My Pygame Journal

By: Jerry Zhu

I have saved backups of previous codes in order to further document my progress on this pygame assignment.
Please refer to the folder which has all my code, sample code, and all additions like sprites and music.
My main program used(the one you run) is the file called Main.py

12/11-12/13:
Worked On Proposal
Designed Game Layout and Game Flow
Made sure game was realistic and doable in time limit
Designed sprite character(Mario)

12/14:
Worked on building the title screen
    Got a blank screen working
Played around with pygame's surface docs, and the surface.rect
Changed display size

12/16:
Ported a sprite onto the screen(Mario)
Started with a simple pygame sprite, but then decided to do a class
Ported a static background(but it looks bad)
Problem: There were parts of mario that were white
Solution: Use paint and trim edges of sprite

12/17:
Made the sprite move using keyboard controls
Problem: How to scale the sprite down
Solution: Use pygame.transform (Make sure to cast to int) and center scaling
Problem: The background color of mario was messed up
Solution: img.convert_alpha() and img.set_colorkey(WHITE) blends in the sprite background color with the background

12/18:
Tackled a few technical problems:

- Drawing sprites to the screen stopped working
    - Fix: Initialize all variables in the self.init function
- Player didn't scale down properly
    - Fix: Scale down every iteration to avoid glitches
- My player stopped moving
    - Fix: Instead of doing only an update function every tick, do a control function, and an update function, so I could control movement better

12/19:
Problem: Sprite kept moving out of the screen
Solution:

12/20:
I learned not to scale ANYTHING. When you scale something, the hitbox is too hard to control, and it also doesn't resize relative to the center. So, I just manually changed the size using photoshop/ms paint.
I added the ground enemy, and tested my game thoroughly
12/23:
Made game loop a function in order to make initializing sprites and variables so much easier
Called a start function to start the program
This will be useful later for my game-over and win screens

12/24:
Scaled sprite hitbox AGAIN
Problem: Sprite disappeared
Solution: DRAW PLAYER LAST -> spent 1 hr debugging

12/25:
Made another class for the bullet bill (second enemy)
Everything stopped working again
Will fix tomorrow

12/26:
Found out where the error was(uninitialized variables)
Added a few global variables to fix the problem
The bullet bill is moving, but there are still some problems with sprite detection
However, the sprite detection for the enemy still works

12/27:
Sprite detection is very annoying
Tried to search up a solution to the problem -> no luck
Don't want to implement a vector for hitbox movement
Played around with pygame.transform -> use inflate
Inflate basically scales a separate "hitbox" bigger or smaller

12/30:
Implemented it
Of course it didn't work
Commented out everything else to test sprite detection
The sprite hitbox is REALLY weird:
- If you stand very far behind it, you get hit
- If you stand in front of it, the hitbox is cropped
- You cannot go under it, but you can go above it
Realization: inflate and deflate warps the CENTER, not the hitbox

(I had to do lots of testing to realize this)
I needed to change the center after every warp

12/31:
I fixed the problem
I just didn't scale the sprite at all
The bullet bill mostly works, though the hitbox is still a bit warped
I figured out how to check for collision

- I wrote a function called def collided()

Problem: I don't know how to pass this into my colliderect function

1/3:
Read pygame docs to figure out what colliderect needs
Needs a pygame.Group() [A lot more work required]
I tried to figure out how to get pygame.Group to work

1/6:
I made each sprite Class into a pygame.Group
I had to make a sprite list in order to get collide_rect to work
I spent a long time debugging in order to get my group to work

1/7:
I changed most of my draw_screen code to implement collide_rect
I made a "hit_list" to figure out which of my enemies were hitting the player
I'm still getting a few errors though

1/8:
WORKING CODE!!!!
I feel very accomplished
The player now disappears when I hit an enemy
I still have to do a few more methods, but I have a big chunk of the code done

1/9:
I added the methods "revert()" and "start()"
I mostly worked on revert(), which would revert the enemy to their previous positions after they got hit
I also had to make the player revert to start position
Problem: the player might revert and automatically get hit

1/10:
I added a bullet bill (stationary sprite) in order for the bullet to look realistic
I just blitted it onto the screen
I play tested my program thoroughly
Problem 1: The player can run too fast and just glitch out of the screen(I'll refer to this as the run glitch)

Problem 2: The player will keep going left if you die while holding left(I'll refer to this as the direction glitch)
Problem 3: When the player hits an enemy close to the boundary, by jumping into the corner very fast, the player can "clip" into the boundary and the enemy will miss hitting the player(I'll refer to this as boundary glitch)

1/13:
I first need to fix the run glitch.
To fix this, I first set a max speed for the player
The player should not be able to accelerate.
I reverted all enemies as well as the player to the start position after each death
That way, the player can't immediately hit another enemy after the first one
The run glitch was one of the easiest ones to solve

1/14:
I added a dash function and a lakitu that was flying in the air
I spent most of the day debugging
It was hard to get the lakitu to move left and right instead of just one direction
I had to use two extra variables to track the lakitu's movement so I know when to start moving in the other direction.

1/15:
I made the egg fall (this time it was much easier)
I fixed the boundary glitch

- I increased the hitbox on the sprite

I realized that I could just move the center to one side, then scale one side drastically
I added a health bar for my sprite
Boundary tracking is hard (very very very hard)
I still need to reformat my code; there are lots of syntaxical or just random errors that pop up whenever I test

1/16:
I tried fixing last glitch(direction glitch), but this one is MUCH harder to fix
This was very depressing for me, as I couldn't get much progress
I tried everything: commenting out code, searching stuff up, printing variables, using the debugger
I even tried writing a seperate file with the code in it, to no avail.
However, I managed to replicate another bug: when my player hits the ground enemy right as it

1/17:
Pygame.quit() is stupid. End of story.
When you have several layers, it is

1. Harder to navigate through code
2. Harder to find stuff
3. HARDER TO BREAK

However, I need lots of layers for my logic to work. I have very complicated logic for my sprites that need lots of layers.
I can’t seem to fix the problem.
I added health bars for both the player and the boss
This was MUCH more easier then the pygame.quit(), but it still took me a significant amount of time before I was able to figure it out
I also had to add/subtract values to the boss health and the player's health, so I just did a frame switch on my program

1/18:
I fixed boundary glitch today
Basically, I wrote my custom quit function using sys and abusing the exit() function by breaking out of the entire program(causing an error)
I also got the boss and the bomb working
I made a final_cnt and boss_appear to count when the boss would appear on the screen
This was hard as I didn’t know how to implement it at the start
I added this into the boss classes so I could track when to make the boss and its related enemies visible to the player
Bug: the player kept dying for no reason if he stood at a particular position
I also had to tell the program not to hit the player if any (off screen) enemies hit the player at any point before the boss is loaded, but to hit the player if the boss is loaded on screen
DId some work with on screen/off screen sprite loading

1/19:
I put music, a title screen, ending screen, and instructions into my program
I had to edit the entire program flow, and added lots of methods in order to let it run smoothly
I couldn’t get the instructions method to call the start method, so I just combined it with the intro_screen method
The music was hard to do, as pygame.mixer wasn’t cooperating
I realized you had to do mixer.Sound to set the theme music
I also trimmed most of the .wav files in order to get them to work
I also forgot to clear the screen(rip 30 min of debugging)
I think I’m almost done, just need to fix logic errors

1/20:
I spent the entire day fixing logic errors
There were errors everywhere: I had to playtest multiple times in every single scenario, and comment out multiple lines of code to get the errors out of my code
Most of these errors were stupid mistakes(unititialized variables, forgot to scale hitbox, pygame.quit() problems, rect value not properly inited)
A big problem i got was when the hitbox of the powerup was warped
This was because when you change self.rect, you have to change self.hitbox too, since it won’t update until the next time you call update.
However, if you never call update, in the case of the powerup, you will never change the hitbox, so the hitbox will be different from the rect

So, for most sprites, just to be safe, I added a hitbox change after the sprite declaration to make the sprite hitbox work the first time
I feel that after extensive testing, there should be no errors left in my code.
(I asked my friends to test my programs too :))
I know that there are a few small things that I could change, and I might even continue working on this project after this course, as I have learned a lot about making games, and would love to continue making cool games for me and my friends to play.

<u>Results:</u>

Objective:
Yes! The objective is to survive as long as possible and destroy the boss!

Good Code:
Yes! My code is formatted properly, and I use lots of classes and methods and sprites and groups to make my code much easier to read, and easier to understand/debug. I use many different logical skills like loops and conditionals to make the game work.

Incentive:
You are rewarded with power ups and health as the game goes on.
You are incentivized with cool music to keep going and get the true ending(beat the boss).
The game is fun as it is easy to beat the first part, but harder to beat the boss.

User Friendly:
Yes! The game starts out easy, but gets harder and harder, and the boss adds on to the difficulty.

Instructions?
Yes! I have a start/instructions screen, a game over screen, and a win screen.
I tell the player how to play the game.

Music/Sound effects?
Yes! I not only have cool theme and boss music, but I also have ending music for each possible stage you get to.
My music incentivizes the player to keep playing and surviving!
I have sound effects for dying in the game too.
Songs Used:
Home – We’re Finally Landing
Green Orbs - At The Fair
Jazz In Paris
Declan DP - We Are Here
Kevin MacLeod - Overworld
Kevin MacLeod - Failing Defense
Sound Effects:

Crash Sound - Youtube Audio Library

Journal?
Yes! :)

Tested? Fun?
Yes and yes! My game has been tested by multiple people, and they all say that my game is very fun to play.
I also think that my game should be a great game to play as it is challenging, fun, and incentivizing.
I really liked making the game as I could see my progress after each point in time, and I would feel a sense of accomplishment as I finish each part of the game. It was a great experience for me, and I hope to make more and better games(or even improve this game) in the future!