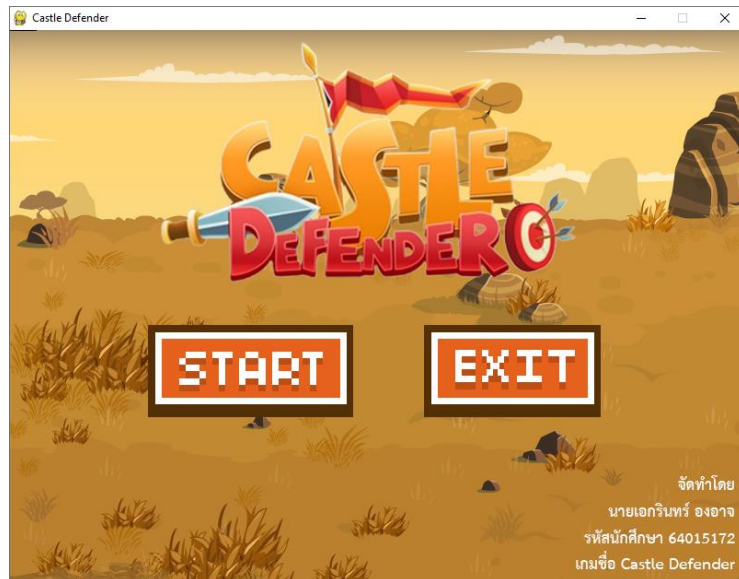


## Castle Defender



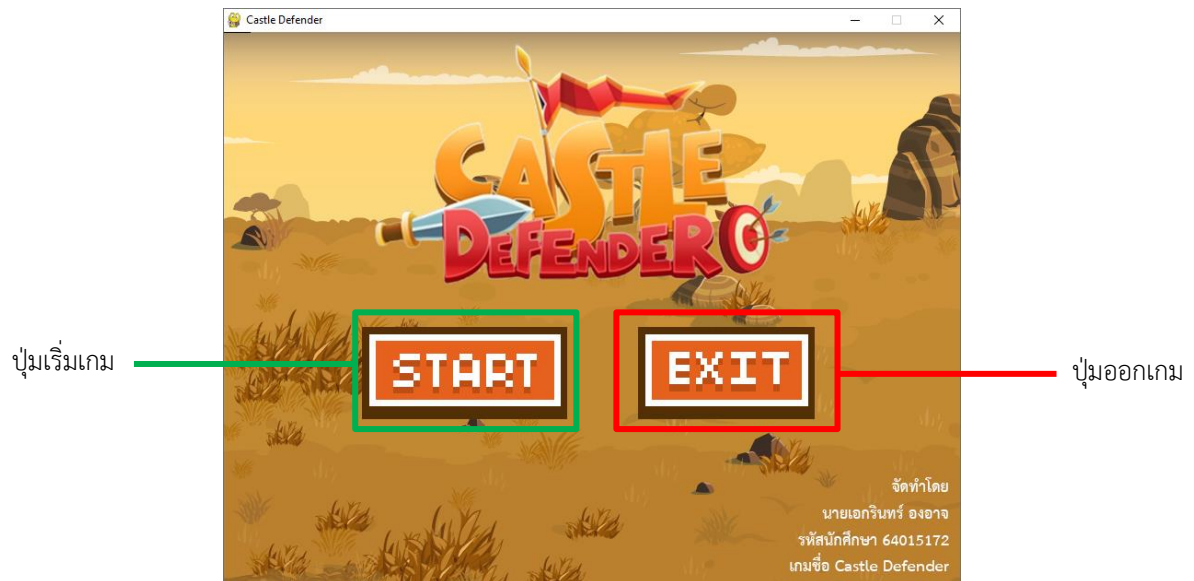
**Castle Defender** เป็นเกมป้องกันปราสาท ซึ่งคุณต้องใช้ปืนยิงหินยิงเพื่อหยุดศัตรูต่าง ๆ ที่พยายามทำลายปราสาทของคุณ คุณต้องใช้ปืนยิงหินยิงของคุณอย่างชาญฉลาดเพื่อหยุดศัตรูที่แข็งแกร่งที่สุดที่เข้ามา พร้อมทั้งจะทำลายปราสาทของคุณ หากศัตรูเข้าไปถึงหน้าปราสาทได้ละก็ มันจะสร้างความเสียหายอย่างมากต่อปราสาท แต่ปราสาทของเราสามารถอัปเกรดบ้อมปืนได้ เพื่อช่วยเราโจมตีศัตรูที่มารุกราน เรามาปกป้องปราสาทกันเถอะ!

เกมนี้มีแนวคิดคล้ายกับเกมชื่อดังอย่าง **plants vs zombie\*** ที่ภายในเกมจะมีระบบการสร้างบ้อมปืน(พืช)เพื่อป้องกันบ้านที่จะถูกซอมบี้มารุกราน แต่เกมของเราจะเป็นการปกป้องปราสาทจากการรุกรานของปีศาจ



plants vs zombie\*

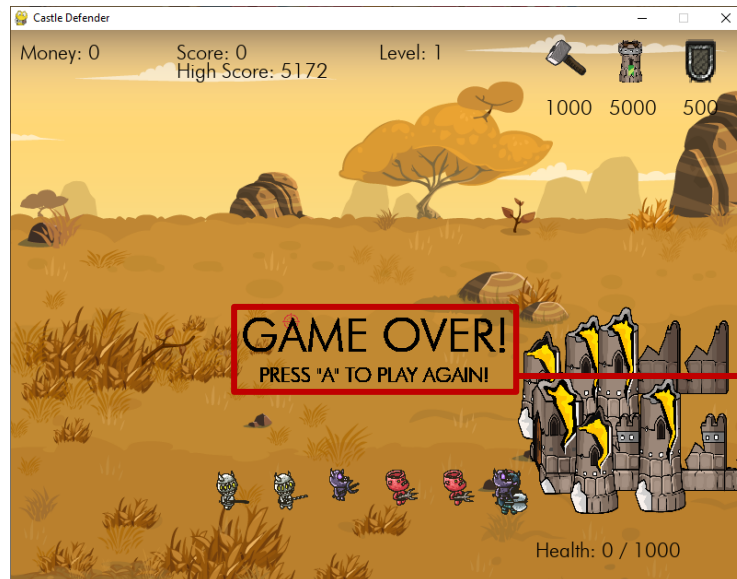
## วิธีเล่น



## ภาพภายในเกม



## ภาพภายในเกม(ต่อ)



Game Over (แพ้)  
กด A เพื่อเล่นใหม่อีกครั้ง

## ไฟล์ภาพ Object ต่าง ๆ



Goblin



knight



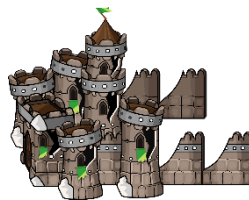
purple\_goblin



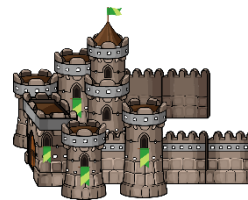
red\_goblin



castle\_25



castle\_50



castle\_100



tower\_25



tower\_50

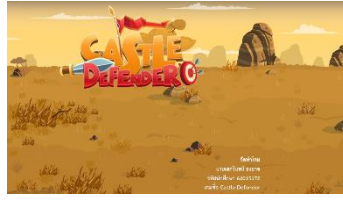


tower\_100

## ไฟล์ภาพ Object ต่าง ๆ (ต่อ)



bg



bgmenu



start\_btn



exit\_btn



## Crosshair



bullet



repair



armour

Credits for assets used:

<https://craftpix.net/freebies/free-castle-2d-game-assets/>

<https://free-game-assets.itch.io/free-stone-tower-defense-game-art>

[https://toppng.com/crosshair-png-cliparts-goal-ico-PNG-free-PNG-Images\\_192784](https://toppng.com/crosshair-png-cliparts-goal-ico-PNG-free-PNG-Images_192784)

ข้อมูลไฟล์โค้ด(ย่อ)



✔ button



✔ enemy



main



 score

button.py

- mouse and button

enemy.py

- enemy animation

- enemy info

main.py

- game play

- Calculate values

- functions, classes, etc.

score.py

- score (high score)

## อธิบายโค้ดไฟล์ main.py

```
main.py > ...
1  #import libraries
2  import pygame
3  import math
4  import sys
5  import random
6  import os
7  from enemy import Enemy
8  import button
9  from pygame import mixer
10
```

#นำเข้าไลบรารี

```
11 #initialise pygame
12 pygame.init()
13
14 #game window
15 SCREEN_WIDTH = 800
16 SCREEN_HEIGHT = 600
17
18 #create game window
19 screen = pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT))
20 pygame.display.set_caption('Castle Defender')
21
22 clock = pygame.time.Clock()
23 FPS = 60
```

#สร้างหน้าต่างเกม

```
25 #define game variables
26 level = 1
27 high_score = 0
28 level_difficulty = 0
29 target_difficulty = 1000
30 DIFFICULTY_MULTIPLIER = 1.1
31 game_over = False
32 next_level = False
33 ENEMY_TIMER = 1000
34 last_enemy = pygame.time.get_ticks()
35 enemies_alive = 0
36 max_towers = 4
37 TOWER_COST = 5000
38 tower_positions = [
39     [SCREEN_WIDTH - 200, SCREEN_HEIGHT - 150],
40     [SCREEN_WIDTH - 150, SCREEN_HEIGHT - 150],
41     [SCREEN_WIDTH - 100, SCREEN_HEIGHT - 150],
42     [SCREEN_WIDTH - 50, SCREEN_HEIGHT - 150],
43 ]
```

#สร้างตัวแปรที่ใช้ภายในเกม

```
46 #load high score
47 if os.path.exists('score.txt'):
48     with open('score.txt', 'r') as file:
49         high_score = int(file.read())
50
```

#ดึงข้อมูลจากไฟล์ high score มาเก็บไว้ในตัวแปร

```
51 #define colours
52 WHITE = (255, 255, 255)
53 BLACK = (0,0,0)
```

#สร้างตัวแปรค่าสี

```

55 #mixer
56 mixer.music.load('sound\sound_bg.mp3')
57 mixer.music.play(-1)

```

#เพลงประกอบพื้นหลังภายในเกม

```

63 #load button images
64 start_img = pygame.image.load('img/start_btn.png').convert_alpha()
65 exit_img = pygame.image.load('img/exit_btn.png').convert_alpha()
66
67 #load images
68 bg = pygame.image.load('img/bg.png').convert_alpha()
69 bgmenu = pygame.image.load('img/bgmenu.png').convert_alpha()
70
71 #castle
72 castle_img_100 = pygame.image.load('img/castle/castle_100.png').convert_alpha()
73 castle_img_50 = pygame.image.load('img/castle/castle_50.png').convert_alpha()
74 castle_img_25 = pygame.image.load('img/castle/castle_25.png').convert_alpha()
75
76 #tower
77 tower_img_100 = pygame.image.load('img/tower/tower_100.png').convert_alpha()
78 tower_img_50 = pygame.image.load('img/tower/tower_50.png').convert_alpha()
79 tower_img_25 = pygame.image.load('img/tower/tower_25.png').convert_alpha()
80
81 #bullet image
82 bullet_img = pygame.image.load('img/bullet.png').convert_alpha()
83 b_w = bullet_img.get_width()
84 b_h = bullet_img.get_height()
85 bullet_img = pygame.transform.scale(bullet_img, (int(b_w * 0.075), int(b_h * 0.075)))
86
87 #load enemies
88 enemy_animations = []
89 enemy_types = ['knight', 'goblin', 'purple_goblin', 'red_goblin']
90 enemy_health = [75, 100, 125, 150]
91
92 animation_types = ['walk', 'attack', 'death']
93 for enemy in enemy_types:
94     #load animation
95     animation_list = []
96     for animation in animation_types:
97         #reset temporary list of images
98         temp_list = []
99         #define number of frames
100         num_of_frames = 20
101         for i in range(num_of_frames):
102             img = pygame.image.load(f'img/enemies/{enemy}/{animation}/{i}.png').convert_alpha()
103             e_w = img.get_width()
104             e_h = img.get_height()
105             img = pygame.transform.scale(img, (int(e_w * 0.2), int(e_h * 0.2)))
106             temp_list.append(img)
107         animation_list.append(temp_list)
108     enemy_animations.append(animation_list)
109
110 #button images
111 #repair image
112 repair_img = pygame.image.load('img/repair.png').convert_alpha()
113 #armour image
114 armour_img = pygame.image.load('img/armour.png').convert_alpha()
115
116 #button images
117 #repair image
118 repair_img = pygame.image.load('img/repair.png').convert_alpha()
119 #armour image
120 armour_img = pygame.image.load('img/armour.png').convert_alpha()
121

```

#นำไฟล์ Object ต่างๆ ใส่ไว้ในตัวแปร

```

112 #function for outputting text onto the screen
113 def draw_text(text, font, text_col, x, y):
114     img = font.render(text, True, text_col)
115     screen.blit(img, (x, y))
116
117
118 #function for displaying status
119 def show_info():
120     draw_text("Money: " + str(castle.money), font, BLACK, 10, 10)
121     draw_text("Score: " + str(castle.score), font, BLACK, 180, 10)
122     draw_text("High Score: " + str(high_score), font, BLACK, 180, 30)
123     draw_text("Level: " + str(level), font, BLACK, SCREEN_WIDTH // 2, 10)
124     draw_text("Health: " + str(castle.health) + " / " + str(castle.max_health), font, BLACK, SCREEN_WIDTH - 230, SCREEN_HEIGHT - 50)
125     draw_text("1000", font, BLACK, SCREEN_WIDTH - 220, 70)
126     draw_text(str(TOWER_COST), font, BLACK, SCREEN_WIDTH - 150, 70)
127     draw_text("500", font, BLACK, SCREEN_WIDTH - 70, 70)

```

#ฟังก์ชันสำหรับแสดงข้อความต่าง ๆ ภายในเกม

```

129 #castle class
130 class Castle():
131     def __init__(self, image100, image50, image25, x, y, scale):
132         self.health = 1000
133         self.max_health = self.health
134         self.fired = False
135         self.money = 0
136         self.score = 0
137
138         width = image100.get_width()
139         height = image100.get_height()
140
141         self.image100 = pygame.transform.scale(image100, (int(width * scale), int(height * scale)))
142         self.image50 = pygame.transform.scale(image50, (int(width * scale), int(height * scale)))
143         self.image25 = pygame.transform.scale(image25, (int(width * scale), int(height * scale)))
144         self.rect = self.image100.get_rect()
145         self.rect.x = x
146         self.rect.y = y
147
148
149     def shoot(self):
150         pos = pygame.mouse.get_pos()
151         x_dist = pos[0] - self.rect.midleft[0]
152         y_dist = -(pos[1] - self.rect.midleft[1])
153         self.angle = math.degrees(math.atan2(y_dist, x_dist))
154         #get mouseclick
155         if pygame.mouse.get_pressed()[0] and self.fired == False and pos[1] > 70:
156             self.fired = True
157             bullet = Bullet(bullet_img, self.rect.midleft[0], self.rect.midleft[1], self.angle)
158             bullet_group.add(bullet)
159         #reset mouseclick
160         if pygame.mouse.get_pressed()[0] == False:
161             self.fired = False
162
163
164     def draw(self):
165         #check which image to use based on health
166         if self.health <= 250:
167             self.image = self.image25
168         elif self.health <= 500:
169             self.image = self.image50
170         else:
171             self.image = self.image100
172
173         screen.blit(self.image, self.rect)
174
175
176     def repair(self):
177         if self.money >= 1000 and self.health < self.max_health:
178             self.health += 500
179             self.money -= 1000
180             if castle.health > castle.max_health:
181                 castle.health = castle.max_health
182
183
184     def armour(self):
185         if self.money >= 500:
186             self.max_health += 250
187             self.money -= 500

```

#คลาสของปราสาท ประกอบด้วยฟังก์ชันของ ปราสาท การคลิก เลือดของปราสาท เลือดสูงสุด



```

189 #tower class
190 class Tower(pygame.sprite.Sprite):
191     def __init__(self, image100, image50, image25, x, y, scale):
192         pygame.sprite.Sprite.__init__(self)
193
194         self.got_target = False
195         self.angle = 0
196         self.last_shot = pygame.time.get_ticks()
197
198         width = image100.get_width()
199         height = image100.get_height()
200
201         self.image100 = pygame.transform.scale(image100, (int(width * scale), int(height * scale)))
202         self.image50 = pygame.transform.scale(image50, (int(width * scale), int(height * scale)))
203         self.image25 = pygame.transform.scale(image25, (int(width * scale), int(height * scale)))
204         self.image = self.image100
205         self.rect = self.image100.get_rect()
206         self.rect.x = x
207         self.rect.y = y
208
209
210     def update(self, enemy_group):
211         self.got_target = False
212
213         for e in enemy_group:
214             if e.alive:
215                 target_x, target_y = e.rect.midbottom
216                 self.got_target = True
217                 break
218
219         if self.got_target:
220             x_dist = target_x - self.rect.midleft[0]
221             y_dist = -(target_y - self.rect.midleft[1])
222             self.angle = math.degrees(math.atan2(y_dist, x_dist))
223
224             shot_cooldown = 1000
225             #fire bullet
226             if pygame.time.get_ticks() - self.last_shot > shot_cooldown:
227                 self.last_shot = pygame.time.get_ticks()
228                 bullet = Bullet(bullet_img, self.rect.midleft[0], self.rect.midleft[1], self.angle)
229                 bullet_group.add(bullet)
230
231         #check which image to use based on health
232         if castle.health <= 250:
233             self.image = self.image25
234         elif castle.health <= 500:
235             self.image = self.image50
236         else:
237             self.image = self.image100
238
239

```

#คลาสของป้อมปืน ประกอบด้วยฟังก์ชันของ ป้อมปืน การยิง เลือดของปราสาท

```

241 #bullet class
242 class Bullet(pygame.sprite.Sprite):
243     def __init__(self, image, x, y, angle):
244         pygame.sprite.Sprite.__init__(self)
245         self.image = image
246         self.rect = self.image.get_rect()
247         self.rect.x = x
248         self.rect.y = y
249         self.angle = math.radians(angle)#convert input angle into radians
250         self.speed = 10
251         #calculate the horizontal and vertical speeds based on the angle
252         self.dx = math.cos(self.angle) * self.speed
253         self.dy = -(math.sin(self.angle) * self.speed)
254
255
256     def update(self):
257         #check if bullet has gone off the screen
258         if self.rect.right < 0 or self.rect.left > SCREEN_WIDTH or self.rect.bottom < 0 or self.rect.top > SCREEN_HEIGHT:
259             self.kill()
260
261         #move bullet
262         self.rect.x += self.dx
263         self.rect.y += self.dy
264

```

#คลาสของลูกกระสุน ประกอบด้วยฟังก์ชันของ การเคลื่อนที่ของกระสุน



```

265 class Crosshair():
266     def __init__(self, scale):
267         image = pygame.image.load('img/crosshair.png').convert_alpha()
268         width = image.get_width()
269         height = image.get_height()
270
271         self.image = pygame.transform.scale(image, (int(width * scale), int(height * scale)))
272         self.rect = self.image.get_rect()
273
274         #hide mouse
275         pygame.mouse.set_visible(False)
276
277     def draw(self):
278         mx, my = pygame.mouse.get_pos()
279         self.rect.center = (mx, my)
280         screen.blit(self.image, self.rect)
281

```

#คลาสของเป้า ประกอบด้วยฟังก์ชันของ การเปลี่ยนเมาส์ให้เป็นรูปเป้าปืน

```

283 #create castle
284 castle = Castle(castle_img_100, castle_img_50, castle_img_25, SCREEN_WIDTH - 250, SCREEN_HEIGHT - 300, 0.2)
285
286 #create buttons
287 repair_button = button.Button(SCREEN_WIDTH - 220, 10, repair_img, 0.5)
288 tower_button = button.Button(SCREEN_WIDTH - 140, 10, tower_img_100, 0.1)
289 armour_button = button.Button(SCREEN_WIDTH - 75, 10, armour_img, 1.5)
290
291 #create groups
292 tower_group = pygame.sprite.Group()
293 bullet_group = pygame.sprite.Group()
294 enemy_group = pygame.sprite.Group()
295
296 #load button images
297 start_img = pygame.image.load('img/start_btn.png').convert_alpha()
298 exit_img = pygame.image.load('img/exit_btn.png').convert_alpha()
299
300 #create button instances
301 start_button = button.Button(150, 320, start_img, 0.8)
302 exit_button = button.Button(450, 320, exit_img, 0.8)
303

```

#เรียกใช้งานคลาสและกำหนดตัวแปรใส่ลงไปในคลาส

```

305 def game_loop():
306     #game loop
307     run = True
308     while run:
309
310         clock.tick(FPS)
311         global game_over, level_difficulty, target_difficulty, last_enemy, next_level, high_score, level
312         if game_over == False:
313             screen.blit(bg, (0, 0))
314
315             #draw castle
316             castle.draw()
317             castle.shoot()
318             #draw towers
319             tower_group.draw(screen)
320             tower_group.update(enemy_group)
321
322             #draw crosshair
323             crosshair.draw()
324
325             #draw bullets
326             bullet_group.update()
327             bullet_group.draw(screen)
328
329             #draw enemies
330             enemy_group.update(screen, castle, bullet_group)
331
332             #show details
333             show_info()
334
335             #draw buttons
336             if repair_button.draw(screen):
337                 castle.repair()

```

#ฟังก์ชันเกี่ยวกับการแสดง object ต่างๆ

```

338     if tower_button.draw(screen):
339         #check if there is enough money and build a tower
340         if castle.money >= TOWER_COST and len(tower_group) < max_towers:
341             tower = Tower(
342                 tower_img_100,
343                 tower_img_50,
344                 tower_img_25,
345                 tower_positions[len(tower_group)][0],
346                 tower_positions[len(tower_group)][1],
347                 0.2
348             )
349             tower_group.add(tower)
350             #subtract money
351             castle.money -= TOWER_COST
352     if armour_button.draw(screen):
353         castle.armour()
354
355     #create enemies
356     #check if max number of enemies has been reached
357     if level_difficulty < target_difficulty:
358         if pygame.time.get_ticks() - last_enemy > ENEMY_TIMER:
359             #create enemies
360             e = random.randint(0, len(enemy_types) - 1)
361             enemy = Enemy(enemy_health[e], enemy_animations[e], -100, SCREEN_HEIGHT - 100, 1)
362             enemy_group.add(enemy)
363             #reset enemy timer
364             last_enemy = pygame.time.get_ticks()
365             #increase level difficulty by enemy health
366             level_difficulty += enemy_health[e]

```

#ตรวจสอบเงินในการสร้างป้อม และ การสร้างศัตรู

```

369     #check if all the enemies have been spawned
370     if level_difficulty >= target_difficulty:
371         #check how many are still alive
372         enemies_alive = 0
373         for e in enemy_group:
374             if e.alive == True:
375                 enemies_alive += 1
376         #if there are none alive the level is complete
377         if enemies_alive == 0 and next_level == False:
378             next_level = True
379             level_reset_time = pygame.time.get_ticks()
380
381     #move onto the next level
382     if next_level == True:
383         draw_text('LEVEL COMPLETE!', font_60, WHITE, 200, 300)
384         #update high score
385         if castle.score > high_score:
386             high_score = castle.score
387             with open('score.txt', 'w') as file:
388                 file.write(str(high_score))
389         if pygame.time.get_ticks() - level_reset_time > 1500:
390             next_level = False
391             level += 1
392             last_enemy = pygame.time.get_ticks()
393             target_difficulty *= DIFFICULTY_MULTIPLIER
394             level_difficulty = 0
395             enemy_group.empty()
396
397     #check game over
398     if castle.health <= 0:
399         game_over = True
400

```

#ตรวจสอบจำนวนศัตรู อัปเดตเลเวล เช็คเกมโอเวอร์

```

401         else:
402             draw_text('GAME OVER!', font_60, BLACK, 250, 300)
403             draw_text('PRESS "A" TO PLAY AGAIN!', font, BLACK, 270, 360)
404             pygame.mouse.set_visible(True)
405             key = pygame.key.get_pressed()
406             if key[pygame.K_a]:
407                 #reset variables
408                 game_over = False
409                 level = 1
410                 target_difficulty = 1000
411                 level_difficulty = 0
412                 last_enemy = pygame.time.get_ticks()
413                 enemy_group.empty()
414                 tower_group.empty()
415                 castle.score = 0
416                 castle.health = 1000
417                 castle.max_health = castle.health
418                 castle.money = 0
419                 pygame.mouse.set_visible(False)
420
421
422
423             #event handler
424             for event in pygame.event.get():
425                 if event.type == pygame.QUIT:
426                     run = False
427
428             #update display window
429             pygame.display.update()
430             pygame.quit()
431             sys.exit()

```

#แสดงเกมโอเวอร์ และรีเซ็ตเกม

```

433 #game menu
434 run = True
435 while run:
436
437     screen.blit(bgmenu, (0, 0))
438
439     if start_button.draw(screen):
440         #create crosshair
441         crosshair = Crosshair(0.025)
442         #start game
443         game_loop()
444     if exit_button.draw(screen):
445         run = False
446
447     #event handler
448     for event in pygame.event.get():
449         #quit game
450         if event.type == pygame.QUIT:
451             run = False
452
453     pygame.display.update()
454
455     pygame.quit()
456     sys.exit()
457

```

#แสดงหน้าเมนูเกม

## อธิบายโค้ดไฟล์ enemy.py

```
enemy.py > Enemy > __init__
1  import pygame
2
3  class Enemy(pygame.sprite.Sprite):
4      def __init__(self, health, animation_list, x, y, speed):
5          pygame.sprite.Sprite.__init__(self)
6          self.alive = True
7          self.speed = speed
8          self.health = health
9          self.last_attack = pygame.time.get_ticks()
10         self.attack_cooldown = 1000
11         self.animation_list = animation_list
12         self.frame_index = 0
13         self.action = 0#0: walk, 1: attack, 2: death
14         self.update_time = pygame.time.get_ticks()
15
16         #select starting image
17         self.image = self.animation_list[self.action][self.frame_index]
18         self.rect = pygame.Rect(0, 0, 25, 40)
19         self.rect.center = (x, y)
20
21
22     def update(self, surface, target, bullet_group):
23         if self.alive:
24             #check for collision with bullets
25             if pygame.sprite.spritecollide(self, bullet_group, True):
26                 #lower enemy health
27                 self.health -= 15
28
29             #check if enemy has reached the castle
30             if self.rect.right > target.rect.left:
31                 self.update_action(1)
32
33             #move enemy
34             if self.action == 0:
35                 #update rectangle position
36                 self.rect.x += self.speed
37
38             #attack
39             if self.action == 1:
40                 #check if enough time has passed since last attack
41                 if pygame.time.get_ticks() - self.last_attack > self.attack_cooldown:
42                     target.health -= 100
43                     if target.health < 0:
44                         target.health = 0
45                     self.last_attack = pygame.time.get_ticks()
46
```

#คลาสและฟังก์ชันข้อมูลศัตรู และ แอนิเมชั่นศัตรู 1/2

```

48         #check if health has dropped to zero
49         if self.health <= 0:
50             target.money += 100
51             target.score += 100
52             self.update_action(2)#death
53             self.alive = False
54
55         self.update_animation()
56
57         #draw image on screen
58         surface.blit(self.image, (self.rect.x - 10, self.rect.y - 15))
59
60
61     def update_animation(self):
62         #define animation cooldown
63         ANIMATION_COOLDOWN = 50
64         #update image depending on current action
65         self.image = self.animation_list[self.action][self.frame_index]
66         #check if enough time has passed since the last update
67         if pygame.time.get_ticks() - self.update_time > ANIMATION_COOLDOWN:
68             self.update_time = pygame.time.get_ticks()
69             self.frame_index += 1
70         #if the animation has run out then reset back to the start
71         if self.frame_index >= len(self.animation_list[self.action]):
72             if self.action == 2:
73                 self.frame_index = len(self.animation_list[self.action]) - 1
74             else:
75                 self.frame_index = 0
76
77
78     def update_action(self, new_action):
79         #check if the new action is different to the previous one
80         if new_action != self.action:
81             self.action = new_action
82             #update the animation settings
83             self.frame_index = 0
84             self.update_date = pygame.time.get_ticks()
85

```

#คลาสและฟังก์ชันข้อมูลศัตรู และ แอนิเมชันศัตรู 2/2

## อธิบายโค้ดไฟล์ button.py

```

button.py > Button > draw
1  import pygame
2
3  #button class
4  class Button():
5      def __init__(self, x, y, image, scale):
6          width = image.get_width()
7          height = image.get_height()
8          self.image = pygame.transform.scale(image, (int(width * scale), int(height * scale)))
9          self.rect = self.image.get_rect()
10         self.rect.topleft = (x, y)
11         self.clicked = False
12
13     def draw(self, surface):
14         action = False
15         #get mouse position
16         pos = pygame.mouse.get_pos()
17
18         #check mouseover and clicked conditions
19         if self.rect.collidepoint(pos):
20             if pygame.mouse.get_pressed()[0] == 1 and self.clicked == False:
21                 self.clicked = True
22                 action = True
23
24             if pygame.mouse.get_pressed()[0] == 0:
25                 self.clicked = False
26
27         #draw button on screen
28         surface.blit(self.image, (self.rect.x, self.rect.y))
29
30     return action

```

#คลาสและฟังก์ชันเกี่ยวกับการเก็บข้อมูลปุ่มต่างๆ ภายในเกม

