ATLASENGINE COMPLETE GUIDE - PART 5

Quick Reference, Tips, Troubleshooting & Command Tables

KEYBOARD SHORTCUTS

Editor Shortcuts

Shortcut	Action
`Ctrl+N`	New script
`Ctrl+O`	Open script
`Ctrl+S`	Save script
`Ctrl+Shift+S`	Save As
`F5`	Run script
`Shift+F5`	Stop script
`Ctrl+R`	Reload script
`Ctrl+K`	Clear output
`Ctrl+B`	Toggle sidebar
`Ctrl+J`	Toggle output panel
`Ctrl+L`	Toggle logs
`F11`	Full screen
`Ctrl+Q`	Quit

Editing Shortcuts

Shortcut	Action
`Ctrl+Z`	Undo
`Ctrl+Y`	Redo
`Ctrl+X`	Cut
`Ctrl+C`	Copy
`Ctrl+V`	Paste
`Ctrl+A`	Select all
`Ctrl+F`	Find
`Ctrl+H`	Replace
`Ctrl+G`	Go to line
`Ctrl+/`	Toggle comment

Shortcut	Action
`Ctrl++`	Zoom in
`Ctrl+-`	Zoom out
`Ctrl+0`	Reset zoom

Help

Shortcut	Action
`F1`	Documentation
`Ctrl+,`	Settings

COMMAND QUICK REFERENCE

Text Output (10)

```
say "text"              # Normal output
shout "TEXT"            # Loud/red output
whisper "text"          # Quiet/gray output
show variable           # Display variable
input "prompt" into var # Get user input
print "text"            # Debug output
clear                   # Clear output
cleargraphics           # Clear graphics
warn "message"          # Warning
error "message"         # Error
```

Variables (15)

```
x is 10                 # Create/set variable
make x 10               # Alias
set x 10                # Alias
x add 5                 # Add to variable
x subtract 3            # Subtract from variable
x multiply 2            # Multiply
x divide 2              # Divide
increment x             # Add 1
decrement x             # Subtract 1
forget x                # Delete variable
```

Math (30)

```
sum is x + y            # Addition
diff is x - y           # Subtraction
product is x * y        # Multiplication
quotient is x / y       # Division
remainder is x modulo y # Modulo

result is power 2 8     # 2^8 = 256
result is root 16       # √16 = 4
result is squared 5     # 5² = 25
result is cubed 3       # 3³ = 27

result is round 3.7     # 4
result is floor 3.9     # 3
result is ceil 3.1      # 4

result is min 10 20     # 10
result is max 10 20     # 20
result is clamp 150 0 100  # 100

result is sin 90        # 1
result is cos 0         # 1
result is tan 45        # 1

result is absolute -10  # 10
result is sign -5       # -1

dice is random 1 6      # Random 1-6
```

Control Flow (15)

```
# Conditionals
when x equals 10 { ... }
when x greater 5 { ... }
when x less 20 { ... }
when x notequals 0 { ... }
when x between 5 10 { ... }

# Loops
repeat 10 times { ... }
while x less 10 { ... }
until x equals 5 { ... }
for i from 1 to 10 { ... }
foreach item in list { ... }
```

```
# Control
break                   # Exit loop
continue                # Next iteration
return                  # Exit function
```

Strings (20)

```
result is join "Hello" " World"
words is split "a,b,c" ","
len is length "Hello"   # 5

result is uppercase "hello"  # HELLO
result is lowercase "HELLO"  # hello
result is titlecase "hello world"  # Hello World

result is trim "  text  "  # "text"
result is replace "I like cats" "cats" "dogs"
result is substring "Hello World" 0 5  # "Hello"

bool is startswith "Hello" "Hel"  # 1
bool is endswith "World" "rld"    # 1
bool is contains "Hello" "ell"    # 1

pos is indexof "Hello" "l"  # 2
count is count "banana" "a"  # 3
result is reverse "Hello"   # "olleH"
```

Graphics 2D (20)

```
# Drawing
drawline from 0, 0 to 100, 100 color "#ffffff"
drawrect at 100, 100 size 50, 50 color "#ff0000"
drawcircle at 300, 300 radius 50 color "#00ff00"
drawtext "Hello" at 200, 200 color "#ffffff"
fillscreen "#000000"

# Sprites
sprite "player" at 100, 100 size 32, 32 color "#00ff00"
movesprite "player" to 150, 150
colorsprite "player" to "#0000ff"
hidesprite "player"
showsprite "player"
deletesprite "player"

# Advanced
arc at 300, 300 radius 100 start 0 end 180 color "#ff0000"
```

```
ellipse at 300, 300 width 200 height 100 color "#0000ff"
triangle 300, 100 200, 300 400, 300 color "#ff0000"
gradient from "#ff0000" to "#0000ff" at 100, 100 size 400, 200
```

3D Basics (10)

```
# Objects
create3d cube at 0, 0, 5 size 1
create3d sphere at 5, 0, 10 size 2
create3d plane at 0, -1, 0 size 10
color3d last3d to "#ff0000"

# Camera
camera at 0, 1.6, 0
lookat 0, 0, 10
zoom 2.0

# Lights
pointlight at 0, 5, 0 color "#ffffff" intensity 1.0
```

Game Mechanics (20)

```
# Stats
health is 100
health subtract 25
heal 50

mana is 100
stamina is 100
armor is 50

# Weapons
weapon "pistol" damage 25 firerate 0.5 ammo 12
equip "pistol"
shoot
reload "pistol"

# Combat
hit "enemy" damage 25
explode at 10, 0, 10 radius 5 damage 50
burn "enemy" duration 5

# Inventory
additem "key"
removeitem "key"
useitem "health_potion"
```

```
has_key is hasitem "key"

# Level/XP
level is 1
xp add 100
levelup
```

Game Flow (10)

```
# Win/Lose
gameover
win
lose

# Checkpoints
checkpoint at 0, 0, 0
respawn at checkpoint

# Score
score add 100
highscore check

# Time
timer start 60
countdown from 10

# Pause
pause
resume
```

COLOR REFERENCE

Common Colors (Hex Codes)

Basic Colors

```
"#ff0000"  # Red
"#00ff00"  # Green (bright)
"#0000ff"  # Blue
"#ffff00"  # Yellow
"#ff00ff"  # Magenta
"#00ffff"  # Cyan
"#ffffff"  # White
"#000000"  # Black
```

Grays

```
"#111111"  # Almost black
"#222222"  # Very dark gray
"#444444"  # Dark gray
"#666666"  # Medium dark gray
"#888888"  # Medium gray
"#aaaaaa"  # Medium light gray
"#cccccc"  # Light gray
"#eeeeee"  # Very light gray
```

Game Colors

```
"#00ff00"  # Player (green)
"#ff0000"  # Enemy (red)
"#ffff00"  # Coin/gold (yellow)
"#00ffff"  # Key/item (cyan)
"#ff8800"  # Health (orange)
"#0088ff"  # Mana (blue)
"#ffd700"  # Goal/exit (gold)
"#8b4513"  # Wood/brown
"#808080"  # Stone/gray
"#1a1a2e"  # Dark background
```

UI Colors

```
"#4caf50"  # Success (green)
"#ff9800"  # Warning (orange)
"#f44336"  # Error (red)
"#2196f3"  # Info (blue)
"#9c27b0"  # Special (purple)
```

TIPS & TRICKS

Performance

Optimize Loops

Bad:

```
repeat 1000 times {
    # Heavy calculation every frame
    result is sqrt(x * x + y * y)
}
```

Good:

```
# Calculate once, reuse
distance is sqrt(x * x + y * y)
repeat 1000 times {
    # Use cached value
    say distance
}
```

Limit Draw Calls

Bad:

```
repeat 100 times {
    cleargraphics  # Clearing every time!
    drawcircle at x, y radius 10 color "#ff0000"
}
```

Good:

```
cleargraphics  # Clear once
repeat 100 times {
    drawcircle at x, y radius 10 color "#ff0000"
}
```

Use Culling

```
# Only draw visible objects
when object_visible equals 1 {
    drawcircle at x, y radius 10 color "#ff0000"
}
```

Debugging

Print Variables

```
say "x = " x
say "y = " y
say "health = " health
```

Check Conditions

```
when x equals 10 {
    say "X is 10!"  # Verify condition triggered
}
```

Log State

```
say "Player position: " player_x " , " player_y
say "Enemy position: " enemy_x " , " enemy_y
say "Distance: " distance
```

Use Comments

```
# Player movement logic
player_x add speed  # Move right
# TODO: Add collision check
```

Game Design

Use Variables for Tuning

```
# Easy to adjust
player_speed is 5
enemy_damage is 25
item_spawn_rate is 10

# Instead of hardcoding everywhere
player_x add 5      #  Hard to change
player_x add player_speed  #  Easy to tune
```

Separate Logic

```
# Health management
function check_health {
    when health less 1 {
        gameover
    }
}

# Collision detection
function check_collision {
    # ...
}

# Keep organized!
```

Test Incrementally

```
# Test movement first
player_x add 5
# Then add collision
# Then add enemies
# Then add combat
# Build piece by piece!
```

Common Patterns

Boundary Check

```
# Keep player in bounds
when player_x less 0 {
    player_x is 0
}
when player_x greater 600 {
    player_x is 600
}
```

Cooldown Timer

```
cooldown is 0

when shoot_pressed equals 1 {
    when cooldown equals 0 {
        shoot
        cooldown is 30  # Frames
    }
}

# Decrease cooldown
when cooldown greater 0 {
    cooldown subtract 1
}
```

Health Bar

```
# Visual health bar
bar_width is health * 2  # 100 health = 200 pixels
drawrect at 50, 50 size bar_width, 20 color "#00ff00"

# Background
drawrect at 50, 50 size 200, 20 color "#333333"
```

Simple AI

```
# Chase player
dx is player_x - enemy_x
dy is player_y - enemy_y

# Normalize
distance is sqrt(dx * dx + dy * dy)
dx divide distance
dy divide distance
```

```
# Move toward player
enemy_x add dx * enemy_speed
enemy_y add dy * enemy_speed
```

TROUBLESHOOTING

Common Errors

"cannot import name 'Viewport3D'"

Problem: Python cache issue **Solution:**

```
rmdir /s /q editor\__pycache__
python editor\main.py
```

"Commands print as text instead of executing"

Problem: Parser issue (was fixed) **Check:** Each line should execute separately **Test:**

```
say "Line 1"
say "Line 2"
```

Should output:

```
Line 1
Line 2
```

NOT:

```
say "Line 1" say "Line 2"
```

"Graphics show black screen"

Problem: Not switching to Graphics tab **Solution:** - Check OUTPUT window at bottom - Click "Graphics" tab (not "Text") - Graphics auto-switch is now enabled

"Input dialog doesn't appear"

Problem: Dialog might be behind window **Solution:** - Click on AtlasEngine window - Check taskbar for dialog - Dialog should pop to front

"Variables not updating"

Problem: Using wrong syntax **Bad:** `x = 10` (Python syntax) **Good:** `x is 10` (T# syntax)

"Syntax error"

Check: - No semicolons needed - One command per line - Proper indentation in blocks - Matching braces { }

Performance Issues

"Script runs slow"

Solutions: 1. Reduce loop iterations 2. Limit draw calls 3. Use simpler graphics 4. Cache calculations 5. Check for infinite loops

"Editor freezes"

Solutions: 1. Stop script (Shift+F5) 2. Close infinite loops 3. Add break conditions 4. Reduce complexity

Graphics Issues

"Graphics not showing"

1. Check Graphics tab is selected
2. Run script again
3. Clear graphics first
4. Check color isn't "#000000" on black background

"Colors wrong"

1. Use hex format: "#RRGGBB"
2. Include quotes: color "#ff0000"
3. Valid hex digits: 0-9, A-F

"Objects off-screen"

1. Check coordinates
2. Canvas is 800x600
3. Origin (0,0) is top-left
4. Stay within bounds

FILE ORGANIZATION

Recommended Structure

```
AtlasEngine1.0/
  editor/
      main.py
```

```
    editor_window.py
    ts_interpreter.py
    viewport_3d.py
    ...
scripts/
    games/
        fps_game.tcc
        rpg_maze.tcc
        adventure.tcc
    examples/
        hello.tcc
        graphics_test.tcc
        physics_test.tcc
    tests/
        test_input.tcc
        test_3d.tcc
docs/
    COMPLETE_GUIDE_PART1.md
    COMPLETE_GUIDE_PART2.md
    COMPLETE_GUIDE_PART3.md
    COMPLETE_GUIDE_PART4.md
    COMPLETE_GUIDE_PART5.md
README.md
```

Naming Conventions

Scripts

- Use descriptive names
- Use underscores: `my_game.tcc`
- Group by category
- Include version: `game_v1.tcc`

Variables

- Lowercase with underscores
- Descriptive: `player_health` not `ph`
- Constants in CAPS: `MAX_HEALTH`

Comments

```
# Single line comment
say "Hello"  # End of line comment

# Multi-line explanation:
# This section handles player movement
# and collision detection
```

COMMAND CATEGORIES SUMMARY

By Category

Text & Output (10)

say, shout, whisper, show, input, print, clear, cleargraphics, warn, error

Variables (15)

is, make, set, create, change, increase, decrease, increment, decrement, remember, forget, recall, exists, typeof, copy

Math (30)

add, subtract, multiply, divide, modulo, power, root, squared, cubed, round, floor, ceil, rounddown, min, max, clamp, sin, cos, tan, absolute, sign, percent, random, exp, ln, log, factorial, sum, average, product

Control Flow (15)

when, if, whenever, equals, notequals, greater, less, between, else, elseif, repeat, while, until, for, foreach, break, continue, return

Strings (20)

join, split, length, uppercase, lowercase, titlecase, trim, replace, substring, startswith, endswith, contains, indexof, count, reverse, padleft, padright, slice, pattern, convert

Lists (25)

list, append, prepend, insert, remove, pop, shift, first, last, indexof, contains, count, length, reverse, sort, unique, slice, merge, join, split, foreach

Graphics 2D (20)

drawline, drawrect, drawcircle, drawtext, fillscreen, sprite, movesprite, colorsprite, hidesprite, showsprite, deletesprite, arc, ellipse, triangle, polygon, bezier, gradient, animation, opacity, zindex

3D Objects (15)

create3d, color3d, scale, rotate, transform, mesh, model, anchor, billboard, wireframe, material, texture, metallic, roughness, emissive, opacity

Camera & View (10)

camera, lookat, zoom, pivot, orbit, clip, fov

Lighting (10)

pointlight, spotlight, directional, hemispheric, skylight, shadow, ambient

Effects (20)

particles, emitter, glow, fog, bloom, vignette, chromatic, grain, pixelate, blur, motionblur, dof, ssao, antialiasing, tonemapping, colorgrading, tint, fade

Physics (25)

gravity, velocity, force, impulse, torque, angular, bounce, drag, inertia, momentum, collision, raycast, buoyancy, magnetism, friction

Animation (15)

tween, ease, elastic, spring, pendulum, ballistic, homing, circular, sine_wave, trajectory, path, curve, spiral, parabola, wave

Game Stats (15)

health, heal, mana, stamina, armor, defense, attack, accuracy, critical, dodge, parry, block

Weapons (20)

weapon, equip, unequip, shoot, fire, cast, reload, ammo, magazine, spread, recoil, firerate, scope, aim, zoom, pistol, shotgun, rifle, bow, sword, laser, rocket, grenade, bomb, bullet, projectile

Combat (15)

hit, explode, burn, poison, freeze, stun, buff, debuff, enchant, cooldown

Inventory (10)

inventory, additem, removeitem, dropitem, useitem, hasitem, pickup, coin, gem, gold

Level/XP (10)

level, levelup, xp, exp, stat, ability, skill, powerup

Quest (5)

quest, objective, completequest, reward

Enemies (10)

enemy, spawn, ai, behavior, chase, flee, attack, patrol, attract, repel, summon

Game Flow (15)

gameover, lose, win, checkpoint, respawn, lives, score, highscore, timer, countdown, pause, resume, key, lock, unlock, door, trigger, zone, teleport, battle, combat, frame, framerate

Total: 334 Commands

QUICK START CHECKLIST

First Time Setup

- ☐ Install Python 3.8+
- ☐ Install Pillow: `pip install pillow`
- ☐ Extract AtlasEngine
- ☐ Run: `python editor\main.py`

Creating Your First Game

- ☐ Click New (Ctrl+N)
- ☐ Type script name
- ☐ Write T# code
- ☐ Run (F5)
- ☐ Check output

Test Each Feature

- ☐ Text output (say "Hello")
- ☐ Variables (x is 10)
- ☐ Math (sum is x + y)
- ☐ Graphics (drawcircle…)
- ☐ Input (input "Name:" into name)
- ☐ 3D (create3d cube…)

Learning Path

1. Hello World (5 min)
2. Variables & Math (10 min)

3. Control Flow (15 min)
4. Graphics 2D (20 min)
5. Text Adventure (30 min)
6. 2D Game (45 min)
7. 3D Basics (45 min)
8. Full Game (2+ hours)

HELP RESOURCES

In This Guide

- Part 1: Interface & Getting Started
- Part 2: Text, Variables, Math, Strings
- Part 3: Graphics 2D/3D, Physics
- Part 4: Game Mechanics, Examples
- Part 5: Reference, Tips, Troubleshooting (you are here)

Built-in Help

- Press F1 in editor
- Check Help menu
- View example scripts

Community

- GitHub: github.com/elianjamal
- Share your games!
- Report bugs
- Contribute

FINAL NOTES

You Now Know:

Every command in T# (334 total) Complete interface guide Graphics 2D and 3D Game mechanics Combat systems Complete examples Tips & tricks Troubleshooting

What You Can Make:

3D FPS Games 2D RPG Games Text Adventures Puzzle Games Physics Games Arcade Games

Next Steps:

1. Create your own game
2. Experiment with commands
3. Combine features
4. Share your creation
5. Help others learn

END OF COMPLETE GUIDE

AtlasEngine - Every Command, Every Feature, Everything!

Created by: elianjamal

You're ready to make amazing games!

Good luck and have fun creating!