF Inputs
  - LF01 (LF Switch 1)
  $$A_1 \leq 3$$
  - LF02 (LF Switch 2)
  $$A_1 \leq 2$$
  - LF10 (LF Stick)
  $$A_1 \leq 1$$
- RF Inputs
  - Same as LF inputs, just with A_2 instead of A_1.
- LH Inputs
  - LH01 (LH Switch 1)
  $$E_1 \leq 3 || F_1 \leq 3$$
  - LH02 (LH Switch 2)
  $$E_1 \leq 2 || F_1 \leq 2$$
  - LH10 (LH Stick)
  $$\Big(E_1 \leq 2 + F_1 \leq 2\Big) \, || \, \Big(E_1 \leq 1\Big) \, || \, \Big(G_1 \leq 3 + (E \leq 2 || F \leq 1)\Big)$$
- RH Inputs, similarly to RF/LF, use 2 instead of 1.
- xH Inputs
  - These inputs can be either mapped to LH or RH, but not both.
  - xH21 (H Pen)
  $$\Big(E_x \leq 2 \, || \, F_x \leq 1\Big) + G_x \leq 3 + H_x \leq 2$$
  - xH31 (H Screen) - Assert class is not X, then:
  $$G_x \leq 3 + H_x \leq 3 + I_x \leq 3$$

  - xH41,42 (H LC, RC)

$$E_x \leq 3 \,||\, (G_x \leq 3 \,+\, H_x \leq 3)$$

- HD Inputs
  - HD00 (C Stick)

    $$J \leq 1$$

  - HD11 (HB)

    $$J \leq 2$$

  - HD12,13 (HBR)

    A and S Classes: $J \leq 3$

    X Class: $J = 0$

  - HD21,22 (H Switch 01, 02)

    A and S Classes: $J = 0$

    X Class: $J \leq 3$

- SP Inputs
  - SP01,02 (Sip s, p)

    $$L \leq 3$$

  - SP03,04 (Sip S, P)

    $$K = 0 \,+\, L \leq 3$$

  - SP11 (M Click)

    $$L \leq 2$$

  - SP21 (Vox)

    $$L \leq 1$$

  - SP31 (M Pen)

    $J \leq 2$ + $L \leq 2$ + Not XA / XC

    OR $J \leq 2$ + $L = 0$

- EE Inputs
  - EE01 (Absolute Pos)
  - EE11,12 (EE L, EE R)
  - EE21,22,23,24,25 (EE Sequence)
  - The above EE inputs are available to anyone, regardless of 14-string.
- Mx Inputs

  - We will go into more detail on Mx inputs in later sections. For now, disregard these. These will appear when the Tech Level is greater than or equal to 3.0.

Note that these inputs are only available for the modes Power Chair and Shell Chair. Other modes may have different control schemes.

We establish the following rules for primary inputs:

1. Two inputs in the same group cannot be input simultaneously. For example, HD21 and HD11 cannot be input simultaneously, while HD21 and EE01 can.
2. SP31 and HD inputs cannot be input simultaneously.
3. SD31 and EE inputs cannot be input simultaneously.
4. When SP31 is ready, SP01,02,03,04,11 are all available.
5. When xH21 is ready, xH01,02 are available.

Then, we establish that each input has certain variables associated with it.

- readyTime: The time it takes to go from the "unready" position to the "ready" position for this input.
- unreadyTime: The time it takes to go from the "ready" position to the "unready" position for this input.
- delaySelf: The time it takes before this same input can be hit consecutively.
- delayGroup: The time it takes before an input in the same subgroup can be input (same tens digit). For example, the time between inputs SP01 and SP03, but **not** SP01 and SP11.

If "chained inputs" are mapped (for example, mapping a double-click of SP11 to mean something different from a single-click), the delaySelf becomes tripled.

The same applies for EE11/12 and EE21-25. If any of EE21-25 are used within a mode, the delaySelf and delayGroup values of EE11-12 are tripled.

Ready and Unready times apply only to SP31 and xH21. When these groups are in the "ready" position, all other inputs in the same group not expressly listed above as available are not available until the group is set back to "unready".

Please see the tech_primary.txt document for information on the above variables.

5.2 Input Information Conveyed

Here we discuss what information these inputs convey.

5.2.1 Unary

- These inputs do not have a time component. They are simply input.
  - SP21
  - EE21,22,23,24,25

5.2.2 Unary On-Off

- These inputs only convey whether or not the input is being pressed or not at a certain moment. They cannot be enhanced by being combined with other inputs in the same group.
  - LF01,02
  - RF01,02
  - LH01,02
  - RH01,02
  - xH41,42
  - SP01,02,03,04
  - SP11
  - EE11,12

5.2.3 Enhanced Unary

- These inputs can be combined with other inputs to create enhanced inputs.
  - HD12,13 + HD11
  - HD21,22 + HD11

5.2.4 Directional

- These inputs provide continuous X and Y values.
  - LF10
  - RF10
  - LH10
  - RH10
  - HD00
- Note that these inputs have "precision" values for r and θ. For example, the default HD00 has precision values of $r = .5$ and $\theta = 45$, indicating that r can take on the values $0, 0.5, 1$, and θ can take on any multiple of 45 degrees. (I would like to use π but I don't trust floating point math).

5.2.5 Absolute Directional

- These inputs are similar to Directional, but their value can be changed while violating Intermediate Value Theorem.
- For example, using HD01,02,03,04, moving from (1,1) to (-1,-1) would require me to pass through X=0 and Y=0. With an Absolute Directional input, however, I can move from (1,1) to (-1,-1) instantly.
  - xH21
  - xH31
  - SP31
  - EE01
- Again, these inputs have precision values, though instead of r and θ they are x and y. Assuming the area of input is $(1, 1)$, a precision of $x = 10$ and $y = 5$ indicates that 50 total areas on the screen are available to be selected.
- As such, we can use Absolute Directional commands to define new Unary commands.

5.2.6 User-Defined Unary On-Off Commands

- Using one of the four Absolute Directional commands, the user can define new unary commands by using the Absolute Directional command to move the cursor a fixed location on the screen.
- For xH21, xH31, and SP31, the amount of unary commands available is equal to $x \cdot y - 7$, as 7 "squares" will be reserved for the universal commands shown later.
- These commands can be turned into on/off by simply holding down the input for at least 0.5 seconds, or by holding LEFT CLICK (if that is also mapped to unary on/off).
- For EE01, there are 6 available unary commands to be mapped.
- These commands can be turned into on/off by holding down the input for at least 1.5 seconds, or by holding LEFT CLICK (if that is also mapped to unary on/off).
- Additionally, SP21 can be used to define new an arbitrary amount of new unary commands, though none of them have on-off capability.

We will explore this later, but each person has a **latency** value associated with them. This determines the "error value" of their input durations and precisions, so to speak. For example, if a person's latency value determines that their input duration can be at most 0.2 seconds precise, and they have that input mapped to a rotation to where they need to be precise to within 10 degrees, then the **velocity** can be at most 1/7.2.

5.3 Secondary Commands

Secondary commands are where the "useful" stuff happens. We use primary inputs to map to secondary commands, in order to make our person do something. **The following information applies only to Power and Shell Chair modes**.

5.3.1 Universal Commands

We have certain **Universal Commands**, where no matter what sub-mode we are in, the same primary inputs will map to the same secondary commands. They are as follows:

- SWITCH: Needs Unary input
- STOP: Needs Unary input
- EMERGENCY: Needs Unary input
- NEXT / TOGGLE: Unary input, takes up to 2.
- LEFT CLICK: Unary input
- RIGHT CLICK: Unary input
- BACK: Unary input

In total, 7 or 8 unary inputs will be reserved for Universal Commands. Left Click and Right Click themselves can be mapped to secondary commands in any sub-mode, while SELECT / TOGGLE can be mapped in only certain sub-modes.

In all subsequent sub-modes, each input-command mapping is only valid in that sub-mode. Additionally, there are sub-submodes we refer to as **axes**, switchable via the SELECT / TOGGLE commands, where input-command mappings can be made valid in only that axis and sub-mode.

We refer to a **2-switch** as a group of two associated unary on-off inputs in the same group: for example, LH01 and LH02. A **1-switch** is a single unary on-off input.

5.3.2 Sub-Mode: Drive

The default mode. Inputs needed:

- Precision of 15 degrees and 0.5 units/s (range -1 to 1) desired. Must be able to reach 0 units/s within 1 second at all times.
- Forwards / Backwards

  - If the user so desires, instead of forwards / backwards, the input can be remapped to increase / decrease current speed, in intervals of 20%. This requires only a unary input.

- Left / Right
- Preferred Scheme: Directional input mapped to both.
- Scheme 2: Two 2-switches mapped.
- Scheme 3: One 2-switch, using TOGGLE to switch.
- Scheme 4: Two unary inputs, using TOGGLE to switch. Each input is fixed to +/- 0.5 units/s or +/- 15 degrees.

5.3.3 Sub-Mode: CPOS

- Base Up / Down: Precision 0.25 units (0 to 1).
- Base Back / Forwards: Precision 10 degrees (0 to 90).
- H Theta Up / Down: Precision 15 degrees (0 to 90).
- Knee Theta Up / Down: Precision 30 degrees (0 to 90).
- Casters Raise / Lower: No precision necessary.
- Preferred: 5 2-switches.
- Scheme 2: 4 2-switches and a unary input for Casters Raise / Lower.
- Scheme 3: 2 2-switches, a unary input for CR/L and TOGGLE (2 axes).
- Scheme 4: 1 2-switch, a unary input for CR/L, and TOGGLE (4 axes).
- Scheme 5: 1 unary input, with each input moving by the precision amount. TOGGLE (10 axes).

5.3.4 Sub-Mode: Camera

- Toggle Camera: Mapped to TOGGLE.
- Rotate Camera CW/CCW. Precision 30 degrees (0 to 360).
- Rotate Camera +Z/-Z. Precision 15 degrees (0 to 180).
- Camera Zoom +/-. Precision 0.5x (1x to 3x).
- Unfortunately, we cannot remap TOGGLE in this mode. Thus the above inputs are according to priority of how many unary inputs are available.
- Note that CCW and -Z are not strictly necessary, since we have 360 degree cameras, and CCW and -Z can simply be held down for however long until it loops back around.

- Preferred: 3 2-switches.
- Scheme 2: 2 1-switches and a 2-switch.
- Scheme 3: 2 2-switches.
- Scheme 4: 1 2-switch and an alternate unary toggle.
- Scheme 5: 1 1-switch and an alternate unary toggle.
- Scheme 6: 3 different unary inputs at precision value.

5.3.5 Sub-Mode: Text Input

- Select Preset Folder: LEFT CLICK
- Select Preset Word/Phrase Within Folder
- Cursor Position (On-Screen Keyboard Entry)
- Select Object at Cursor Position: LEFT CLICK
- Quick Backspace 1 Character
- Quick Backspace 1 Word
- Quick Backspace 1 Sentence
- Return Current Phrase
- Clear Everything: BACK
- One of the Absolute Directional inputs will be mapped to Cursor Position. Then, 4 unary inputs will be mapped to the quick backspaces and ”Return Current Phrase”.
- If four unary inputs are not available, the order of priority is Return, Back 1 word, Back 1 sentence, Back 1 char.

It is generally preferred to have minimal overlap between the inputs used in the sub-modes, as then multiple commands can be issued without having to switch modes, giving the user much more flexibility.

5.4 Hooks

We now shift our focus to hooks. This is an additional sub-mode, unlocked at tech level 1. The hook consists of two metal rods of length 3 feet each, connected by a ball-and-socket joint. At the end of the second rod is a gripper that can hold objects up to a reasonable size and weight. By default, the hook will be controlled through the camera; thus the hook will likely inherit many of the camera sub-mode’s arguments.

The inputs are as follows, for **Hook 1**:

- Move Hook Forwards / Backwards (towards Camera Crosshair): Precision 0.5 inches.
- Rotate Crosshair CW / CCW: Precision 5 degrees (0 to 360).
- Rotate Crosshair +Z / -Z: Precision 5 degrees (0 to 360).
- Grip / Release Object: Unary input.
- Rotate Head CW / CCW: Precision 10 degrees.
- Scheme 1: 4 2-switches and a unary input.
- Scheme 2: 2 2-switches and a unary input. 2xTOGGLE axes.
- Scheme 3: 1 2-switch and a unary input. 4xTOGGLE axes.
- If Scheme 3 cannot be provided, the user does not receive Hook 1.

6 Picking a Control Scheme

Below I will detail the methodology to select a control scheme for each of the sub-modes. I will be working with 5 sample people, with the following constraints:

Person A: Class X 55.3

Person B: Class X 44.3

Person C: Class X 21.2

Person D: Class S $C4.9$

Person E: Class S $C6.3$

Every single person has the inputs EE01, EE11/12, and EE21-25. Below are the additional inputs for each person:
Person A: RF01
Person B: LH01-02, RH01, SP01-04
Person C: RF01-02, RF10, HD21-22, SP01-04, SP11, SP21
Person D: HD00, HD11-13, HD21-22, SP01-04, SP11, SP21, SP31
Person E: LH01-02, LH10, RH01-02, RH10, RH21, RH41-42, HD00, HD11-13, HD21-22, SP01-04, SP11, SP21, SP31

6.1 Terminology and Notation

First, some terminology and notation. To expand our range of possible schemes we have:

- [n] - Hold input for n seconds. Valid for unary on/off. Example: SP02[2.5]
- (A) - Valid only in a certain axis. Used when sub-modes rely on axes. Example: RF01(2)
- >A - An additional unary assignment operator. Useful for: LH/RH21, LH/RH21, SP21, SP31, and EE01. Example: SP31>1, indicating that this is using slot 1 of the avilable user-defined unary inputs.
- a,b - Inputs a and b are used in that order. Used as shorthand. Example: Drive Forward/Backward - RF01,RF02
- xn - Repeat input n times. This changes the effective delaySelf value to 3 times, and inputs must be repeated within 1 - 2x the original delaySelf window. Example: HD11x2
- a=b - Input a and b back to back. If in the same group, the delayGroup value is tripled.
- a+b - Input a and b simultaneously.

Out of the above, only >, xn, =, and + are truly useful, and all serve to simply turn some small amount of unary inputs into a large amount of unary inputs; none of them can create unary on-off inputs.

6.2 Cursor Position

Recall that there are six possible Absolute Directional commands. These are LH/RH21, LH/RH31, SP31, and EE01. We first assign each person an Absolute Directional input according to this order of priority:

1. RH21
2. LH21
3. RH31
4. LH31
5. SP31
6. EE01

Then for our example five people, we have:

- A: EE01
- B: EE01
- C: EE01
- D: SP31
- E: RH21

Note that the other unused Absolute Directional commands will be discarded, made unavailable for use.

6.3 Universal Commands

There are seven universal commands, none of which can conflict with each other. In order of most to least used, we have:

1. NEXT / TOGGLE
2. LEFT CLICK
3. SWITCH
4. BACK
5. RIGHT CLICK
6. STOP
7. EMERGENCY

All of these take unary input. It is probably best to use up all the unary non-on-off inputs on these universal commands. Recall from above that to avoid any amount of toggling axes during sub-modes, we would need 4 2-switches and a unary input. To toggle minimally (CPOS and Hook only), we would need 2 2-switches and a unary input. Essentially, we want to "preserve" up to 4 2-switches and a unary input for the sub-modes that won't be used on the universal commands. A 1-switch or half of a 2-switch can be subsituted for a unary input, though not vice versa. We have, in order of priority, to be reserved:

1. 10 degrees of freedom + 1 unary
2. 8 degrees of freedom + 1 unary
3. 6 degrees of freedom + 1 unary
4. 4 degrees of freedom + 1 unary
5. 2 degrees of freedom + 1 unary
6. 1 degree of freedom + 1 unary
7. 1 unary

However, this isn't always possible. Our order of priority for preservation is:

1. All directionals are "worth" 2 2-switches.
   (a) RH10, LH10
   (b) HD00
   (c) RF10, LF10
2. HD11+12+13 (Enhanced 2-switch)
3. Complete 2-switches (if both exist)
   (a) RH41+42, LH41+42 (2-switches)
   (b) RH01+02, LH01+02 (2-switches)
   (c) SP01+02, SP03+04 (2-switches)
   (d) HD21+22 (2-switch)
   (e) HD12+13 (2-switch)
   (f) RF01+02, LF01+02 (2-switch)
4. Single 1-switches
   (a) LH41, LH42, RH41, RH42
   (b) SP01, SP02, SP03, SP04
   (c) LH01, LH02, RH01, RH02
   (d) SP11

   (e) HD21, HD22
   (f) LF01, LF02, RF01, RF02
   (g) HD12, HD13

5. SP21 mapped to some arbitrary integer

We also note that certain Cursor Position selections force certain Universal Command mappings.

- SP31 forces SP11 for LEFT CLICK
- LH21 forces LH41 for LEFT CLICK and LH42 for RIGHT CLICK
- RH21 forces RH41 for LEFT CLICK and RH42 for RIGHT CLICK

The Absolute Directional mapping already has 7 pre-reserved slots for Universal Commands to provide a fail-safe.

Looking at people A, B, C, D, and E: here is a list of what we would preserve:
A: RF01 (1-switch)
B: LH01-02, SP01-02, RH01 (2 2-switches and a unary)
    SP03-04 available for Universal.
C: RF10, HD21-22, SP01-02, SP11 (Directional, 2 2-switches, and a unary), with SP21 available
    RF01-02, SP03-04 available for Universal.
D: HD00, HD12-13, SP01-02, HD11 (Directional, 2 2-switches, and a unary), with SP21 available.
    SP11 reserved for LEFT CLICK.
    HD21-22, SP03-04 available for Universal.
E: LH10, RH10, RH01 (2 Directional and a unary), with SP21 available.
    RH41-42 reserved for LEFT CLICK and RIGHT CLICK.
    LH01-02, RH02, all HD, all SP available for Universal.

Then to assign our universal commands, we have the following, ranked in order of speed of input:

1. LH41-42, RH41-42, SP02
2. EE11, 12 (when EE21-25 are not used) (at least 5 non-HD or SP reserved)
3. SP04
4. SP01
5. EE11, 12 (when EE21-25 are not used) (at least 5 reserved)
6. SP03
7. LH01-02, RH01-02, SP11

8. HD21-22
9. LF01-02, RF01-02, HD12-13
10. EE11, EE12 (when any of EE21-25 are used)
11. EE21
12. HD11
13. EE22, 23, 24, 25 in that order

We would assign:

Person	TOGGLE	LEFT CLICK	SWITCH	BACK	RIGHT CLICK	STOP	EMER
A	EE11	EE12	EE21	EE22	EE23	EE24	EE25
B	SP04	SP01	EE11	EE12	EE21	EE22	EE23
C	SP04	SP03	RF01	RF02	EE11	EE12	EE21
D	SP04	SP11	EE11	EE12	SP03	HD21	HD22
E	RH02	RH41	LH01	LH02	RH42	EE11	EE12

6.4 Text Input Speed

If SP21 (Voice) is available, text can be input at whatever speed the user can talk, generally around 150 words per minute.

If not, then text must be input manually. We consider the Directional Abs for input. There are $xy - 7$ available "squares" of words to select. There is almost no case in which someone would have xH21/31 without SP21, so we won't consider that for this purpose.

6.4.1 EE01 Text Input

First we consider the EE01 Absolute Directional case. There is a 8×5 grid available, with the following preset mappings:

- Squares 1 through 7 reserved for the 7 universal commands.
    1. TOGGLE - Scroll to the next page.
    2. LEFT CLICK - Confirm selection.
    3. SWITCH - Enter Letter by Letter mode.
    4. BACK - Go back one menu.
    5. RIGHT CLICK - Enter editing mode, where the user can add, remove, and edit certain words.
    6. STOP - Clears the input queue.
    7. EMER - Calls the nearest secondary. Used in event of system failure.

- Squares 8 through 24 for other selections.

The average conversation requires around 3000 words, and I'm going to guess that there are around 1500 common phrases of length 2 to 4. Thus there will be 66 sub-menus spread across 2 separate pages, and 66 words and phrases in each sub-menu spread across those 2 pages, for a total of $66^2 = 4356$ words and phrases. To select a word, assuming every selection has an equal chance of happening, we observe that it takes on average:

$$\text{TOGGLE} + 2\,\text{BACK} + 2\,\text{LEFT CLICK} + 5\,\text{EE01DELAY}$$

Of course, the usage of words is not equally distributed, so we conjecture that the user will frontload the more common words, and instead it will take:

$$\frac{3}{4}\text{TOGGLE} + 2\,\text{BACK} + 2\,\text{LEFT CLICK} + \frac{19}{4}\,\text{EE01DELAY}$$

Using in-built UNIV BACK: $\frac{3}{4}\text{TOGGLE} + 4\,\text{LEFT CLICK} + \frac{19}{4}\,\text{EE01DELAY}$ seconds to select each word or phrase. The average word length of a "word or phrase" is 2 in our case, so the above represents the time it takes to input 2 words. Running calculations on the above people A and B yields a text input speed of:

A: TOGGLE EE11, BACK EE22, LEFT CLICK EE12, speeds 0.4, 1.6, 0.4 respectively, for 45.9 words per minute using UNIV BACK.

B: TOGGLE SP04, BACK EE12, LEFT CLICK SP01, speeds 0.25, 1.2, 0.35 respectively, for 52.1 words per minute using UNIV BACK.

We also need to evaluate the speed of raw character input. The screen can only display 17 characters at once, and the space key must be on both screens. Thus we consider the frequency of the 16 most popular characters as opposed to the last 10 characters, and use the estimate of 5 characters per word.

The 16 most common characters make up roughly 90 percent of all characters. Monte Carlo methods suggest that a single keypress has a 3/20 probability of needing a switch beforehand. Thus each word has on average 9/10 switches and 6 left clicks, thus the time per word is:

$$\frac{9}{10}\,\text{TOGGLE} + 6\,\text{LEFT CLICK} + \frac{69}{10}\,\text{EE01DELAY}$$

Running calculations on A and B yields text input speed of:

A: TOGGLE EE11, LEFT CLICK EE12, speeds 0.4, 0.4, for 15.8 words per minute.

B: TOGGLE SP04, LEFT CLICK SP01, speeds 0.25, 0.35, for 17.9 words per minute.

6.4.2 SP31 Text Input

This type of setup is quite rare, occuring in the following cases for the presets I have laid out:

- 1C archetype, S Class, any seed
- 1B archetype, S Class, 0 seed
- 33 archetype, X Class (Upper), 5 seed, Not XA or XC

There is a 12×8 grid, and 7 are reserved, leaving 89 open slots. We let there be $89^2 = 7921$ possible words in the word bank through selection, leading to a time per word or phrase of:

$$2\,\text{BACK} + 2\,\text{LEFT CLICK} + 4\,\text{SP31DELAY}$$

$$= 4\,\text{LEFT CLICK using UNIV BACK} + 4\,\text{SP31DELAY}$$

Because this setup has twice as many words and phrases as the EE01 text input, the average phrase length is 3.

6.5 Driver, SPT, GPT Conditions

I know Driver was introduced above, but I will specify the conditions further. There are several ways one can be a Driver:

1. Standard: All the below conditions must be met.
   (a) Either LF10 or RF10
   (b) Either LH21 or RH21
2. Modified: Rocket Complex, Rudder.
   (a) Rocket Complex Type 1: Continuous. Must be mapped to some Directional (not Absolute) control.
   (b) Rocket Complex Type 2: Cruise. Conditions below:
      i. An input to act as the ”Friction” input, out of the following: SP03, SP04, LH/RH01-02, HD21/22, LF/RF01-02.
      ii. An input to act as the ”Increase” input, one of the aforementioned inputs. Cannot share a group with Friction.
      iii. An input to act as the ”Decrease” input, one of the aforementioned inputs. Cannot share a group with Friction.
   (c) Rocket Complex Type 3: Discrete. Conditions below:
      i. 5 unary inputs to act as ”Friction”, ”Rocket”, ”Increase Step”, ”Decrease Step”, ”Cruise”. Friction cannot share a group with any of the other 4 inputs.
   (d) Rudder: Must be a 2-switch.

(e) If the person has $J \leq 3$ as a S or A class, or $J = 0$ as a X class, then they are a Driver.

    i. If not, then the person must be able to construct a Rocket Complex and Rudder without the usage of any HD inputs. Then, the person must assign an extra 2-switch to ”Left” and ”Right”.

Then, the SPT conditions are comparatively much simpler. The Text Input speed must be at least 100 words per minute. At Tech levels 1 and 2, this simply entails having SP21.

Finally the GPT conditions are that the user has a Hook 1, which means having a 2-switch and unary input left over after assigning universal variables.

We will explore these conditions more when we introduce **Locations** and **Residencies**.

7 Tech Level 2: The TECH TABLE

We introduce the concept of a TECH TABLE, a residence-specific device. It is comprised of a table within a room, and several hooks, cameras, and bands. All primary inputs available to the user outside of the TECH TABLE will also be available within the tech table.

Recall that **Chair** was the mode that a person would spend the most time in. Now, for the TECH TABLE, we refer to this as being in **Zone** mode. Only certain residences are endowed with a TECH TABLE.

We now describe the Zone in detail.

- We define the X-axis as crossing through the midline of the Seat, Back and Head. The Z-axis runs from the ground to the sky.
- The Person: What information do we need?
  - The 21-string
  - Their latency. Call this σ.
  - Their BLBW seed
- The Table
  - Composed of three components, smaller sub-tables. Call these components the Seat, the Back, and the Head.
  - Seat-Back joint:
    - Can drift up to 6 inches apart.
    - Axis of rotation is the Y-axis. Angle ranges from 45 to 180 degrees.
    - Must maintain alignment in the XY plane.
  - Back-Head joint:
    - Angle ranges from 120 to 180 degrees about the Y-axis.
    - The Head itself can be rotated -90 to 90 degrees about the midline of the seat.
- The Bands
  - There are three types of bands: table bands, free bands, and connectors. The fewer bands there are, the less time it will take.
  - Table bands: Attach to the table or a joint of the table.
    - SL01, SR01: $\sigma > 4$
    - SL02, SR02: $\sigma > 10$
    - BB01: All
    - BB02: $\sigma > 3$ and $D \geq 3$

    * BB03: $\sigma > 8$ and $D = 4$
    * BB10, BB11: $\sigma > 12$
    * HP01: $J \geq 2$ and HD commands are in the user's input scheme.
  - Free Bands: Attach to person and perhaps the wall too.
    * NK01: $J \geq 1$ and no HD commands are in the user's input scheme.
    * FL01, FR01: User does not have LF or RF in their scheme, respectively. $A < 4$, no A classes or any hybrids.
    * FL02, FR02: User does not have LF or RF in their scheme, respectively. $A < 4$ and $\sigma > 14$, no A classes or any hybrids.
    * LL01, LR01: $K < 4$, no A classes or any hybrids.
    * LL02, LR02: $K < 4$, no A classes or any hybrids.
    * LL03, LR03: $K < 4$, no A classes or any hybrids, $\sigma > 10$.
    * LL04, LR04: $K < 4$, no A classes or any hybrids, $\sigma > 20$.
    * AL01, AR01: No OHP capability.
    * AL02, AR02: No OHP capability.
    * AL03, AR03: No OHP, $\sigma > 5$.
    * AL04, AR04: No OHP, $\sigma > 12$.
  - Connectors: Connect between two bands.
    * BB21: $\sigma > 17.5$. Connects BB01, 02, and 03.
    * HR01, HR02: Used when deploying Left or Right Shell. Connects SL02, SR02, and BB02. $\sigma > 12$.

The secondary inputs to be mapped are:

- Universal Commands: TOGGLE, LEFT CLICK, SWITCH, BACK, RIGHT CLICK, STOP, EMER
- List of Modes (Available from screen select on Main Menu)
  - Active Hook
  - CPOS Enhanced
  - Camera
  - PLC Input
  - Inventory
  - Torrent
  - Desert
  - Procedure
- Active Hook:
  - Inherits Hook 1 controls.

  - Grip / Release should be a 2-switch. If the user does not have a 2-switch available, Active Hook's motion features are disabled.
  - Get Item from Inventory
  - Return Item to Inventory
  - Return to Resting
  - Switch Active Hook

- CPOS Enhanced Mode:
  - Select Preset POS
    * FLAT, VAL, VAL+, MISS, CHR, CHRB, LC, RC, LO, RO, BACK, FORWARD, FLOAT
  - Select Rotating Axis of Motion
    * Unit YZ CW / CCW
    * Unit XZ CW / CCW
    * Seat-Back Increase / Decrease
    * Back-Head Increase / Decrease
    * Head Left / Right
  - Select Translating Axis of Motion
    * Seat-Back Dist Increase / Decrease
    * Unit +Z / -Z
  - Deploy Front Shell
  - Deploy Back Shell
  - Deploy Left Shell
  - Deploy Right Shell

- Camera:
  - Camera Zoom + / -
  - Switch to Preset Camera Position (1 through 16)
    * Camera CW / CCW
    * Camera Rotate +Z / -Z
  - Switch to Free Camera
    * Camera +Axis1 / -Axis1
    * Camera +Axis2 / Axis2 (The camera is on the surface of the wall, so only 2 degrees of freedom)

- PLC Input:
  - Inherits all Camera controls.

  - Crosshair to absolute position X, Y. Precision to be specified by specific procedure.
  - Begin Drawing - either unary or 1-switch.
    * If the Absolute Directional input is not EE01, then the Crosshair position can be mapped to that.
    * Otherwise, the crosshair has a few options:
      · 2 2-switches or a directional.
      · TOGGLE and a 2-switch.
      · 3 TOGGLEs and a 1-switch.
      · A single unary input that moves the crosshair by 1/10 of the screen in a set direction, with 3 TOGGLEs.

- Inventory:
  - Simply select an item placed in the Inventory, and it swaps it onto the active hook. "Unlimited" storage space.
- Torrent:
  - Inherits parts of CPOS and Camera Controls.
    * CPOS: Switch to Preset Position: FLAT, FORWARD, BACK, MISS
    * Camera: Camera +/- Axis1, Axis2
  - Increase / Decrease Heat
  - Increase / Decrease Pressure
- Desert:
  - Inherits parts of CPOS and PLC controls.
    * CPOS: Switch to Preset Position: FLAT, FORWARD, BACK, MISS
    * PLC: All
  - Switch Head: Base, Dry
  - Use this mode by using the PLC mode to draw closed shapes.
- Procedure:
  - Enters a preset Procedure. We will discuss these in the following sections.