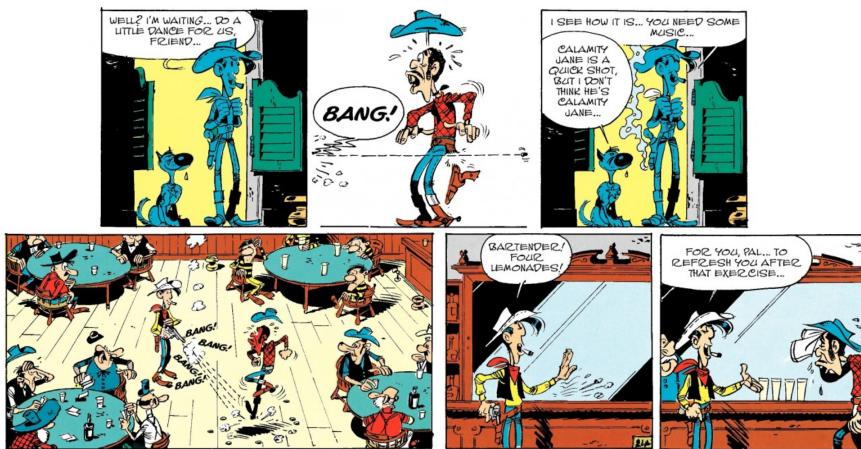


# BLG453E

## Homework-3

21.12.2020

Res. Asst. Yusuf H. Sahin  
sahinyu@itu.edu.tr



Lucky Luke : The Daltons Redeem Themselves, Morris & Goscinny

- You should write all your code in Python language.
- Cheating is highly discouraged. If you are planning to use different libraries or functions, please ask me about it.
- You should run the game in fullscreen mode.

### 1 - Part 1: Sobel Filter (10 pts.)

In this homework, we will make a 3D character dance according to the given shape inputs. I created a small game using Unity3D<sup>1</sup> game engine. You can download the Windows built of the game from <sup>2</sup> and Web built from <sup>3</sup>.

<sup>1</sup><https://unity3d.com/unity>

<sup>2</sup>[https://web.itu.edu.tr/sahinyu/saturdaynightfilter\\_pc.rar](https://web.itu.edu.tr/sahinyu/saturdaynightfilter_pc.rar)

<sup>3</sup>[https://web.itu.edu.tr/sahinyu/saturdaynightfilter\\_web.rar](https://web.itu.edu.tr/sahinyu/saturdaynightfilter_web.rar)

If you want to play the game in Firefox web browser, you can start the game by opening index.html. However, since it is a local file, you should reach "about:config" page on Firefox and change "privacy.file unique origin" to "false".

**Your final project will also contain some games which can be played using computer vision. Thus, this homework will be a good exercise before the project.**

The main interface of the game is given in Figure 1. We will be working on two songs: Vabank and Shame.



Figure 1: Main interface

If you click the "All Shapes" button, you will be directed to a page containing example shapes and their corresponding keyboard inputs as given in Figure 2.



Figure 2: "All Shapes" page.

For this homework, you will benefit from pyautogui library which is used to simulate mouse and keyboard interactions with Python. The example script given below first takes a screenshot of the game, then clicks random buttons.

```
1 import pyautogui
import time
```

```

3 time.sleep(5)
5 #In this 5 seconds you should switch to game screen to transfer the
   simulated keyboard inputs to the game.

7 myScreenshot = pyautogui.screenshot()
myScreenshot.save('test.png')
9 # An example screenshot is obtained. We will work on screenshots like this
   for this homework.

11 pyautogui.keyDown('shift')

13 pyautogui.keyDown('w')
time.sleep(1)
15 pyautogui.keyUp('w')

17 pyautogui.keyUp('shift')

19 #pyautogui.keyUp and pyautogui.keyDown functions are used to simulate
   holding a button. For simple presses, pyautogui.press can be used.

```

Write a script which takes a screenshot from the All Shapes page, applies Sobel filter and saves the result.

## 2 - Part 2: Canny Edge Detector (10 pts.)

Use Canny Edge detection<sup>4</sup> algorithm to obtain an edge map of the All Shapes screenshot. You can use OpenCV's default detector. Try to find optimum parameters. Also you can use "cv2.findContours" function to find the contours which will ease your job in the last part.

## 3 - Part 3: Minimum Eigenvalue Corner Detector (30 pts.)

Implement Minimum Eigenvalue corner detector on game screenshots to obtain the corners of the shapes given in All Shapes page.

## 4 - Part 4: Dance Game (50 pts.)

Clicking "Vabank" or "Shame" buttons will direct you to a game page as given in Figure 3. You can press ESC to go back to the main screen.

The game pages consist of our dancing 3D model and a 2D canvas at the bottom showing the shapes on a timeline. According to the dance, some shapes are flown from right to left. The player should press the corresponding button on time to obtain a point. Using

---

<sup>4</sup>Canny, John, "A Computational Approach to Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, No. 6, 1986, pp. 679-698

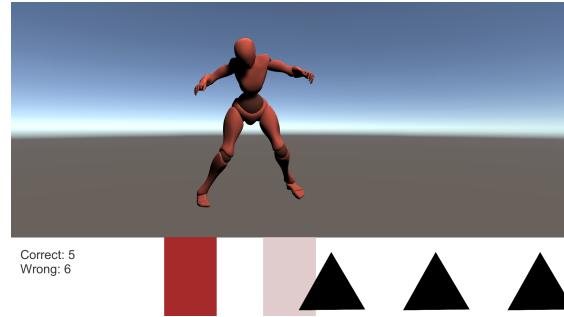


Figure 3: A game page

your findings in previous parts, write a script which beats the game. For this part, you are free to use OpenCV equivalents of the mentioned functions.