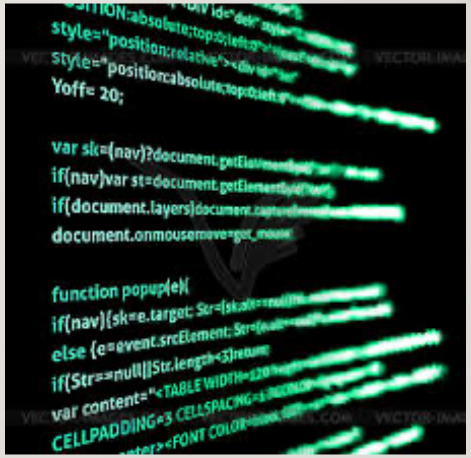
LEARNING ABOUT PYGAME

JAVIER NG

WHAT IS PYGAME?

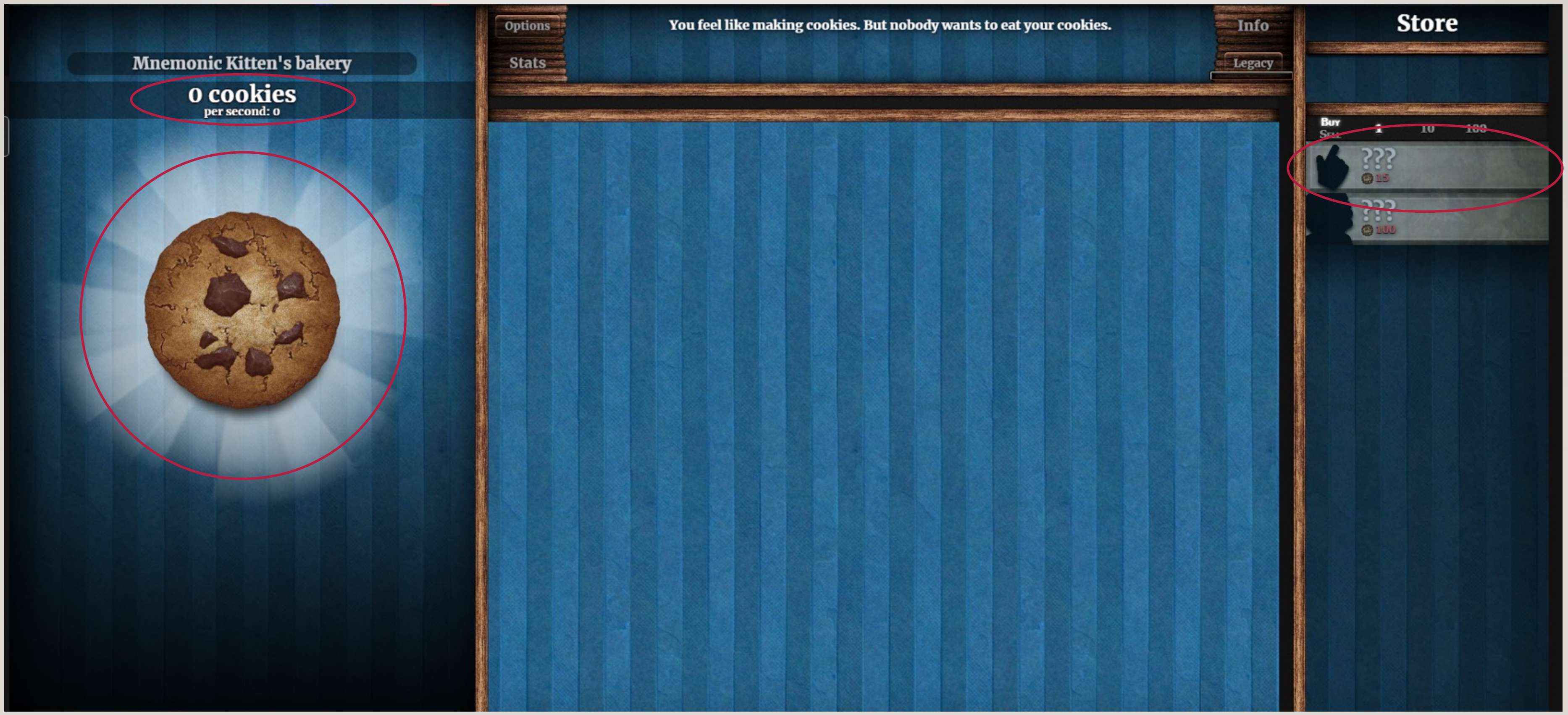
WHAT IS PYGAME?

WHAT IS PYGAME?

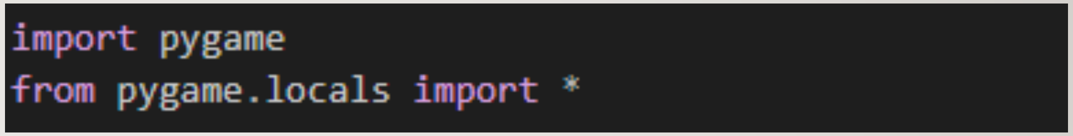


```
style="position:absolute;top:0;left:0
style="position:relative">
style="position:absolute;top:0;left:
Yoff= 20;

var sk=(nav)?document.getElement
if(nav)var st=document.getElement
if(document.layers)document.capture
document.onmousemove=get_mouse

function popup(e){
if(nav){sk=e.target; Str=
else {e=event.srcElement; Str=
if(Str==null||Str.length<3)return;
var content="<TABLE WIDTH=120
CELLPADDING=3 CELLSPACING=
<FONT COLOR=
```

You feel like making cookies. But nobody wants to eat your cookies.

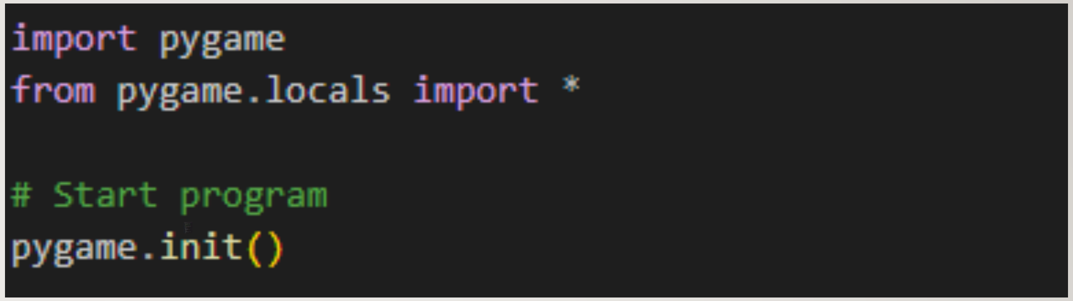

Mnemonic Kitten's bakery
0 cookies
per second: 0
Options
Stats
You feel like making cookies. But nobody wants to eat your cookies.
Info
Legacy
Store
Buy
Sell
1
10
100
???
15
???
100

MAKING A SCREEN



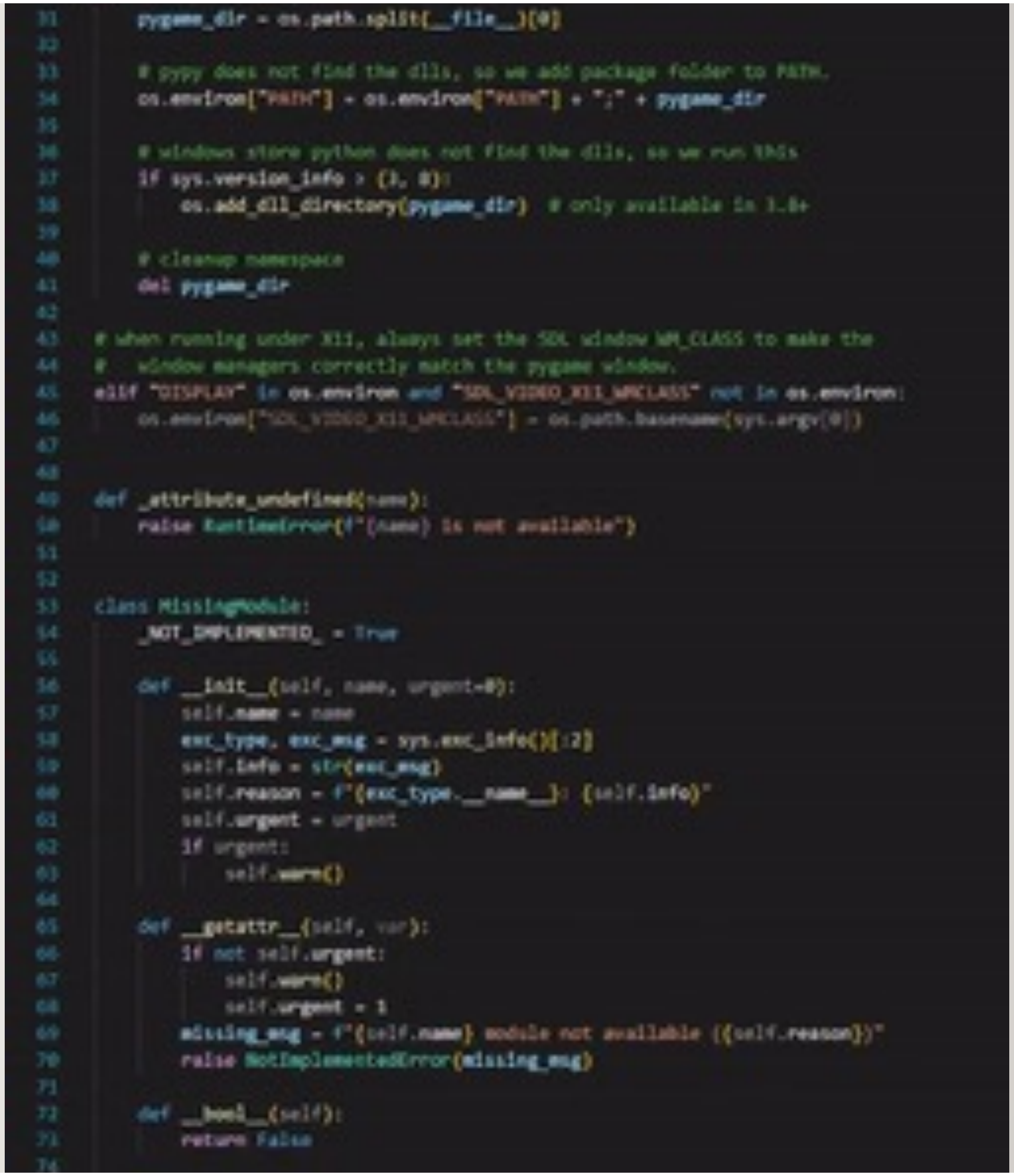
```
import pygame
from pygame.locals import *
```

MAKING A SCREEN



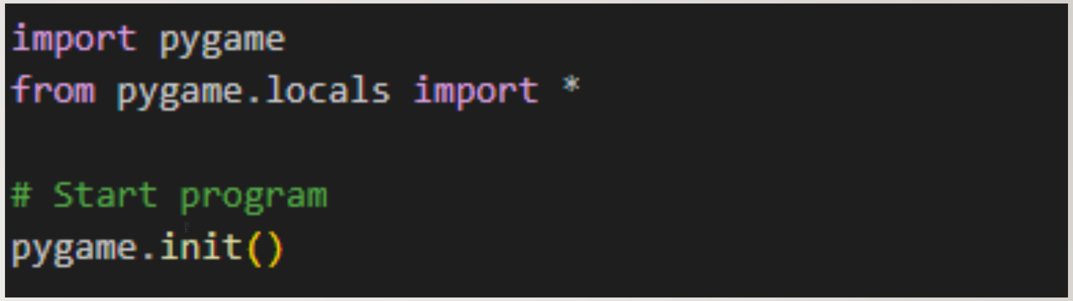
```
import pygame
from pygame.locals import *

# Start program
pygame.init()
```

MAKING A SCREEN



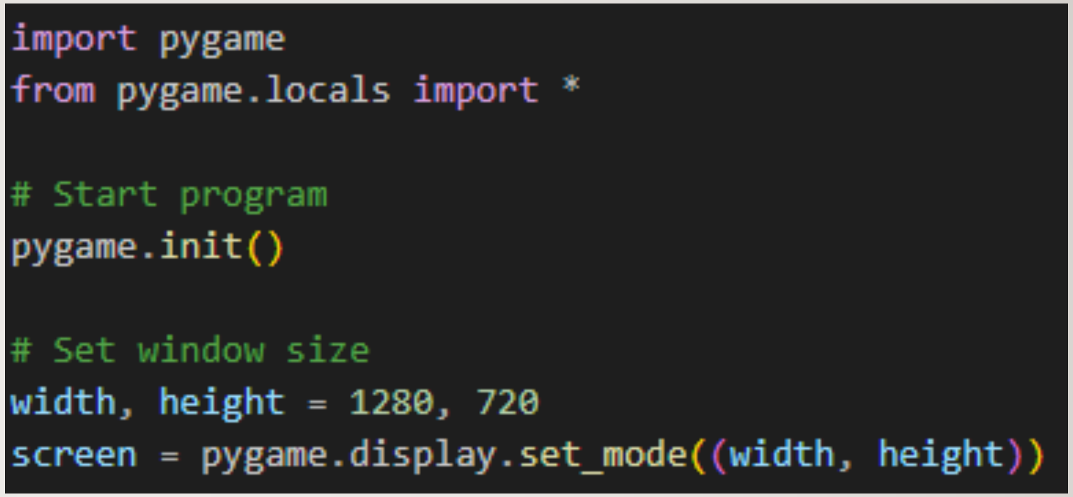
```
        pygame_dir = os.path.split(__file__)[0]

        # pypy does not find the dlls, so we add package folder to PATH.
        os.environ["PATH"] = os.environ["PATH"] + ";" + pygame_dir

        # windows store python does not find the dlls, so we run this
        if sys.version_info > (3, 8):
            os.add_dll_directory(pygame_dir)  # only available in 3.8+

        # cleanup namespace
        del pygame_dir

# when running under X11, always set the SDL window WM_CLASS to make the
#   window managers correctly match the pygame window.
elif "DISPLAY" in os.environ and "SDL_VIDEO_X11_WMCLASS" not in os.environ:
    os.environ["SDL_VIDEO_X11_WMCLASS"] = os.path.basename(sys.argv[0])


def _attribute_undefined(name):
    raise RuntimeError(f"{name} is not available")


class MissingModule:
    _NOT_IMPLEMENTED_ = True

    def __init__(self, name, urgent=0):
        self.name = name
        exc_type, exc_msg = sys.exc_info()[:2]
        self.info = str(exc_msg)
        self.reason = f"{exc_type.__name__}: {self.info}"
        self.urgent = urgent
        if urgent:
            self.warn()

    def __getattr__(self, var):
        if not self.urgent:
            self.warn()
            self.urgent = 1
        missing_msg = f"{self.name} module not available ({self.reason})"
        raise NotImplementedError(missing_msg)

    def __bool__(self):
        return False

```

MAKING A SCREEN



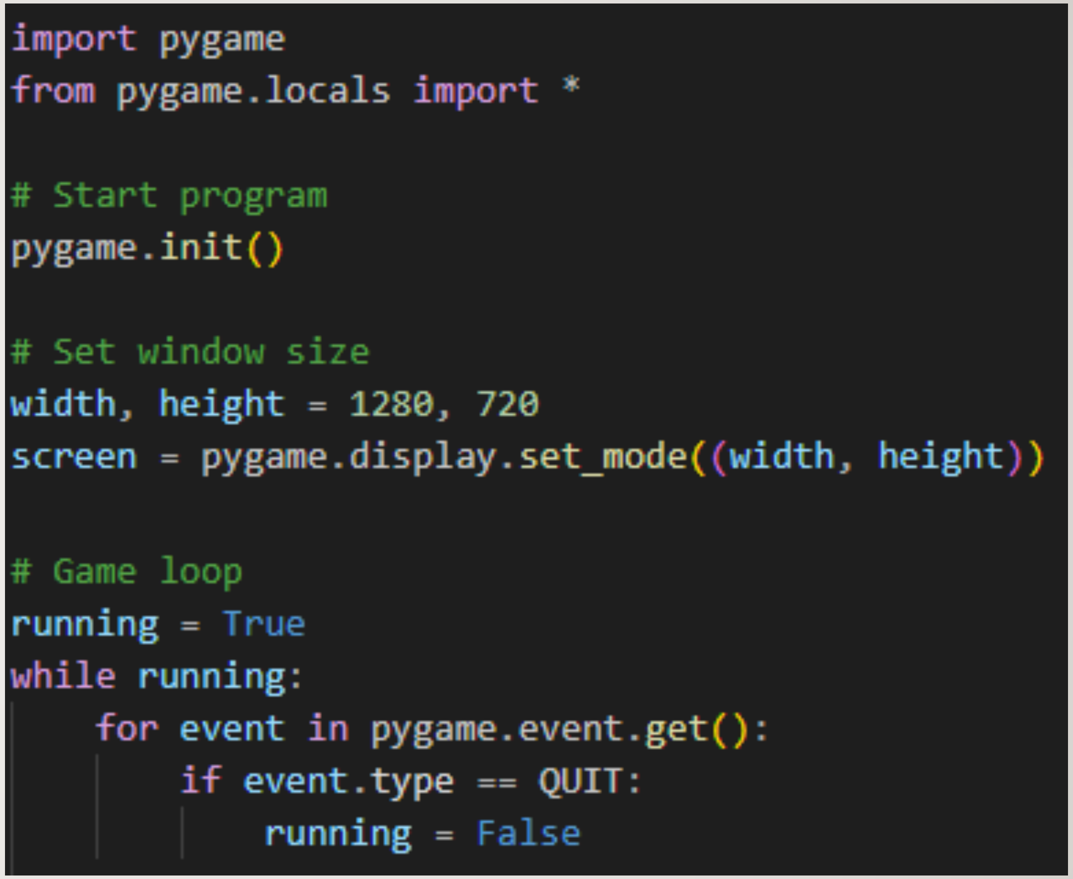
```
import pygame
from pygame.locals import *

# Start program
pygame.init()
```

MAKING A SCREEN



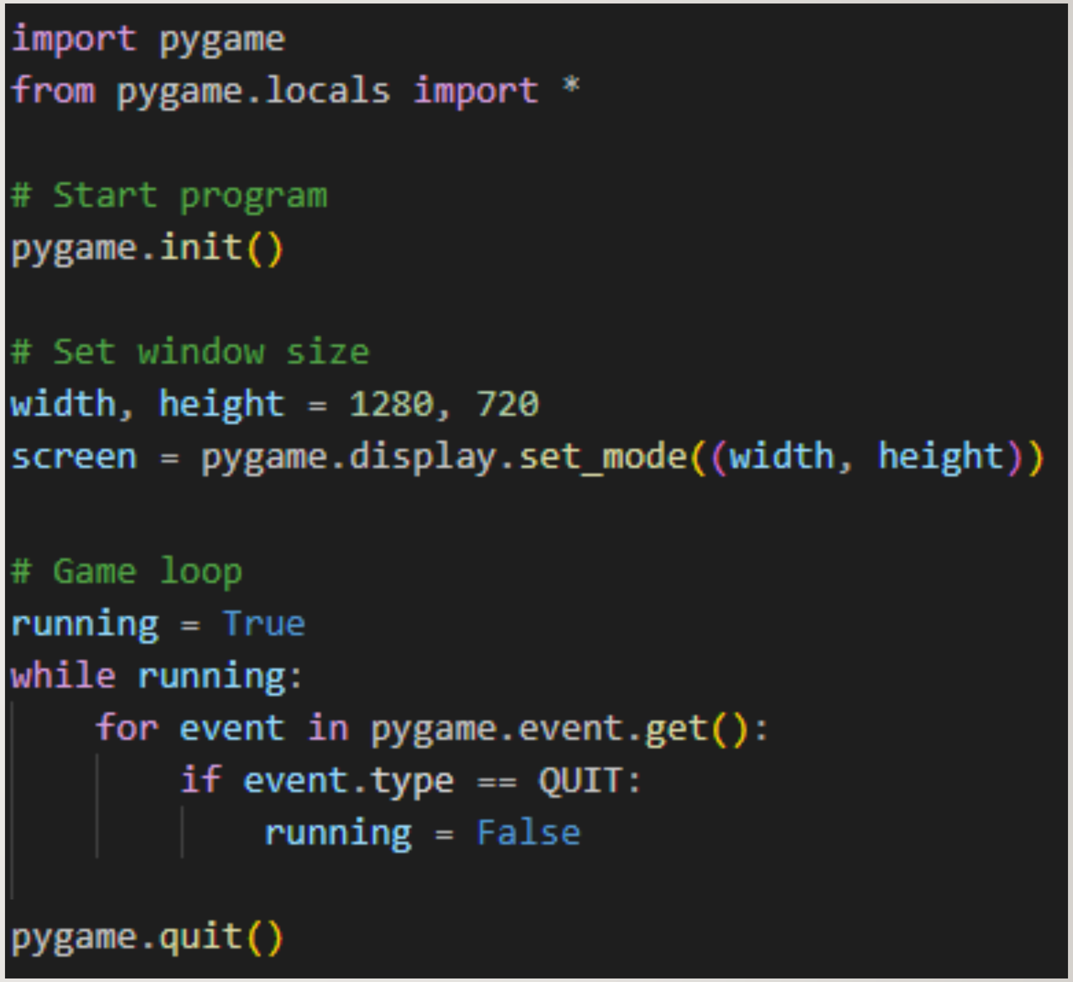
```
import pygame
from pygame.locals import *

# Start program
pygame.init()

# Set window size
width, height = 1280, 720
screen = pygame.display.set_mode((width, height))
```

MAKING A SCREEN



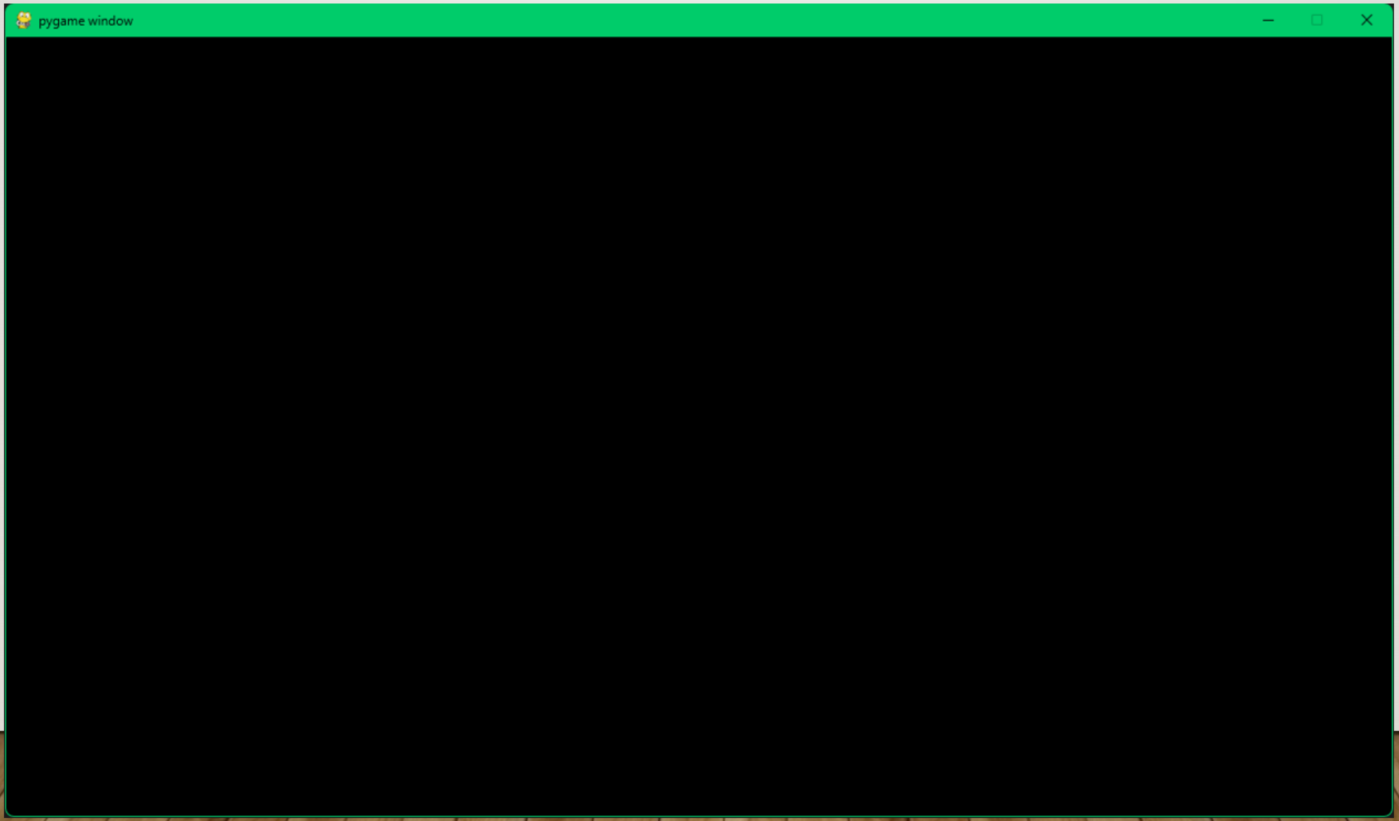
```python
import pygame
from pygame.locals import *

# Start program
pygame.init()

# Set window size
width, height = 1280, 720
screen = pygame.display.set_mode((width, height))

# Game loop
running = True
while running:
    for event in pygame.event.get():
        if event.type == QUIT:
            running = False
```

MAKING A SCREEN



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

# Start program
pygame.init()

# Set window size
width, height = 1280, 720
screen = pygame.display.set_mode((width, height))

# Game loop
running = True
while running:
    for event in pygame.event.get():
        if event.type == QUIT:
            running = False

pygame.quit()
```

MAKING A SCREEN



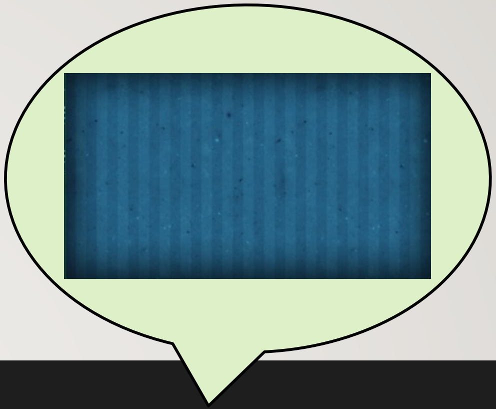
You feel like making cookies. But nobody wants to eat your cookies.

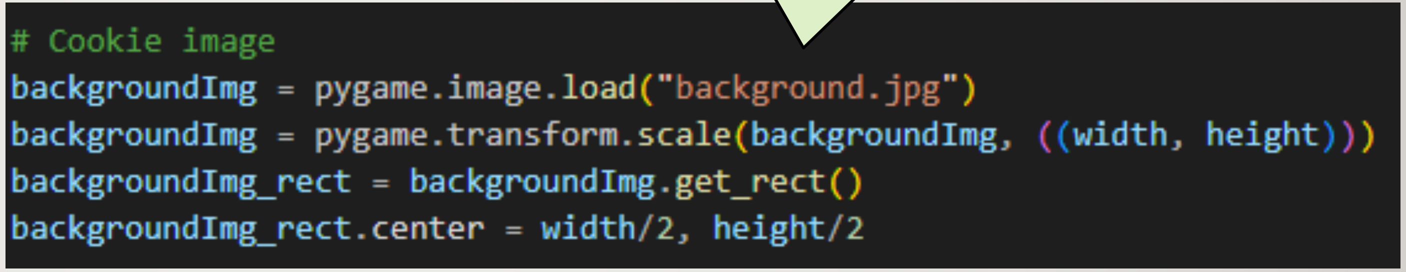
Mnemonic Kitten's bakery

0 cookies

per second: 0

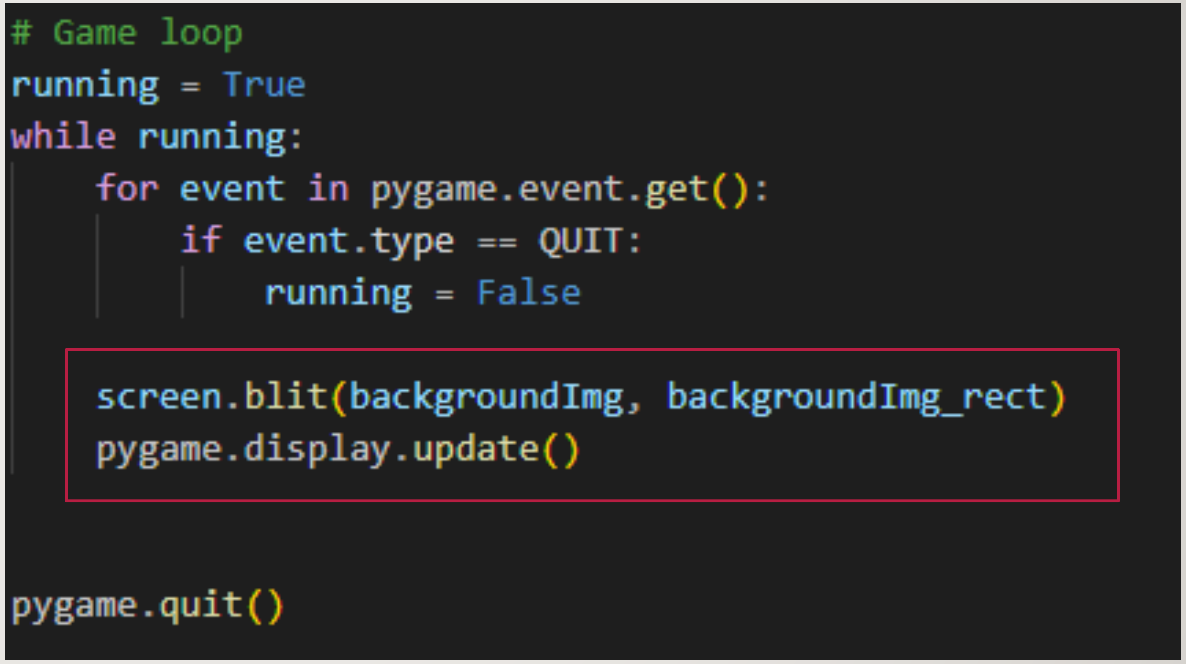
THE BACKGROUND





```
# Cookie image
backgroundImg = pygame.image.load("background.jpg")
backgroundImg = pygame.transform.scale(backgroundImg, ((width, height)))
backgroundImg_rect = backgroundImg.get_rect()
backgroundImg_rect.center = width/2, height/2
```

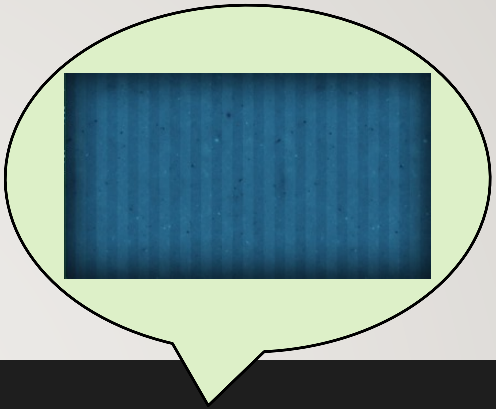
THE BACKGROUND

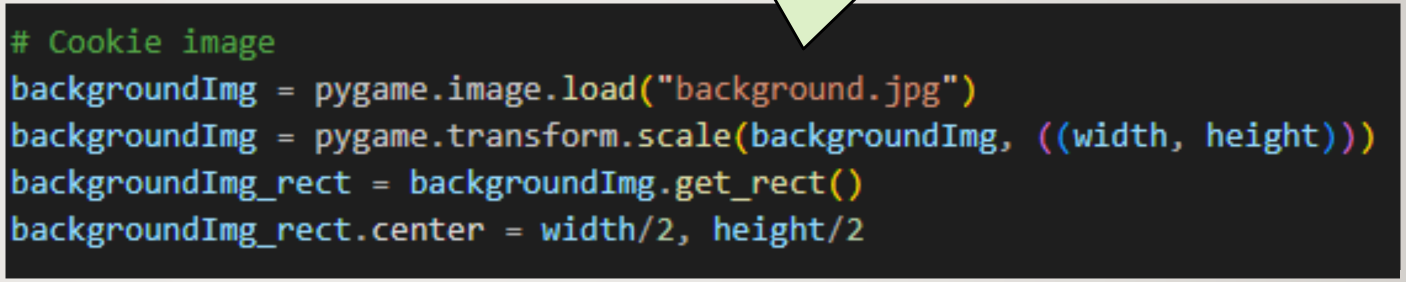


```
# Game loop
running = True
while running:
    for event in pygame.event.get():
        if event.type == QUIT:
            running = False

    screen.blit(backgroundImg, backgroundImg_rect)
    pygame.display.update()


pygame.quit()
```

THE BACKGROUND

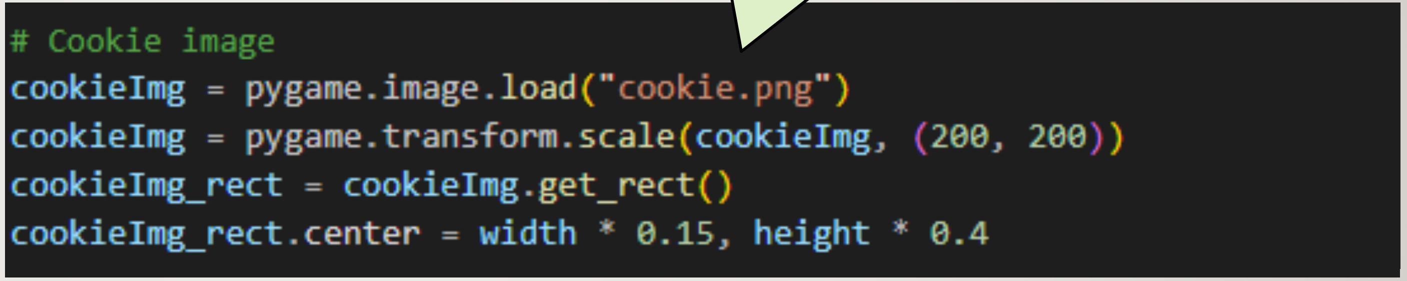
THE COOKIE





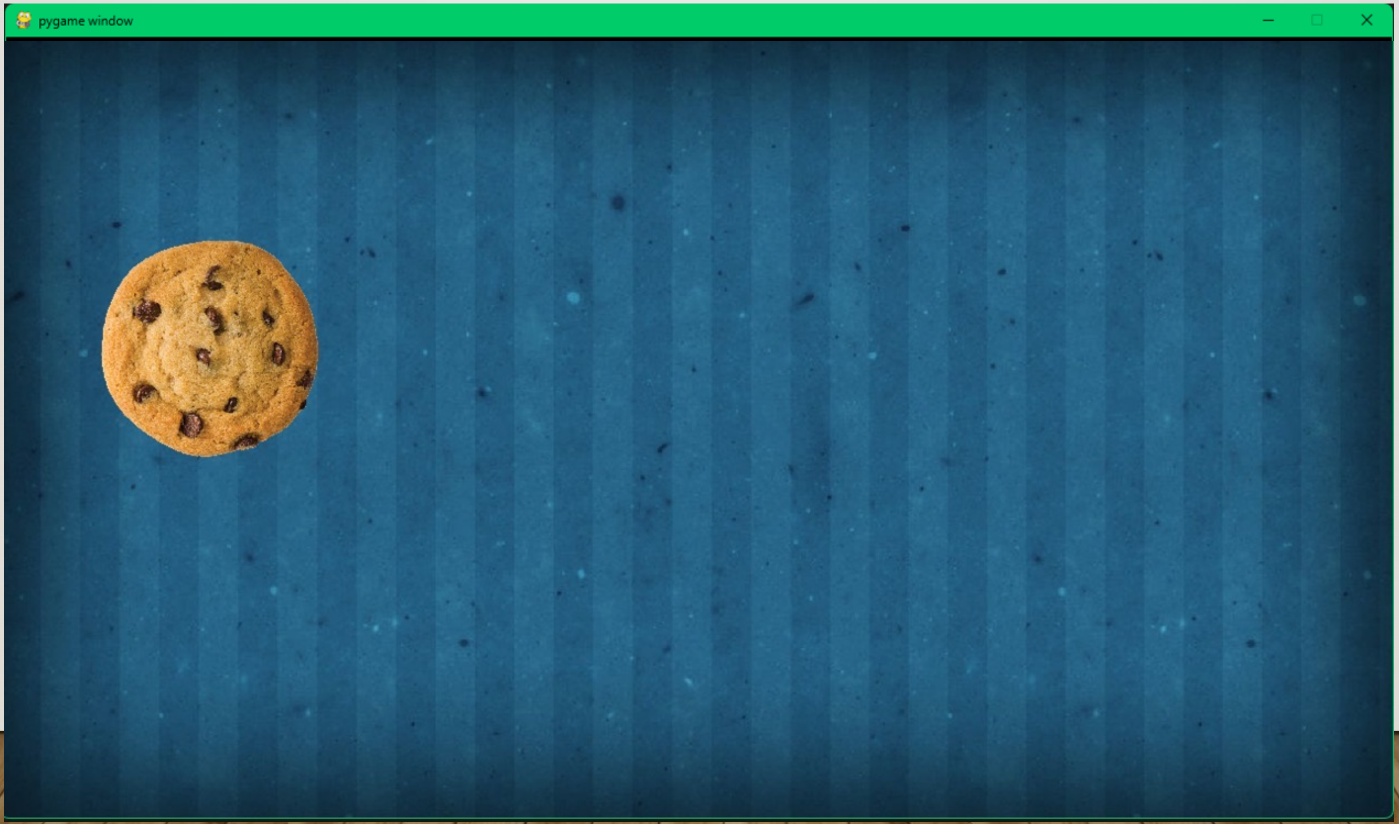
```
# Cookie image
backgroundImg = pygame.image.load("background.jpg")
backgroundImg = pygame.transform.scale(backgroundImg, ((width, height)))
backgroundImg_rect = backgroundImg.get_rect()
backgroundImg_rect.center = width/2, height/2
```

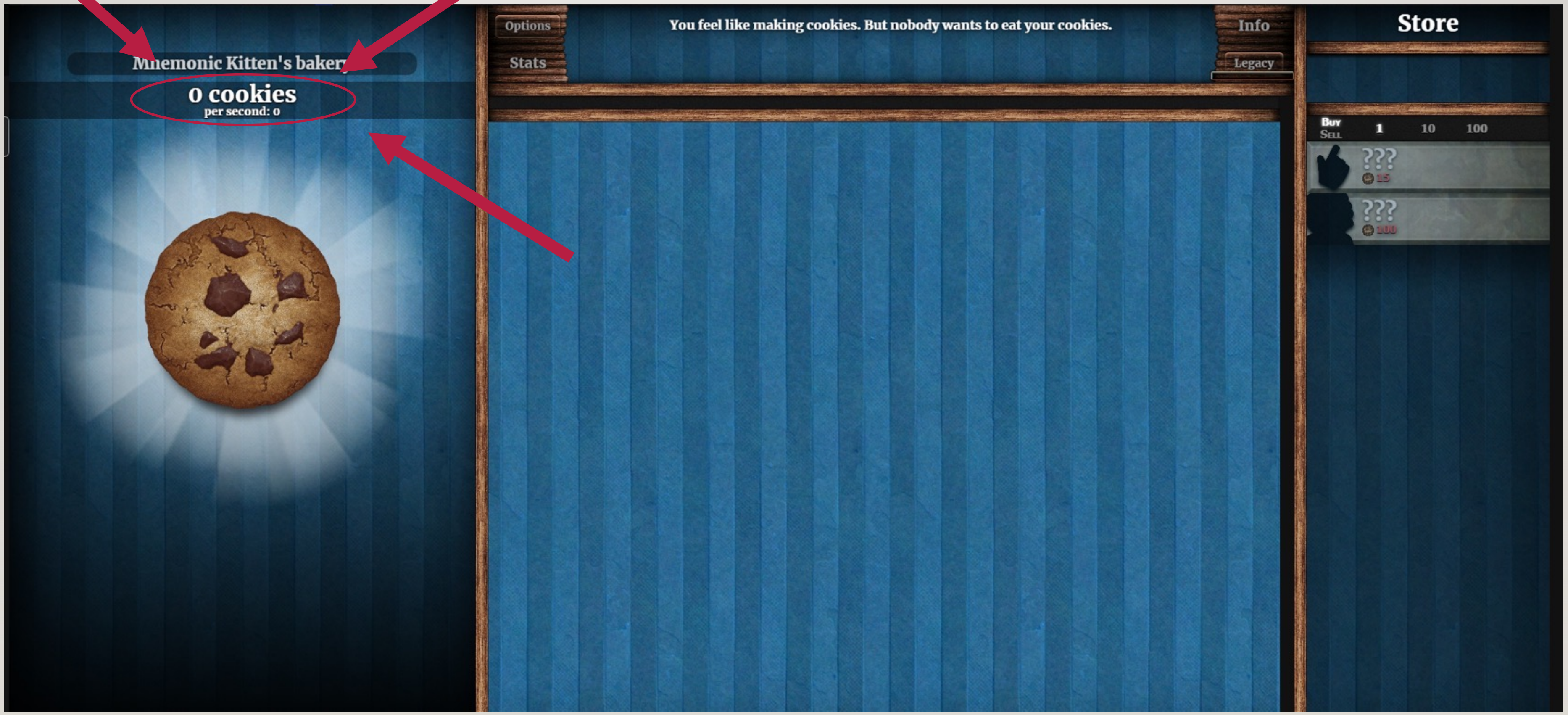
THE COOKIE



```
# Cookie image
cookieImg = pygame.image.load("cookie.png")
cookieImg = pygame.transform.scale(cookieImg, (200, 200))
cookieImg_rect = cookieImg.get_rect()
cookieImg_rect.center = width * 0.15, height * 0.4
```

THE COOKIE



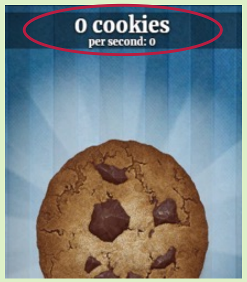


Mnemonic Kitten's bakery

0 cookies
per second: 0

You feel like making cookies. But nobody wants to eat your cookies.

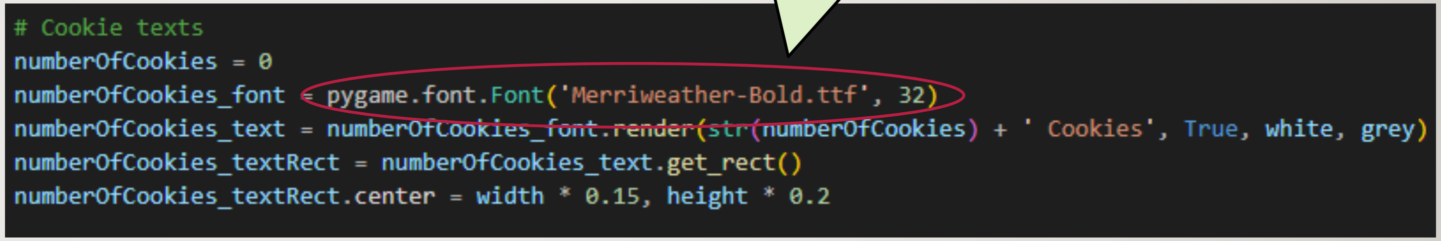
THE NUMBER OF COOKIES



```
# Cookie texts
numberOfCookies = 0
numberOfCookies_font = pygame.font.Font('Merriweather-Bold.ttf', 32)
numberOfCookies_text = numberOfCookies_font.render(str(numberOfCookies) + ' Cookies', True, white, grey)
numberOfCookies_textRect = numberOfCookies_text.get_rect()
numberOfCookies_textRect.center = width * 0.15, height * 0.2
```

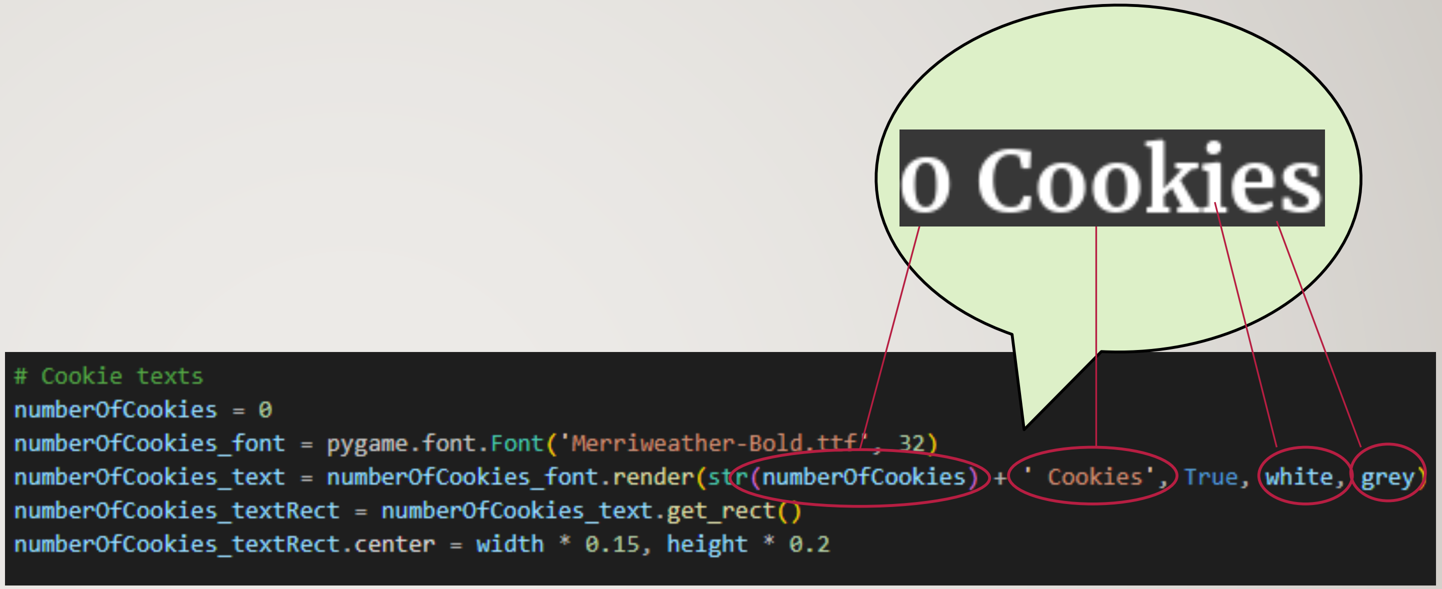
THE NUMBER OF COOKIES

Aa Bb Cc Dd Ee Ff Gg
Hh Ii Jj Kk Ll Mm Nn
Oo Pp Qq Rr Ss Tt Uu
Vv Ww Xx Yy Zz



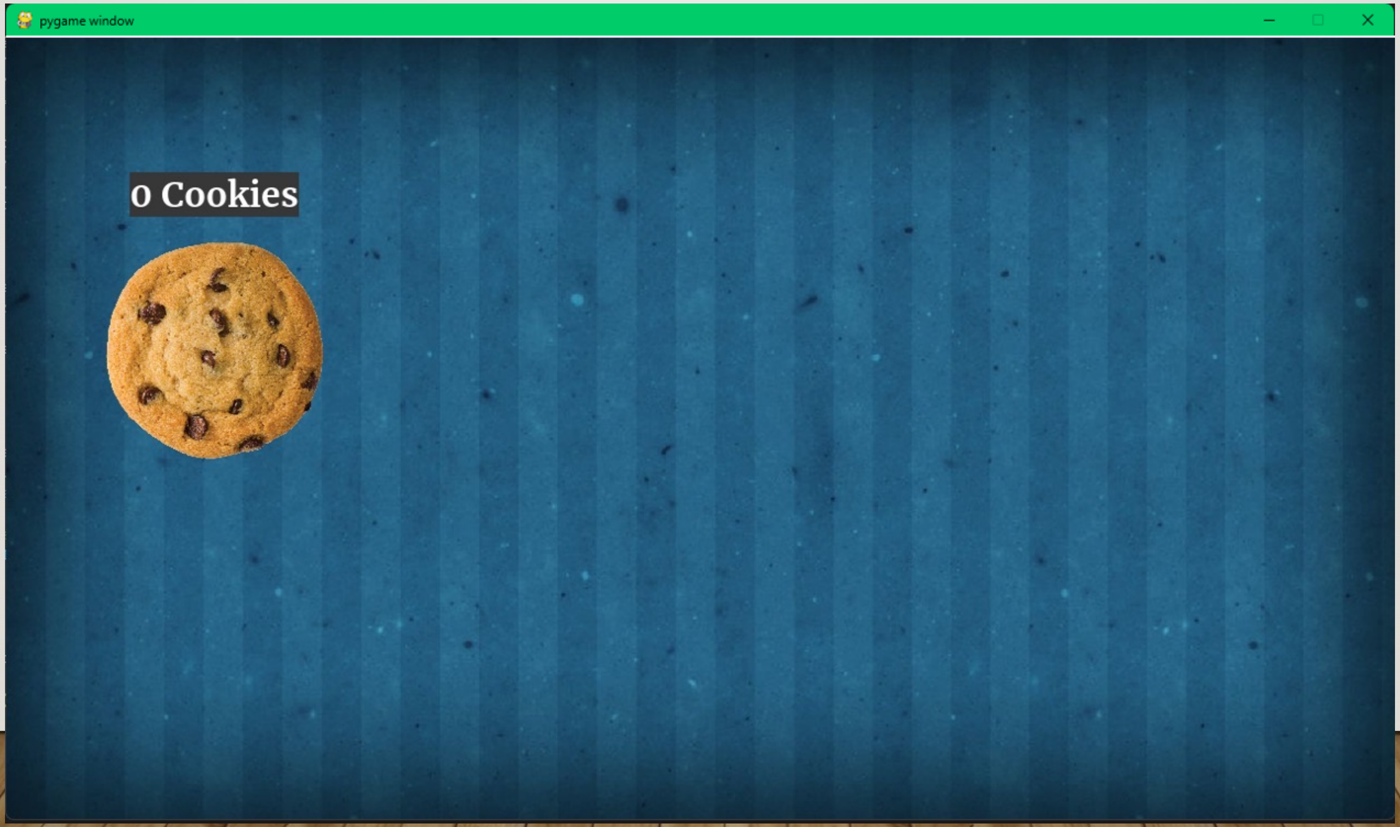
```
# Cookie texts
numberOfCookies = 0
numberOfCookies_font = pygame.font.Font('Merriweather-Bold.ttf', 32)
numberOfCookies_text = numberOfCookies_font.render(str(numberOfCookies) + ' Cookies', True, white, grey)
numberOfCookies_textRect = numberOfCookies_text.get_rect()
numberOfCookies_textRect.center = width * 0.15, height * 0.2
```

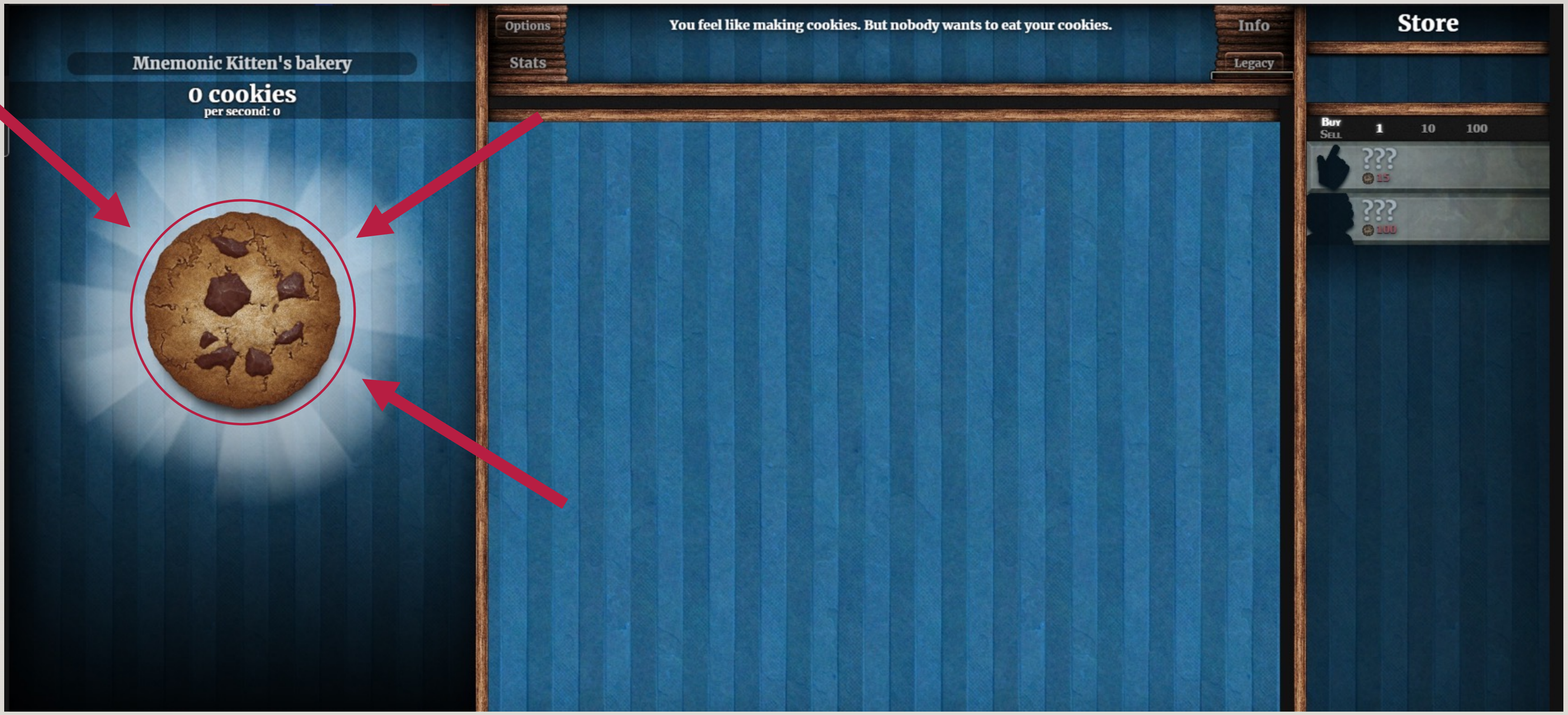
THE NUMBER OF COOKIES



```
# Cookie texts
numberOfCookies = 0
numberOfCookies_font = pygame.font.Font('Merriweather-Bold.ttf', 32)
numberOfCookies_text = numberOfCookies_font.render(str(numberOfCookies) + ' Cookies', True, white, grey)
numberOfCookies_textRect = numberOfCookies_text.get_rect()
numberOfCookies_textRect.center = width * 0.15, height * 0.2
```

THE NUMBERS

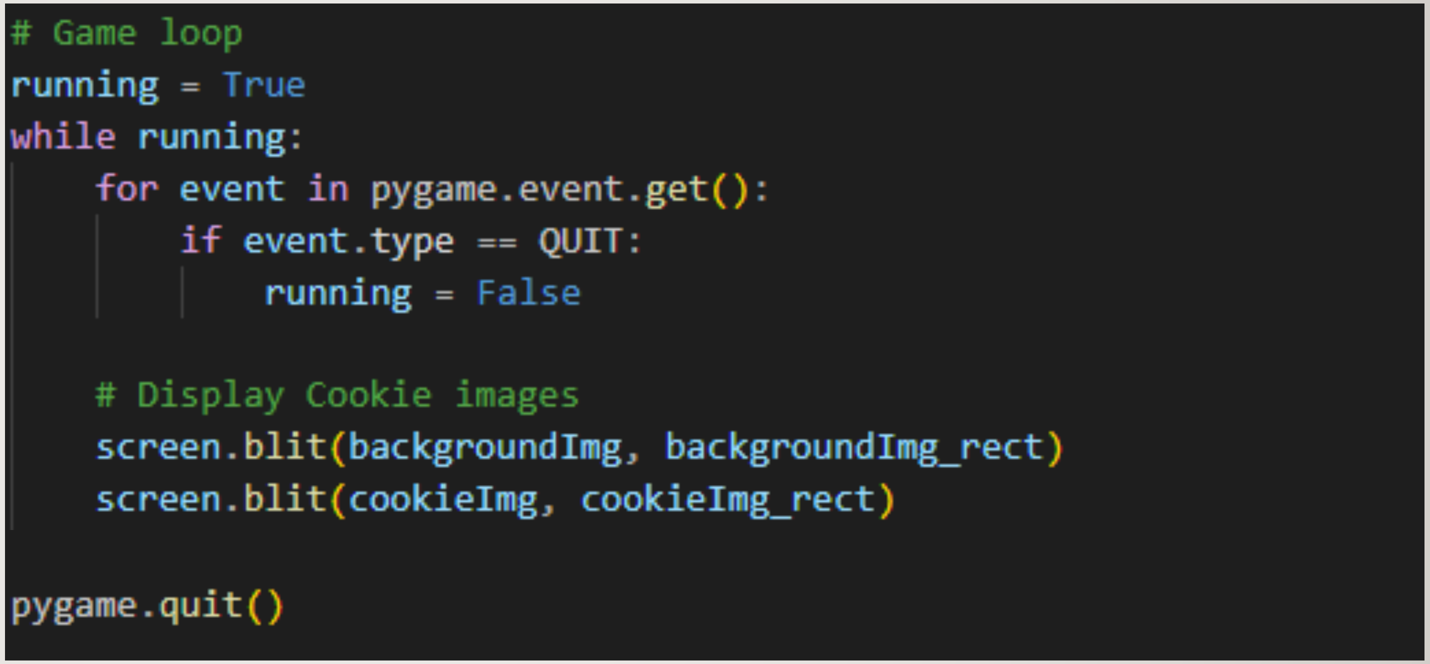




You feel like making cookies. But nobody wants to eat your cookies.

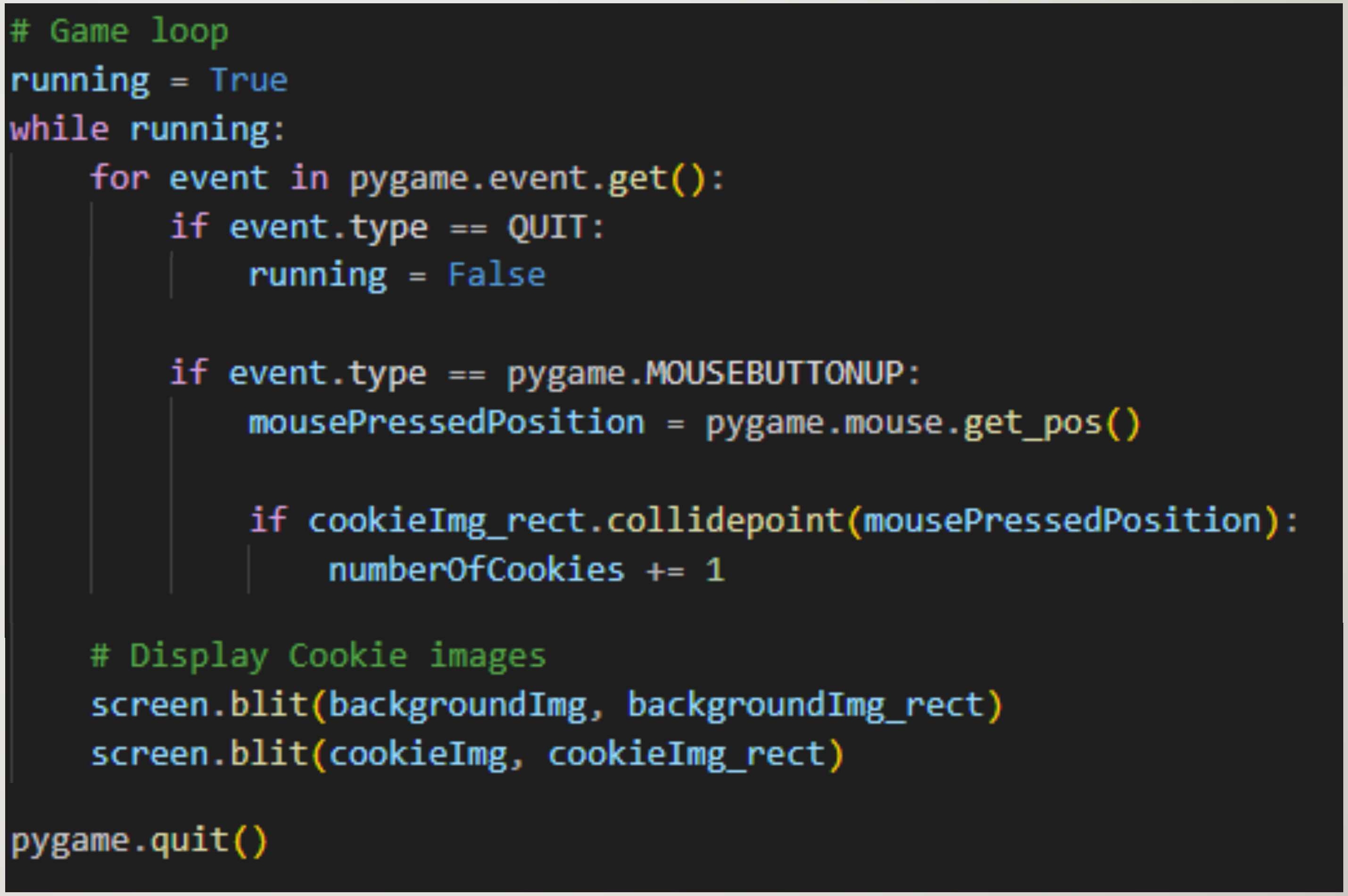
THE CLICKING

CURRENT GAME LOOP



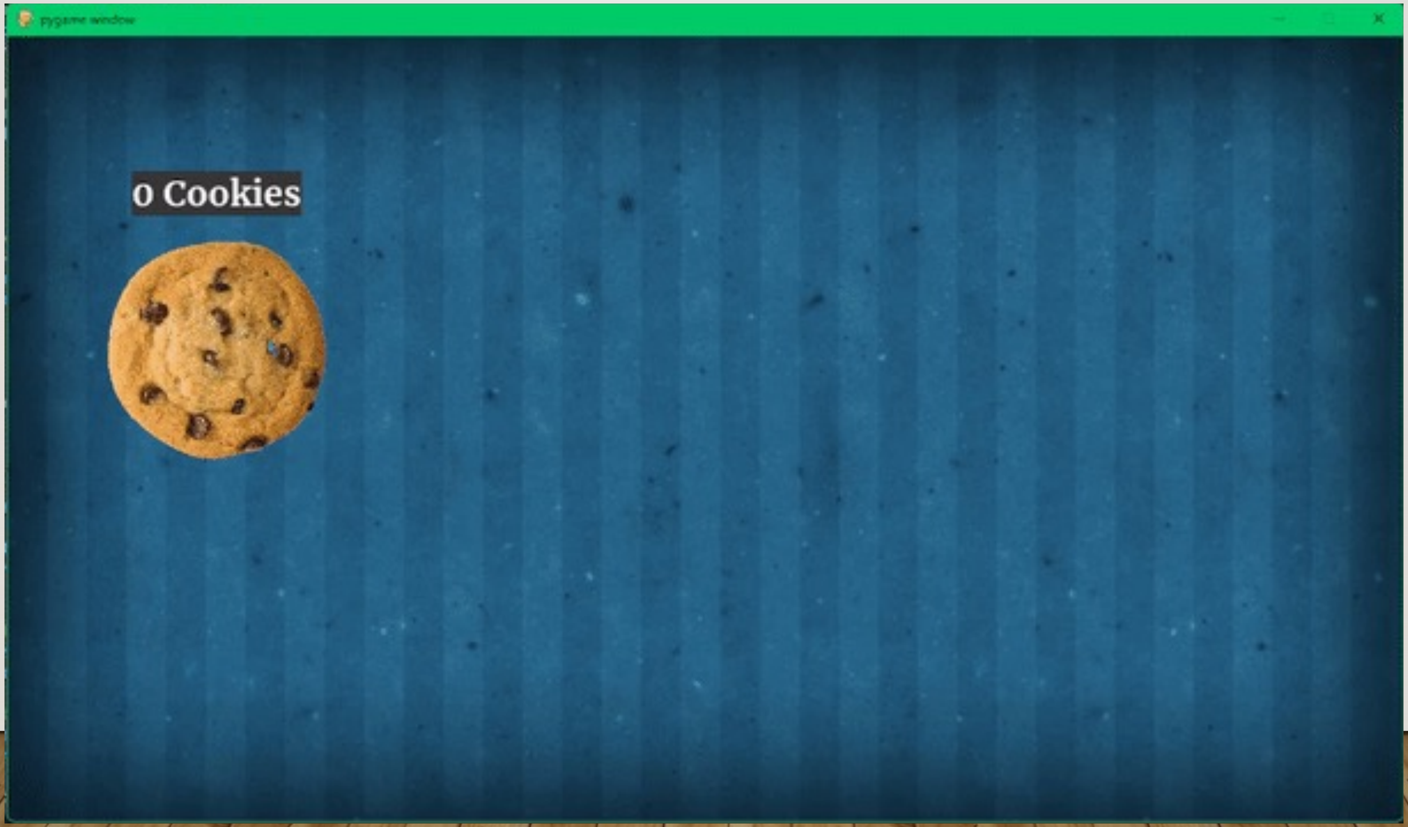
```
# Game loop
running = True
while running:
    for event in pygame.event.get():
        if event.type == QUIT:
            running = False

    # Display Cookie images
    screen.blit(backgroundImg, backgroundImg_rect)
    screen.blit(cookieImg, cookieImg_rect)

pygame.quit()
```

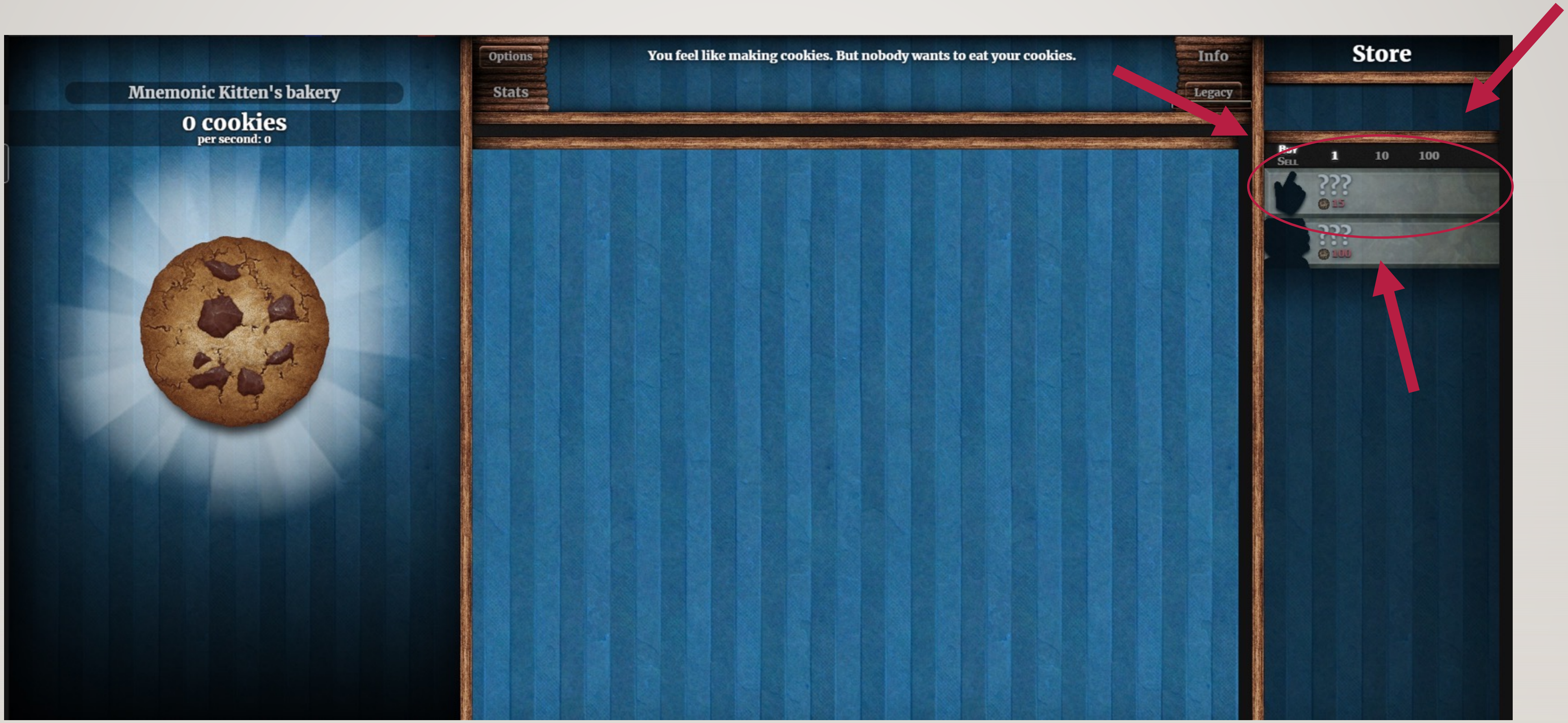
THE CLICKING



```
# Game loop
running = True
while running:
    for event in pygame.event.get():
        if event.type == QUIT:
            running = False

        if event.type == pygame.MOUSEBUTTONUP:
            mousePressedPosition = pygame.mouse.get_pos()

            if cookieImg_rect.collidepoint(mousePressedPosition):
                numberOfCookies += 1

    # Display Cookie images
    screen.blit(backgroundImg, backgroundImg_rect)
    screen.blit(cookieImg, cookieImg_rect)

pygame.quit()
```

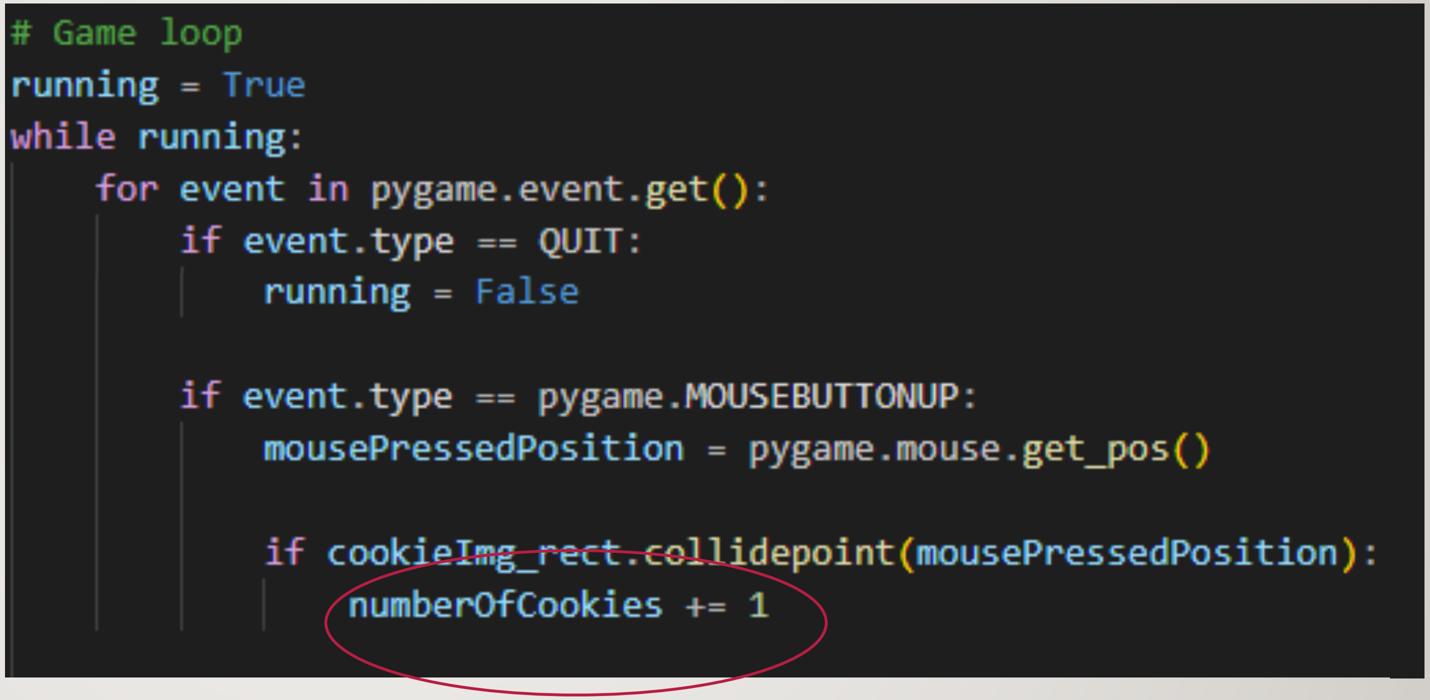
THE CLICKING




Options
Stats
You feel like making cookies. But nobody wants to eat your cookies.
Info
Legacy
Store
Mnemonic Kitten's bakery
0 cookies
per second: 0
Buy
Sell
1
10
100
???
15
???
100

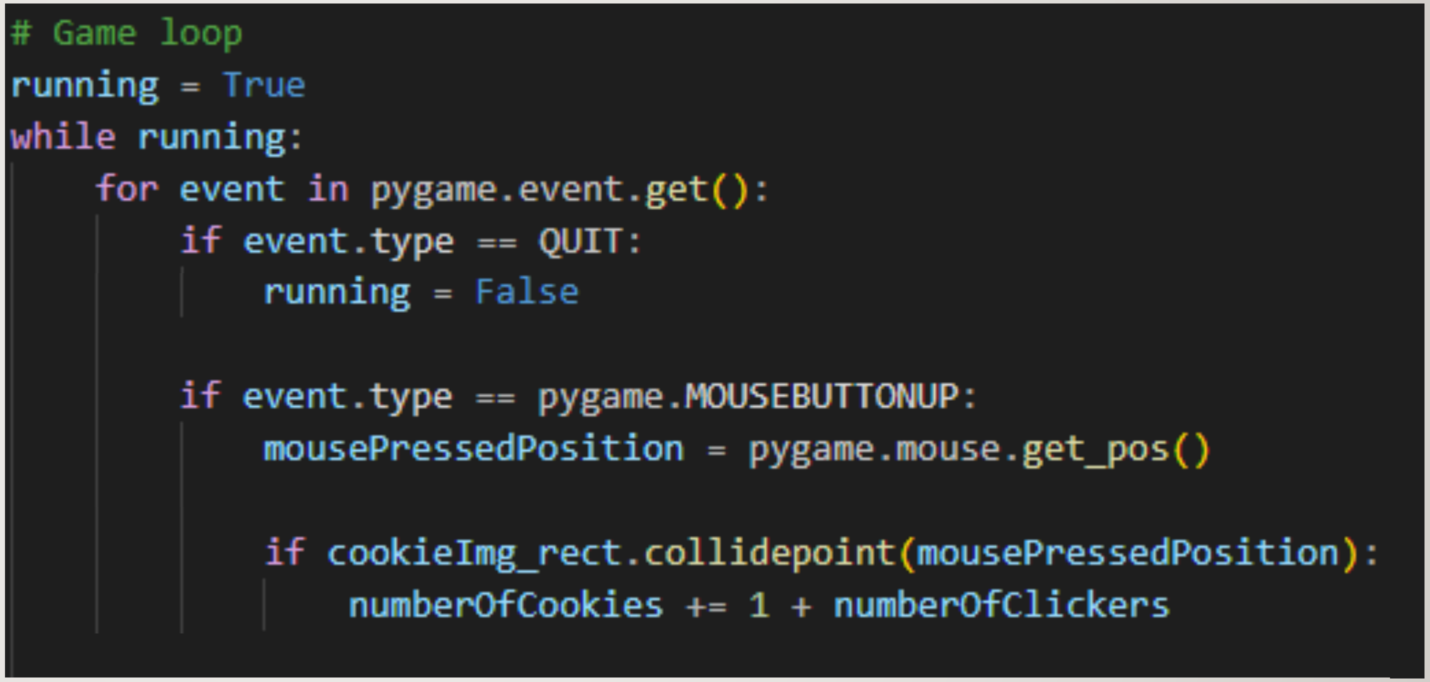
THE UPGRADES

CURRENT MOUSE EVENT



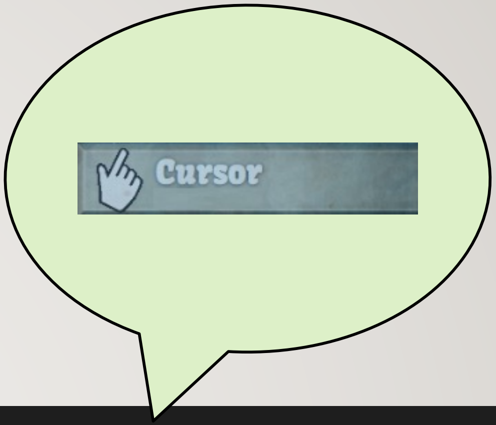
```
# Game loop
running = True
while running:
    for event in pygame.event.get():
        if event.type == QUIT:
            running = False

        if event.type == pygame.MOUSEBUTTONUP:
            mousePressedPosition = pygame.mouse.get_pos()

            if cookieImg_rect.collidepoint(mousePressedPosition):
                numberOfCookies += 1
```

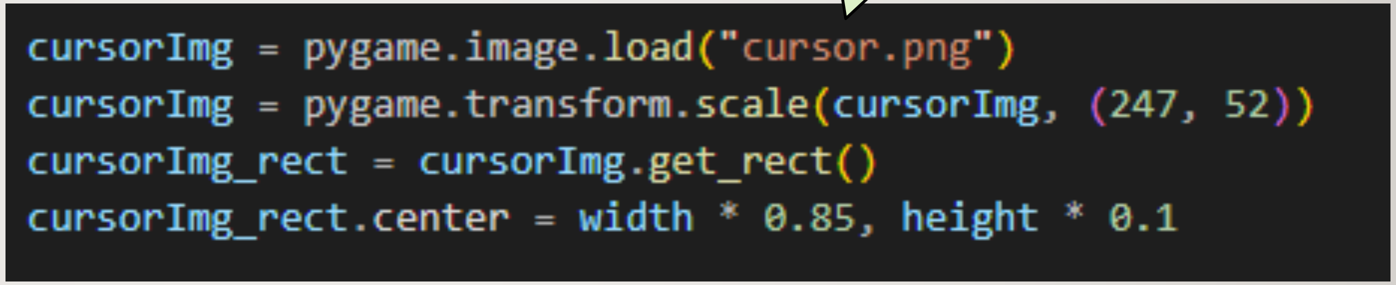
THE UPGRADES



```
# Game loop
running = True
while running:
    for event in pygame.event.get():
        if event.type == QUIT:
            running = False

        if event.type == pygame.MOUSEBUTTONUP:
            mousePressedPosition = pygame.mouse.get_pos()

            if cookieImg_rect.collidepoint(mousePressedPosition):
                numberOfCookies += 1 + numberOfClickers
```

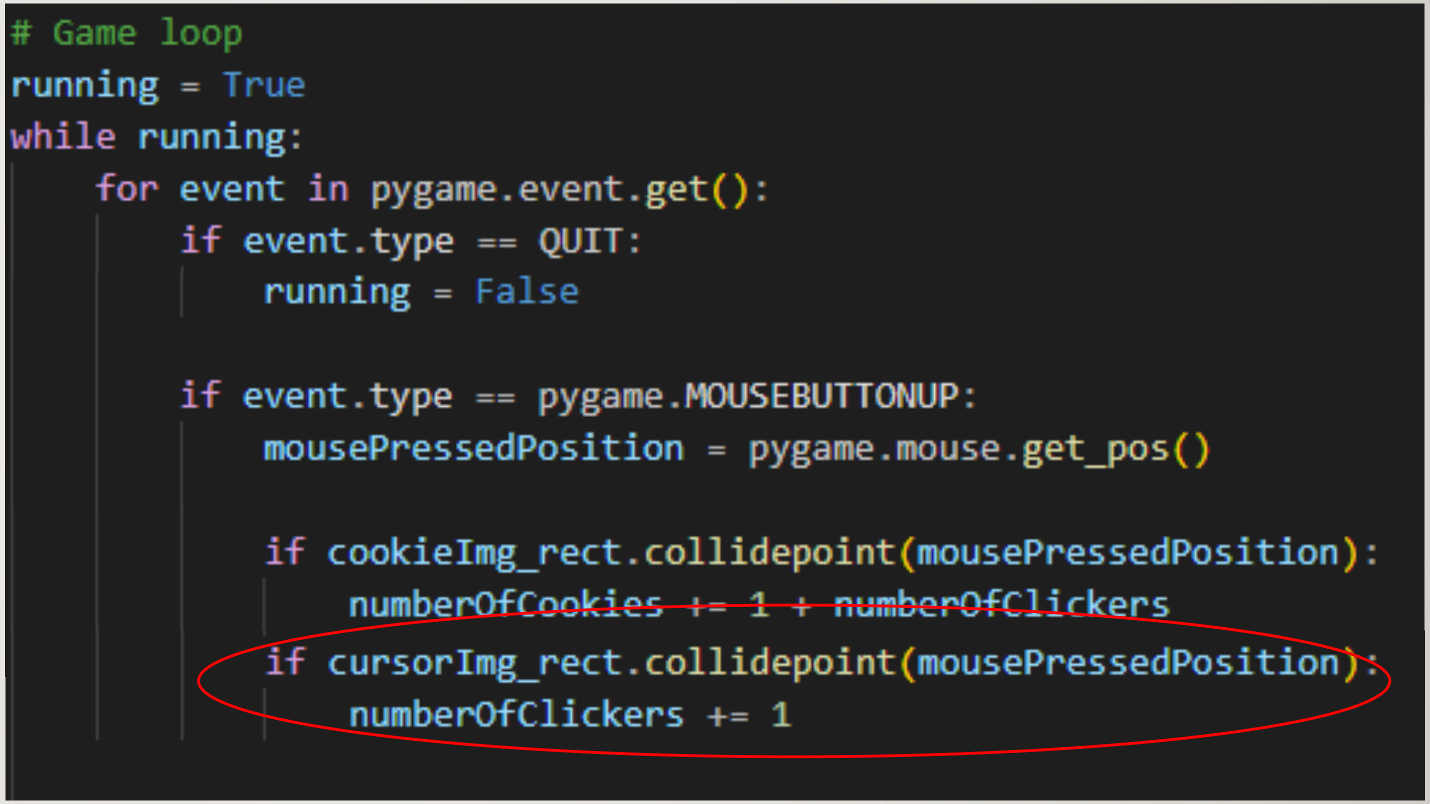
THE UPGRADES





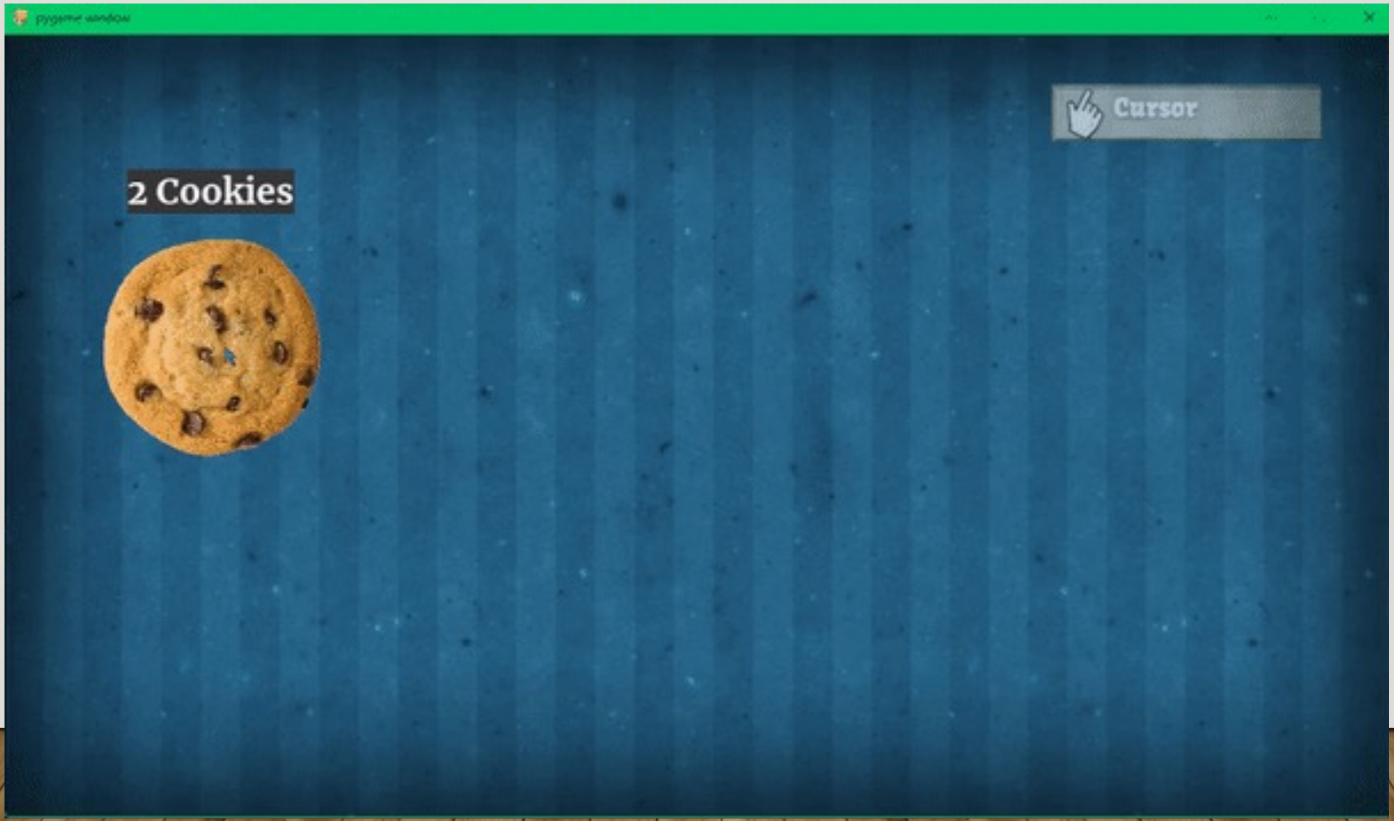
```
cursorImg = pygame.image.load("cursor.png")
cursorImg = pygame.transform.scale(cursorImg, (247, 52))
cursorImg_rect = cursorImg.get_rect()
cursorImg_rect.center = width * 0.85, height * 0.1
```

THE UPGRADES

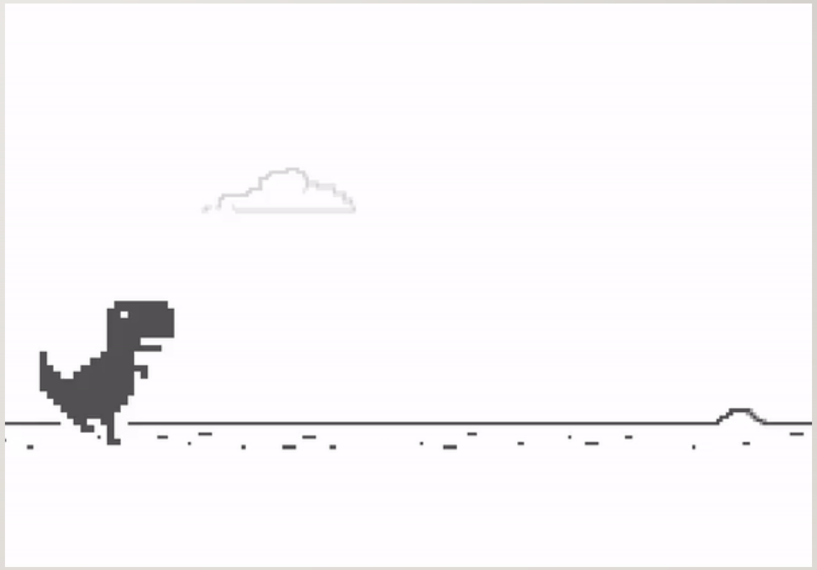


```
# Game loop
running = True
while running:
    for event in pygame.event.get():
        if event.type == QUIT:
            running = False

        if event.type == pygame.MOUSEBUTTONUP:
            mousePressedPosition = pygame.mouse.get_pos()

            if cookieImg_rect.collidepoint(mousePressedPosition):
                numberOfCookies += 1 + numberOfClickers
            if cursorImg_rect.collidepoint(mousePressedPosition):
                numberOfClickers += 1
```

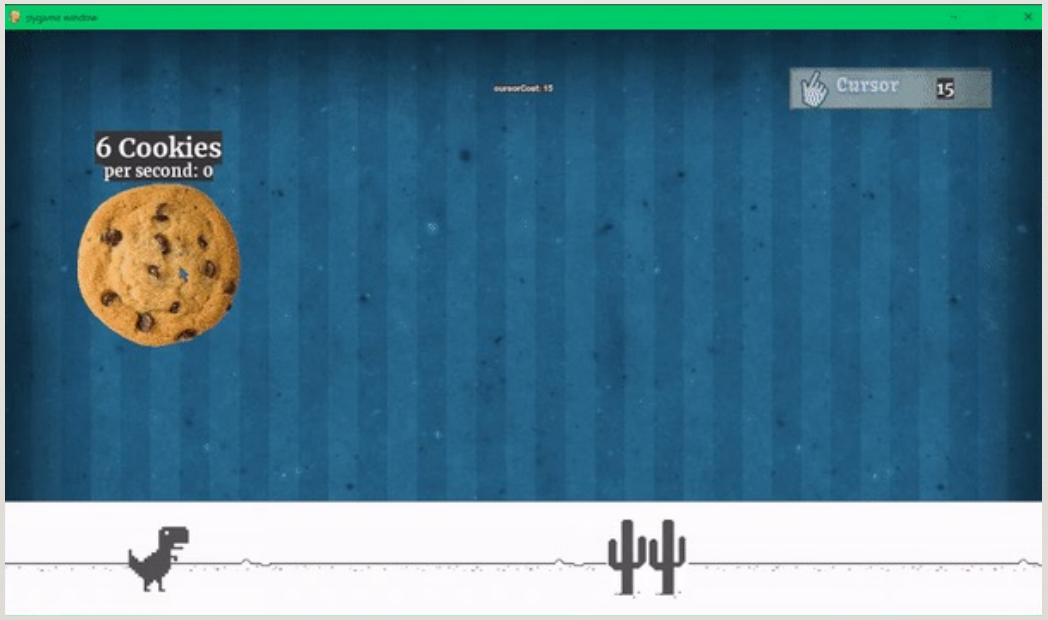
THE UPGRADES



WHAT ELSE CAN I DO?



WHAT ELSE CAN I DO?



THANK YOU

PROMPTING PYGAME

JOHN TAN CHONG MIN

GET CHATGPT TO GENERATE THE GAME CODE FOR YOU!

- Reference:
  - https://github.com/tanchongmin/TensorFlow-Implementations/tree/main/Fireball_Dodger

- Prompt:
  - I would like to create a Python game in PyGame. The game's objective is to dodge fireballs coming from all four directions. You control a player that can move up, down, left, right. You have three lives and your aim is to collect cupcakes spawned at a random position. The score is based on the number of cupcakes you collect.

PYGAME INSTRUCTIONS

- Things to note:
  - pygame (just do pip install pygame)
  - Place assets like images and audio in the same folder as the PyGame code
  - Run code and enjoy the game

ERROR HANDLING

- In order to correct any errors in the code, do this
- Prompt:
  - <Game Code so far>
  - <Error Message from Python>
  - Generate me the corrected game code in a Python block.