Bomber Game bug report

Design and Professional Skills: Assignment 2

Charlie Barber

Bug 1

Report

After missing a target you are unable to fire again

When the plane drops a bomb and does not hit a building, they are unable to drop any more bombs - even after a restart.

I think the plane should be able to drop more bombs after a missed shot so that the player can continue on with the game.

Cause

I believe the problem lies inside the `move_bomb()` function which determines whether a bomb hit the building:

```
def move_bomb(pos):
    global bomb_falling
    if bomb_falling:
        pos[1] = pos[1] + 8 * speed
```

When the `explode()` function is called, it sets a variable named `bomb_falling` to *False*. This enables the player to then continue to drop more bombs thanks to if statements which keep the game running.

Thanks to debugging with print statements, I know that it is not an issue with the control inputs as the game successfully recognises that the space button has been pressed while this bug is occurring. I can also confirm that the `drop_bomb()` function is getting called during the bug. This means that the bomb was not getting set to not falling when it does not explode.

Fix

To fix this, I had to use an if statement to check if the y coordinate of the bomb was more than the bottom of the towers (more than 700):

```
    if pos[1] > 700:
        bomb_falling = False
```

Bug 2

Report

Towers can never be flattened

Towers will go below the playing screen once they have been fully bombed.

This can be reproduced when the screen is stretched larger than the default option.

Instead, the tower should be deleted when they have been fully bombed.

Cause

When a bomb hits the building, the `shrink_building()` function is called which as the name suggests makes the building smaller. This will *'delete'* the building and then create a smaller one on top seamlessly so the user thinks that the building has shrunk.

```
def shrink_building(canvas, building_num, building_width, building_heights,
building_xpos, building_rects):
    building_heights[building_num] = building_heights[building_num] - 50
    canvas.delete(building_rects[building_num])
    x = building_xpos[building_num]
    building_rects[building_num] = canvas.create_rectangle(x, CANVAS_HEIGHT, x +
building_width,CANVAS_HEIGHT-building_heights[building_num], fill="brown")
```

However, it never checks if the building has been destroyed and creates a new one either way. To fix this I needed to use selection with an if statement to check if the building height is more than 0 before creating a *'new'* building.

```
    if (building_heights[building_num] > 0):
        building_rects[building_num] = canvas.create_rectangle(x, CANVAS_HEIGHT,
x + building_width,CANVAS_HEIGHT-building_heights[building_num], fill="brown")
```

Bug 3

Report

Restarts cause the game to crash

When you restart the game before having finished your first game, the game will crash.

I expect the game to restart whenever the R key is pressed without any errors or crashes

Cause

The error message produced by the Python interpreter is the following:

```
Traceback (most recent call last):
  File "/home/charlie/CS/ENGF2/ENGF2-
2021/assignments/assignment2/bomber_proc.py", line 380, in <module>
    run_game()
  File "/home/charlie/CS/ENGF2/ENGF2-
2021/assignments/assignment2/bomber_proc.py", line 364, in run_game
    (score, level) = restart(canvas, plane_pos, building_heights,
  File "/home/charlie/CS/ENGF2/ENGF2-
2021/assignments/assignment2/bomber_proc.py", line 281, in restart
    canvas.delete(msg_text)
NameError: name 'msg_text' is not defined
```

This suggests that the error is caused by a variable called `msg_text` not being defined. I can see in the code that `msg_text` gets defined when the user loses and it is set to `GAME OVER!`.

Fix

To fix this, I had to set the value of `msg_text` when the `restart()` function is called to an empty string.

```
def restart(canvas, plane_pos, building_heights,
            building_xpos, building_rects):
    global building_width, won, game_running, plane, msg_text
    msg_text = ''
    canvas.delete(msg_text)
```

Bug 4

Report

It is impossible to hit the first tower

It is not physically possible to hit the first tower as the plane spawns on top of it.

The player should be able to hit the first tower or they will never be able finish the game.

Cause

The obvious cause is the spawn point of the plane is too close to the first tower.

Fix

There are two ways that this could be fixed. Either, make the playing screen wider so that the plane starts before the first tower or make the plane spawn at the nose rather than at the tail.

The easiest way to fix was to make the screen wider so that is what I did. This is done by editing the global configuration for Tkinter.

```
CANVAS_WIDTH = 1500
```

Bug 5

Report

Restarting the game does not delete last game's towers

When the R button is pressed, the previous game's towers are not deleted and new ones are added on top of them.

Instead, the towers should be cleared before new ones are generated.

Cause

When creating new buildings, the function includes code to remove all old buildings:

```
def create_buildings(canvas, building_width, building_heights, building_xpos,
building_rects):
    #remove any old buildings
    if len(building_rects) > 0:
        for building_num in range(0, 1200//SPACING):
            delete_building(canvas, building_num, building_rects)
    building_heights.clear()
    building_xpos.clear()
```

When the for loop is called it loops over a range of building numbers. The problem is that none of the buildings on screen actually have a building number in that range.

Fix

Unfortunately I did not find a fix for this bug.