



Comparing 2D game in Lua with game engine of love and in python with pygame

Rakshit Tyagi¹ | Prof. Shreya Kapoor²

¹B.Tech Scholar , Department of IT ,Maharaja Agrasen Institute of Technology, Delhi, India

²Assistant Professor, Department of IT , Maharaja Agrasen Institute of Technology, Delhi, India,

To Cite this Article

Rakshit Tyagi and Prof. Shreya Kapoor, "Comparing 2D game in Lua with game engine of love and in python with pygame", *International Journal for Modern Trends in Science and Technology*, 6(12): 370-372, 2020.

Article Info

Received on 16-November-2020, Revised on 09-December-2020, Accepted on 12-December-2020, Published on 17-December-2020.

ABSTRACT

The game flappy bird is used in the process of research .The game is used in both LUA and python programming language and the main aim of the user in game is to move forward by avoiding any contact with tunnel sprite .Also ,Flappy has to stay in air all the time without touching with ground and collects coins in between to increase the score. The game is 2D.

[1]Flappy Bird is an arcade-style game in which the player controls the bird Flappy, which moves persistently to the right. The player is tasked with navigating Flappy through pairs of pipes that have equally sized gaps placed at random heights. Flappy automatically descends and only ascends when the player taps control . [2]Each successful pass through a pair of pipes awards the player one point. Colliding with a pipe or the ground ends the gameplay. During the game over screen, the player is awarded a bronze medal if they reached ten or more points, a silver medal from twenty points, a gold medal from thirty points, and a platinum medal from forty points.

KEYWORDS: flappy , Lua , python

I. INTRODUCTION

The game flappy bird is used in the process of research .The game is used in both LUA and python programming language and the main aim of the user in game is to move forward by avoiding any contact with tunnel sprite .Also ,Flappy has to stay in air all the time without touching with ground and collects coins in between to increase the score. The game is 2D.

[1]Flappy Bird is an arcade-style game in which the player controls the bird Faby, which moves persistently to the right. The player is tasked with navigating Faby through pairs of pipes that have equally sized gaps placed at random heights. Faby automatically descends and only ascends when the player taps control . [2]Each successful pass

through a pair of pipes awards the player one point. Colliding with a pipe or the ground ends the gameplay. During the game over screen, the player is awarded a bronze medal if they reached ten or more points, a silver medal from twenty points, a gold medal from thirty points, and a platinum medal from forty points.

Pygame is a cross-platform set of Python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used with the Python programming language. Pygame uses the Simple DirectMedia Layer (SDL) library,[a] with the intention of allowing real-time computer game development without the low-level mechanics of the C programming language and its

derivatives. This is based on the assumption that the most expensive functions inside games can be abstracted from the game logic, making it possible to use a high-level programming language, such as Python, to structure the game.

Other features that SDL doesn't have include vector math, collision detection, 2d sprite scene graph management, MIDI support, camera, pixel-array manipulation, transformations, filtering, advanced freetype font support, and drawing.

Applications using pygame can run on Android phones and tablets with the use of pygame Subset for Android (pgs4a). Sound, vibration, keyboard, and accelerometer are supported on Android.

LÖVE (or Love2D) is an open-source cross-platform engine for developing 2D video games. The engine is written in C++ and uses Lua as its scripting language. It is published under the zlib license.

The API provided by the engine gives access to the video and sound functions of the host machine through the libraries SDL and OpenGL, or since version 0.10 also OpenGL ES 2 and 3. Fonts can be rendered by the FreeType engine.[4] A version of the engine called piLöve has been specifically ported to Raspberry Pi.

It also provides a basic "sandbox" management of the files in order to avoid giving access to all its disk to the executed games

II. METHODOLOGY

This study is an applied research in terms of purpose and is a quasi-experiment in terms of data collection method. The comparison is based on four parameters

Speed

During game play, delays in processing often have severe consequences, providing large incentive for players to increase speed. Accordingly, there is anecdotal evidence that avid game players react more readily to their environment. However, it remains unknown whether any reduction in reaction time (RT) really generalizes to tasks beyond video-game playing and, if it does, whether it makes gamers more impulsive and prone to making errors. The 25-meter-long path with the sub-maximal running (c) and the second 25 m at full speed (d); then, they go the rest of the path slowly to recover at the starting line (e).

Graphics

A variety of computer graphic techniques have been used to display video game content throughout the history of video games. The predominance of individual techniques have evolved over time, primarily due to hardware advances and restrictions such as the processing power of central or graphics processing units

Libraries

Different libraries have ability of different usage physics models which plays a vital role in developing a game

Game handling

Game lags are when there's a delay between the action of the players and the reaction of the game server. Alongside the **internet connection** cutting out, they're one of the most frustrating things a gamer will experience

III. RESULTS

1 SPEED

Game was slower in fps performance in python with respect to lua. In Pygame, you need(!) to catch every Quad that changed in the Frame before and draw only the Changes, otherwise it's impossible to draw Background and Sprites on a Resolution like 1024x768 and 60 FPS.

Lua's performance compares very well to other languages, If performance needs to be further improved you can:

- (i) Implement critical parts in C
- (ii) Use the LuaJIT compiler. The LuaJIT compiler is a drop in replacement for the stock compiler and provides significant performance improvements.

[6] As we know, Python is an interpreted language. Interpreted code is always slower than direct machine code because it takes a lot more instructions in order to implement an interpreted instruction than to implement an actual machine instruction.

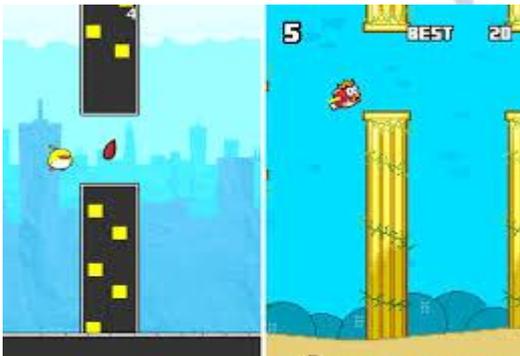
2 Graphics

Python has a huge collection of open-source libraries that do everything, from graphics, physics, and charts to machine learning. There are at least ten really well-documented and widely used libraries that help with machine learning alone.

IV. CONCLUSION

As we know because of more complex and steady graphics in today's world it is easy to make games based on pygame. Although syntax of lua programming language is much easier to learn in comparison to python because of its messy design, but with demand of new features and graphics pygame is ahead of love.

Pygame is a great tool for developers. It isn't set up to be easily compatible with mobile devices, and due to the nature of Python it isn't great for commercial products in general; however, its strengths include its ease of use and accessibility. The flow of the Pygame workflow is an amazing way for beginners to learn game programming at a deeper level because it is straightforward while also having a fair amount of depth. It prepares you thoroughly to begin using much bigger and better libraries that are used commercially like libGDX with Java, and even the LÖVE framework in Lua.



([5]graphics love vs pygme)

3 Libraries

Python ships with a large standard library, including modules for everything from writing graphical applications, running servers, and doing unit testing. This means that beginners won't need to spend time searching for tools and libraries just to get started on their projects.

Lua has very few libraries.

```
def main():
    # Create the window
    screen = pygame.display.set_mode((400, 600))
    pygame.display.set_caption("Flappy Bird")
    # Create the bird
    bird = Bird(50, 50)
    # Create the pipes
    pipes = []
    # Create the ground
    ground = pygame.Rect(0, 580, 400, 600)
    # Create the score
    score = 0
    # Create the best score
    best_score = 0
    # Create the font
    font = pygame.font.Font('freesansbold.ttf', 32)
    # Create the text
    text = font.render("Flappy Bird", True, black)
    # Create the background
    background = pygame.Surface(screen.get_size())
    background.convert_alpha()
    background.fill((255, 255, 255))
    # Create the clouds
    clouds = []
    # Create the sun
    sun = pygame.Rect(200, 100, 100, 150)
    # Create the game loop
    while True:
        # Get the events
        for event in pygame.event.get():
            # Quit if the user clicks the close button
            if event.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
            # If the user clicks the spacebar, the bird jumps
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_SPACE:
                    bird.jump()
        # Update the bird
        bird.move()
        # Update the pipes
        pipes.move()
        # Update the ground
        ground.move()
        # Update the score
        score += 1
        # Update the best score
        if score > best_score:
            best_score = score
        # Update the text
        text = font.render(str(score), True, black)
        # Update the background
        background.fill((255, 255, 255))
        # Update the clouds
        clouds.move()
        # Update the sun
        sun.move()
        # Draw the window
        screen.blit(background, (0, 0))
        screen.blit(text, (200, 100))
        screen.blit(sun, (200, 100))
        # Update the display
        pygame.display.update()
        # Wait for the next frame
        pygame.time.wait(60)
```

([8]Different libraries of pygame)

4 Game Handling

With love game engine the game was lagging a lot initially until the code was fully executed on the other hand with pygame game handling was pretty smooth.

REFERENCES

- [1] "Contributors to Pygame". GitHub.
- [2] Shinnars, Pete. "Python Pygame Introduction - History". Pygame.org. Archived from the original on 2019-09-17. Retrieved 2017-04-28.
- [3] "Downloads - Pygame - Python game development". Pypi.python.org.
- [4] Williams, Rhiannon. "What is Flappy Bird? The game taking the App Store by storm". The Daily Telegraph. Archived from the original on January 30, 2014. Retrieved January 30, 2014.
- [5] McCracken, Harry (February 9, 2014). "Where to Get Flappy Bird: On eBay, for \$900. Cheap". Time. Archived from the original on March 7, 2014. Retrieved February 9, 2014.
- [6] Molina, Brett (February 10, 2014). "Phones with 'Flappy Bird' app fetching big bids on eBay". USA Today. Archived from the original on February 10, 2014. Retrieved February 10, 2014.
- [7] Thier, Dave (February 10, 2014). "'Flappy Bird' Price Skyrocketing on eBay". Forbes. Archived from the original on February 12, 2014. Retrieved February 10, 2014.
- [8] Watts, Jack (December 31, 2014). "Google looks back at 2014 trending topics with New Year's Eve 2014 Google Doodle". Daily Express. Archived from the original on December 31, 2014. Retrieved January 1, 2015.
- [9] Jump up to:^a ^b "Flappy Bird Reviews on Metacritic". Metacritic. Archived from the original on March 2, 2014. Retrieved February 9, 2014.
- [10] Quigley, Joseph (1 June 2007). "A Look at Lua". Linux Journal.
- [11] Hamilton, Naomi (11 September 2008). "The A-Z of Programming Languages: Lua". Computerworld. IDG. Archived from the original on 8 July 2018. Retrieved 7 July 2018. Interview with Roberto Ierusalimsky.
- [12] Ierusalimsky, Roberto; de Figueiredo, Luiz Henrique; Celes, Waldemar (12 May 2011). "Passing a Language through the Eye of a Needle". ACM Queue. **9** (5): 20–29. doi:10.1145/1978862.1983083. S2CID 19484689. Archived from the original on 8 July 2018. Retrieved 7 July 2018. How the embeddability of Lua impacted its design.