

3D DYNA DRAW INSTRUCTIONS

This dynamic drawing tool will have you creating wild and crazy scenes that you will love to share with others. The single letter programming commands are like the LOGO Turtle graphics of 'old' and a great educational platform for learning three-dimensional spatial concepts and coding.

You can share with anyone on any platform including smart phones and tables, but if you want to roll up your sleeves and create your own, a keyboard is needed.

Happy Drawing!

SINGLE LETTER MOTION & ROTATION

The first command you should learn is 'f' which draws a short line forward. If you enter more than one 'f' the line gets longer, 'ffff' is four times as long. If you use a capital 'F', the line dynamically grows and shrinks in both directions.

f/F: Line Forward

This is a special command because it actually draws something on the screen. The only other commands that draw on the screen are the **3D Object** commands, for example **<World>**, or **<Smoke>**.

All the other commands will give you control over the position, rotation, size, and color of your creation.

Some special commands let you interact with your creation by aiming and shooting cannon balls.

Some really special commands give you programmatic type controls of your drawing scripts, to create your own functions, encapsulate transforms, and make repetitive scripts easier.

' s: as a multiplier (Maximum 25)

#'s Give a multiplier to a single letter command. For example, 5f is the same as 'fffff', 6r is a 90 ° turn!

Note, this also works with all step and dynamic rotate commands as well as c (copy), and z (zoom)

2D MOTION - Lower case gives a single step,

UPPER case gives dynamic motion/rotation for all these commands:

	m/M: Move forward	
r/R: Rotate Right 15 °		l/L: Rotate Left 15 °
	b/B: Move Back	

3D MOTION - Lower case gives a single step,

UPPER case gives dynamic motion/rotation for all these commands:

	u/U: Rotate Up 15 °	
o/O: Rotate Out – spin clockwise 15 °		i/I: Rotate In – spin counter-clock wise 15 °
	d/D: Rotate Down 15 °	

3D OBJECTS TO ADD TO YOUR SCENE – THIS IS THE F U N STUFF!!

- **<World>** **<Rocket>** **<Asteroid>**
- **<Sword>** **<Disk>** **<Torus>**
- Lights: **<Point>** **<Spot>** for a point light or a spotlight.
- Drawing **<Smoke>** **<Draw>** must be placed somewhere that is moving. It will leave a smoke trail behind. Smoke is the same as Draw.
- " " : Quotes. Anything inside quotes will be drawn as 3D text. **Will not draw until opening and closing Quotes are in place.
- **<Circletext(5)>** Will put your 3D Text on a Circle Path with a Radius of 5 units.
- **<Linetext()>** This will return you back to standard Line Text. This is the default, so this command is only needed if you have started with some Circle Text and want to return to Line Text.

POINT AND SHOOT

- **<Shoot>**: Shoots a cannon ball when you press Space Bar, or the Fire button on your Game Controller.
- **<Tilter>**: Arrow key on the keyboard and Game Controller control the tilt – Also try **<Tilter><Tilter>** for 360 degrees of tilting.
- **<Atilter>**: Just like Tilter, but in the opposite direction.

COMMANDS TO ADD PIZZAZ

- **z**: Zoom Smaller, **Z**: Zoom Larger
- **C**: Dynamic Rotate colors of anything drawn after this – “Taste the Rainbow!”
- **<Color(C)>**: Same as the above single letter ‘C’ command.
- **<Color(1)>**: Set a specific color:
0: black, 1: white, 2: red, 3: green, 4: blue, 5: yellow, 6: magenta, 7: cyan, 8: gray, 9: black

SPECIAL PROGRAMMING COMMANDS

- **() c**: Copy stuff found in last (...), example: **(fr) cccc**

You can also have copies inside copies: **((6r5f) ccc6r) ccc**

- **[]**: Branch. Any motion/rotation or zooms set inside the brackets are reset to where they started at the beginning of the branch. This is nice for branching from the same point
Example: **ffff[Rffff] [Lffff]**, both arms start from the same point. You can do branches inside branches as well.
- **< >**: Create and call your own functions
 - Step 1, define the function **<rightbend=ffrrffrr>**
 - Step 2, call the function **<rightbend>** results in **ffrrff**
- You can also pass a parameter to your function. The key word X gets replaced by what you put inside the () when you call the function:
 - **<say()="X">** **<say>Hello>** results in **"Hello"**

MORE PROGRAMMING EXAMPLES

Create Function – X gets replaced	Use Your Function
<code><square () = (Xf6r) ccc></code>	<code><square (3)></code>
<code><triangle () = (Xf8r) 2c></code>	<code><triangle (4)></code>
<code><circle () = (Xfr) 24c></code>	<code><circle (1)></code>
<code><rotdisk () =XI<Disk>></code>	<code><rotdisk (6)></code>
<code><spinthis () =IX></code>	<code><spinthis (<Torus>)></code>

Note that when you create a function, the capital letter 'X' will be replaced with whatever you pass in the parentheses (). You can pass multiplier #'s, Text for 3D Text, a 3D object like <Disk>, or even call a function inside a function. A function can even call itself, for some cool recursion!

Stare at this for a while and figure out what it is doing:

```
<recurse () =X[RX] [LX]><recurse (<recurse (fff)>)>
```

SOME BASIC CONCEPTS

All these are the same, and draw a square:

```
ffffrrrrrrrrffffrrrrrrrrffffrrrrrrrrffffrrrrrrr
4f6r4f6r4f6r4f6r
(4f6r) ccc
<square () = (Xf6r) 3c><square (4)>
```

A fill circle is 360 degrees, an 'r' command only rotates 15 degrees. So, it takes 24 'r' commands to make a full circle.

```
(fr) 24c
```

Start with a circle and make it wacky by adding dynamic motion:

<code>(fR) 24c</code>	** we change the 'r' to 'R'
<code>(fIr) 24c</code>	** we added a dynamic spin
<code>(fOUR) 24c</code>	** Really Wacky
<code>(fOUR) 24c<Rocket></code>	** Now that's just hilarious!