**Cropping Images**

It is often useful to take a part of an image. To do this, we use the subsurface() method. Here’s how it works:

Begin with an image. For this example, I will use my elephant.png image (found in data folder):

import pygame

width = 500

height = 400

size = width, height

screen = pygame.display.set\_mode(size) # set the dimensions of screen

green = 0, 255, 0 # create a RGB tuple for green colour

screen.fill(green)

pic = pygame.image.load("elephant.png") # load the pic from the file

screen.blit(pic,(0,0)) # put it on the screen

pygame.display.flip() # flip screen on to window

This program does the following:

* sets up the screen to 500 by 400
* fills the background with green
* loads the png image from a file
* puts it on the screen

The nice thing about this png file is that the background of the elephant is transparent – you can’t see the rectangle surrounding the picture. This is not possible with some formats, such as JPEG or raw BMP.

Now let's crop the elephant.

To get a small sample of the elephant, we use the subsurface() method. The added lines are in blue:

import pygame

width = 500

height = 400

size = width, height

screen = pygame.display.set\_mode(size) # set the dimensions of screen

green = 0, 255, 0 # create a RGB tuple for green colour

screen.fill(green)

pic = pygame.image.load("elephant.png") # load the pic from the file

screen.blit(pic,(0,0)) # put it on the screen

newpic = pic.subsurface(200,200, 80, 80) # get a small selection of pic

screen.blit(newpic,(100,100)) # add it to screen

pygame.display.flip() # flip screen on to window

The first line in blue gets a sample (called a subsurface). It takes the sample starting from a top-left corner at (200,200) and 80 pixels wide by 80 pixels high. The second line in blue places that sample at 100,100.

A common trick is to put sections of a picture into a list of images. Heres how you could split the elephant into 4 pieces:

import pygame

screen\_width = 640

screen\_height = 520

size = screen\_width, screen\_height

screen = pygame.display.set\_mode(size) # set the dimensions of screen

green = 0, 255, 0 # create a RGB tuple for green colour

screen.fill(green)

pic = pygame.image.load("elephant.png") # load the pic from the file

pic\_width, pic\_height = pic.get\_size() # get dimensions of pic

slice\_width = pic\_width/4 # size of each slice

slice\_height = pic\_height/4 # size of each slice

pics = [] # start with empty list of pics

for i in range (4):

pics.append( pic.subsurface(i\*slice\_width,0, slice\_width, pic\_height)) # put pics into list

screen.blit(pics[0],(100,0)) # add slice to screen

screen.blit(pics[2],(500,0)) # add slice to screen

pygame.display.flip() # flip screen on to window

The key here is that the dimensions of the whole pic is found and then each slice is calculated for width and height.