

비트맵 그래픽스와 사운드

순천향대학교 컴퓨터공학과
이 상 정

GUI 설계기법

학습 내용

- 비트맵 그래픽스
 - 사각형, 원 등 단순한 모양이 아닌 **이미지로 작업**
 - 사진, 그리기 프로그램 등으로 작성된 이미지
- 배경 이미지 그리기
- 이미지 이동
- 사운드 연주
- **타일 이미지 추출**

배경 이미지 그리기

□ 파일 이미지 적재 (load)

- `pygame.image.load(filename): return Surface`
 - 예: `background_image = pygame.image.load("saturn_family1.jpg")`
- 적재된 이미지는 파이게임의 픽셀 형식으로 변환 되어야 함
 - `Surface.convert(): return Surface`
 - 예


```
background_image=pygame.image.load("saturn_family1.jpg").convert()
```

□ 이미지의 디스플레이

- `Dst_Surf.blit(src_surf, dst_pos, src_area=None, special_flags = 0): return Rect`
- `source` Surface 를 현재 Surface의 `dest` 위치에 그리기
- 예


```
screen = pygame.display.set_mode([800, 600])
background_image=pygame.image.load("saturn_family1.jpg").convert()
screen.blit(background_image, [0,0])
```



시험주행



배경 이미지 그리기 코드 (1)

```
import pygame

white=[255,255,255]
black=[0,0,0]

pygame.init()

# 윈도우 설정
screen = pygame.display.set_mode([800, 600])
pygame.display.set_caption('배경 이미지')

# 배경색 설정
screen.fill(black)
# 배경 이미지의 스크린 적재 위치
background_position=[0,0]
# 이미지 적재
background_image = pygame.image.load("saturn_family1.jpg").convert()
```

배경 이미지 그리기 코드 (2)

```
clock = pygame.time.Clock()
done = False

while done==False:
    clock.tick(10)

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done=True

    # 배경 이미지를 스크린에 디스플레이
    screen.blit(background_image, background_position)

    pygame.display.flip()

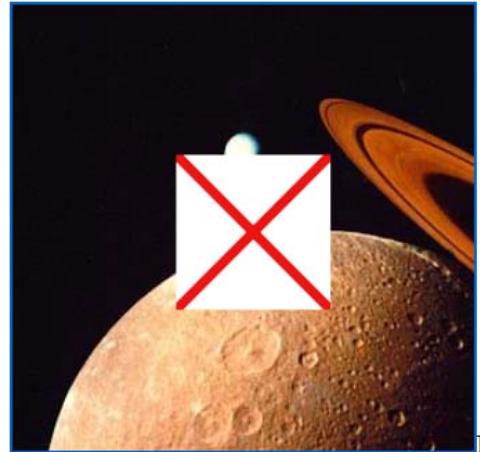
pygame.quit ()
```

이미지의 이동

- ❑ 마우스 움직임에 따라 이미지 이동
 - 마우스 위치를 blit() 메서드의 위치 인수로 전달

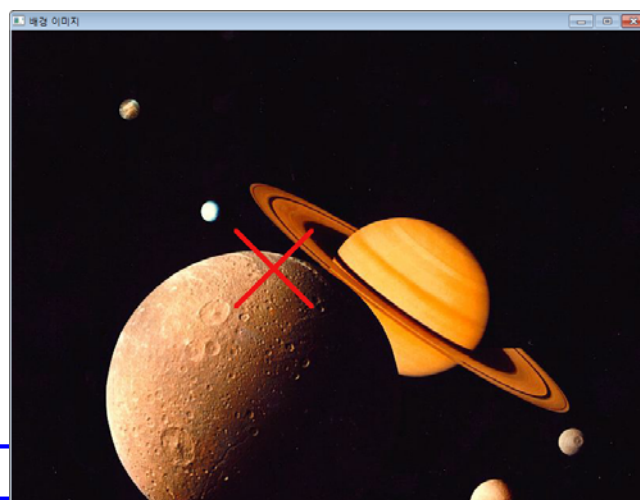
- ❑ 예

```
player_image = pygame.image.load("player.png").convert()
# 현재 마우스 위치 읽기
player_position = pygame.mouse.get_pos()
x=player_position[0]
y=player_position[1]
# 마우스 위치에 이미지 그리기
screen.blit(player_image, [x,y])
```



투명한 색 (Transparent Color)

- ❑ 앞의 예에서 배경이 흰색으로 가려짐
 - X 표시만 남고 흰색은 투명으로 처리
 - `Surface.set_colorkey(Color, flags=0): return None`
 - Surface의 Color 색을 투명으로 처리
 - 예: `player_image.set_colorkey(white)`



이미지 이동 코드 (1)

```
import pygame

white=[255,255,255]
black=[0,0,0]

pygame.init()

# 윈도우 설정
screen = pygame.display.set_mode([800, 600])
pygame.display.set_caption('이미지 이동')

screen.fill(black)

# 배경 이미지의 스크린 적재 위치
background_position=[0,0]
# 이미지 적재
background_image = pygame.image.load("saturn_family1.jpg").convert()
player_image = pygame.image.load("player.png").convert()
```

이미지 이동 코드 (2)

```
# 색의 투명화
player_image.set_colorkey(white)

clock = pygame.time.Clock()
done = False

while done==False:
    clock.tick(10)

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done=True

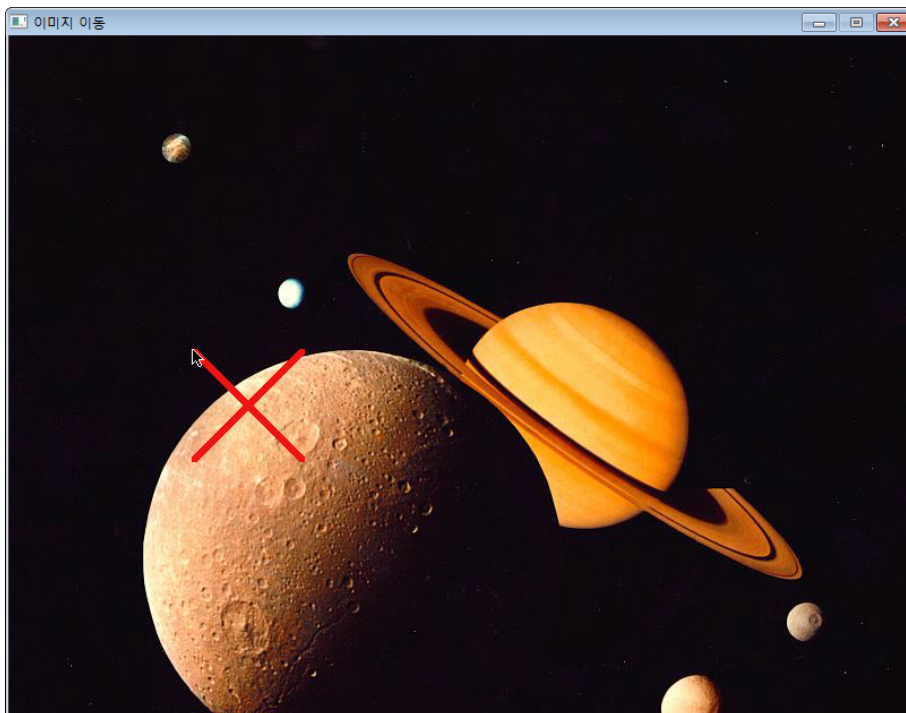
    # 배경 이미지를 스크린에 디스플레이
    screen.blit(background_image, background_position)

    # 현재 마우스 위치 읽기
    player_position = pygame.mouse.get_pos()
    x=player_position[0]
    y=player_position[1]
```

```
# 마우스 위치에 이미지 그리기
screen.blit(player_image, [x,y])

pygame.display.flip()

pygame.quit ()
```



- ❑ 마우스 클릭 시 “클릭 소리” 연주
- ❑ Sound 객체 생성
 - `pygame.mixer.Sound(filename)`: return Sound
- ❑ Sound 연주
 - `Sound.play(loops=0, maxtime=0, fade_ms=0)`: return Channel
 - `loops`: 추가 반복 연주 횟수, `maxtime`: 최대 연주 시간, `fade_ms`: 최대 볼륨까지 서서히 상승하는 시간
- ❑ 예


```
# 사운드 객체 생성
click_sound = pygame.mixer.Sound("click.wav")

if event.type == pygame.MOUSEBUTTONDOWN:
    click_sound.play()
```

이미지 코드: 클릭 사운드 (1)

```

import pygame

white=[255,255,255]
black=[0,0,0]

pygame.init()

# 윈도우 설정
screen = pygame.display.set_mode([800, 600])
pygame.display.set_caption('배경 이미지')

# 배경 이미지의 Surface 생성 및 배경색 설정
background =
    pygame.Surface(screen.get_size())
background.fill(black)

# 사운드 객체 생성
click_sound = pygame.mixer.Sound("click.wav")

# 배경 이미지의 스크린 적재 위치
background_position=[0,0]

# 이미지 적재
background_image =
    pygame.image.load("saturn_family1.jpg").convert()
player_image =
    pygame.image.load("player.png").convert()
# 색의 투명화
player_image.set_colorkey(white)

clock = pygame.time.Clock()
done = False

while done==False:
    clock.tick(10)

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done=True
        if event.type == pygame.MOUSEBUTTONDOWN:
            click_sound.play()

```

비트맵 그래픽스와 사운드

이미지 코드: 클릭 사운드 (2)

```

# 배경 이미지를 스크린에 디스플레이
screen.blit(background_image, background_position)

# 현재 마우스 위치 읽기
player_position = pygame.mouse.get_pos()
x=player_position[0]
y=player_position[1]

# 마우스 위치에 이미지 그리기
screen.blit(player_image, [x,y])

pygame.display.flip()

pygame.quit ()

```



타일 이미지(스프라이트 시트)

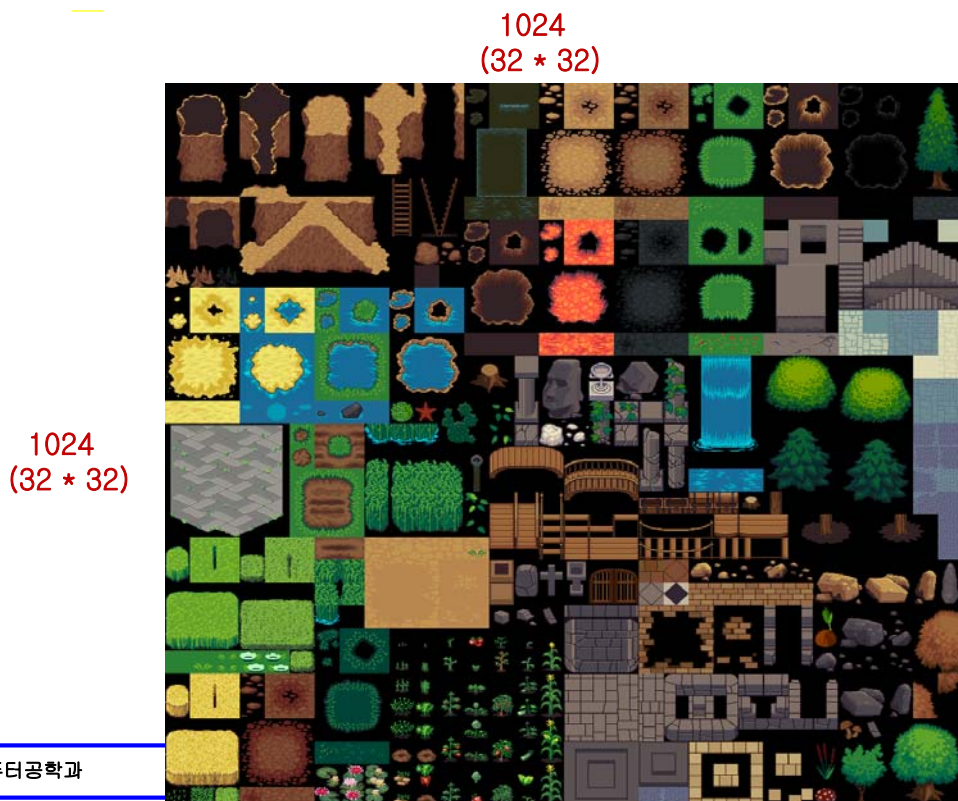
- 플랫폼(어드벤처) 게임 등은 많은 그래픽 이미지를 사용
 - 플랫폼 게임은 플레이어가 여러 장애물을 통과하여 여러 플랫폼(배경 화면) 사이로 아바타를 안내하여 진행하는 비디오 게임
 - 예, 슈퍼 마리오
- 타일 이미지 (Tiled Image), 스프라이트 시트 (sprite sheet)
 - 각 이미지를 독립된 하나의 파일로 생성하고 관리하지 않음
 - 여러 작은 이미지들로 구성된 하나의 큰 이미지를 생성하여 관리
 - 타일 이미지는 Tiled 맵 편집기(Map Editor) 등을 사용하여 생성
<http://www.mapeditor.org/>

타일 이미지 추출

- 이미지 파일에 그림 중 일부를 추출하여 그리기
 - 많은 그림들을 일정한 크기의 타일 영역에 배치
 - 어드벤처 게임 등의 배경에 사용되는 그림들을 추출하는데 이용
 - `Surface.blit()` 함수 이용하여 그리기

- 1024 x 1024 이미지 파일 예, `terrain_atlas.png`
 - 32 픽셀 x 32 픽셀 크기의 타일로 분할
 - x, y 축이 각각 32개 타일로 표시
 - $1024 = 32 \text{ 개 타일} \times 32 \text{ 픽셀/타일}$

1024 x 1024 이미지 파일 예



타일 이미지 그리기

□ 3개의 그림 추출하여 그리기

□ 나무1

- 타일 이미지 위치: 30번째 타일 열, 0번째 타일 행 => [32*30, 32*0]
- 타일 이미지 크기: 2 타일 x 5 타일 => 64 픽셀 x 160 픽셀
- 스크린 표시 위치: [100, 100]

□ 나무2

- 타일 이미지 위치: 29번째 타일 열, 28번째 타일 행 => [32*29, 32*28]
- 타일 이미지 크기: 3 타일 x 4 타일 => 96 픽셀 x 128 픽셀
- 스크린 표시 위치: [300, 250]

□ 벽돌 장애물

- 타일 이미지 위치: 16번째 타일 열, 23번째 타일 행 => [32*16, 32*23]
- 타일 이미지 크기: 3 타일 x 3 타일 => 96 픽셀 x 96 픽셀
- 스크린 표시 위치: [150, 450]

타일 이미지 예



스크린 표시 예



이미지 추출 예

- `Surface.blit()` 함수의 인수 중 **소스 이미지의 영역(area)**을 표시하여 특정 그림 추출
 - `Dst_Surf.blit(src_surf, dst_pos, src_area=None, special_flags = 0):`
return Rect
 - 영역은 Rect로 표시
- 나무1 그리기 예


```
# 이미지 적재
atlas_image = pygame.image.load("terrain_atlas.png").convert()
# 색의 투명화
atlas_image.set_colorkey(black)
# 스크린에 그리기
screen.blit(atlas_image, [100,100], [32*30, 32*0, 64, 160])
```

타일 이미지 추출 코드 (1)

```

import pygame

white=[255,255,255]
black=[0,0,0]
back_color = [255,255,153]

# 이미지 추출 그리기 함수
def setGraphic(tilex,tiley,tilewidth,tileheight,x,y):
    # 이미지 적재
    atlas_image = pygame.image.load("terrain_atlas.png").convert()
    atlas_image.set_colorkey(black)
    # 배경 이미지를 스크린에 디스플레이
    screen.blit(atlas_image, [x,y], [tilex,tiley, tilewidth, tileheight])

pygame.init()

# 윈도우 설정
screen = pygame.display.set_mode([800, 600])
pygame.display.set_caption('어드벤처 이미지')
clock = pygame.time.Clock()
done = False

```

타일 이미지 추출 코드 (2)

```

while done==False:
    clock.tick(10)

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done=True

    # 스크린 배경
    screen.fill(back_color);

    # 배경 이미지를 스크린에 디스플레이
    # 나무
    setGraphic(32*30, 32*0, 64, 160, 100,100) # 나무 1
    setGraphic(32*29, 32*28, 96, 128, 300,250) # 나무 2
    # 벽돌 장애물
    setGraphic(32*16, 32*23, 96, 96, 150, 450) # 벽돌 장애물 1

    pygame.display.flip()

pygame.quit ()

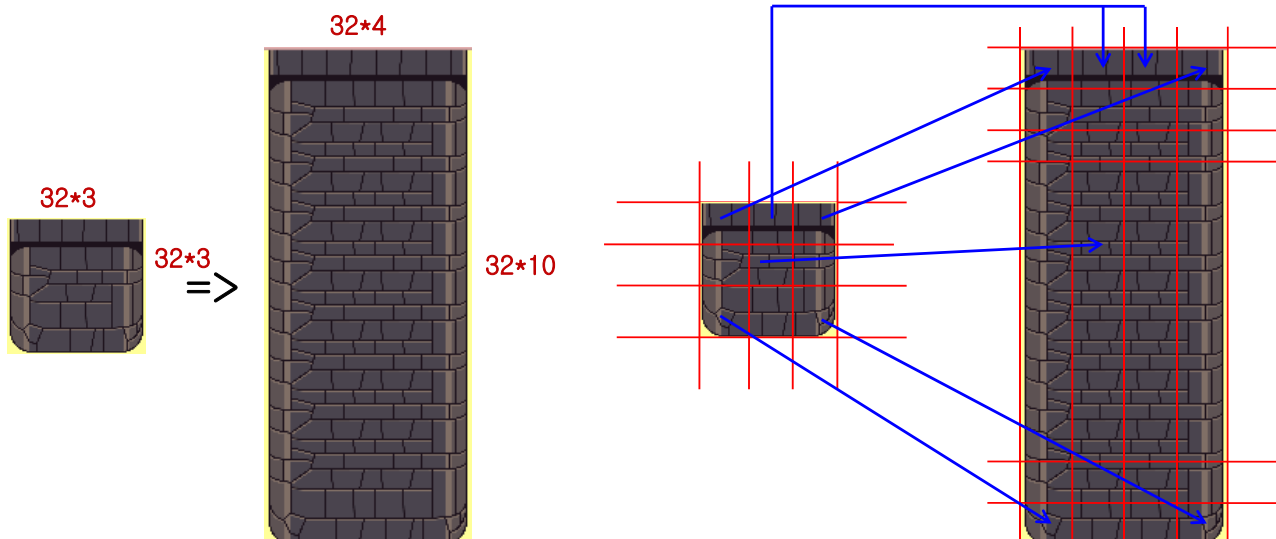
```



타일 이미지 확장 추출 예

□ 이미지 파일의 특정 그림을 확장

- 테두리 부분은 그대로 유지하고 중간 부분을 확장
 - 스크린 이미지 크기: 4 타일 x 10 타일 => 128 픽셀 x 320 픽셀
 - 위치: [600, 0]



타일 이미지 확장 추출 코드 (1)

```
import pygame

white=[255,255,255]
black=[0,0,0]
back_color = [255,255,153]

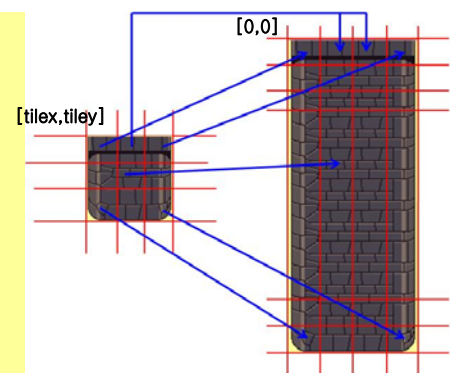
# 이미지 추출 그리기 함수
def setGraphic(tilex,tiley,tilewidth,tileheight,x,y):
    # 이미지 적재
    atlas_image = pygame.image.load("terrain_atlas.png").convert()
    atlas_image.set_colorkey(black)
    # 배경 이미지를 스크린에 디스플레이
    screen.blit(atlas_image, [x,y], [tilex,tiley, tilewidth, tileheight])

# 확장된 타일 이미지 그리기 함수
def setGraphic2(tilex,tiley, x, y, width, height):
    # 이미지 적재
    atlas_image = pygame.image.load("terrain_atlas.png").convert()
    # 스크린에 표시될 이미지 Surface 생성
    image = pygame.Surface([width, height])
    image.set_colorkey(black)
```

타일 이미지 확장 추출 코드 (2)

```
## 1행 타일
# 1행 1열 타일
image.blit(atlas_image,[0,0],[tilex,tiley,32,32])
# 1행 중간열 타일
for column in range(width//32-2):
    image.blit(atlas_image,[(column+1)*32,0],[tilex+32,tiley,32,32])
# 1행 마지막열 타일
image.blit(atlas_image,[(width//32-1)*32,0],[tilex+64,tiley,32,32])

## 중간행 타일
for row in range(height//32-2):
    # row 행의 1열 타일
    image.blit(atlas_image,[0,(row+1)*32],[tilex,tiley+32,32,32])
    # row 행의 중간 열 타일
    for column in range(width//32-2):
        image.blit(atlas_image,[(column+1)*32,(row+1)*32],[tilex+32,tiley+32,32,32])
    # row 행의 마지막 열 타일
    image.blit(atlas_image,[(width//32-1)*32,(row+1)*32],[tilex+64,tiley+32,32,32])
```



타일 이미지 확장 추출 코드 (3)

```

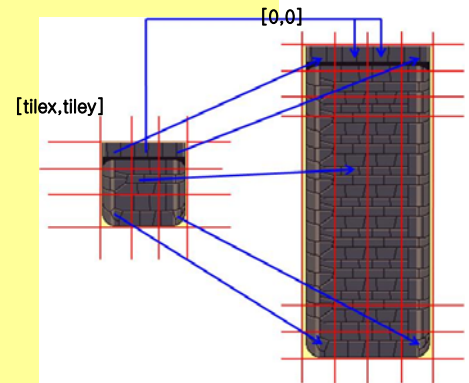
## 마지막행 타일
# 마지막행 1열 타일
image.blit(atlas_image, [0, (height//32-1)*32], [tilex, tiley+64, 32, 32])
# 마지막행 중간열 타일
for column in range(width//32-2):
    image.blit(atlas_image, [(column+1)*32, (height//32-1)*32], [tilex+32, tiley+64, 32, 32])
# 마지막행 마지막열 타일
image.blit(atlas_image, [(width//32-1)*32, (height//32-1)*32], [tilex+64, tiley+64, 32, 32])

# 스크린에 이미지 그리기
screen.blit(image, [x, y])

pygame.init()

# 윈도우 설정
screen = pygame.display.set_mode([800, 600])
pygame.display.set_caption('어드벤처 이미지')
clock = pygame.time.Clock()
done = False

```



타일 이미지 확장 추출 코드 (4)

```

while done==False:
    clock.tick(10)

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done=True

    # 스크린 배경
    screen.fill(back_color);
    # 배경 이미지를 스크린에 디스플레이
    # 나무
    setGraphic(32*30, 32*0, 64, 160, 100,100) # 나무 1
    setGraphic(32*29, 32*28, 96, 128, 300,250) # 나무 2
    # 벽돌 장애물
    setGraphic(32*16, 32*23, 96, 96, 150, 450) # 벽돌 장애물 1
    # 확장된 벽돌 장애물
    setGraphic2(32*16, 32*23, 600, 0, 128, 320)

    pygame.display.flip()

```




순천향대학교 컴

사운드

1. 앞에서 소개된 클릭 사운드 이미지 프로그램을 작성하고 실행
2. 앞에서 배운 내용을 사용한 임의의 프로그램 작성
 - 프로그램 설명
 - 프로그램 소스
 - 실행 결과