

Revving Up Memories: Building a Classic Car Racing Game Using Amazon Q CLI in a Single Prompt

Introduction

Remember those pixelated, top-down car racing games on your old feature phone? The thrill of dodging traffic and hitting perfect turns sparked a passion for speed and strategy that few modern games can replicate.

I became completely captivated by those early racing experiences!

Now, I'm recreating a classic car racing game using Python and Pygame, with a modern twist—I'm harnessing Amazon Q CLI to accelerate development.

So, fasten your seatbelts! We're building this racing game in just one prompt. Yes, just one!

Installation & Setup: Amazon Q Developer CLI on Windows (WSL)

If you're building a game and prefer working in a Linux environment while on Windows, this guide walks you through setting up WSL (Windows Subsystem for Linux) and installing the Amazon Q Developer CLI using Ubuntu.

Step 1: Install WSL on Windows PowerShell

Open PowerShell as Administrator and run the following command to install WSL:

wsl --install

This command performs the following actions:

- Enables required optional components:
 - Windows Subsystem for Linux
 - Virtual Machine Platform
- Downloads and installs the latest Linux kernel
- Sets WSL 2 as the default version
- Installs the Ubuntu Linux distribution
- Prompts for a system reboot (if necessary)
- After reboot, you'll be prompted to:

- Set your Linux username
- Set a password

Once completed, you can run Linux commands directly from PowerShell or Ubuntu Terminal.

To switch to the root user, use:

wsl -u root

Step 2: Install Amazon Q Developer CLI in Ubuntu (via WSL)

- Navigate to the Home Directory: **cd**
- Download the Amazon Q CLI (latest release): **sudo curl --proto '=https' --tlsv1.2 -sSf https://desktop-release.codewhisperer.us-east-1.amazonaws.com/latest/q-x86_64-linux-musl.zip -o q.zip**
- Install unzip (if not already installed): **sudo apt install unzip**
- Extract the ZIP file: **1) unzip q.zip 2) cd q**
- Make the installer executable and run it: **1) chmod +x install.sh 2) ./install.sh**
- Reload the shell: **bash**
- **Step 3: Authenticate Amazon Q**

Run the following command to log in: **q login**

This will authenticate your session and allow you to use Amazon Q CLI features, including chat, code generation, and project assistance.

Step 3: Prompt for a Racing Car Game with Amazon Q CLI

Here's a prompt you can use with Amazon Q CLI to build a simple Racing Car game:

Prompt:

Create a Python game using pygame where a player controls a car that must avoid incoming obstacles. The car should move left and right using arrow keys. Add a scoring system that increases over time as the player survives. End the game if the car collides with an obstacle. Use modular, well-commented code.

To run the game, you'll need to have pygame installed. If you don't have it yet, you can install it with:

pip install pygame

Then run the game with:

python ~/car_game/car_game.py

Did you know?

You can resume the last conversation from your current directory by launching with `q chat --resume`

creating: q/bin/

`help` all commands • `ctrl + j` new lines • `ctrl + s` fuzzy search

I'll create a Python game using pygame with a car that avoids obstacles. Let's break this down into modular

components with well-commented code, where a player controls a car that must avoid incoming obstacles. The car should move left and right using arrow keys. Add a scoring system that increases over time as the player survives. Efirst, let's create the main game file: obstacle. Use modular, well-commented code.

constants defined at the top for easy customization

To run the game, you'll need to have pygame installed. If you don't have it yet, you can install it with:

```
pip install pygame
```

Then run the game with:

```
python ~/car_game/car_game.py
```

I've created a Python car game using pygame where the player controls a car that must avoid incoming obstacles. The game has been saved to `/home/q/car_game/car_game.py` and is ready to run.

Here's a breakdown of the game's features:

- 1. Player Control:**
 - The car moves left and right using arrow keys
 - The car stays at the bottom of the screen
- 2. Obstacles:**
 - Random-sized obstacles appear from the top of the screen
 - Obstacles move downward at increasing speeds as the game progresses
- 3. Scoring System:**
 - Score increases over time as the player survives
 - Score is displayed in the top-left corner
- 4. Game Over Condition:**
 - Game ends if the car collides with any obstacle
 - A "GAME OVER" message appears with instructions to restart
- 5. Code Structure:**
 - Modular design with separate classes for Player, Obstacle, and Game
 - Well-commented code explaining each function and major section
 - Constants defined at the top for easy customization

To run the game, you'll need to have pygame installed. If you don't have it yet, you can install it with:

```
pip install pygame
```

Then run the game with:

```
python ~/car_game/car_game.py
```

The game features:

- Simple but effective visual design with colored rectangles
- Gradually increasing difficulty as obstacles speed up over time
- Restart functionality by pressing the space bar after game over
- Exit options via the Escape key or window close button

The code is organized into three main classes:

- 1. Player** - Handles the car's appearance, position, and movement
- 2. Obstacle** - Manages obstacle creation, appearance, and movement
- 3. Game** - Controls the main game loop, collision detection, and scoring

To install pygame, you'll need to run that command in your terminal. Let me help you with that:

Great! I've successfully installed pygame using the system package manager. The pygame package is now installed and ready to use.

To run the car game, you can use the following command:

```
bash
python3 ~/car_game/car_game.py
```

The game is now ready to play! Here's a summary of what the game includes:

- 1. Player Control:**
 - Use the left and right arrow keys to move your car
 - Avoid the red obstacles coming from the top of the screen
- 2. Game Features:**
 - Score increases as you survive longer
 - Obstacles appear randomly with different sizes
 - Game difficulty increases over time as obstacles move faster
 - Visual road markings and car details for better appearance
- 3. Game Over:**
 - The game ends when your car collides with an obstacle
 - Press SPACE to restart after game over
 - Press ESC to quit the game at any time

Note that the game requires a graphical display to run. If you're running this in a headless environment or over SSH without X forwarding, you might encounter errors related to the display.

Enjoy playing the car obstacle avoidance game!

```
> python3 /c
/clear                /context show --expand    /context rm          /context clear --global /context hooks rm      /context hooks enable-all /compact help
/context help          /context add                /context rm --global /context hooks help    /context hooks enable  /context hooks disable-all
/context show          /context add --global       /context clear       /context hooks add     /context hooks disable  /compact
> python3 ~/car_game/car_game.py
I'll help you run the car game. Let me execute that command for you:
```

• I will run the following shell command:

```
python3 ~/car_game/car_game.py
```

❏ **Purpose:** Running the car game

```
<frozen importlib_bootstrap>:488: RuntimeWarning: Your system is avx2 capable but pygame was not built with support for it. The performance of some of your blits could be adversely affected. Consider enabling compile time detection with
environment variables like PYGAME_DETECT_AVX2=1 if you are compiling without cross compilation.
```

```
pygame 2.5.2 (SDL 2.26.0; Python 3.12.3)
```

```
Hello from the pygame community. https://www.pygame.org/contribute.html
```

```
❏
• Completed in 209.744s
```

The game has started running! You should now see a game window with:

- A blue car at the bottom of the screen that you can control with the left and right arrow keys
- Red obstacles that will start falling from the top of the screen
- A score counter in the top-left corner that increases as you play

Game controls:

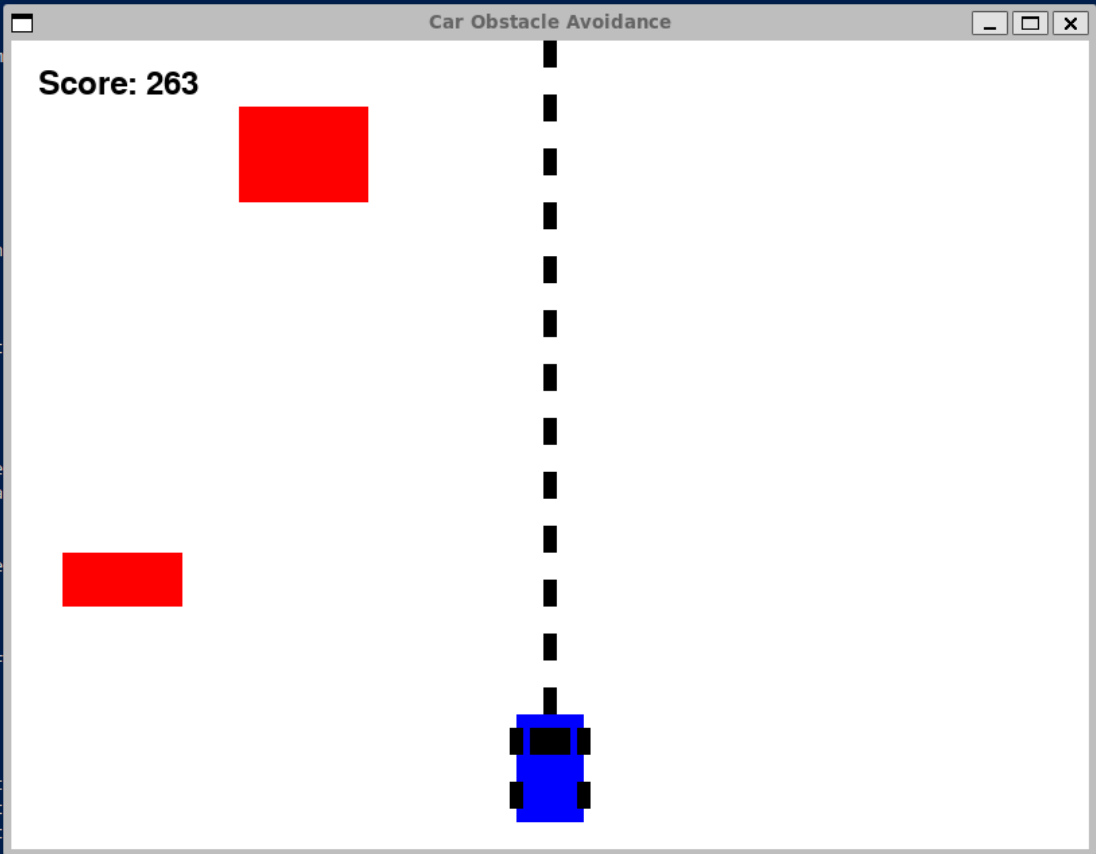
- Left Arrow: Move car left
- Right Arrow: Move car right
- ESC: Quit the game
- SPACE: Restart after game over

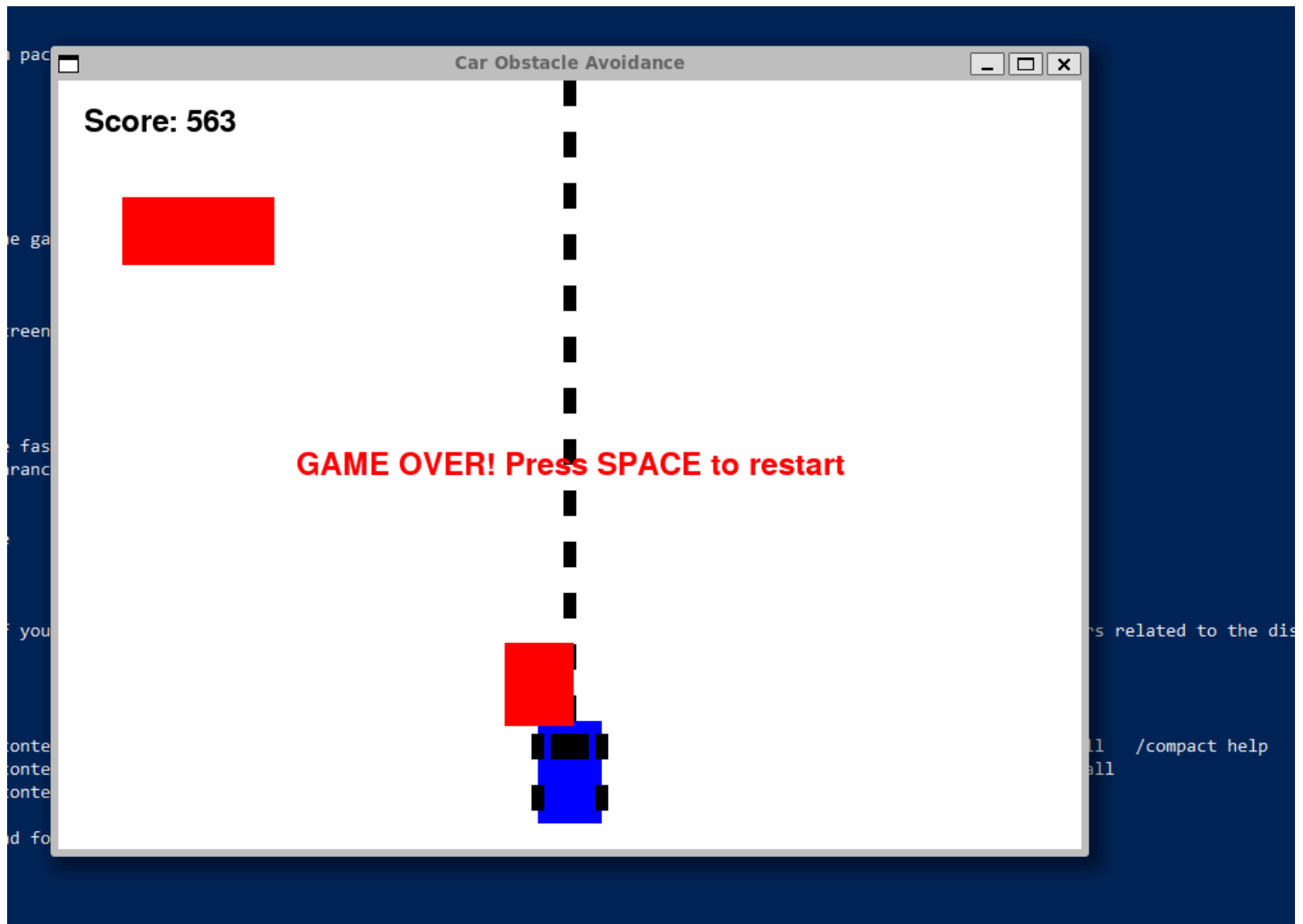
The objective is to avoid the red obstacles for as long as possible. As your score increases, the obstacles will move faster, making the game progressively more challenging.

If your car collides with an obstacle, the game will display a "GAME OVER" message, and you can press SPACE to restart.

Note: The warning message about AVX2 support is just informational and doesn't affect gameplay - it's just indicating that pygame could potentially run even faster with different compilation settings.

Enjoy the game!





Reference :

1) for WSL [🌐 Set up a WSL development environment](#)

2) Installing Amazon Q for the command line

[🌐 The essential guide to installing Amazon Q Developer CLI on Windows](#)

[🌐 Installing Amazon Q for command line - Amazon Q Developer](#)

[🌐 Installing with a zip file - Amazon Q Developer](#)