Scratch-Pygame Documentation (PIEXGame)

Gordon.py is a file that manages and combines complex Pygame commands into a series of easy Scratch-like code that will help student progress to scratch easier.

Check base.py to understand the base of how to use this module.

# **Basic Game Design Command**

**Make new game manager and set screen to 800 and 600**

pygame = Pygame(800,600)

**Set Game Title**

pygame.setGameTitle(“My Game”)

**Add new sprite**

my\_sprite = pygame.addSprite(‘sprite1.png’, 50)

Make a new sprite with image ‘sprite1.png’ at initial scale size of 50 (Half of original size)

**Add Costume**

my\_sprite.switchCostume(‘sprite2.png’)

**Set Background/Backdrop Image**

pygame.changeBackgroundImage(“background.jpg”)

**Go to new scene / Remove all sprites in current screen**

pygame.newScene()

# **Motion**

**Move 10 steps**

my\_sprite.move(10)

**Turn Right 15 degrees**

my\_sprite.turnRight(15)

**Turn Left 15 degrees**

my\_sprite.turnLeft(15)

**Go to random position**

my\_sprite.goToRandomPosition()

**Go to mouse pointer**

Not implemented

**Go to another sprite**

Not implemented

**Go to specific position (x,y)**

my\_sprite.goTo(100,100)

**Glide to random position in 1 sec**

Not implemented

**Glide to specific position in 1 sec**

Not implemented

**Point in direction 90 / Set direction to 90**

1. my\_sprite.setDirection(90)
2. my\_sprite.setRotation(90)

**Point towards mouse**

my\_sprite.pointTowardMouse()

**Point towards another sprite**

my\_sprite.pointTowardSprite(another\_Sprite)

**Change x by 10**

my\_sprite.changeX(10)

**Set x to 100**

my\_sprite.setX(100)

**Change y by 10**

my\_sprite.changeY(10)

**Set y to 100**

my\_sprite.setY(100)

**If on edge, bounce**

my\_sprite.stopOnEdge()

**Set Rotation Style**

my\_sprite.setRotationStyle(‘all-around’)

my\_sprite.setRotationStyle(‘left-right’)

my\_sprite.setRotationStyle(‘don’t rotate)

**Get X Position**

X = my\_sprite.getXPosition()

**Get Y Position**

Y = my\_sprite.getYPosition()

**Get Direction**

Rotation = my\_sprite.getDirection()

# **Looks**

**Say “Hello!” for 2 seconds**

my\_sprite Not Implemented

**Say “Hello!”**

my\_sprite.say(“Hello”)

**Think “Hmm..” for 2 seconds**

Not Implemented

**Think “Hmm..”**

Not Implemented

**Switch costume to costume1**

my\_sprite.switchCostume(‘costume1.png’)

**Next costume**

Not Implemented

**Switch background/backdrop to backdrop1**

pygame.changeBackgroundImage(“background.jpg”)

**Next backdrop**

Not Implemented

**Change size by 10**

my\_sprite.changeSize(10)

**Set size to 100%**

my\_sprite.setSize(100)

**Change color effect by 25**

Not Implemented

**Set color effect to 0**

Not Implemented

**Clear graphic effects**

Not Implemented

**Show**

my\_sprite.show()

**Hide**

my\_sprite.hide()

**Go to Front Layer**

Not Implemented

**Go Forward 1 Layers**

Not Implemented

**Get Costume Number**

Not Implemented

**Get Costume Name**

Not Implemented

**Get Background Number**

Not Implemented

**Get Background Name**

Not Implemented

**Get Image Size**

image\_size = my\_sprite.getSize()

# **Sound**

**Play sound Meow until done**

Not Implemented

**Start sound Meow**

pygame.playSoundEffect(‘effect1.mp3’)

pygame.playBackgroundMusic(‘music1.mp3’)

**Stop all sounds**

pygame.stopSound()

**Change pitch effect by 10**

Not Implemented

**Set pitch effect to 100**

Not Implemented

**Clear sound effects**

Not Implemented

**Change volume by -10**

Not Implemented

**Set volume to 100%**

Not Implemented

**Get Volume**

Not Implemented

# **Sensing**

**Touching mouse**

result = my\_sprite.mouseHoveredOnSprite() [TRUE/FALSE]

result = my\_sprite.mouseClickedOnSprite() [TRUE/FALSE]

**Touching edge**

Not Implemented

**Touching another sprite**

result = my\_sprite.touch(another\_sprite) [TRUE/FALSE]

**Distance to mouse**

Not Implemented

**Distance to another sprite**

Not Implemented

**Key ‘Space’ pressed**

result = keyHold(‘Space’) [TRUE/FALSE]

result = keyPressed(‘Space’) [TRUE/FALSE]

result = keyReleased(‘Space’) [TRUE/FALSE]

**Mouse down?**

result = checkMouseClicked(self) [TRUE/FALSE]

**Mouse x**

Not Implemented

**Mouse y**

Not Implemented