COMP120: Creative Computing: Tinkering

6: Tinkering Graphics III

Learning Outcomes

- **Apply** iteration **and** nested iteration to **write** a program that manipulates part of a raster image
- **Copy** an image from one array to another

FALMOUTH
UNIVERSITY

Question

Source Code: Moonflower

```
def make_flowers_moon_colour(picture):
  moon_pixel_matrix = get_pixels(MOON)
  pixel_matrix = get_pixels(picture)
  for pixel in pixel_matrix:
    # (1) if distance(get_colour(pixel), RED) > 200:
    # (2) if get_red(pixel) > 100 and ( get_blue(p) +
        get_green(p) < 100 ):
    # (3) if get_red(pixel) > 150 and ( get_blue(p) +
        get_green(p) < 200):
    # (4) if get_red(pixel) > ( get_blue(p) +
        get_green(p) ):
        x = get_x(pixel)
        y = get_y(pixel)
        moon_colour = get_colour(get_pixel(
            moon_pixel_matrix, x, y))
        set_pixel_colour(pixel, moon_colour)
```

Note: This source code excerpt will not work in PyGame.

Question: Moon Flower

Socrative room code: ---

Which of the below conditions were used to generate the moon flower effect:

- (1) `if distance(get_colour(pixel), RED)> 200:`
- (2) `if get_red(pixel)> 100 and ( get_blue(p)+ get_green(p)< 100 ):`
- (3) `if get_red(pixel)> 150 and ( get_blue(p)+ get_green(p)< 200):`
- (4)

```
if get_red(pixel)> ( get_blue(p)+ get_green(p)):
```

FALMOUTH
UNIVERSITY

General Assignment Support

Questionnaire

- Check your emails!
- Please complete the mid-term questionnaire that has been sent to you
- These will feeds-forward into the design for next semester
- Forward any concerns and/or issues and/or suggestions to your student reps

General Assessment Support

- Continue work on your tinkering graphics assignment with your pair programming partner
- Ensure that both partners have pushed new code to the repository
- Create a pull-request before the end of the session

FALMOUTH
UNIVERSITY

Question

Source Code: Manipulation (1)

```
def manipulate(picture):
  width = get_width(picture)
  height = get_height(picture)
  for x in xrange(0, width):
      for y in xrange(0, height / 2):
          pixel = get_pixel(picture, x, y)
          red = get_red(pixel)
          set_red(pixel, red / 2)
      for y in xrange(height / 2, height):
          pixel = get_pixel(picture, x, y)
          red = get_red(pixel)
          set_red(pixel, red * 2)
```

Note: This source code excerpt will not work in PyGame.

Source Code: Manipulation (2)

```
def manipulate(picture):
  width = get_width(picture)
  height = get_height(picture)
  for x in xrange(0, width / 2):
      for y in xrange(0, height / 2):
          pixel = get_pixel(picture, x, y)
          red = get_red(pixel)
          set_red(pixel, red * 2)
      for y in xrange(height / 2, height):
          pixel = get_pixel(picture, x, y)
          red = get_red(pixel)
          set_red(pixel, red / 2)
```

Note: This source code excerpt will not work in PyGame.

Source Code: Manipulation (3)

```
def manipulate(picture):
  width = get_width(picture)
  height = get_height(picture)
  for x in xrange(0, width):
      for y in xrange(0, height / 3):
          pixel = get_pixel(picture, x, y)
          red = get_red(pixel)
          set_red(pixel, red * 2)
      for y in xrange(height / 2, height):
          pixel = get_pixel(picture, x, y)
          red = get_red(pixel)
          set_red(pixel, red / 2)
```

Note: This source code excerpt will not work in PyGame.

Question: Manipulation

Socrative room code: ---

Which of the code listings manipulated the lecture hall image:

- (1)
- (2)
- (3)

FALMOUTH
UNIVERSITY

Segmentation and Collages


mediasources\blueMotorcycle.jpg
Zoom
X:
0
Y:
0
R: 73 G: 81 B: 32 Color at location:

Source Code: Mirroring (1)

```
def mirror_vertical(picture):
  width = get_width(picture)
  height = get_height(picture)
  mirror_point = width / 2
  for y in xrange(0, height):
      for x in xrange(0, mirror_point):
          left_pixel = get_pixel(picture, x, y)
          right_pixel = get_pixel(picture, width - x - ↩
              1, y)
          set_colour(right_pixel, get_colour( ↩
              left_pixel))
```

Source Code: Mirroring (2)

```
def mirror_vertical(picture):
  width = get_width(picture)
  height = get_height(picture)
  mirror_point = height / 2
  for x in xrange(0, width):
      for y in xrange(0, mirror_point):
          top_pixel = get_pixel(picture, x, y)
          bottom_pixel = get_pixel(picture, x, height
              - y - 1)
          set_colour(bottom_pixel, get_colour(
              top_pixel))
```

Activity: Mirroring

In pairs:

- Integrate mirroring into your tinkering graphics project
- Add an argument to change which side of the mirror is rendered (i.e., left-into-right, or right-into-left)
- 20 minutes

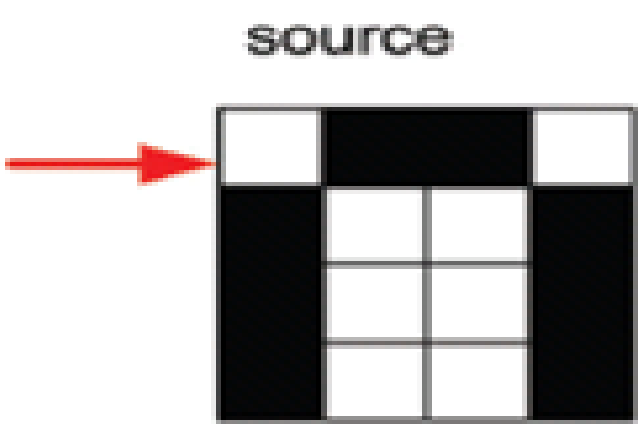
Activity: Mirroring

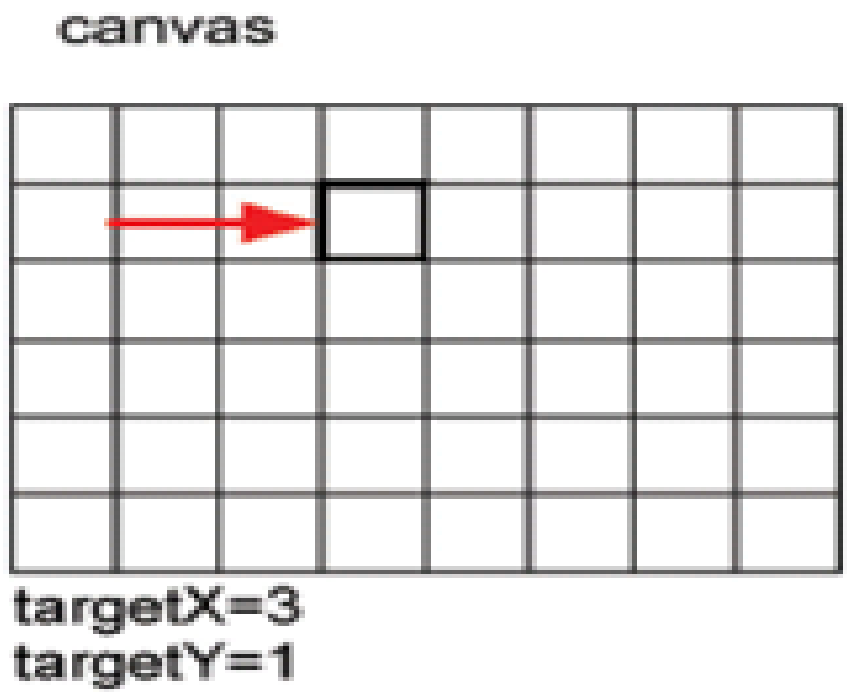
In pairs:

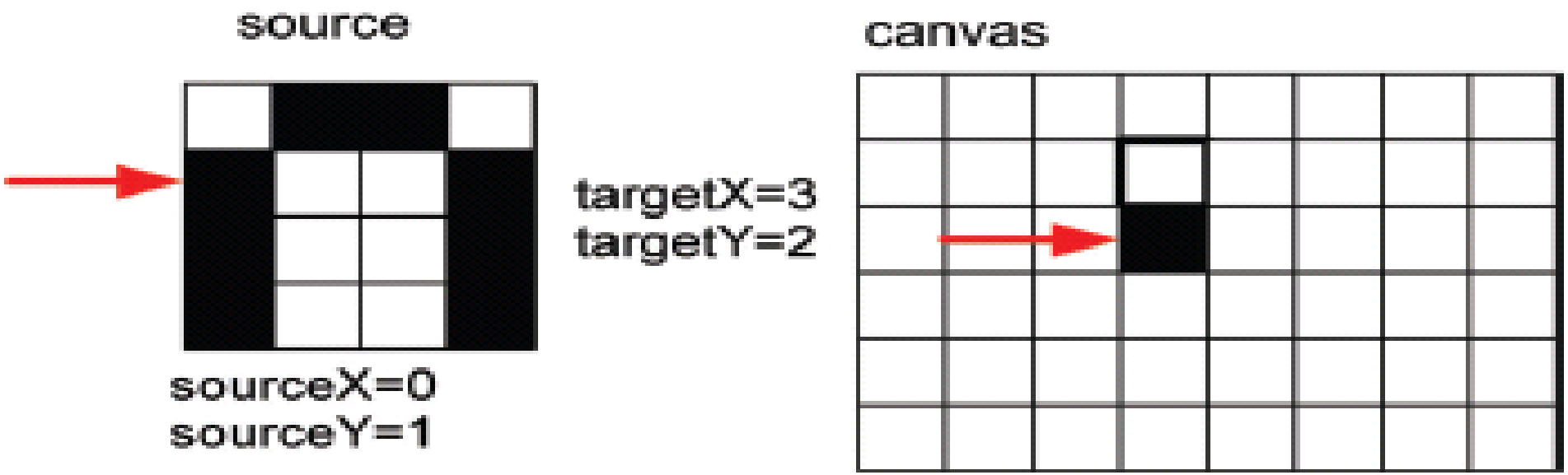
- Use your function to repair the temple
- 20 minutes

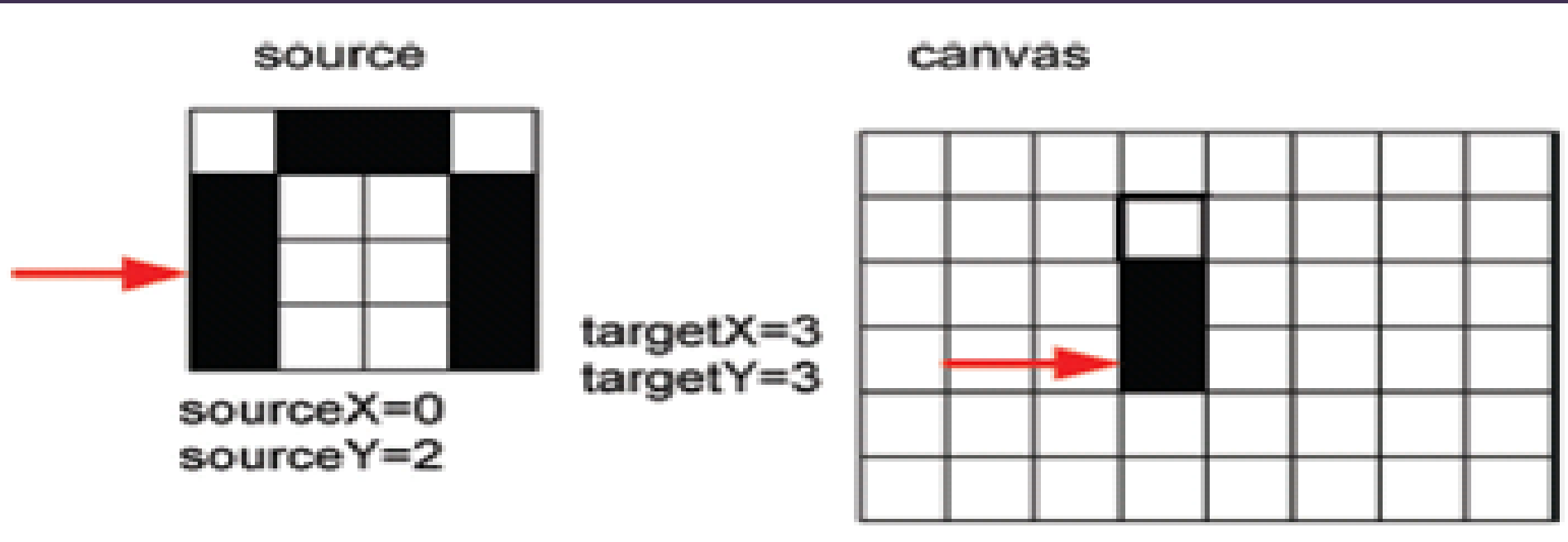
Source Code: Collage

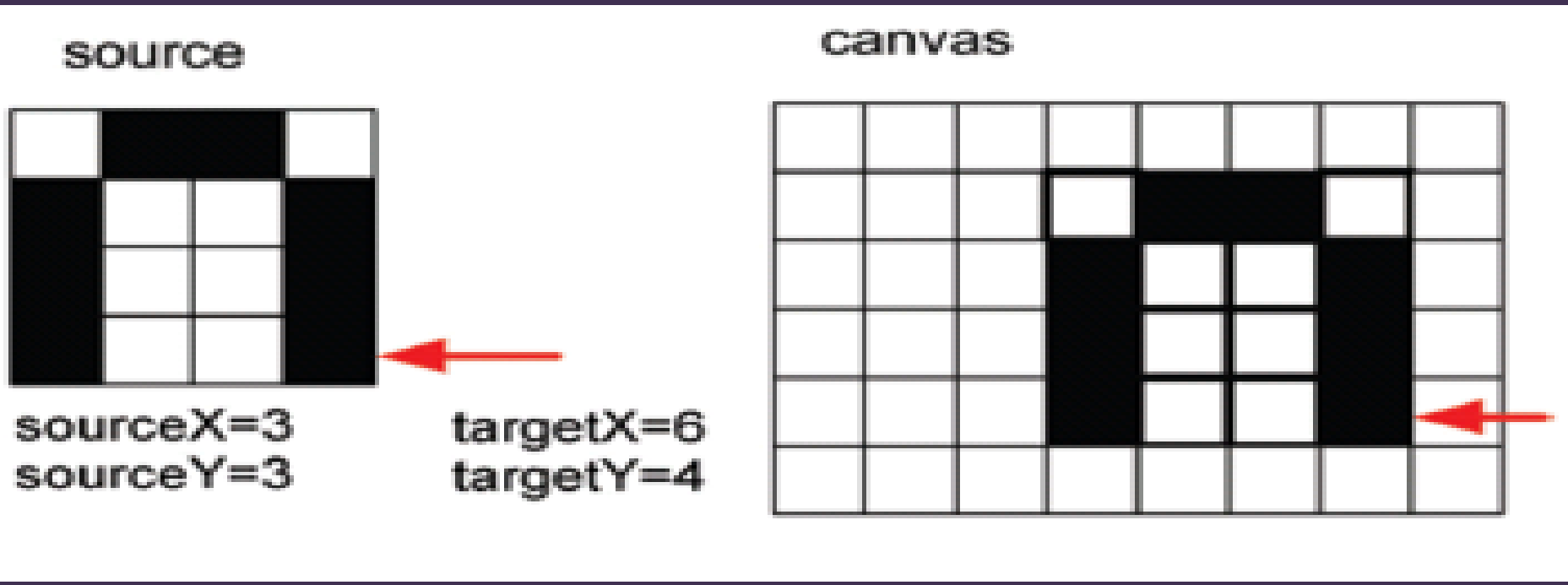
```
def copyBarb():
  # Set up the source and target pictures
  barbf=getMediaPath("barbara.jpg")
  barb = makePicture(barbf)
  canvasf = getMediaPath("7inX95in.jpg")
  canvas = makePicture(canvasf)
  # Now, do the actual copying
  targetX = 0
  for sourceX in range(0,getWidth(barb)):
    targetY = 0
    for sourceY in range(0,getHeight(barb)):
      color = getColor(getPixel(barb,sourceX,sourceY))
      setColor(getPixel(canvas,targetX,targetY), color
          )
      targetY = targetY + 1
    targetX = targetX + 1
  show(barb)
  show(canvas)
  return canvas
```

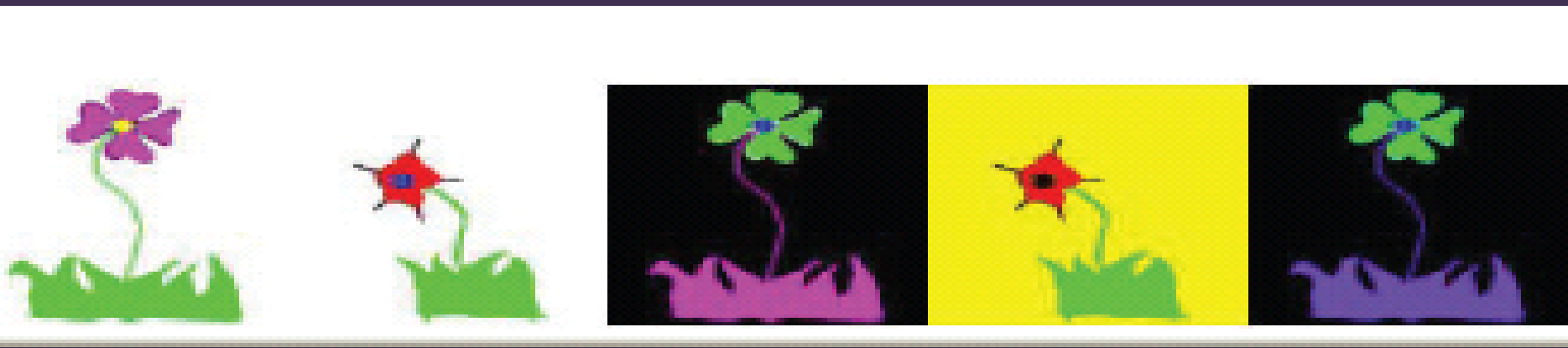

source
sourceX=0
sourceY=0


canvas
targetX=3
targetY=1


source
canvas
targetX=3
targetY=2
sourceX=0
sourceY=1


source
canvas
targetX=3
targetY=3
sourceX=0
sourceY=2


source
canvas
sourceX=3
sourceY=3
targetX=6
targetY=4



Activity: Collage

In pairs:

- Find some smaller images online
- Integrate the copy algorithm into your tinkering graphics project
- Create a collage of the images you found
- 20 minutes

Sprite Sheets and Animations

Review Al Swigart's pyganim python module:

`http://inventwithpython.com/pyganim/`

Activity: Sprite Sheets

In pairs:

- Find a sprite sheet online
- Integrate pyganim into your tinkering graphics project
- Animate something
- 20 minutes