

# PYTHON: GAMES

# PYGAME FRAMEWORK

```
$ pip install pygame
```

# Game loop



# Game loop

```
while True:
```

# Game loop

```
while True:  
    # are some keys pressed? has the mouse moved?  
    handle_input(keys, mouse)
```

# Game loop

```
while True:
    # are some keys pressed? has the mouse moved?
    handle_input(keys, mouse)

    # move objects, check collisions, update physics
    for obj in objects:
        obj.update(delta)
```

# Game loop

```
while True:
    # are some keys pressed? has the mouse moved?
    handle_input(keys, mouse)

    # move objects, check collisions, update physics
    for obj in objects:
        obj.update(delta)

    # render graphics to screen
    for obj in objects:
        obj.draw(screen)
```

What is delta?



# What is delta?

```
x += speed * delta
```

# What is delta?

```
x += speed * delta  
# frame took 1/60 s -> small movement
```

# What is delta?

```
x += speed * delta  
# frame took 1/60 s -> small movement  
# frame took 1 s -> large movement
```

# What is delta?

```
x += speed * delta  
# frame took 1/60 s -> small movement  
# frame took 1 s -> large movement
```

- adjustment for computer speed

# What is delta?

```
x += speed * delta  
# frame took 1/60 s -> small movement  
# frame took 1 s -> large movement
```

- adjustment for computer speed
- usually capped at 60 FPS

Rectangle



# Rectangle

```
pos = (1, 1) # points are represented as tuple (x, y)
```

# Rectangle

```
pos = (1, 1) # points are represented as tuple (x, y)  
size = (30, 30)
```



# Rectangle

```
pos = (1, 1) # points are represented as tuple (x, y)
size = (30, 30)
r = pygame.Rect(pos, size)
```

# Rectangle

```
pos = (1, 1) # points are represented as tuple (x, y)
size = (30, 30)
r = pygame.Rect(pos, size)
r.center = (50, 60) # sets center to (50, 60)
```

# Rectangle

```
pos = (1, 1) # points are represented as tuple (x, y)
size = (30, 30)
r = pygame.Rect(pos, size)
r.center = (50, 60) # sets center to (50, 60)
moved = r.move(50, 30) # new rectangle at (x + 50, y + 30)
```

Drawing image to screen



## Drawing image to screen

```
image = load_image('images/enemy1.gif')
```

## Drawing image to screen

```
image = load_image('images/enemy1.gif')  
rect = image.get_rect() # image rectangle
```

## Drawing image to screen

```
image = load_image('images/enemy1.gif')  
rect = image.get_rect() # image rectangle  
rect.center = (30, 30) # change position
```

## Drawing image to screen

```
image = load_image('images/enemy1.gif')
rect = image.get_rect() # image rectangle
rect.center = (30, 30) # change position

def draw(self, screen):
    screen.blit(image, rect)
```



# Object collections



# Object collections

```
enemy = Enemy((50, 50)) # inherits from pygame.sprite.Sprite
```

# Object collections

```
enemy = Enemy((50, 50)) # inherits from pygame.sprite.Sprite  
enemies = pygame.sprite.Group() # list of game objects
```

# Object collections

```
enemy = Enemy((50, 50)) # inherits from pygame.sprite.Sprite  
enemies = pygame.sprite.Group() # list of game objects  
enemies.add(enemy)
```

# Object collections

```
enemy = Enemy((50, 50)) # inherits from pygame.sprite.Sprite
enemies = pygame.sprite.Group() # list of game objects
enemies.add(enemy)

enemies.update(p1, ...) # update(p1, ...) on all enemies
```

# Object collections

```
enemy = Enemy((50, 50)) # inherits from pygame.sprite.Sprite
enemies = pygame.sprite.Group() # list of game objects
enemies.add(enemy)

enemies.update(p1, ...) # update(p1, ...) on all enemies
enemies.draw(p1, ...) # draw(p1, ...) on all enemies
```

# Object collections

```
enemy = Enemy((50, 50)) # inherits from pygame.sprite.Sprite
enemies = pygame.sprite.Group() # list of game objects
enemies.add(enemy)

enemies.update(p1, ...) # update(p1, ...) on all enemies
enemies.draw(p1, ...) # draw(p1, ...) on all enemies

len(enemies) # 1
```

# Object collections

```
enemy = Enemy((50, 50)) # inherits from pygame.sprite.Sprite
enemies = pygame.sprite.Group() # list of game objects
enemies.add(enemy)

enemies.update(p1, ...) # update(p1, ...) on all enemies
enemies.draw(p1, ...) # draw(p1, ...) on all enemies

len(enemies) # 1
enemy.kill() # remove enemy from all collections
```



# Object collections

```
enemy = Enemy((50, 50)) # inherits from pygame.sprite.Sprite
enemies = pygame.sprite.Group() # list of game objects
enemies.add(enemy)

enemies.update(p1, ...) # update(p1, ...) on all enemies
enemies.draw(p1, ...) # draw(p1, ...) on all enemies

len(enemies) # 1
enemy.kill() # remove enemy from all collections
len(enemies) # 0
```

# Testing collisions



# Testing collisions

```
cols = pygame.sprite.spritecollide(player, enemies, False)  
# `cols` now contains `enemies` that collide with `player`
```

# Testing collisions

```
cols = pygame.sprite.spritecollide(player, enemies, False)
# `cols` now contains `enemies` that collide with `player`
if len(cols) > 0:
    for item in cols:
        item.kill()
```

Cooldown



# Cooldown

```
cd = Cooldown(500) # CD for 500 ms
```

# Cooldown

```
cd = Cooldown(500) # CD for 500 ms

def update(self, engine, delta):
    cd.update(delta)
    if cd.reset_if_ready():
        # fire in the hole!
```

# Pseudo-random generator





# Pseudo-random generator

```
import random
```

# Pseudo-random generator

```
import random  
r = random.Random()
```

# Pseudo-random generator

```
import random
r = random.Random()
r.random()          # random number between in range (0, 1)
```

# Pseudo-random generator

```
import random
r = random.Random()
r.random()           # random number between in range (0, 1)
r.randint(3, 5)      # random number in range [3, 5]
```

# Pseudo-random generator

```
import random
r = random.Random()
r.random()           # random number between in range (0, 1)
r.randint(3, 5)      # random number in range [3, 5]
r.choice([1, 2, 3])  # randomly selects an item from iterable
```