Name : Pinball

A normal pinball game with a twist:

instead of using the keyboard for the flipper inputs, my game would rely on object detection through a live video feed to control the flippers.

The game that I’m trying to replicate is the pinball game that came along with the old windows xp. While I haven’t found the exact pinball game online (in python), I’ve found several aspects of it as “mini-projects” such as a bouncing ball game or a projectile motion simulator.

I plan on having a main file that would integrate all the different elements of my game. I would store aspects such as the ball, flippers and the board in different files and import them to the main file.

I feel the trickiest part of my game would be the physics that goes behind making the ball move and the collision detection. I will be deriving all the equations myself for the ball as this is something that I’ve learnt as a part of my physics courses. For the collision detection, I’m relying on multiple online guides to tackle it.

I’d like to complete the core part of this game by TP2 - The physics that goes behind the ball, the flipper movements and collision detection.

I’m using GitHub to actively back my code up

By using GitHub’s desktop app, all the changes that I make to my code in VSCode are automatically picked up by GitHub and all I have to do is push the changes to the main file once I’m done.

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

I plan on using numpy and OpenCV for my TP.

Storyboard:

To play my game, a user would first hold 2 objects and assign the objects to the left and the right flipper. The user then gets to choose a level of difficulty. The game would detect when the chosen objects are visible and would move the paddles accordingly. The player plays the game till the ball is inside the board and they get 3 lives.

Sources:

* AI. “LEARN OPENCV in 3 HOURS with Python | Including 3x Example Projects.” *YouTube*, 25 Mar. 2020, www.youtube.com/watch?v=WQeoO7MI0Bs. Accessed 20 Apr. 2021.
* ‌“Gui Features in OpenCV — OpenCV-Python Tutorials 1 Documentation.” *Readthedocs.io*, 2013, opencv-python-tutroals.readthedocs.io/en/latest/py\_tutorials/py\_gui/py\_table\_of\_contents\_gui/py\_table\_of\_contents\_gui.html. Accessed 20 Apr. 2021.
* ‌Rajathithan Rajasekar. “OpenCV Series — 2— Configure VSCode for OpenCV Development in MacOS.” *Medium*, Medium, 7 May 2020, rajathithanrajasekar.medium.com/opencv-series-2-configure-vscode-for-opencv-development-in-macos-4a2a06e144fa. Accessed 20 Apr. 2021.
* ‌“Multiple Color Detection in Real-Time Using Python-OpenCV - GeeksforGeeks.” *GeeksforGeeks*, 24 Apr. 2020, www.geeksforgeeks.org/multiple-color-detection-in-real-time-using-python-opencv/. Accessed 20 Apr. 2021.
* ‌“Python Programming Tutorials.” *Pythonprogramming.net*, 2021, pythonprogramming.net/morphological-transformation-python-opencv-tutorial/. Accessed 20 Apr. 2021.
* ‌“15-112: Fundamentals of Programming.” *Cmu.edu*, 2021, www.cs.cmu.edu/~112/notes/notes-animations-part2.html. Accessed 20 Apr. 2021.
* ‌Canu, Sergio. “Detecting Colors (Hsv Color Space) - Opencv with Python - Pysource.” *Pysource*, 15 Feb. 2019, pysource.com/2019/02/15/detecting-colors-hsv-color-space-opencv-with-python/. Accessed 20 Apr. 2021.
* ‌“YaflaColor RGB - HSV Color Conversion.” *Archive.org*, 2012, web.archive.org/web/20130806191424/www.yafla.com/yaflacolor/ColorRGBHSL.html. Accessed 20 Apr. 2021.
* “Projectile Motion Formula.” *101 Computing*, 3 July 2014, www.101computing.net/projectile-motion-formula/. Accessed 26 Apr. 2021.
* ‌TokyoEdtech. “Python Bouncing Ball Simulator 2.” *YouTube*, 12 June 2018, www.youtube.com/watch?v=ibdICVK0W3Q. Accessed 26 Apr. 2021.
* “How to Check If a given Point Lies inside or Outside a Polygon? - GeeksforGeeks.” *GeeksforGeeks*, 11 July 2013, www.geeksforgeeks.org/how-to-check-if-a-given-point-lies-inside-a-polygon/#:~:text=1)%20Draw%20a%20horizontal%20line,true%2C%20then%20point%20lies%20outside.. Accessed 1 May 2021.
* ‌“How to Check If Two given Line Segments Intersect? - GeeksforGeeks.” *GeeksforGeeks*, 10 July 2013, www.geeksforgeeks.org/check-if-two-given-line-segments-intersect/. Accessed 1 May 2021.
* ‌“Intersection of a Line and a Circle.” *DoubleRoot.in*, Dec. 2020, doubleroot.in/lessons/circle/intersection-line-circle-1/#:~:text=To%20determine%20the%20position%20of,circle%20at%20two%20distinct%20points.. Accessed 1 May 2021.
* ‌“Check If a Line Touches or Intersects a Circle - GeeksforGeeks.” *GeeksforGeeks*, 17 Sept. 2017, www.geeksforgeeks.org/check-line-touches-intersects-circle/. Accessed 1 May 2021.
* ‌Batash. “Calculate Angle Change after Hitting a Tilted Wall.” *Stack Overflow*, 17 July 2017, stackoverflow.com/questions/45154176/calculate-angle-change-after-hitting-a-tilted-wall. Accessed 1 May 2021.
* ‌“Dot Product.” *Mathsisfun.com*, 2017, www.mathsisfun.com/algebra/vectors-dot-product.html. Accessed 1 May 2021.
* ‌“Distance between Point and Line | Brilliant Math & Science Wiki.” *Brilliant.org*, 2016, brilliant.org/wiki/distance-between-point-and-line/#:~:text=The%20formula%20for%20this%20one,%2B%20b%202%20%2B%20c%202%20.. Accessed 1 May 2021.