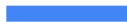


Game



part6

Count time

```
import pygame
...
pygame.init()
text = font.render('bulabula', False, (0, 255, 0), (0, 0, 255))

done = False
while not done:
    ...
    screen.fill(background_colour)
    allspriteslist.draw(screen)
    screen.blit(text, (400, 100))

    time = font.render(str(pygame.time.get_ticks()//1000), False, (0, 255, 0), (0, 0, 255))
    screen.blit(time, (100, 100))

    pygame.display.flip()
    allspriteslist.update()

    clock.tick(120)
pygame.quit()
```

Recap:
We have a second counter.
Before we make it count
down, let's take a closer to
the time. What do you see?
Does it count from zero?

Count time

```
import pygame
...
pygame.init()
print(pygame.time.get_ticks())
font = pygame.font.SysFont("comicsansmsttf", 72)
text = font.render('bulabula', False, (0, 255, 0), (0, 0, 255))

done = False
while not done:
    ...
    screen.fill(background_colour)
    allspriteslist.draw(screen)
    screen.blit(text, (400, 100))

    time = font.render(str(pygame.time.get_ticks()//1000), False, (0, 255, 0), (0, 0, 255))
    screen.blit(time, (100, 100))

    pygame.display.flip()
    allspriteslist.update()

    clock.tick(120)
```

It is weird right?
How come it doesn't count from 0?
Let's print out the elapsed time after we call `pygame.init()`
What do you see?

Count time

```
import pygame
...
pygame.init()
print(pygame.time.get_ticks())
font = pygame.font.SysFont("comicsansmsttf", 72)
print(pygame.time.get_ticks())
text = font.render('bulabula', False, (0, 255, 0), (0, 0, 255))
```

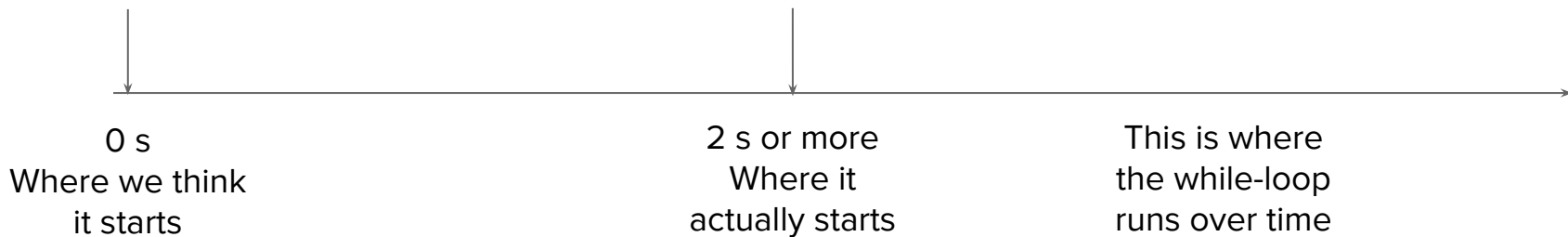
```
done = False
while not done:
```

```
...
    screen.fill(background_colour)
    allspriteslist.draw(screen)
    screen.blit(text, (400, 100))
    time = font.render(str(pygame.time.get_ticks()//1000), False, (0, 255, 0), (0, 0, 255))
    screen.blit(time, (100, 100))
    pygame.display.flip()
    allspriteslist.update()
```

```
    clock.tick(120)
pygame.quit()
```

Let's put one more after the `font =` statement. This line loads a font from your computer.

So what does it mean?



We thought the time should be counting up from zero, but in fact it starts at later than that. That is because loading the font is a heavy-lifting work. By the time it gets to have the game running, it has already been some time.

Count time

```
import pygame
```

```
...
```

```
text = font.render(str(pygame.time.get_ticks()), False, (0, 255, 0), (0, 0, 255))
```

```
print(text.get_width())
```

```
print(text.get_height())
```

```
done = False
```

```
while not done:
```

```
...
```

```
    screen.fill(background_colour)
```

```
    allspriteslist.draw(screen)
```

```
    time = font.render(str(pygame.time.get_ticks()//1000), False, (0, 255, 0), (0, 0, 255))
```

```
    screen.blit(time, (100, 100))
```

```
    pygame.display.flip()
```

```
    allspriteslist.update()
```

```
    clock.tick(120)
```

```
pygame.quit()
```

Q&A:

What is

`pygame.time.get_ticks()`?

Count time

```
import pygame
```

```
...
```

```
text = font.render(str(pygame.time.get_ticks()), False, (0, 255, 0), (0, 0, 255))
```

```
print(text.get_width())
```

```
print(text.get_height())
```

```
done = False
```

```
while not done:
```

```
...
```

```
    screen.fill(background_colour)
```

```
    allspriteslist.draw(screen)
```

```
    time = font.render(str(pygame.time.get_ticks()//1000), False, (0, 255, 0), (0, 0, 255))
```

```
    screen.blit(time, (100, 100))
```

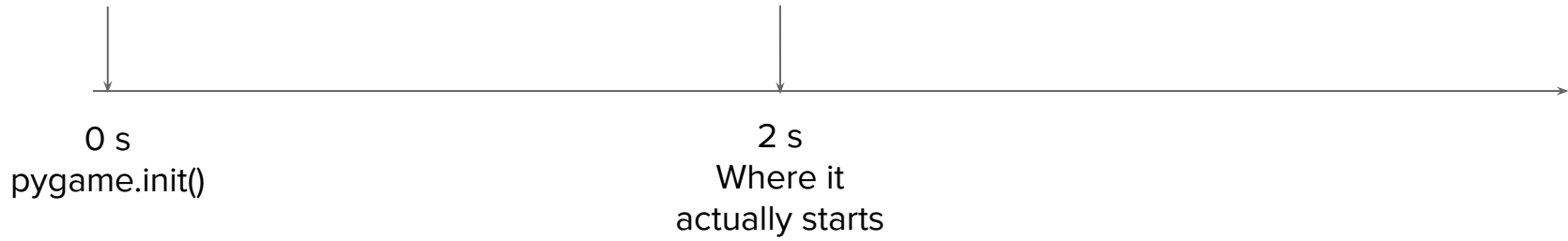
```
    pygame.display.flip()
```

```
    allspriteslist.update()
```

```
    clock.tick(120)
```

```
pygame.quit()
```

At which second it
start? Why?



Challenge

How to count from 0s?

Count time from 0s

```
import pygame
```

```
...
```

```
text = font.render(str(pygame.time.get_ticks()//1000), False, (0, 255, 0), (0, 0, 255))
```

```
print(text.get_width())
```

```
print(text.get_height())
```

```
done = False
```

```
while not done:
```

```
...
```

```
    screen.fill(background_colour)
```

```
    allspriteslist.draw(screen)
```

```
    screen.blit(text, (400, 100))
```

```
    time = font.render(str(pygame.time.get_ticks()//1000), False, (0, 255, 0), (0, 0, 255))
```

```
    screen.blit(time, (100, 100))
```

```
    pygame.display.flip()
```

```
    allspriteslist.update()
```

```
    clock.tick(120)
```

```
pygame.quit()
```

Let's remove these lines as we don't need them.

Count time from 0s

```
import pygame
```

```
...
```

```
done = False
```

```
while not done:
```

```
...
```

```
    screen.fill(background_colour)
```

```
    allspriteslist.draw(screen)
```

```
    time = font.render(str(pygame.time.get_ticks()//1000), False, (0, 255, 0), (0, 0, 255))
```

```
    screen.blit(time, (100, 100))
```

```
    pygame.display.flip()
```

```
    allspriteslist.update()
```

```
    clock.tick(120)
```

```
pygame.quit()
```

This part is counting up, but just does not start at a right number. So we will modify this.

Count time from 0s

```
import pygame
```

```
...
```

```
start_time = pygame.time.get_ticks()  
print(start_time)
```

```
done = False  
while not done:
```

```
...
```

```
    screen.fill(background_colour)  
    allspriteslist.draw(screen)
```

```
    time = font.render(str(pygame.time.get_ticks()//1000), False, (0, 255, 0), (0, 0, 255))  
    screen.blit(time, (100, 100))
```

```
    pygame.display.flip()  
    allspriteslist.update()
```

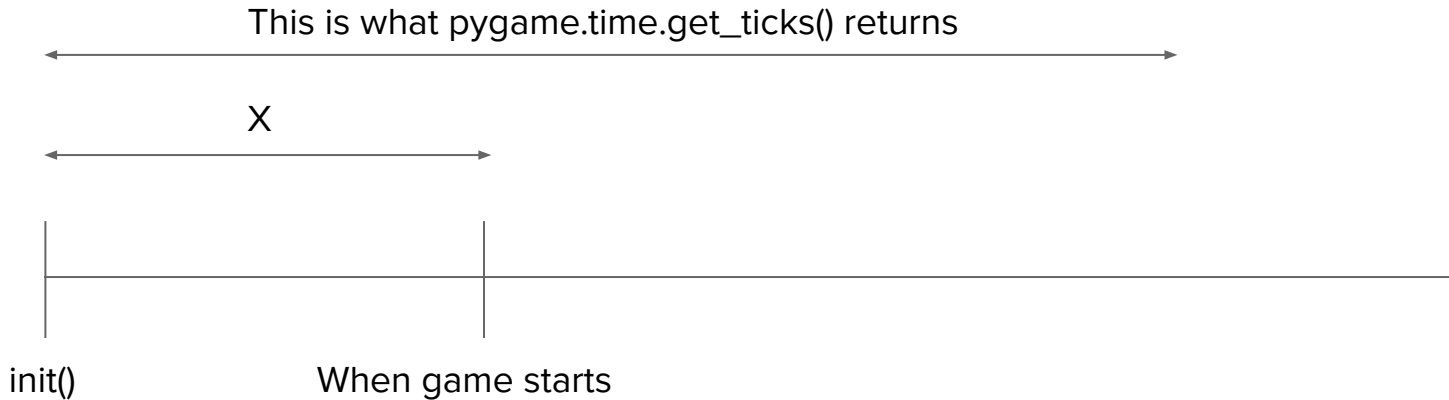
```
    clock.tick(120)  
pygame.quit()
```

Before we modify that, we need to understand how to offset the time it takes for loading the font.

We need to know at what time the font has finished loading.

Count time from 0s

Let say it takes x amount of time to finish loading the font.



The time we want to display is:

Total elapsed time since `pygame.init()` MINUS the time it took for font

In other word, `pygame.time.get_ticks()` - x

Count time from 0s

```
import pygame
```

```
...
```

```
start_time = pygame.time.get_ticks()
```

```
print(start_time)
```

```
done = False
```

```
while not done:
```

```
...
```

```
    screen.fill(background_colour)
```

```
    allspriteslist.draw(screen)
```

```
    time = font.render(str((pygame.time.get_ticks()-start_time)//1000), False, (0, 255, 0), (0, 0, 255))
```

```
    screen.blit(time, (100, 100))
```

```
    pygame.display.flip()
```

```
    allspriteslist.update()
```

```
    clock.tick(120)
```

```
pygame.quit()
```

Question

Time to count down!

Count time ----

```
import pygame
```

```
...
```

```
start_time = pygame.time.get_ticks()
```

```
time_left = 3000
```

```
done = False
```

```
while not done:
```

```
...
```

```
    screen.fill(background_colour)
```

```
    allspriteslist.draw(screen)
```

```
    time = font.render(str((pygame.time.get_ticks()-start_time)//1000), False, (0, 255, 0), (0, 0, 255))
```

```
    screen.blit(time, (100, 100))
```

```
    pygame.display.flip()
```

```
    allspriteslist.update()
```

```
...
```

```
pygame.quit()
```

To count down, we must have a number we want to count down from.

Count time ----

```
import pygame
...
start_time = pygame.time.get_ticks()
time_left = 3000

done = False
while not done:
    ...

    screen.fill(background_colour)
    allspriteslist.draw(screen)

    remaining_time = (time_left-(pygame.time.get_ticks()-start_time))//1000
    print(remaining_time)
    time = font.render(str(remaining_time), False, (0, 255, 0), (0, 0, 255))

    screen.blit(time, (100, 100))
    pygame.display.flip()
    allspriteslist.update()

    ...
pygame.quit()
```

time_left will always be the same.
We should use 3 second minus whatever time has passed.
If time has passed is 1 second.
 $3-1 = 2$ we should show 2 seconds remaining

If time has passed is 2 second.
 $3-2 = 1$ we should show 1 second remaining

Don't forget to render

Count time ----

```
import pygame
```

```
...
```

```
start_time = pygame.time.get_ticks()
```

```
time_left = 3000
```

```
done = False
```

```
while not done:
```

```
...
```

```
    screen.fill(background_colour)
```

```
    allspriteslist.draw(screen)
```

```
    remaining_time = (time_left-pygame.time.get_ticks()+start_time + 500)//1000
```

```
    print(remaining_time)
```

```
    time = font.render(str(remaining_time), False, (0, 255, 0), (0, 0, 255))
```

```
    screen.blit(time, (100, 100))
```

```
    pygame.display.flip()
```

```
    allspriteslist.update()
```

```
...
```

```
pygame.quit()
```

**Do you really see the number 3
when your code is running?**

Clicked count

```
import pygame
```

```
...
```

```
start_time = pygame.time.get_ticks()
```

```
time_left = 3000
```

```
clicked_count = 0
```

```
done = False
```

```
while not done:
```

```
...
```

```
    screen.fill(background_colour)
```

```
    allspriteslist.draw(screen)
```

```
    remaining_time = (time_left-pygame.time.get_ticks()-start_time + 500)//1000
```

```
    print(remaining_time)
```

```
    time = font.render(str(remaining_time), False, (0, 255, 0), (0, 0, 255))
```

```
    screen.blit(time, (100, 100))
```

```
    pygame.display.flip()
```

```
...
```

```
pygame.quit()
```

We are going to keep track of how many squares the player has clicked. So we will need a variable.

Clicked count ----

```
import pygame
```

```
...
```

```
start_time = pygame.time.get_ticks()
```

```
time_left = 3000
```

```
clicked_count = 0
```

```
done = False
```

```
while not done:
```

```
...
```

```
    screen.fill(background_colour)
```

```
    allspriteslist.draw(screen)
```

```
    remaining_time = (time_left-pygame.time.get_ticks()-start_time + 500)//1000
```

```
    print(remaining_time)
```

```
    time = font.render(str(remaining_time), False, (0, 255, 0), (0, 0, 255))
```

```
    screen.blit(time, (100, 100))
```

```
    clicked_count_text = font.render(str(clicked_count), False, (0, 255, 0), (0, 0, 255))
```

```
    screen.blit(clicked_count_text, (300, 100))
```

```
...
```

```
pygame.quit()
```

**Create another text and display it.
The text will get changed from
time to time, but when?**

Clicked count ----

.....

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

```
    if event.type == pygame.MOUSEBUTTONDOWN:
```

```
        pos = pygame.mouse.get_pos()
```

```
    for sprite in allspriteslist:
```

```
        if sprite.rect.collidepoint(pos):
```

```
            clicked_count = clicked_count + 1
```

```
            sprite.remove(allspriteslist)
```

....

If clicked_count is changed, the displayed text should also be changed too, give that the previous setup is correct

Question

Draw a new screen when remaining_time is less than 0

Draw new screen

```
import pygame
```

```
...
```

```
while not done:
```

```
...
```

```
    screen.fill(background_colour)  
    allspriteslist.draw(screen)
```

```
    remaining_time = (time_left-(pygame.time.get_ticks()-start_time + 500)//1000
```

```
    if remaining_time<=0:  
        print(remaining_time)
```

```
...
```

```
    clicked_count_text = font.render(str(clicked_count), False, (0, 255, 0), (0, 0, 255))  
    screen.blit(clicked_count_text, (300, 100))
```

```
...
```

```
pygame.quit()
```

We are going to keep track of how many squares the player has clicked. So we will need a variable.

We will just print the remaining time it is zero or negative.

Draw new screen

```
import pygame
```

```
...
```

```
while not done:
```

```
...
```

```
    screen.fill(background_colour)
```

```
    allspriteslist.draw(screen)
```

```
    remaining_time = (time_left-(pygame.time.get_ticks()-start_time + 500))//1000
```

```
    if remaining_time <= 0:
```

```
        print(remaining_time)
```

```
    else:
```

```
        allspriteslist.draw(screen)
```

```
...
```

```
    clicked_count_text = font.render(str(clicked_count), False, (0, 255, 0), (0, 0, 255))
```

```
    screen.blit(clicked_count_text, (300, 100))
```

```
...
```

```
pygame.quit()
```

For else, meaning there is still remaining time. The game should go on. And we keep drawing the Squares!

Draw new screen

```
import pygame
```

```
...
```

```
while not done:
```

```
...
```

```
    screen.fill(background_colour)
```

```
    remaining_time = (time_left-(pygame.time.get_ticks()-start_time + 500)//1000
```

```
    if remaining_time <= 0:
```

```
        remaining_time = 0
```

```
        over_text = font.render("Game Over", False, (0, 255, 0), (0, 0, 255))
```

```
        screen.blit(over_text, (200,300))
```

```
    else:
```

```
        allspriteslist.draw(screen)
```

```
...
```

```
    clicked_count_text = font.render(str(clicked_count), False, (0, 255, 0), (0, 0, 255))
```

```
    screen.blit(clicked_count_text, (300, 100))
```

```
...
```

```
pygame.quit()
```

First we try not to show negative time as it doesn't make sense.

Then we display another text!

For GAME OVER!

Challenge

When “game over” show up, we can still click squares even though we can not see them.

How to fix it?