Day 4 Ninja Guide

You will be adding the code in green and deleting any code in red. Be sure to use the code in grey to help you find the correct spot to insert new code or delete old code.

INVISIBLE GRID

--

```
import pygame
from pygame import *
from random import randint
pygame.init()

clock = time.Clock()

WINDOW_WIDTH = 1100
WINDOW_HEIGHT = 600
WINDOW_RES = (WINDOW_WIDTH, WINDOW_HEIGHT)

WIDTH = 100
HEIGHT = 100

WHITE = (255, 255, 255)

SPAWN_RATE = 360
FRAME_RATE = 60

GAME_WINDOW = display.set_mode(WINDOW_RES)
display.set_caption('Attack of the Vampire Pizzas!')

background_img = image.load('restaurant.jpg')
background_surf = Surface.convert_alpha(background_img)
BACKGROUND = transform.scale(background_surf, WINDOW_RES)

pizza_img = image.load('vampire.png')
pizza_surf = Surface.convert_alpha(pizza_img)
VAMPIRE_PIZZA = transform.scale(pizza_surf, (WIDTH, HEIGHT))

class VampireSprite(sprite.Sprite):
    def __init__(self):
        super().__init__()
        self.speed = 2
        self.lane = randint(0, 4)
        all_vampires.add(self)
        self.image = VAMPIRE_PIZZA.copy()
        y = 50 + self.lane * 100
        self.rect = self.image.get_rect(center = (1100, y))
    def update(self, game_window):
        game_window.blit(BACKGROUND, (self.rect.x, self.rect.y), self.rect)
        self.rect.x -= self.speed
        game_window.blit(self.image, (self.rect.x, self.rect.y))

class BackgroundTile(sprite.Sprite):
    def __init__(self, rect):
        super().__init__()
        self.effect = False
        self.rect = rect

all_vampires = sprite.Group()

tile_grid = []

tile_color = WHITE
for row in range(6):
    row_of_tiles = []
    tile_grid.append(row_of_tiles)
    for column in range(11):
        tile_rect = Rect(WIDTH * column, HEIGHT * row, WIDTH, HEIGHT)
        new_tile = BackgroundTile(tile_rect)
        row_of_tiles.append(new_tile)
        draw.rect(BACKGROUND, tile_color, (WIDTH*column, HEIGHT*row, WIDTH, HEIGHT), 1)

GAME_WINDOW.blit(BACKGROUND, (0, 0))

game_running = True
while game_running:

    for event in pygame.event.get():
        if event.type == QUIT:
            game_running = False

    if randint(1, SPAWN_RATE) == 1:
        VampireSprite()

    for vampire in all_vampires:vampire.update(GAME_WINDOW)

    display.update()

    clock.tick(FRAME_RATE)

pygame.quit()
```

GARLIC TRAP

```
import pygame
from pygame import *
from random import randint
pygame.init()

clock = time.Clock()

WINDOW_WIDTH = 1100
WINDOW_HEIGHT = 600
WINDOW_RES = (WINDOW_WIDTH, WINDOW_HEIGHT)

WIDTH = 100
HEIGHT = 100

WHITE = (255, 255, 255)

SPAWN_RATE = 360
FRAME_RATE = 60

REG_SPEED = 2
SLOW_SPEED = 1

GAME_WINDOW = display.set_mode(WINDOW_RES)
display.set_caption('Attack of the Vampire Pizzas!')

background_img = image.load('restaurant.jpg')
background_surf = Surface.convert_alpha(background_img)
BACKGROUND = transform.scale(background_surf, WINDOW_RES)

pizza_img = image.load('vampire.png')
pizza_surf = Surface.convert_alpha(pizza_img)
VAMPIRE_PIZZA = transform.scale(pizza_surf, (WIDTH, HEIGHT))

class VampireSprite(sprite.Sprite):
    def __init__(self):
        super().__init__()
        self.speed = 2
        self.speed = REG_SPEED
        self.lane = randint(0, 4)
        all_vampires.add(self)
        self.image = VAMPIRE_PIZZA.copy()
        y = 50 + self.lane * 100
        self.rect = self.image.get_rect(center = (1100, y))
    def update(self, game_window):
        game_window.blit(BACKGROUND, (self.rect.x, self.rect.y), self.rect)
        self.rect.x -= self.speed
        game_window.blit(self.image, (self.rect.x, self.rect.y))

class BackgroundTile(sprite.Sprite):
    def __init__(self, rect):
        super().__init__()
        self.effect = False
        self.rect = rect

all_vampires = sprite.Group()

tile_grid = []

tile_color = WHITE
for row in range(6):
    row_of_tiles = []
    tile_grid.append(row_of_tiles)
    for column in range(11):
        tile_rect = Rect(WIDTH * column, HEIGHT * row, WIDTH, HEIGHT)
        new_tile = BackgroundTile(tile_rect)
        row_of_tiles.append(new_tile)
        draw.rect(BACKGROUND, tile_color, (WIDTH*column, HEIGHT*row, WIDTH, HEIGHT), 1)

GAME_WINDOW.blit(BACKGROUND, (0, 0))

game_running = True
while game_running:

    for event in pygame.event.get():
        if event.type == QUIT:
            game_running = False

        elif event.type == MOUSEBUTTONDOWN:
            coordinates = mouse.get_pos()
            x = coordinates[0]
            y = coordinates[1]
            tile_y = y//100
            tile_x = x//100
            tile_grid[tile_y][tile_x].effect = True
            print(tile_x, tile_y)

    if randint(1, SPAWN_RATE) == 1:
        VampireSprite()

    for vampire in all_vampires:vampire.update(GAME_WINDOW)
    for vampire in all_vampires:
        tile_row = tile_grid[vampire.rect.y//100]
        vamp_left_side = vampire.rect.x//100
        vamp_right_side = (vampire.rect.x+ vampire.rect.width)//100
        if 0 <= vamp_left_side <= 10:
            left_tile = tile_row[vamp_left_side]
        else:
            left_tile = None
        if 0 <= vamp_right_side <= 10:
            right_tile = tile_row[vamp_right_side]
        else:
            right_tile = None
        if bool(left_tile) and left_tile.effect:
            if right_tile != left_tile:
                vampire.speed = SLOW_SPEED
        if vampire.rect.x <= 0:
            vampire.kill()

    for vampire in all_vampires:
        vampire.update(GAME_WINDOW)

    display.update()

    clock.tick(FRAME_RATE)

pygame.quit()
```

PIZZA BUCKS

```
import pygame
from pygame import *
from random import randint
pygame.init()

clock = time.Clock()

WINDOW_WIDTH = 1100
```

```
WINDOW_HEIGHT = 600
WINDOW_RES = (WINDOW_WIDTH, WINDOW_HEIGHT)

WIDTH = 100
HEIGHT = 100

WHITE = (255, 255, 255)

SPAWN_RATE = 360
FRAME_RATE = 60

STARTING_BUCKS = 15
BUCK_RATE = 120
STARTING_BUCK_BOOSTER = 1

REG_SPEED = 2
SLOW_SPEED = 1

GAME_WINDOW = display.set_mode(WINDOW_RES)
display.set_caption('Attack of the Vampire Pizzas!')

background_img = image.load('restaurant.jpg')
background_surf = Surface.convert_alpha(background_img)
BACKGROUND = transform.scale(background_surf, WINDOW_RES)

pizza_img = image.load('vampire.png')
pizza_surf = Surface.convert_alpha(pizza_img)
VAMPIRE_PIZZA = transform.scale(pizza_surf, (WIDTH, HEIGHT))

class VampireSprite(sprite.Sprite):
    def __init__(self):
        super().__init__()
        self.speed = REG_SPEED
        self.lane = randint(0, 4)
        all_vampires.add(self)
        self.image = VAMPIRE_PIZZA.copy()
        y = 50 + self.lane * 100
        self.rect = self.image.get_rect(center = (1100, y))

    def update(self, game_window):
        game_window.blit(BACKGROUND, (self.rect.x, self.rect.y), self.rect)
        self.rect.x -= self.speed
        game_window.blit(self.image, (self.rect.x, self.rect.y))

class Counters(object):
    def __init__(self, pizza_bucks, buck_rate, buck_booster):
        self.loop_count = 0
        self.display_font = font.Font('pizza_font.ttf', 25)
        self.pizza_bucks = pizza_bucks
        self.buck_rate = buck_rate
        self.buck_booster = buck_booster
        self.bucks_rect = None
    def increment_bucks(self):
        if self.loop_count % self.buck_rate == 0:
            self.pizza_bucks += self.buck_booster
    def draw_bucks(self, game_window):
        if bool(self.bucks_rect):
            game_window.blit(BACKGROUND, (self.bucks_rect.x, self.bucks_rect.y), 
self.bucks_rect)
        bucks_surf = self.display_font.render(str(self.pizza_bucks), True, WHITE)
        self.bucks_rect = bucks_surf.get_rect()
        self.bucks_rect.x = WINDOW_WIDTH - 50
        self.bucks_rect.y = WINDOW_HEIGHT - 50
        game_window.blit(bucks_surf, self.bucks_rect)
    def update(self, game_window):
        self.loop_count += 1
        self.increment_bucks()
        self.draw_bucks(game_window)

class BackgroundTile(sprite.Sprite):
    def __init__(self, rect):
        super().__init__()
        self.effect = False
        self.rect = rect

all_vampires = sprite.Group()

counters = Counters(STARTING_BUCKS, BUCK_RATE, STARTING_BUCK_BOOSTER)

tile_grid = []

tile_color = WHITE
for row in range(6):
    row_of_tiles = []
    tile_grid.append(row_of_tiles)
    for column in range(11):
        tile_rect = Rect(WIDTH * column, HEIGHT * row, WIDTH, HEIGHT)
        new_tile = BackgroundTile(tile_rect)
        row_of_tiles.append(new_tile)
        draw.rect(BACKGROUND, tile_color, (WIDTH*column, HEIGHT*row, WIDTH, HEIGHT), 1)

GAME_WINDOW.blit(BACKGROUND, (0, 0))

game_running = True
while game_running:
    for event in pygame.event.get():
        if event.type == QUIT:
            game_running = False

        elif event.type == MOUSEBUTTONDOWN:
            coordinates = mouse.get_pos()
            x = coordinates[0]
            y = coordinates[1]
            tile_y = y//100
            tile_x = x//100
            tile_grid[tile_y][tile_x].effect = True

    if randint(1, SPAWN_RATE) == 1:
        VampireSprite()

    for vampire in all_vampires:
        tile_row = tile_grid[vampire.rect.y//100]
        vamp_left_side = vampire.rect.x//100
        vamp_right_side = (vampire.rect.x+ vampire.rect.width)//100
        if 0 <= vamp_left_side <= 10:
            left_tile = tile_row[vamp_left_side]
        else:
            left_tile = None
        if 0 <= vamp_right_side <= 10:
            right_tile = tile_row[vamp_right_side]
        else:
            right_tile = None
        if bool(left_tile) and left_tile.effect:
            if right_tile != left_tile:
                vampire.speed = SLOW_SPEED
        if vampire.rect.x <= 0:
            vampire.kill()

    for vampire in all_vampires:
        vampire.update(GAME_WINDOW)

    counters.update(GAME_WINDOW)

    display.update()

    clock.tick(FRAME_RATE)

pygame.quit()
```