UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

NAAC GRADE A+ ACCREDITED UNIVERSITY

A PROJECT REROART
ON
CAR GAME IN LINUX

Submitted by

Gourav (24MCA20114)
Section – 24MCA- 2(B)

in partial fulfilment for the award of the degree of

Master of Computer Application

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

Chandigarh University

Table Of Content

INTRODUCTION

1.1. Identification of Client / Need / Relevant Contemporary Issue

The gaming industry is rapidly evolving, with an increasing demand for engaging and interactive experiences. Statistics show that the global gaming market is expected to reach $256.97 billion by 2025, demonstrating the relevance of creating new games that capture players' attention. This report addresses the need for a simple yet effective car racing game that not only entertains but also serves as an introductory project for budding game developers.

1.2. Identification of Problem

The broad problem identified is the lack of accessible game development projects for beginners, which often leads to frustration and discouragement in learning programming skills. The issue at hand is to create an engaging car racing game using Pygame, which will help new developers understand the fundamental concepts of game development.

1.3. Identification of Tasks

The tasks required to identify, build, and test the solution are as follows:

- **Task 1**: Define game mechanics and controls.
- **Task 2**: Design the user interface, including menus and instructions.

- **Task 3**: Implement game logic, including collision detection and scoring systems.
- **Task 4**: Test the game for bugs and user experience issues.
- **Task 5**: Document the development process and prepare a final report.

1.4. Timeline

A Gantt chart will be provided in the appendices to illustrate the project timeline, detailing the completion of each task over a span of 4 weeks.

1.5. Organization of the Report

- **Chapter 1**: Introduction - Overview of the project, client needs, and identified problems.
- **Chapter 2**: Literature Review - Background study and existing solutions in game development.
- **Chapter 3**: Design Flow/Process - Specification evaluation, design constraints, and implementation plan.
- **Chapter 4**: Results Analysis and Validation - Implementation details and analysis of results.
- **Chapter 5**: Conclusion and Future Work - Summarization of findings and suggestions for further development.

LITERATURE REVIEW/BACKGROUND STUDY

2.1. Timeline of the Reported Problem

Game development has seen significant milestones since the 1970s, with various frameworks and languages emerging over the decades. The first racing game, "Gran Trak 10," was released in 1974, paving the way for future developments.

2.2. Existing Solutions

Various game engines such as Unity and Unreal Engine dominate the market for game development. However, Pygame remains a popular choice for beginners due to its simplicity and ease of use for 2D games.

2.3. Bibliometric Analysis

Key features of existing solutions include:

- **Unity**: Robust graphics and physics engine; high learning curve.
- **Unreal Engine**: Advanced visual capabilities; not beginner-friendly.
- **Pygame**: Lightweight; excellent for learning programming concepts; limited graphics capabilities.

2.4. Review Summary

The literature review indicates a gap in beginner-friendly resources that focus on 2D game development. This project aims to bridge that gap by creating a simple car racing game using Pygame.

2.5. Problem Definition

The problem at hand is to design and implement a 2D car racing game that is accessible for beginners. The game must incorporate basic mechanics such as user input for car movement, collision detection, and scoring systems.

2.6. Goals/Objectives

- **Objective 1**: Develop a functional car racing game using Pygame.
- **Objective 2**: Ensure the game is user-friendly and engaging.
- **Objective 3**: Document the development process for educational purposes.

DESIGN FLOW/PROCESS

3.1. Evaluation & Selection of Specifications/Features

Key features required for the solution include:

- User controls (arrow keys for movement, A for accelerate, B for brake).
- Scoring system based on obstacles dodged.
- Pause and restart functionality.

3.2. Design Constraints

Design constraints considered:

- **Standards**: Adherence to software development best practices.

- **Health/Safety**: Ensuring the game does not promote reckless driving behavior.
- **Cost**: Utilization of free resources and software.

3.3. Analysis of Features and Finalization Subject to Constraints

Features were evaluated based on feasibility within Pygame and adjusted to meet project constraints, such as performance and usability.

3.4. Design Flow

Two alternative designs were considered:

- **Design A**: Basic obstacle dodging mechanics.
- **Design B**: Advanced features like power-ups and levels.

3.5. Design Selection

Design A was selected for its simplicity and focus on core gameplay mechanics, making it more suitable for beginners.

3.6. Implementation Plan/Methodology

The implementation was carried out using a modular approach, with a flowchart detailing the game loop, event handling, and rendering process.

RESULTS ANALYSIS AND VALIDATION

4.1. Implementation of Solution

The solution was implemented using the following tools:

- **Pygame** for game development.
- **Python** as the programming language.
- Testing was conducted to ensure functionality and user experience.

Entering to root user.

```
liveuser@localhost-live:~$ su root
Password:
```

Check the python version an then install pip manager.

```
root@localhost-live:/home/liveuser# python3 --version
pip3 --version
Python 3.12.2
bash: pip3: command not found...
root@localhost-live:/home/liveuser# dnf install python3-pip
Last metadata expiration check: 0:15:51 ago on Fri 01 Nov 2024 06:15:56 AM EDT.
Dependencies resolved.
================================================================================
 Package             Architecture   Version               Repository       Size
================================================================================
Installing:
 python3-pip         noarch         23.3.2-2.fc40         updates         3.2 M

Transaction Summary
================================================================================
Install  1 Package

Total download size: 3.2 M
Installed size: 14 M
Is this ok [y/N]: y
Downloading Packages:
python3-pip-23.3.2-2.fc40.noarch.rpm           510 kB/s | 3.2 MB     00:06
--------------------------------------------------------------------------------
Total                                          497 kB/s | 3.2 MB     00:06
```

Now install pygame library.

```
root@localhost-live:/home/liveuser# pip3 install pygame
Collecting pygame
  Downloading pygame-2.6.1-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl.metadata (12 kB)
Downloading pygame-2.6.1-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.
whl (14.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 14.0/14.0 MB 1.7 MB/s eta 0:00:00
Installing collected packages: pygame
Successfully installed pygame-2.6.1
WARNING: Running pip as the 'root' user can result in broken permissions and con
flicting behaviour with the system package manager. It is recommended to use a v
irtual environment instead: https://pip.pypa.io/warnings/venv
root@localhost-live:/home/liveuser#
```

Create a file name cargame.py using touch command and to write in file use nano command.

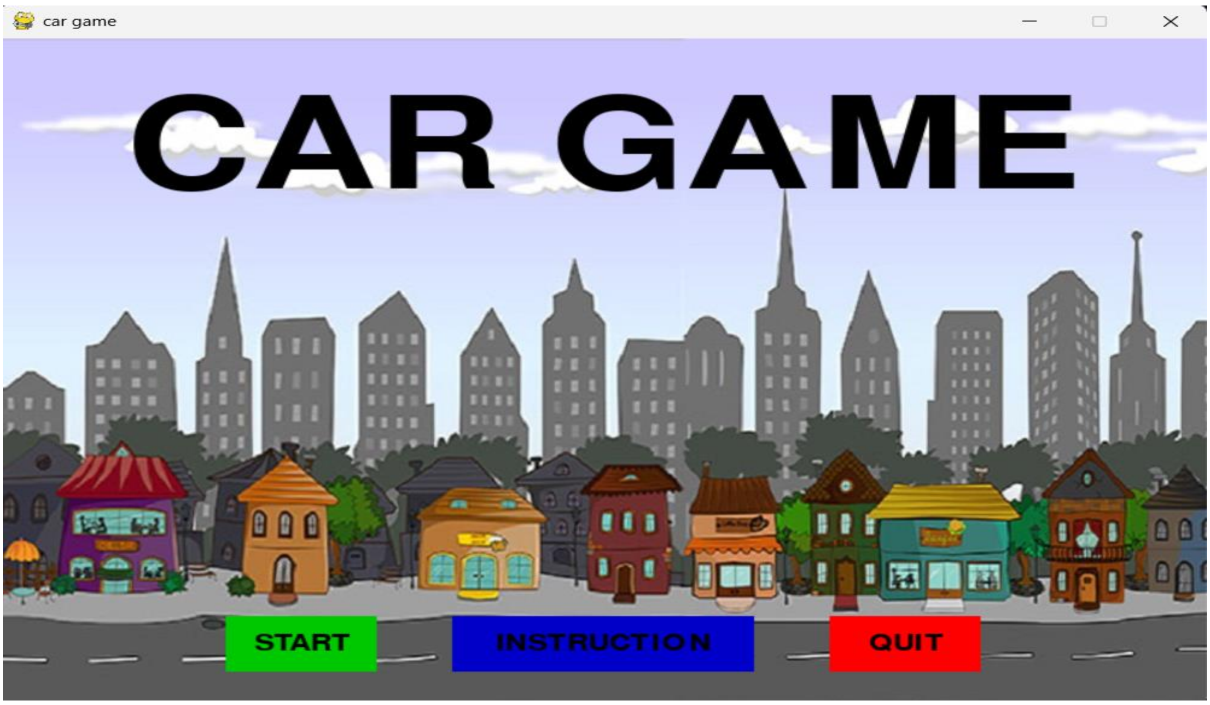
```
root@localhost-live:/home/liveuser# touch cargame.py
root@localhost-live:/home/liveuser# nano cargame.py
```

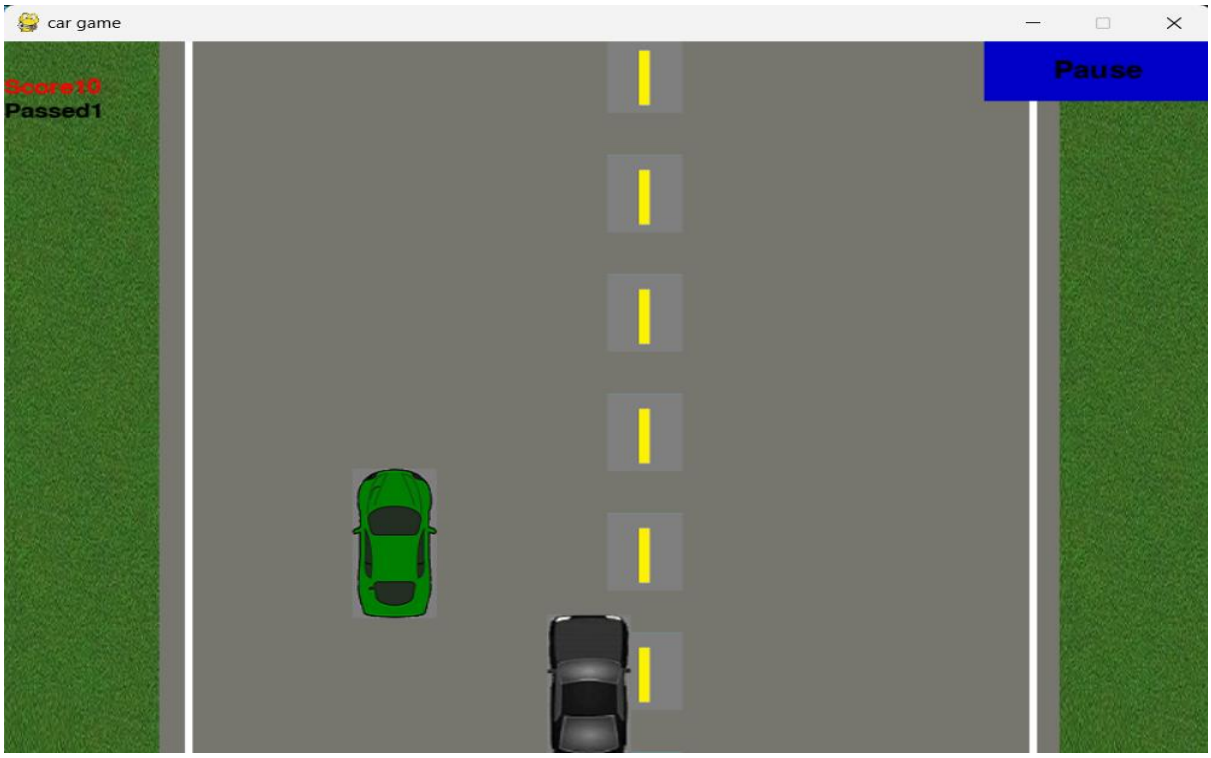
Enter your code and save it.

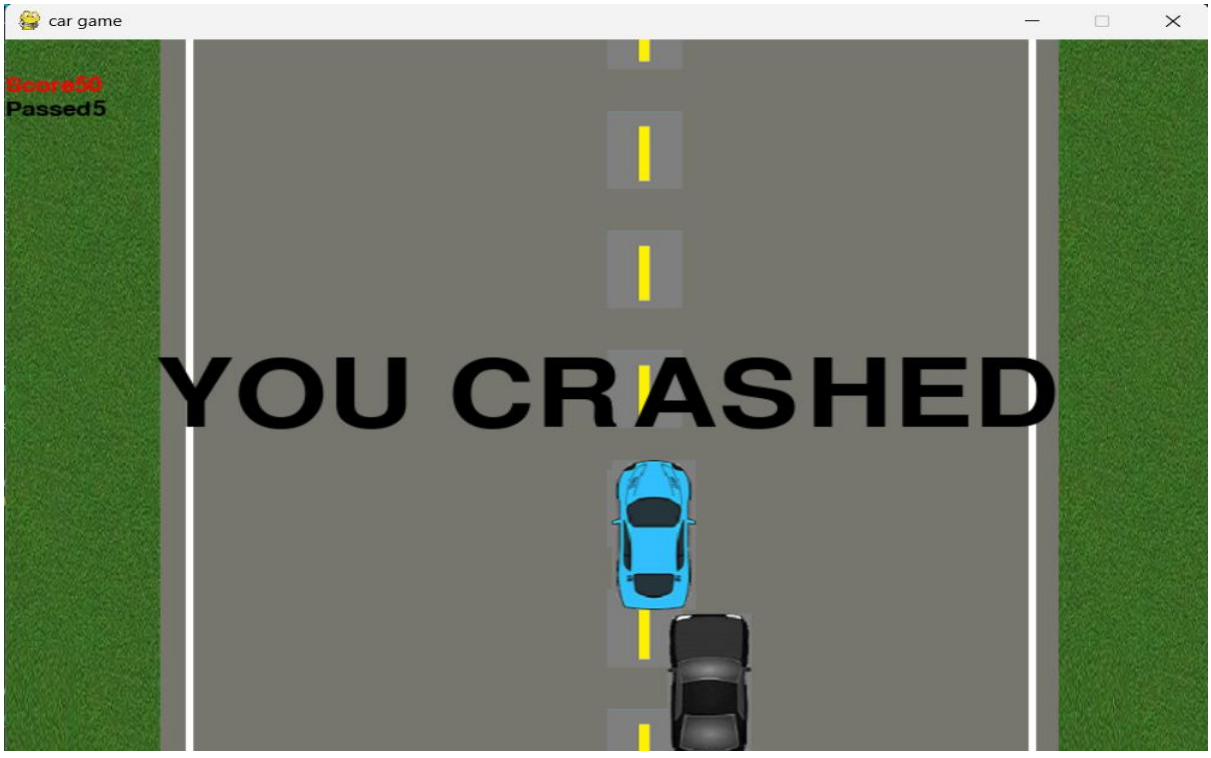
```
  GNU nano 7.2                          cargame.py
import pygame
pygame.init()
gray=(119,118,110)
black=(0,0,0)
red=(255,0,0)
green=(0,200,0)
blue=(0,0,200)
bright_red=(255,0,0)
bright_green=(0,255,0)
bright_blue=(0,0,255)
display_width=800
display_height=600
import time
import random


gamedisplays=pygame.display.set_mode((display_width,display_height))
pygame.display.set_caption("car game")
clock=pygame.time.Clock()
carimg=pygame.image.load('car1.jpg')

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

Run the file.

```
root@localhost-live:/home/liveuser# python3 cargame.py
pygame 2.6.1 (SDL 2.28.4, Python 3.12.2)
Hello from the pygame community. https://www.pygame.org/contribute.html
XDG_RUNTIME_DIR (/run/user/1000) is not owned by us (uid 0), but by uid 1000! (T
his could e.g. happen if you try to connect to a non-root PulseAudio as a root u
ser, over the native protocol. Don't do that.)
ALSA lib pcm.c:8740:(snd_pcm_recover) underrun occurred
ALSA lib pcm.c:8740:(snd_pcm_recover) underrun occurred
ALSA lib pcm.c:8740:(snd_pcm_recover) underrun occurred
ALSA lib pcm.c:8740:(snd_pcm_recover) underrun occurred
ALSA lib pcm.c:8740:(snd_pcm_recover) underrun occurred
ALSA lib pcm.c:8740:(snd_pcm_recover) underrun occurred
ALSA lib pcm.c:8740:(snd_pcm_recover) underrun occurred
ALSA lib pcm.c:8740:(snd_pcm_recover) underrun occurred
ALSA lib pcm.c:8740:(snd_pcm_recover) underrun occurred
ALSA lib pcm.c:8740:(snd_pcm_recover) underrun occurred
ALSA lib pcm.c:8740:(snd_pcm_recover) underrun occurred
ALSA lib pcm.c:8740:(snd_pcm_recover) underrun occurred
ALSA lib pcm.c:8740:(snd_pcm_recover) underrun occurred
ALSA lib pcm.c:8740:(snd_pcm_recover) underrun occurred
ALSA lib pcm.c:8740:(snd_pcm_recover) underrun occurred
ALSA lib pcm.c:8740:(snd_pcm_recover) underrun occurred
ALSA lib pcm.c:8740:(snd_pcm_recover) underrun occurred
ALSA lib pcm.c:8740:(snd_pcm_recover) underrun occurred
```




car game
Score10
Passed1
Pause


car game
Score50
Passed5
YOU CRASHED

CONCLUSION AND FUTURE WORK

5.1. Conclusion

The expected outcome was achieved, resulting in a functional car racing game that serves as a learning tool for beginners. Minor deviations from expected results were addressed during testing.

5.2. Future Work

Future enhancements could include:

- Adding more levels and obstacle types.
- Implementing online leaderboards for competitive play.
- Expanding the game with enhanced graphics and sound.

Source:

Drive: https://drive.google.com/drive/folders/1FyLSFpD9XuN_Z7SgKhG75q_ApbXiXSX?usp=drive_link

GitHub: https://github.com/Gourav2811/Linux-project