Pong and Shooter game.

Programming a pong and a shooter game with a little help from an LLM.

Exercise.

A pong and a shooter game in Python, with help from an LLM.

Sila. October 24th, 2024.

In this exercise: Work with an LLM to create a pong and a shooter game. Focus on the pong (shooter) game functionality. Notice: Writing clean code is obviously always important. But here you are asked to "play along" with the code you are given by the LLM. And only refactor the code, when you think the code given to you, from the LLM, becomes too "smelly". You can even ask the LLM to help you with refactoring.

I.e. After having worked on a Car game, it is now time to work on a Pong and a Shooter game.

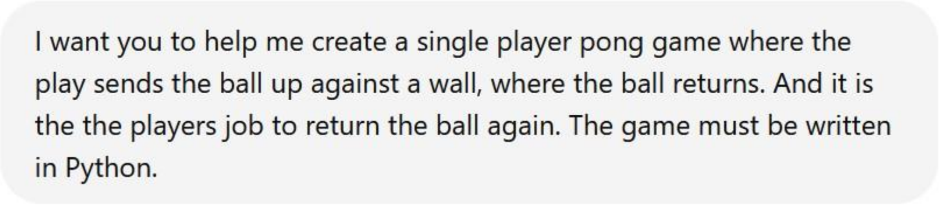
Exercise 1 and 2: We are going to create a pong game with the help of an LLM. Exercise 3: Next, we will create a shooter game. Exercise 4. Finally, we will discuss whether it was all really a little bit too easy.

So, here we go.

Exciting.

Exercise 1.

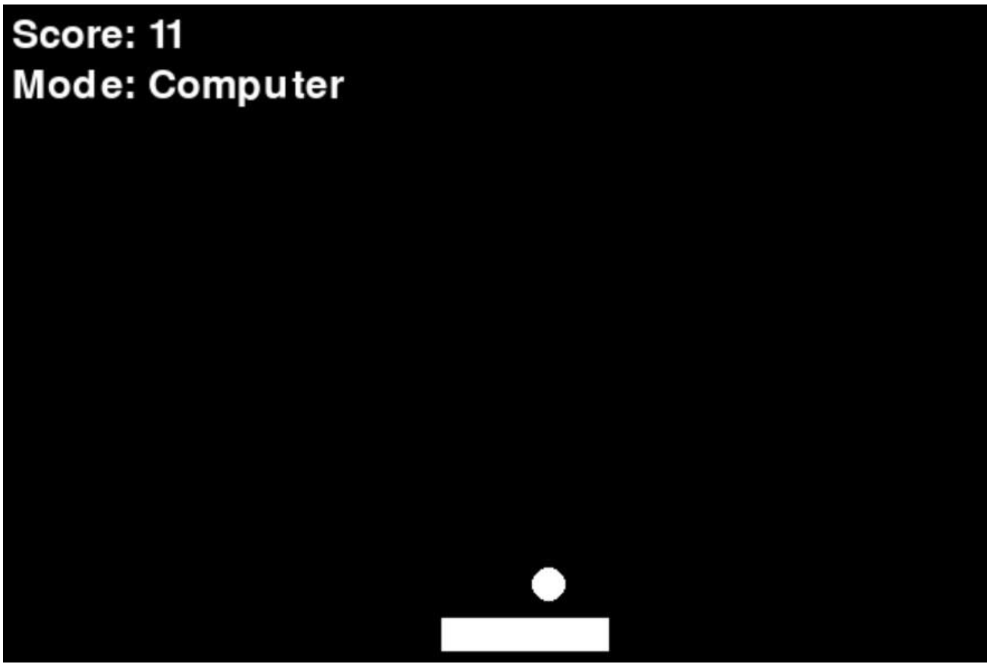
We begin chatting with ChatGPT (version ChatGPT o1-preview).



I want you to help me create a single player pong game where the play sends the ball up against a wall, where the ball returns. And it is the the players job to return the ball again. The game must be written in Python.

Apparently, that was an easy thing to do. So, ChatGPT returns with code that works out of the box.

Running the game, it looks like this:



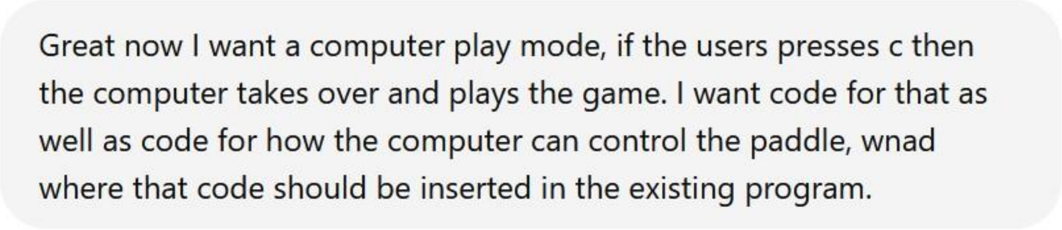
The code can be found in the file PongInitialVersion.py

Now, try this yourself, by creating a similar prompt that asks for Python Pong game.

Exercise 2.

Next, you should try to make a small change to the game.

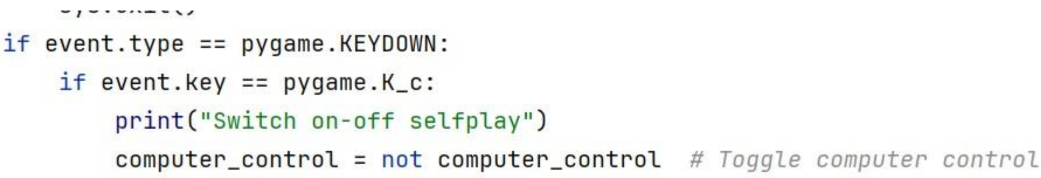
One possibility is to ask for a version with “self-play”, where the computer also controls the bat.



Great now I want a computer play mode, if the users presses c then the computer takes over and plays the game. I want code for that as well as code for how the computer can control the paddle, wnad where that code should be inserted in the existing program.

This almost worked right out of the box.

Except for a minor bug with one control.



```
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_c:
                print("Switch on-off selfplay")
                computer_control = not computer_control  # Toggle computer control
```

A little debugging found the problematic lines, where a small change in an if-statement made everything run smoothly.

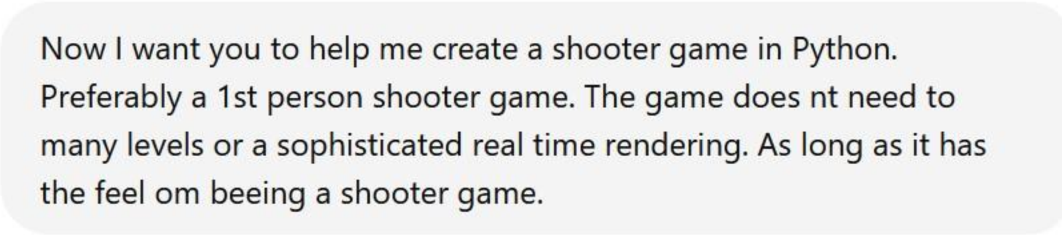
Try out the code in PongSelfPlay.py

And follow-up by asking for your own small change to game. Which you can then then debug, if necessary, to make it work properly.

Start my asking ChatGPT for you own small change to existing code.

Exercise 3.

After the small changes to Pong game (exercise 1 and 2), you can then move on to a shooter game.

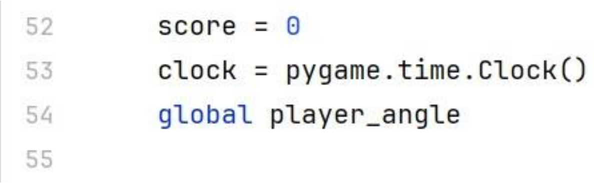


Now I want you to help me create a shooter game in Python. Preferably a 1st person shooter game. The game does nt need to many levels or a sophisticated real time rendering. As long as it has the feel om beeing a shooter game.

Again, ChatGPT quickly provides some code. Which unfortunately crashes.

Making one of the variables global seemed to fix the problem though.

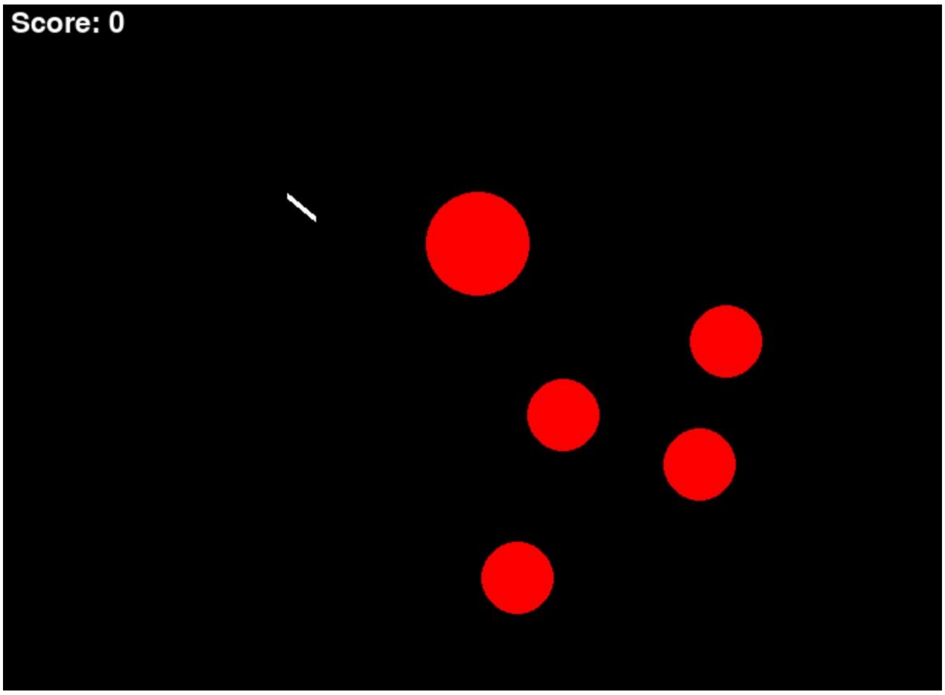
Line 54.



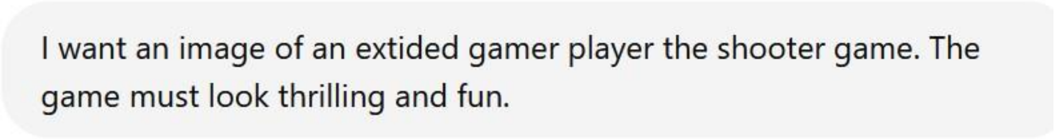
```
52        score = 0
53        clock = pygame.time.Clock()
54        global player_angle
55
```

Try out the code in the file Shooter.py

And continue by making some small changes to that code.



ChatGPT is also happy to give you an image of what it feels like playing that game …


I want an image of an extided gamer player the shooter game. The game must look thrilling and fun.

Well, well…

Exercise 4.

Discuss in groups of 2-3.

Finally, discuss whether it is all too easy to make small games like this?

Look at this study, that explores the impact of LLMs on human creativity.

https://techxplore.com/news/2024-10-explores-impact-llms-human-creativity.html

https://arxiv.org/abs/2410.03703

The authors argue that:

"However, there are growing concerns, which have not yet received sufficient attention, about the long-term impact of these tools on human creativity. We hypothesized that the repeated use of large language models (LLMs) might be impairing our ability to think creatively on our own, despite enhancing performance while the tool is in use—similar to the temporary boost provided by performance-enhancing steroids in sports."

Indeed:

“The researchers found that across both experiments, the use of GPT-4o had improved the performance of participants during the exposure phase. Interestingly, however, participants who did not initially use GPT-4o were found to outperform participants who had access to the model during the test phase.”

And:

“The researchers were also surprised to find that the previously reported homogenization effect (i.e., the reported tendency of LLM users to come up with less diverse ideas over time) persisted even after participants had stopped using GPT-4o.”

Discuss in the group:

Based on your experiments with building Car, Pong and Shooter games in this and earlier exercises does the conclusions in the (Harsh Kumar, Jonathan Vincentius, Ewan Jordan, Ashton Anderson) article ring true to you?