

DAILY ONLINE ACTIVITIES SUMMARY

Date:	03-06-2020	Name:	SAFNAAZ
Sem & Sec	VIII Semester & B Section	USN:	4AL16CS081
Online Test Summary			
Subject	-		
Max. Marks	-	Score	-
Certification Course Summary			
Course	Game development using Pygames		
Certificate Provider	Ui Path	Duration	3 Hours
Coding Challenges			
Problem Statement: Find an array of positive integers for the inversion count of array.			
Status: COMPLETED			
Uploaded the report in Github		YES	
If yes Repository name		Safnaazsheikh	
Uploaded the report in slack		YES	

Online Test Details:

Certification Course Details:

The screenshot displays a web-based learning interface. At the top, a navigation bar includes a 'Learn on the Go' button and an 'Install App' button. Below this, a breadcrumb trail shows the path: 'Course' > 'Game development using PyGame' > '003 Event Loop PyGame'. The main content area features a code editor with the following Python code:

```
13 pygame.display.flip()
14
15 while True:
16     # main game loop
17     for event in pygame.event.get():
18         print(event)
19         if event.type == QUIT:
20             pygame.quit()
21             sys.exit()
22     pygame.display.update()
```

Below the code editor, there are tabs for 'Notes', 'Summary', 'Activity', and 'Discussion'. The 'Notes' tab is active, showing a text area with the placeholder 'Add your notes ...' and a green 'Submit' button. To the right of the notes area is a 'Table of content' sidebar. This sidebar shows a progress bar for the 'Beginner Module' at 30%. The table of contents lists the following modules:

- 001 Introduction to PyGame
- 002 Intro Pygame Code
- 003 Event Loop Pygame
- 004 Even Loop DeepDive
- 005 Shapes Pygame
- 006 Slither Snake Game Part 1
- 007 Slither Snake Game Part 2
- 008 Slither Snake Game Part 3
- 009 Pong Part1
- 010 Pong Part2

At the bottom of the sidebar, there is a 'Get Certified' button.

Learn on the Go

Install App

CourseGame development using PyGame004 Even Loop DeepDive

235Rank: 10%

Copy icon

004 Even Loop DeepDive

```
17 def events:
18     if event.type == QUIT:
19         quit()
20     elif event.type == KEYDOWN:
21         if event.key == K_ESCAPE:
22             quit()
23         elif event.key == K_LEFT:
24             data.x -= 5
25             print("Move Rect left")
26         elif event.key == K_RIGHT:
27             data.x += 5
28             print("Move Rect right")
29         elif event.key == K_SPACE:
30             print(event.key)
31 screen.fill((0,0,0))
32 pygame.display.flip()
33 pygame.time.Clock().tick(60)
```

YouTube

NotesSummaryActivityDiscussion

Add your notes ...

Submit

Table of content

Beginner Module

40%

001 Introduction to PyGame

002 Intro Pygame Code

003 Event Loop Pygame

004 Even Loop DeepDive

005 Shapes Pygame

006 Slither Snake Game Part 1

007 Slither Snake Game Part 2

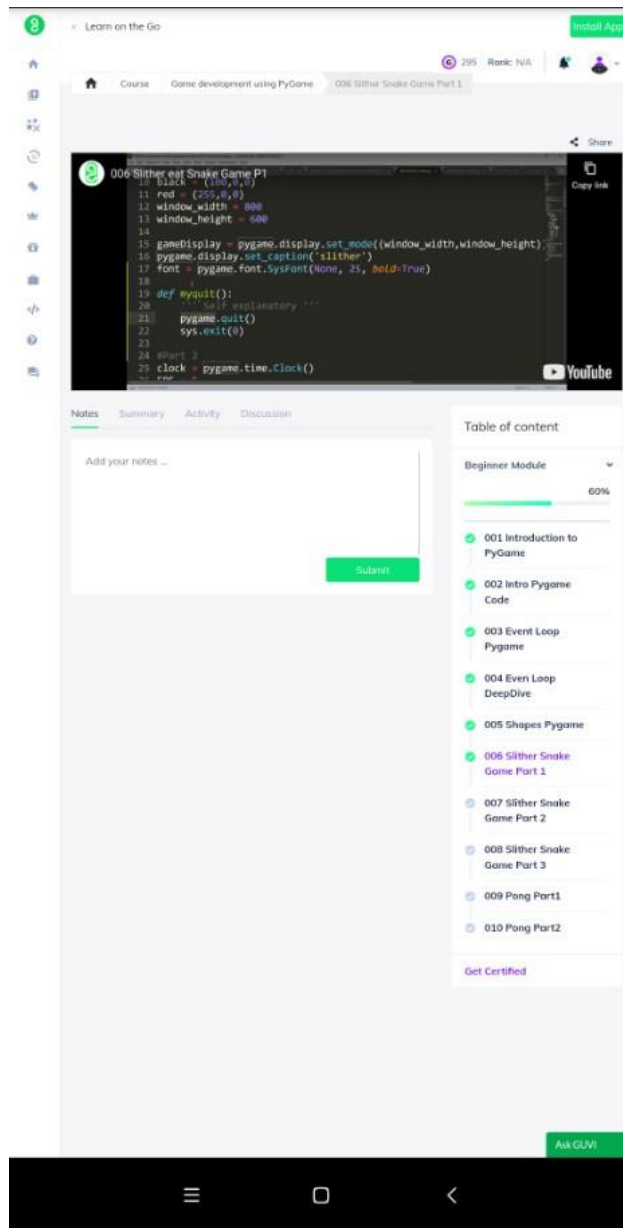
008 Slither Snake Game Part 3

009 Pong Part1

010 Pong Part2

Get Certified

Ask GUY!

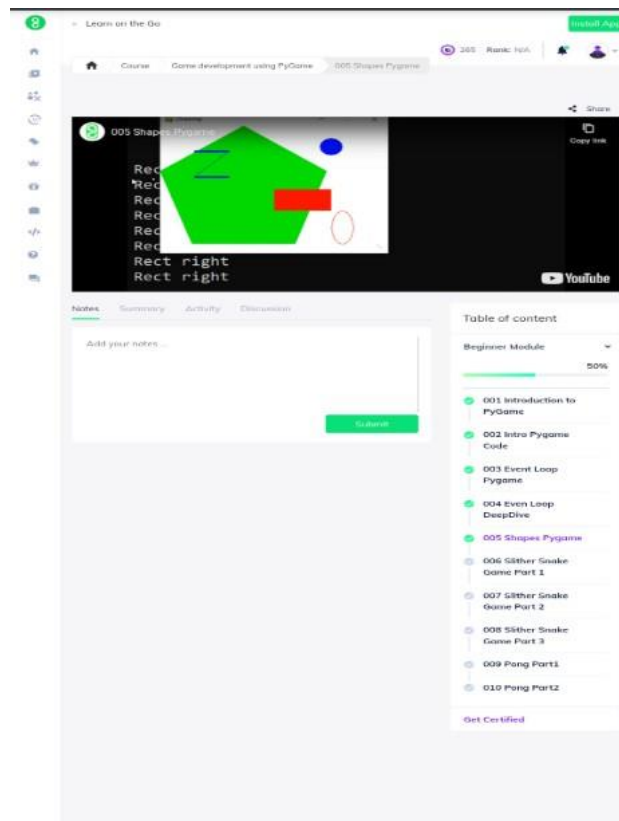


Pygame uses the Simple Direct Media Layer (SDL) library, [a] with the intention of allowing real-time computer game development without the low-level mechanics of the C programming language and its derivatives. This is based on the assumption that the most expensive functions inside games can be abstracted from the game logic, making it possible to use a high-level programming language, such as Python, to structure the game.

Other features that SDL doesn't have include vector math, collision detection, 2d sprite scene graph management, MIDI support, camera, pixel-array manipulation,

transformations, filtering, advanced freetype font support, and drawing.

Applications using pygame can run on Android phones and tablets with the use of pygameSubset for Android. Sound, vibration, keyboard, and accelerometer are supported on Android.



Coding challenges online details :

```
#include<stdio.h>
```

```
int getInvCount(int arr[], int n)
{
    int inv_count = 0;
    for (int i = 0; i < n - 1; i++)
        for (int j = i + 1; j < n; j++)
```

```
        if (arr[i] > arr[j])
            inv_count++;

    return inv_count;
}

int main(int argv, char** args)
{
    int arr[] = { 2,4,1,3,5 };
    int n = sizeof(arr) / sizeof(arr[0]);
    printf(" Number of inversions are %d \n", getInvCount(arr, n));
    return 0;
}
```