**Event Detection Part I: Keyboard**

An **event** is an action (from the user, usually) such as a mouse click, a mouse move, a key click. Pygame has an event class that allows us to detect these events using the **get()** function.

Inside your while loop (i.e. your game loop) we can detect various events. We have already used this to detect the exit button being pressed:

gameOn = True

while gameOn:

for event in pygame.event.get(): #what events are happening now?

if event.type == pygame.QUIT: # is it the quit button?

gameOn = False

break

The while loop allows the program to continuously check for events. Events (such as mouse clicks, keys pressed) are logged into an array of events.

The for loop goes through all the events listed in the array.

The if statement is used to see if the event was to quit. There are several event types, but here are the ones we are most interested in here are:

pygame.QUIT

pygame.KEYDOWN

pygame.KEYUP

**Event Codes**

Why are these events upper case? In most programming languages, **constants** are uppercase. Each event is really just a number representing the event. For example, QUIT is 12. In the pygame module, each event is given a code:

QUIT = 12

KEYDOWN = 2 etc.

You do not need to know the codes, but we do need to know the names of them. These codes are constants, not variables, so they are always uppercase. A reminder about case in programming:

modules – lower case - example: pygame.event

variables – lower case - example: x = 3

functions – lower case but ending with parenthesese – example: event.get()

classes – first letter upper case – example: mixer.Sound - Sound is a class.

constants – all upper case – example: even.QUIT

Now let’s add a check for a key being pressed:

gameOn = True

while gameOn:

for event in pygame.event.get():

if event.type == pygame.QUIT:

sys.exit()

elif event.type == pygame.KEYDOWN:

sound.play()

This example assumes you have a sound object created called “sound”. When any key is pressed, the sound is played.

We usually want to check for a particular key, so we add a condition:

gameOn = True

while gameOn:

for event in pygame.event.get():

if event.type == pygame.QUIT:

sys.exit()

elif event.type == pygame.KEYDOWN:

if event.key == pygame.K\_ESCAPE: # check the escape key

sound.play()

**Exercise:**

Try detecting various events – keys pressed (or released), mouse buttons, mousemovement.

For keys, each key has a key code and a constant that goes with it. Here is the complete list (it’s long!)

**KeyASCII ASCII Common Name**

K\_BACKSPACE \b backspace

K\_TAB \t tab

K\_CLEAR clear

K\_RETURN \r return

K\_PAUSE pause

K\_ESCAPE ^[ escape

K\_SPACE space

K\_EXCLAIM ! exclaim

K\_QUOTEDBL " quotedbl

K\_HASH # hash

K\_DOLLAR $ dollar

K\_AMPERSAND & ampersand

K\_QUOTE quote

K\_LEFTPAREN ( left parenthesis

K\_RIGHTPAREN ) right parenthesis

K\_ASTERISK \* asterisk

K\_PLUS + plus sign

K\_COMMA , comma

K\_MINUS - minus sign

K\_PERIOD . period

K\_SLASH / forward slash

K\_0 0 0

K\_1 1 1

K\_2 2 2

K\_3 3 3

K\_4 4 4

K\_5 5 5

K\_6 6 6

K\_7 7 7

K\_8 8 8

K\_9 9 9

K\_COLON : colon

K\_SEMICOLON ; semicolon

K\_LESS < less-than sign

K\_EQUALS = equals sign

K\_GREATER > greater-than sign

K\_QUESTION ? question mark

K\_AT @ at

K\_LEFTBRACKET [ left bracket

K\_BACKSLASH \ backslash

K\_RIGHTBRACKET ] right bracket

K\_CARET ^ caret

K\_UNDERSCORE \_ underscore

K\_BACKQUOTE ` grave

K\_a a a

K\_b b b

K\_c c c

K\_d d d

K\_e e e

K\_f f f

K\_g g g

K\_h h h

K\_i i i

K\_j j j

K\_k k k

K\_l l l

K\_m m m

K\_n n n

K\_o o o

K\_p p p

K\_q q q

K\_r r r

K\_s s s

K\_t t t

K\_u u u

K\_v v v

K\_w w w

K\_x x x

K\_y y y

K\_z z z

K\_DELETE delete

K\_KP0 keypad 0

K\_KP1 keypad 1

K\_KP2 keypad 2

K\_KP3 keypad 3

K\_KP4 keypad 4

K\_KP5 keypad 5

K\_KP6 keypad 6

K\_KP7 keypad 7

K\_KP8 keypad 8

K\_KP9 keypad 9

K\_KP\_PERIOD . keypad period

K\_KP\_DIVIDE / keypad divide

K\_KP\_MULTIPLY \* keypad multiply

K\_KP\_MINUS - keypad minus

K\_KP\_PLUS + keypad plus

K\_KP\_ENTER \r keypad enter

K\_KP\_EQUALS = keypad equals

K\_UP up arrow

K\_DOWN down arrow

K\_RIGHT right arrow

K\_LEFT left arrow

K\_INSERT insert

K\_HOME home

K\_END end

K\_PAGEUP page up

K\_PAGEDOWN page down

K\_F1 F1

K\_F2 F2

K\_F3 F3

K\_F4 F4

K\_F5 F5

K\_F6 F6

K\_F7 F7

K\_F8 F8

K\_F9 F9

K\_F10 F10

K\_F11 F11

K\_F12 F12

K\_F13 F13

K\_F14 F14

K\_F15 F15

K\_NUMLOCK numlock

K\_CAPSLOCK capslock

K\_SCROLLOCK scrollock

K\_RSHIFT right shift

K\_LSHIFT left shift

K\_RCTRL right ctrl

K\_LCTRL left ctrl

K\_RALT right alt

K\_LALT left alt

K\_RMETA right meta

K\_LMETA left meta

K\_LSUPER left windows key

K\_RSUPER right windows key

K\_MODE mode shift

K\_HELP help

K\_PRINT print screen

K\_SYSREQ sysrq

K\_BREAK break

K\_MENU menu

K\_POWER power

K\_EURO euro