Pygame으로 게임만들기 1

시작하기 전에

- 본 수업은
  python과 pygame으로 게임만들기
  (Making Games with Python & pygame)
  와 인터넷 자료를 참조하였다.
- 인터넷 자료는 슬라이드 노트에 출처를 명시하였다.
- 영문판 책은 공개되어 있으니 다운로드받아 읽을 수 있다.
  - http://inventwithpython.com/pygame/
- pygame documentation
  - http://www.pygame.org/docs/

Pygame의 설치

- pip를 통한 설치
  - pip란?
    - python으로 작성된 패키지를 설치 및 관리하는 시스템
    - 업데이트도 관리해준다.
  - pip업데이트
    - 패키지를 설치할 때, 오류를 방지하기 위해서 pip를 먼저 업데이트 해야 된다.

pip

- 모든 명령어는 명령프롬프트에서 입력한다.
- pip list
  - 설치되어 있는 패키지 리스트가 나타난다.

```
C:₩Users₩KJH>pip list
DEPRECATION: The default format will switch to columns in the future. You can us
e --format=(legacy|columns) (or define a format=(legacy|columns) in your pip.con
f under the [list] section) to disable this warning.
pip (9.0.1)
setuptools (28.8.0)
```

pip

- pip list의 목록 확인방식
  - pip list --format=legacy
  - pip list --format=columns

```
C:₩Users₩WKJH>pip list --format=legacy
pip (9.0.1)
setuptools (38.4.0)

C:₩Users₩WKJH>pip list --format=columns
Package    Version
---------- -------
pip        9.0.1
setuptools 38.4.0
```

pip

- pip 업데이트

  pip install --upgrade pip

- 설치된 패키지 업데이트 확인

```
C:\Users\WKJH>pip list --outdated
DEPRECATION: The default format will switch to columns in the future. You can us
e --format=(legacy|columns) (or define a format=(legacy|columns) in your pip.con
f under the [list] section) to disable this warning.
setuptools (28.8.0) - Latest: 38.4.0 [wheel]
```

pip

- 설치된 패키지의 업그레이드
  - pip install --upgrade <패키지명>

  ex) setuptools의 업그레이드

```
C:₩Users₩KJH>pip install --upgrade pip
Requirement already up-to-date: pip in c:₩users₩kjh₩appdata₩local₩programs₩pytho
n₩python36₩lib₩site-packages

C:₩Users₩KJH>pip install --upgrade setuptools
Collecting setuptools
  Downloading setuptools-38.4.0-py2.py3-none-any.whl (489kB)
    100% |████████████████████████████████| 491kB 669kB/s
Installing collected packages: setuptools
  Found existing installation: setuptools 28.8.0
    Uninstalling setuptools-28.8.0:
      Successfully uninstalled setuptools-28.8.0
Successfully installed setuptools-38.4.0
```

pip

- pip를 통한 패키지 설치
  - pip install <패키지명>

  ex) pip install pygame

```
C:₩Users₩KJH>pip install pygame
Collecting pygame
  Downloading pygame-1.9.3-cp36-cp36m-win_amd64.whl (4.2MB)
    100% |████████████████████████████████| 4.2MB 224kB/s
Installing collected packages: pygame
Successfully installed pygame-1.9.3
```

pip

- pip 패키지 삭제
  - pip uninstall <패키지명>

CLI vs CUI

- CLI(Command Line Interface, 명령어 인터페이스)
  - 문자열을 입력하여 조작하는 것
  - ex) pip
- GUI(Graphical User Interface, 그래픽 사용자 인터페이스)
  - 프로그램의 조작을 그래픽으로 하는 것
  - ex) windows

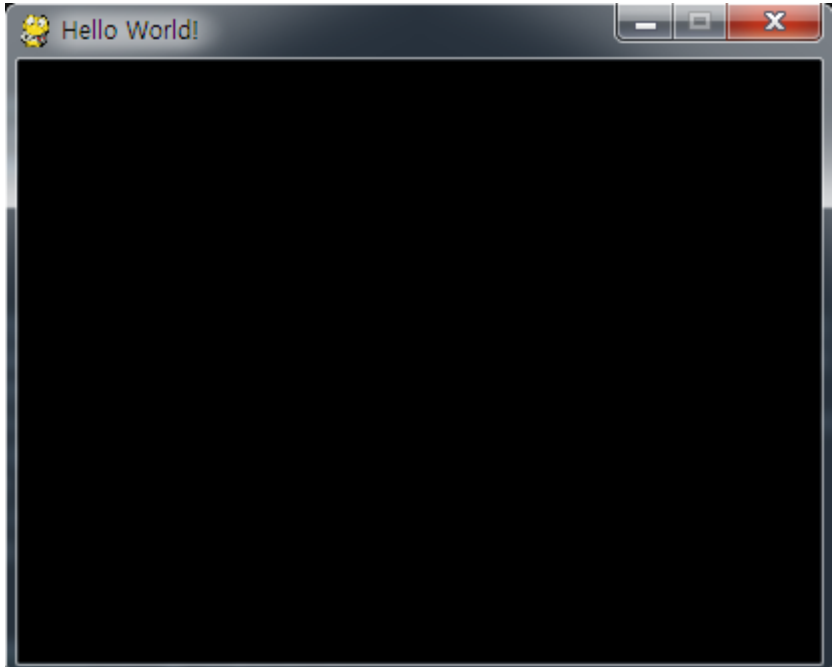
Hello World

- Pycharm에 입력해보자
  - 슬라이드 노트에 입력 값이 있다.

```
1. import pygame, sys
2. from pygame.locals import *
3.
4. pygame.init()
5. DISPLAYSURF = pygame.display.set_mode((400, 300))
6. pygame.display.set_caption('Hello World!')
7. while True: # main game loop
8.     for event in pygame.event.get():
9.         if event.type == QUIT:
10.            pygame.quit()
11.            sys.exit()
12.    pygame.display.update()
```

Hello World

- 실행하면 다음과 같은 창이 뜬다.
- 상단타이틀바(캡션)의 이름이 Hello World 이다.
- X버튼을 누르면 종료



Hello world

```
1. import pygame, sys
2. from pygame.locals import *
```

- 첫 번째 행은 pygame과 sys모듈을 가져온다. Pygame내의 모든 모듈을 가져오게 된다.
- 두 번째 행은 모듈의 이름을 생략하고 사용하도록 import하는데, locals내에 있는 상수가 빈번하게 사용되기 때문이다.

Hello World

4. pygame.init()

- pygame함수를 호출하기 전에 반드시 수행해야 되는 함수이다.

5. DISPLAYSURF =₩

pygame.display.set_mode((400, 300))

- pygame.Surface객체(정확하게는 display surface)를 반환하는 함수.
- list나 tuple로 넘겨주어야 된다.

Hello World

- Surface 객체
  - 정사각형 2D이미지를 나타내는 객체
  - 이 객체의 픽셀은 pygame그리기 함수를 호출해서 바꿀 수 있다.
  - set_mode()를 통해 반환된 surface객체인 display surface는 pygame.display.update()를 호출했을 때, 위에 그려져 있는 것을 전부 보이게 만들어준다.

Hello World

- Surface 객체
  - surface객체를 화면에 표시하는 것 보다는 surface객체에 그리는 것 메모리 값만 바꾸는 것이라 훨씬 빠르다.
  - 여러 개의 물체를 그릴 때, surface 객체 위에 필요한 것을 다 그리고 스크린에 그리는 것이 표현속도상 좋다.
  - 이 한 장면을 프레임이라고 하는데, 이번에 만들 프로그램들은 30-60fps(Frame per second)를 가진다.

Hello World

- 게임 루프와 게임 스테이트
  - while True # main game loop
  - 메인 게임 루프에서는 3가지 일을 수행한다.
    - 이벤트 다루기
    - 게임 state 업데이트
    - 게임 state를 스크린에 그리기

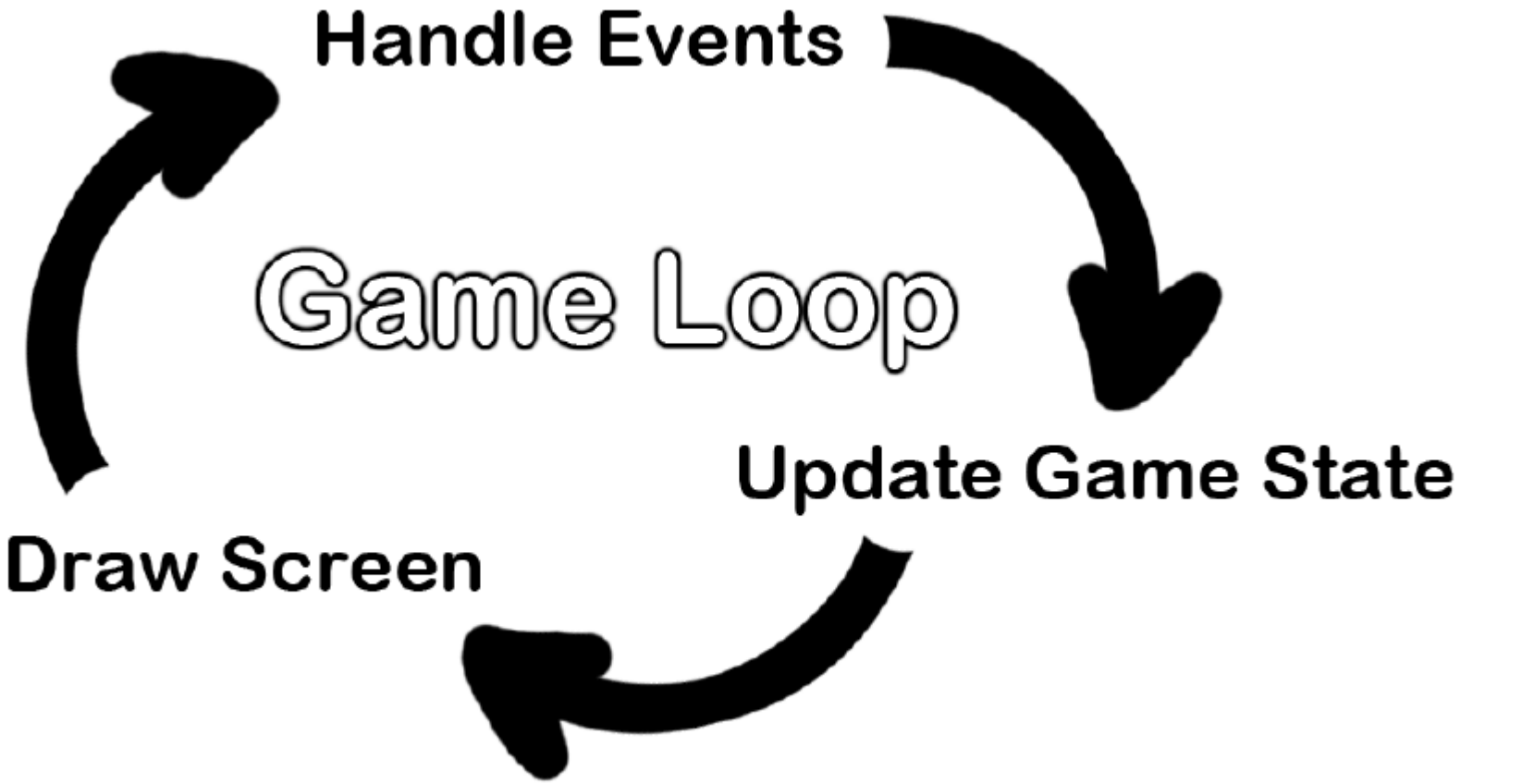
Hello World

- 게임 루프와 게임 스테이트
  - 게임 스테이트는 게임 프로그램 안에서 사용하는 모든 변수의 값을 통칭한다.
  - 플레이어의 상태 정보나 위치, 적의 상태 등을 포함되고, 보통 이런 정보들은 점수나 차례를 결정하게 된다.
  - 플레이어나 적이 피해를 입거나 위치가 움직이거나 이벤트가 발생했으면 이를 모두 게임스테이트가 바뀌었다고 말한다.
  - 게임을 저장할 때 저장되는 정보들이라 생각하면 된다.

Hello World

- 게임 루프와 게임 스테이트
  - 게임 스테이트는 마우스를 클릭하거나 키보드를 누르는 특정이벤트나 시간이 지나면 바뀐다.
  - 따라서 게임 루프에서는 새로운 이벤트가 발생했는지 항상 검사해야 된다.
  - 또한, 게임 루프에서는 이벤트에 따라서 게임 스테이트를 업데이트한다.
  - 이를 이벤트 핸들링이라 한다.

Hello World



Hello World

- pygame.event.Event객체(Event객체)
  - 사용자가 마우스를 움직이거나 키보드를 누르는 행동을 기록하기 위해 pygame에서는 event객체를 만든다.
  - pygame.event.get()을 호출하면 Event객체의 목록을 반환하고, Event객체의 목록은 본 함수가 호출된 다음 발생한 이벤트를 가지고 있다
  - Event객체에는 사용자가 발생시킨 이벤트를 각 객체로 가지고 있다.(마우스를 클릭하면 마우스 클릭 관련 객체 등)

Hello World

8 for event in pygame.event.get():

- Event객체가 가지고 있는 목록에서 각 이벤트 객체가 event변수에 할당된다.

9 if event.type == QUIT:

- 각 이벤트 객체는 type이라는 맴버 변수를 가지고 있어 이벤트 객체의 종류를 알 수 있다. pygame에서는 발생할 수 있는 이벤트 타입들을 pygame.locals에 정의해 두었다.

Hello World

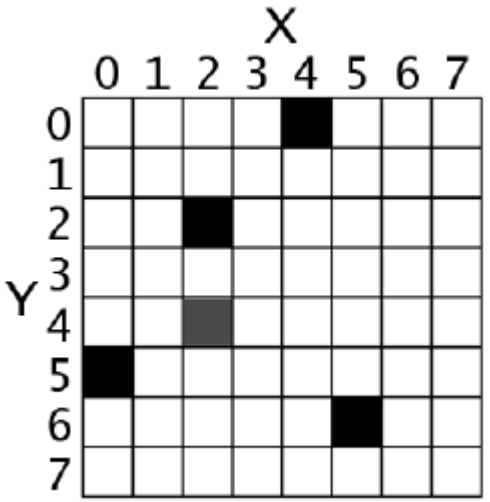
9 pygame.quit()

10 sys.exit()

- pygame.quit()은 pygame.init()의 반대되는 함수로 pygame라이브러리를 종료한다.
- sys.exit()는 파이썬을 종료한다.
- 일반적으로 프로그램이 종료되면, pygame 라이브러리가 종료되지만, 오류가 있을 가능성이 있어 명시적으로 pygame을 종료 해준다.

픽셀좌표

- 화면을 구성하는 하나의 단위를 픽셀(화소, pixel)이라고 한다.
- 수학과 유사하지만, y축은 내려가는 방향이 커지는 방향이다.
- 직교좌표계를 사용하여 나타내며 2개의 숫자 쌍으로 표현한다.
  (4, 0), (2, 2)



색

- 빛의 삼원색은 빨강, 초록, 파랑(RGB)이다.
- pygame에서는 색을 세 쌍의 정수값으로 표현하며, 0에서 255까지 값을 가진다.
- 따라서 각 변수에 색상정보를 저장시켜놓고 사용할 수 있다.

색

Color	RGB Values
Aqua	(0, 255, 255)
Black	(0, 0, 0)
Blue	(0, 0, 255)
Fuchsia	(255, 0, 255)
Gray	(128, 128, 128)
Green	(0, 128, 0)
Lime	(0, 255, 0)
Maroon	(128, 0, 0)
Navy Blue	(0, 0, 128)
Olive	(128, 128, 0)
Purple	(128, 0, 128)
Red	(255, 0, 0)
Silver	(192, 192, 192)
Teal	(0, 128, 128)
White	(255, 255, 255)
Yellow	(255, 255, 0)

투명

- pygame에서는 투명도를 조정할 수 있다. (보통 비트맵이나 jpg에서는 알파값이 없어 투명도 조정이 안된다.)
- 투명한 값을 알파 값(alpha value)라고 하고 이것은 색이 얼마나 불투명한지 나타낸다.
- ex) (0, 255, 0, 128)이면 반투명한 녹색이다.

투명

- 투명도를 조절하기 위해서는 convert_alpha() 메소드로 surface객체를 만들어야 된다.

anoterSurface=DISPLAYSURF..convert_alpha()

- surface객체에 무엇인가를 그리고 anotherSurface변수에 저장하면another surface는 DISPLAYSURF로 복사되어 화면에 보이게 된다.

pygame.Color 객체

- pygame의 그리기 함수를 쓰려면 어떤 색으로 그림을 그리기 위해서는 색상정보를 넘겨주어야 된다.
- 앞과 같이 세 쌍의 정수나 네 쌍의 정수를 사용할 수 있지만, pygame.Color 객체를 사용할 수도 있다.
- pygame.Color()생성자 함수를 호출할 때, 세 쌍이나 네 쌍의 정수를 넘겨줌으로써 color객체를 만들 수 있다.

pygame.Color 객체

```
>>> import pygame
>>> pygame.Color(255, 0, 0)
(255, 0, 0, 255)
>>> myColor = pygame.Color(255, 0, 0, 128)
>>> myColor == (255, 0, 0, 128)
True
>>>
```

- 정수 쌍과 color객체는 데이터 타입이 다르지만, 동일한 색이면 동일한 값을 가지고 있는 것을 알 수 있다.

Rect객체

- 색에서 2가지 방법을 사용한 것처럼, 사각형을 그릴 때도 2가지 방법을 사용할 수 있다.
- (왼쪽 상단 모서리 x좌표, 왼쪽 상단 모서리 y좌표, 사각형의 너비(픽셀), 사각형의 높이(픽셀))
- 또는 pygame.Rect객체를 사용한다.

```
>>> import pygame
>>> spamRect = pygame.Rect(10, 20, 200, 300)
>>> spamRect == (10, 20, 200, 300)
True
```

Rect객체

- Rect객체는 편리하게 다른 필요한 좌표들을 자동으로 계산하고, 다른 값으로 바뀌면 다른 좌표도 자동으로 계산된다.

```
>>> spamRect.right
210
```

```
>>> spam.right = 350
>>> spam.left
150
```

Attribute Name	Description
`myRect.left`	The int value of the X-coordinate of the left side of the rectangle.
`myRect.right`	The int value of the X-coordinate of the right side of the rectangle.
`myRect.top`	The int value of the Y-coordinate of the top side of the rectangle.
`myRect.bottom`	The int value of the Y-coordinate of the bottom side.
`myRect.centerx`	The int value of the X-coordinate of the center of the rectangle.
`myRect.centery`	The int value of the Y-coordinate of the center of the rectangle.
`myRect.width`	The int value of the width of the rectangle.
`myRect.height`	The int value of the height of the rectangle.
`myRect.size`	A tuple of two ints: (width, height)
`myRect.topleft`	A tuple of two ints: (left, top)
`myRect.topright`	A tuple of two ints: (right, top)
`myRect.bottomleft`	A tuple of two ints: (left, bottom)
`myRect.bottomright`	A tuple of two ints: (right, bottom)
`myRect.midleft`	A tuple of two ints: (left, centery)
`myRect.midright`	A tuple of two ints: (right, centery)
`myRect.midtop`	A tuple of two ints: (centerx, top)
`myRect.midbottom`	A tuple of two ints: (centerx, bottom)

기본 형태 그리기

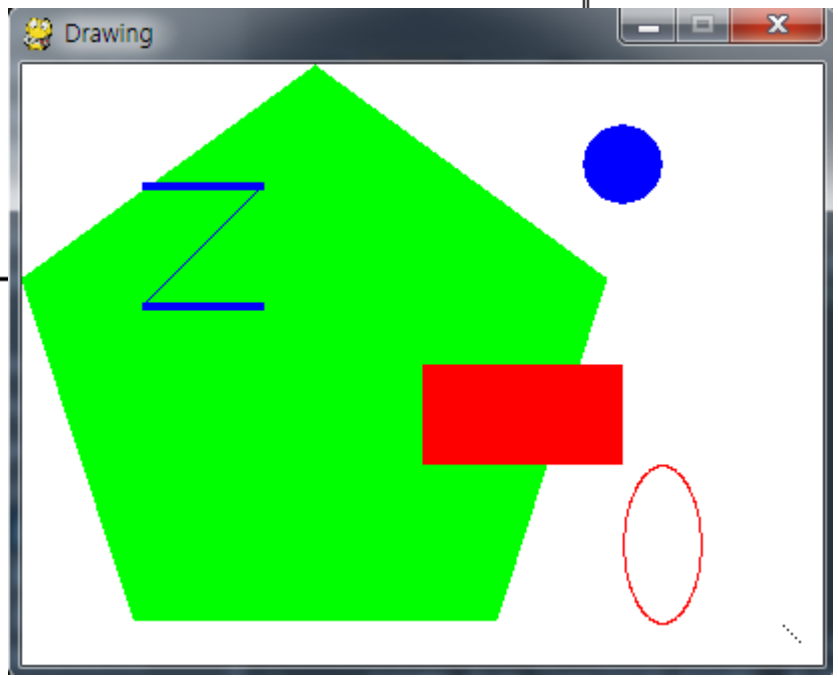
- drawing.py

```
1. import pygame, sys
2. from pygame.locals import *
```

```
3.
4. pygame.init()
5.
6. # set up the window
7. DISPLAYSURF = pygame.display.set_mode((500, 400), 0, 32)
8. pygame.display.set_caption('Drawing')
9.
10. # set up the colors
11. BLACK = (  0,   0,   0)
12. WHITE = (255, 255, 255)
13. RED   = (255,   0,   0)
14. GREEN = (  0, 255,   0)
15. BLUE  = (  0,   0, 255)
16.
17. # draw on the surface object
18. DISPLAYSURF.fill(WHITE)
19. pygame.draw.polygon(DISPLAYSURF, GREEN, ((146, 0), (291, 106), (236, 277),
(56, 277), (0, 106)))
20. pygame.draw.line(DISPLAYSURF, BLUE, (60, 60), (120, 60), 4)
21. pygame.draw.line(DISPLAYSURF, BLUE, (120, 60), (60, 120))
22. pygame.draw.line(DISPLAYSURF, BLUE, (60, 120), (120, 120), 4)
23. pygame.draw.circle(DISPLAYSURF, BLUE, (300, 50), 20, 0)
24. pygame.draw.ellipse(DISPLAYSURF, RED, (300, 250, 40, 80), 1)
25. pygame.draw.rect(DISPLAYSURF, RED, (200, 150, 100, 50))
26.
```

기본 형태 그리기

```
26.
27. pixObj = pygame.PixelArray(DISPLAYSURF)
28. pixObj[480][380] = BLACK
29. pixObj[482][382] = BLACK
30. pixObj[484][384] = BLACK
31. pixObj[486][386] = BLACK
32. pixObj[488][388] = BLACK
33. del pixObj
34.
35. # run the game loop
36. while True:
37.     for event in pygame.event.get():
38.         if event.type == QUIT:
39.             pygame.quit()
40.             sys.exit()
41.     pygame.display.update()
```



기본 형태 그리기

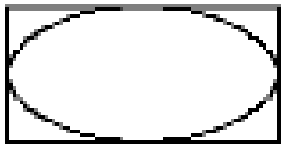
- pygame.surface.fill(color)(7번)
  - 인자로 넘겨준 색으로 surface객체를 모두 채운다
- pygame.draw.polygon(surface, color, pointlist, width)
  - 다각형을 그린다.
  - surface인자는 그릴 surfce객체, color는 색이다.
  - pointlist는 점이나 점의 목록이다. 점의 쌍 대신 리스트를 넘겨줄 수도 있다.
  - width는 옵션이고, 주시 않거나 0을 주면 안에 색을 채운다. 정수값을 전달하면 테두리만 그린다. 모든 pygame.draw함수는 동일하게 작동한다.

기본 형태 그리기

- pygame.draw.line(surface, color, start_point, end_point, width)
  - start_point와 end_point사이의 선을 그린다.
- pygame.draw.lines(surface, color, closed, polintlist, width)
  - polygon처럼 점에서 점으로 선을 연속적으로 그리나, closed 인자를 False를 넘기면 마지막 점에서 시작점으로 선을 긋지 않는다. True를 넘기면 끝에서 시작점까지 선을 긋는다.

기본 형태 그리기

- pygame.draw.circle(surface, color, center_point, radius, width)
  - 원을 그리며, 중심은 cneter_point, 반지름은 radius이다.
- pygame.draw.ellipse(surface, color, bounding_rectangle, width)
  - 타원형을 그리며, 감싸는 사각형을 정의하여 그린다. bounding_rectangle은 pygame.Rect 객체이거나 4개의 정수쌍이다.



기본 형태 그리기

- pygame.draw.rect(surface, color, rectangle_tuple, width)
  - 사각형을 그린다. rectangle_tuple은 4개의 정수 쌍이거나, pygame.Rect객체이다.

기본 형태 그리기

- pygame.PixelArray객체(PixelArray객체)
  - 함수 하나로 픽셀 하나의 색을 지정할 수는 없다. (편법으로 pygame.draw.line()의 시작과 끝점을 같게 하여 할 수는 있다.)
  - 이는 surface객체에 무엇인가를 그리면, 그리기 전과 후에 돌려야 되는 코드가 있어서 점 하나 하나의 색을 지정하는 코드는 불가능하기 때문이다. (약 2-3배 느려짐)

기본 형태 그리기

- pygame.PixelArray객체(PixelArray객체)
  - 대신 surface객체의 PixelArray객체를 만들어서 각각의 픽셀을 조정할 수 있다.
  - 이 때, PixelArray객체를 만들면 surface를 잠그기 때문에(lock) 그리기 함수는 호출 가능하지만, blit()를 통한 이미지를 surface객체상에 만들 수는 없다.
  - surface객체가 잠겨있는지 확인하려면 surface.get_locked()
    로 확인할 수 있다. (잠겨있으면 True)

기본 형태 그리기

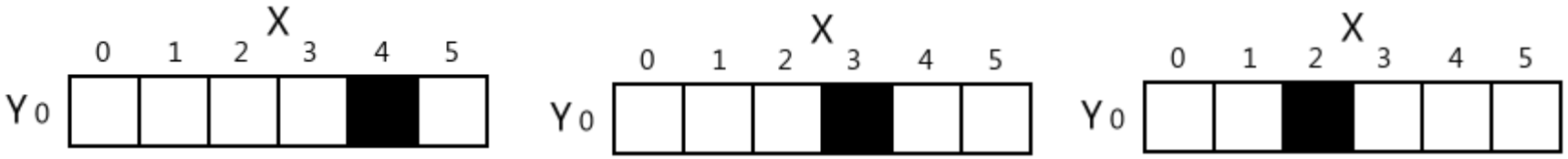
- pygame.PixelArray객체(PixelArray객체)
  - 따라서 각 픽셀에 대한 그리기를 끝냈으면 del을 통해 PixelArray객체를 지워 잠긴 상태를 풀어준다.

기본 형태 그리기

- pygame.display.update()함수
  - display surface객체를 화면에 보여주게 된다.
  - 따라서 한 프레임을 다 그린 후에는 본 함수를 통해 화면에 보여주게 된다.
  - 다른 surface 객체 이미지를 화면에 보여주고 싶으면, 이미지를 복사해서 blit()메소드로 Surface객체상에 보여주면 된다.

애니메이션

- 애니메이션은 1초를 몇 개의 구간으로 나눈 후, 조금씩 변화가 일어난 이미지를 화면에 보여주는 것이다.



- 컴퓨터는 각 점이 찍힌 그림 두 장을 보여준 것이지만, 사용자에게는 이 그림이 검정색 점이 왼쪽으로 움직인 것처럼 보인다.
- 모든 영상은 이러한 방식으로 보여진다.

애니메이션

- catanimation.py

```
 1. import pygame, sys
 2. from pygame.locals import *
 3.
 4. pygame.init()
 5.
 6. FPS = 30 # frames per second setting
 7. fpsClock = pygame.time.Clock()
 8.
 9. # set up the window
10. DISPLAYSURF = pygame.display.set_mode((400, 300), 0, 32)
11. pygame.display.set_caption('Animation')
12.
13. WHITE = (255, 255, 255)
14. catImg = pygame.image.load('cat.png')
15. catx = 10
16. caty = 10
17. direction = 'right'
18.
19. while True: # the main game loop
20.     DISPLAYSURF.fill(WHITE)
21.
22.     if direction == 'right':
23.         catx += 5
24.         if catx == 280:
25.             direction = 'down'
26.     elif direction == 'down':
```

애니메이션

```
26.     elif direction == 'down':
27.         caty += 5
28.         if caty == 220:
29.             direction = 'left'
30.     elif direction == 'left':
31.         catx -= 5
32.         if catx == 10:
33.             direction = 'up'
34.     elif direction == 'up':
35.         caty -= 5
36.         if caty == 10:
37.             direction = 'right'
38.
39.     DISPLAYSURF.blit(catImg, (catx, caty))
40.
41.     for event in pygame.event.get():
42.         if event.type == QUIT:
43.             pygame.quit()
44.             sys.exit()
45.
46.     pygame.display.update()
```

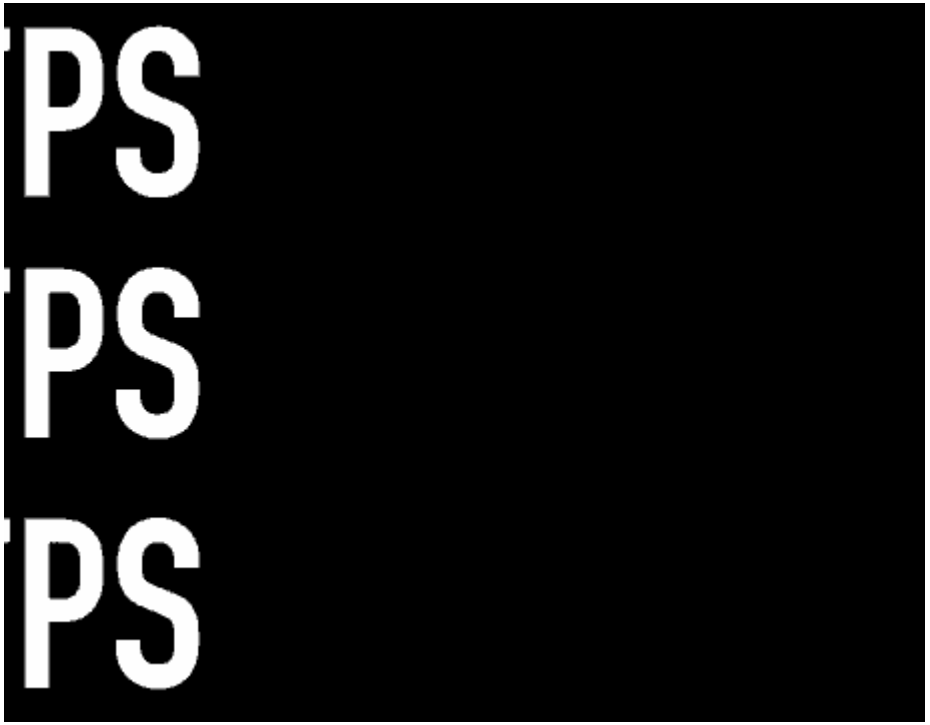
- cat.png가 같은 폴더에 있어야 한다.

애니메이션

- 초당 프레임과 pygame.time.Clock 객체
  - 프레임 레이트(frame rate) or 리프레시 레이트(refresh rate)는 프로그램이 1초당 그려내는 그림의 숫자를 말하며 단위는 FPS 즉, 초당 프레임(frames per second)이다.
  - 게임에서 프레임 레이트가 낮으면 게임이 툭툭 튀어보이는데, 만약 프로그램이 움직이는 하면을 보여줄 때, 너무 많은 코드가 필요하면 FPS는 낮아질 수 밖에 없다.

애니메이션

- 초당 프레임과 pygame.time.Clock 객체



애니메이션

- 초당 프레임과 pygame.time.Clock 객체
  - pygame.time.Clock객체는 프로그램이 최대의 FPS로 작동하게 해주며, Clock객체는 게임 루프를 돌 때, 약간씩 간격을 줘서 프로그램이 너무 빠르게 수행되지 않도록 한다.
  - 이렇게 간격을 주지 않으면 게임 프로그램은 컴퓨터의 성능에 따라 빨리 수행되고 끝나게 된다.
  - 따라서 clock객체의 tick()메소드를 통해서 컴퓨터의 성능과 상관없이 작동하도록 한다.

애니메이션

- 초당 프레임과 pygame.time.Clock 객체
  - 게임 루프 끝 부분에서 pygame.display.update()를 호출한 다음, Clock객체의 tick을 호출하는 것이 좋은데, 간격을 주는 방식이 이전에 tick()을 호출한 다음 얼마나 시간이 지났는지에 따라 계산해서 호출하기 때문이다.
  - FPS에 따라 속도가 조절되는 것을 볼 수 있다.

애니메이션

- pygame.image.load()와 blit()로 이미지 그리기
  - 게임에서는 보통 이미지를 사용하는데, 이 이미지를 스프라이트(sprite)라고 한다. pygame에서는 PNG, JPG, GIF, BMP이미지 파일들을 surface객체로 올릴 수 있다.
  - 이미지를 surface객체로 만들기 위해서는 pygame.image.load()함수에 파일 이름을 넘겨준다.

애니메이션

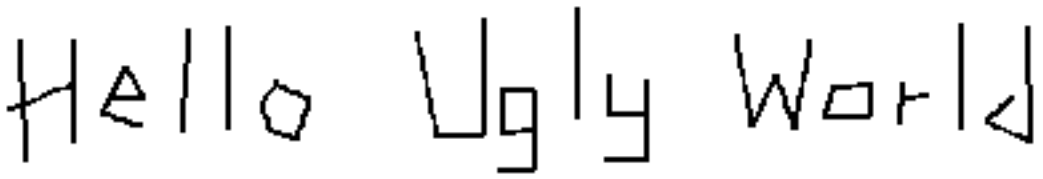
- pygame.image.load()와 blit()로 이미지 그리기
  - 이 surface객체는 display surface객체와 다른 객체이기 때문에 이미지용 surface 객체를 display surface객체로 복사(blit)해야 된다.
  - blitting은 한 surface객체의 내용을 다른 surface객체 위에 그리는 것을 말한다.

애니메이션

- pygame.image.load()와 blit()로 이미지 그리기
  - DISPLAYSURF.blit(catImag, (catx, caty))
  - 첫 번째 인자는 소스가 되는 surface객체, 두 번째 인자는 이미지가 복사될 위치의 맨 왼쪽 상단의 좌표이다.
  - 잠겨있는 surface 객체로는 복사할 수 없다.

폰트

- 글씨를 pygame.draw.line()을 여러 번 사용해서 쓸 수 있으나, 그리 좋지 않다.



폰트

- fonttext.py

```
1. import pygame, sys
2. from pygame.locals import *
3.
4. pygame.init()
5. DISPLAYSURF = pygame.display.set_mode((400, 300))
6. pygame.display.set_caption('Hello World!')
7.
8. WHITE = (255, 255, 255)
9. GREEN = (0, 255, 0)
10. BLUE = (0, 0, 128)
11.
12. fontObj = pygame.font.Font('freesansbold.ttf', 32)
13. textSurfaceObj = fontObj.render('Hello world!', True, GREEN, BLUE)
14. textRectObj = textSurfaceObj.get_rect()
15. textRectObj.center = (200, 150)
16.
17. while True: # main game loop
18.     DISPLAYSURF.fill(WHITE)
19.     DISPLAYSURF.blit(textSurfaceObj, textRectObj)
20.     for event in pygame.event.get():
21.         if event.type == QUIT:
22.             pygame.quit()
23.             sys.exit()
24.     pygame.display.update()
```

폰트

- 12줄: pygame.font.Font 객체를 만든다.
- 13줄: Font객체의 render() 메소드로 써서 텍스트를 그릴 surface객체를 만든다.
- 14줄: get_rect() 메소드를 써서 Rect객체를 만든다. 텍스트가 꼭 맞게 들어갈 폭과 높이를 가지지만, 왼쪽 상단 좌표는 0이다.
- 15줄: center를 변경한다.
- 19줄: 텍스트가 있는 surface를 surface객체로 복사한다.

폰트

- 윈도우의 font폴더를 접근해서 폰트를 고를 수 있다.
- fontObj = pygame.font.Font('c:₩₩Windows₩₩fonts₩₩gulim.ttc', 32)
- freesansbold.ttf는 pygame에 존재한다.

폰트

- 상대경로와 절대경로
  - 절대경로
    - 고유한 경로
    - 절대적인 위치를 말한다.
    - Ex) “c:₩windows”, “d:₩”
  - 상대경로
    - 현재 디렉토리를 바탕으로 경로를 확인한다.
    - Ex) “./python.py”, “../update.txt”
    - /: 루트, ./: 현재 위치, ../: 현재 위치의 상단폴더

폰트

- 현재 폴더가 “c:₩python₩index”일 경우
  - /는 c:
  - ./는 index
  - ../는 python
- 경로를 지정할 때, ₩는 escape문자임으로 ₩₩를 적어야 된다.

폰트

- 환경변수
  - C:₩Widnows는 실행창에서 사용할 때, %SYSTEMROOT%를 통해서 사용할 수 있다.
  - 이렇게 %SYSTEMROOT%를 통해서 사용하는 이유는 os가 꼭 c드라이브에 설치되지 않았을 가능성이 있기 때문이다.
  - %HOMEPATH%는 현재 사용자의 home 디렉토리로 간다.
    C:₩Users₩사용자이름
  - 기본적인 변수는 슬라이드 노트의 주소를 참조

폰트

- 파이썬에서 환경변수 사용하기

```
import pygame, sys
from pygame.locals import *
import os

pygame.init()
DISPLAYSURF = pygame.display.set_mode((400, 300))
pygame.display.set_caption('Hello World!')

WHITE = (255, 255, 255)
GREEN = (  0, 255,   0)
BLUE  = (  0,   0, 128)

fontObj = pygame.font.Font(os.environ['systemroot']+'\\fonts\\gulim.ttc', 32)
textSurfaceObj = fontObj.render('Hello world!', True, GREEN, BLUE)
textRectObj = textSurfaceObj.get_rect()
textRectObj.center = (200, 150)

while True: # main game loop
    DISPLAYSURF.fill(WHITE)
    DISPLAYSURF.blit(textSurfaceObj, textRectObj)
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit()
            sys.exit()
    pygame.display.update()
```

폰트

- 안티 엘리어싱(anti-aliasing)
  - 테스트나 도형이 울퉁불퉁해 보이지 않도록 둘레선을 약간 번진 것처럼 보이는 기술
  - 좋게 보이나, 프로그램이 느려질 수 있다.
  - Aliasing는 픽셀조각이 튀는 현상

폰트

- 안티앨리어싱
  - Pygame에서 안티앨리어싱 텍스트를 사용하기 위해선 render()메소드의 두 번째 파라미터에 True값을 넘겨줘야 한다.

소리 재생하기

- 소리를 재생하는 것은 pygame에서 어렵지 않다.
- Pygame.mixer.Sound()생성자를 통해 pygame.mixer.Sound객체(Sound객체)를 만든다.
- 본 생성자는 오디오 파일의 이름을 파라미터로 받는다.
- Wav, mp3, ogg가 가능하다.

소리재생하기

- Sound.py

```python
import pygame
import sys
import time
from pygame.locals import *

pygame.init()
DISPLAYSURF = pygame.display.set_mode((400, 300))
pygame.display.set_caption('Sound')

while True: # main game loop
    soundObj = pygame.mixer.Sound("beep1.ogg")
    soundObj.play()
    time.sleep(1)  # 1초 동안 소리를 재생하도록 둔다.
    soundObj.stop()
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit()
            sys.exit()
    pygame.display.update()
```

소리재생하기

- Sound객체는 적을 죽이거나, 공격하거나, 아이템을 주었을 때, 사용할 수 있다.
- 플레이어의 동작에 상관없이 배경음악이 계속 나와야하는 경우
  - Pygame.mixer.music.load() 함수에 파라미터를 파일의 인자로 넘겨준다.
  - Wav.mp3, midi를 사용한다.
- 배경 음악으로 사운드 파일을 로드에서 재생할 때, pygame.mixer.music.play(-1, 0.0)을 통해서 재생한다.

소리재생하기

- 배경 음악
  - -1이면 무한 반복한다.
  - 0이상의 정수이면, 그 숫자만큼 반복한다
  - 0.0은 파일의 처음부터 재생한다.
  - 0.0이상의 실수는 그 부분부터 재생하게 만든다.
  - 배경 음악 재생을 멈추기 위해서는 pygame.mixer.music.stop()함수를 호출한다.

소리 재생하기

- Sound.py 변경

```python
import pygame
import sys
import time
from pygame.locals import *

pygame.init()
DISPLAYSURF = pygame.display.set_mode((400, 300))
pygame.display.set_caption('Sound')
soundObj = pygame.mixer.Sound("beep1.ogg")
pygame.mixer.music.load("Shining.mp3")
pygame.mixer.music.play(-1, 0.0)

while True: # main game loop
    soundObj.play()
    time.sleep(1)  # 1초 동안 소리를 재생하도록 둔다.
    soundObj.stop()
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit()
            sys.exit()
    pygame.display.update()

```