



VIT[®]
BHOPAL

vityarthi
YOUR LEARNING DESTINATION

Flappy Bird Project Report (Manny Bird)

Title : Manny Bird

Name : Mann Sharma

Course : CSE

Date : November 2025

Registration Number : 25BAI10379

Introduction ~

This project is a remake of the popular mobile game flappy Bird using Python and the pygame library. The main idea of the game is very simple: the player controls a bird that must fly between pipes without hitting them. Even though the gameplay looks easy, it is actually very challenging and addictive.

functional requirements ~

- bird moves with spacebar
- pipes spawn with random gaps
- collision detection
- score tracking
- reset after game over

Non-functional Requirements ~

- Smooth gameplay at 60 FPS
- Responsive controls
- Minimal lag or glitches
- Simple UI and sound effects

System Architecture ~

The game uses a modular structure with separate files for main loop, assets, and logic. Pygame handles rendering, input, and timing.

- main loop: events + rendering
- bird class: position + image
- pipe class: position + scoring
- event system: key presses + timers

implementation details ~ Programming Language & Tools ~

- **Language:** Python 12.7
- **Library:** Pygame (for graphics, sound, input handling)

Game Logic -

- Gravity and jump physics for the bird
- Pipes move leftward; bird stays in place
- Collision detection using bounding boxes
- Score increases when bird passes a pipe

Assets Used -

- png sprites for bird, pipes, background
- Fonts for score display

screenshots & results -

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** Shows a project named "MANNY BIRD PROJECT cse" containing files: "project report", "bottompipe.png", "flappybird.png", "flappybird.py", "flappybirdbg.png", and "toppipe.png".
- Code Editor:** The main editor window displays the "flappybird.py" file content, which includes imports for pygame, sys, and random, and defines classes for Bird and Pipe.
- Terminal:** The bottom terminal shows the message: "Hello from the pygame community. https://www.pygame.org/contribute.html".
- Suggested Actions:** A sidebar on the right lists actions such as "Build Workspace", "Show Config", and "Explore and understand" for the "flappybird.py" file.
- System Status:** The bottom status bar shows system information like temperature (23°C), battery level, and date (22-11-2025).

A screenshot of a Microsoft Visual Studio Code (VS Code) interface showing a Python project for a Flappy Bird clone.

The left sidebar shows the Explorer view with files in the "MANNY BIRD PROJECT CSE" folder:

- project report
- bottompipe.png
- flappybird.png
- flappybird.py**
- flappybirddbg.png
- toppipe.png

The "TERMINAL" tab at the bottom shows the command history:

```
4  
6  
6  
6  
2  
2  
4  
6
```

The main code editor window displays the "flappybird.py" file:

```
1 import pygame  
2 from sys import exit  
3 import random  
4  
5 game_width = 360  
6 game_Height = 640  
7  
8 # setting mannie bird xD  
9 bird_x = game_width/8  
10 bird_y = game_Height/2  
11 bird_width = 34  
12 bird_height = 24  
13  
14 class Bird(pygame.Rect):  
15     def __init__(self, img):  
16         pygame.Rect.__init__(self, bird_x, bird_y, 0, 0)  
17         self.img = img  
18  
19     pipe_x = game_width  
20     pipe_y = 0  
21     pipe_width = 64  
22     pipe_height = 512  
23  
24  
25 class Pipe(pygame.Rect):  
26     def __init__(self, img):  
27         pygame.Rect.__init__(self, pipe_x, pipe_y, pipe_width, pipe_height)  
28         self.img = img  
29         self.passed = False  
30  
31  
32 #game images
```

The center of the screen shows a window titled "Manny Bird" displaying the Flappy Bird game. A yellow bird is positioned between two green pipes. The background features a blue sky with white clouds and a green ground with brown patches.

The status bar at the bottom right indicates:

- Ln 1, Col 1
- Spaces: 3
- UTF-8
- CRLF
- { } Python
- Python 3.12 (64-bit)
- 17:04
- 22-11-2025

A sidebar on the right contains an AI interface:

- Ask about your code
- AI responses may be inaccurate.
- Generate Agent
- Instructions to onboard AI onto your codebase.

Suggested actions include:

- Build Workspace
- Show Config

A screenshot of a Microsoft Visual Studio Code (VS Code) interface showing a Flappy Bird game project.

The left sidebar shows the Explorer view with files in the "MANNY BIRD PROJECT CSE" folder:

- project report
- bottopipe.png
- flappybird.png
- flappybird.py**
- flappybirddbg.png
- toppipe.png

The center-left area displays the Python code for "flappybird.py":

```
138  
139  
140  
141  
142     while True: #game loop inmaking  
143         for event in pygame.event.get():  
144             if event.type == pygame.QUIT :  
145                 pygame.quit()  
146                 exit()  
147             if event.type == create_pipes_timer and not GAME_OVER:  
148                 create_pipes()  
149  
150             if event.type == pygame.KEYDOWN:  
151                 if event.key in ( pygame.K_SPACE , pygame.K_UP ):  
152                     velocity_y = -5  
153  
154                     #reset game  
155                     if GAME_OVER:  
156                         bird.y = bird_y  
157                         pipes.clear()  
158                         score = 0  
159                         GAME_OVER = False  
160  
161  
162             if not GAME_OVER :  
163  
164                 move()  
165                 draw()  
166                 pygame.display.update()  
167  
168                 clock.tick(60) #60fps  
169  
170
```

The center-right area shows a window titled "Manny Bird" displaying the Flappy Bird game. The screen shows two green pipes, a yellow bird flying between them, and the text "MaNn Over 0" indicating the player has died. The background features a blue sky with white clouds and green ground with small buildings.

The bottom status bar shows:

- Ln 1, Col 1
- Spaces: 3
- UTF-8
- CRLF
- { } Python
- Python 3.12 (64-bit)
- 23°C Sunny
- 22-11-2025
- 17:05

The right sidebar includes:

- SUGGESTED ACTIONS
 - Build Workspace
 - Show Config
- A speech bubble icon with the text "Ask about your code". Below it, smaller text says "AI responses may be inaccurate." and "Generate Agent", "Instructions to onboard AI onto your codebase."

Testing approach -

Manually played it for some time tested every possible way and yeah initially there were some bugs like the score wasn't increasing ,there was a time when it couldn't detect if the bird touched the ground or not but used a bit of youtube and google and fixed it .

challenges faced -

Looping was very hard and settings pole were very hard to used a bit of youtube . It was quite challenging it took me like 2days to understand how pygame works but yeah with the help of yt and google I managed to get it done .

Learnings -

Learned how newtons is applicable in coding lol like gravity . Also learned about how to use pygame , how to use random function ,how to use include external png's , how to resize them etc.

future enhancements -

add sounds, menu, high scores, bird animation, mobile controls . This was a basic game in future I might make it 3d using unreal engines maybe

References- Youtube tutorials and Github examples.

