Sudoku game using Python

A PROJECT REPORT

Scheduled by

A. Muhil Raghavandhar

In Partial fulfillment for the award of the degree of in

CSC

Computer software college

An IOS 9001:2000 certified institution

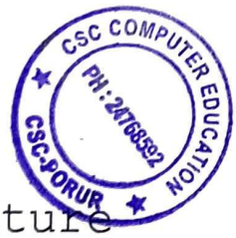
COMPUTER SOFTWARE ARE COLLEGE

PORUR

CHENNAI-600116

BONAFIDE CERTIFICATE

Certified that this project report titled as "**Sudoku game using python**" is the bonafide work of A. Muhil Raghavandhar who carried out the project work under my supervision
MS. J.VIJAYALASHMI



Signature

Mr.A.RAMAKRISHNAN

HEAD OF THE CSC

PORUR

Signature

Ms.J.VIJAYALAKSMI

PROJECT-GUIDE/TRAINER

PORUR

Acknowledgment

At the outset thank god for having brought this throughout as in computer this project. We convey our thanks to A. RAMAKRISHNAN(MANAGER), CSC Computer education, porur branch for giving an opportunity to embark on this project successfully.

With almost pleasure we here by express our faculty

MS.J.VIJAYALAKSHMI

Done by

A. Muhil Raghavandhar

EXPLAINATION ABOUT PYTHON

Python is an **object-oriented, high-level programming language** with integrated dynamic semantics primarily for web and app development. It is extremely attractive in the field of Rapid Application Development because it offers dynamic typing and dynamic binding options.

TOP 7PRACTICAL APPLICATION USE IN PYTHON

Operating Systems

The robust standard library of Python makes it perfect for building entire operating systems. The object-oriented design of the language ensures large projects are easily managed. Python is compatible with most operating systems and can be easily used to build native applications for Windows and Mac computers.

Web and Internet Development

Python offers several choices for complex web development projects. HTML and JavaScript are the main languages used to build the front end of an application. But Python-based web frameworks like Django, Pyramid, and Flask are used to handle backend or server-side functionality of sites and services like Spotify, Reddit, and Mozilla. Giant platforms like Google and YouTube depend largely on Python for critical infrastructure.

The standard library of Python also supports many Internet protocols like HTML and XML, JSON, Email processing, FTP, and IMAP.

Game Development

Just like for web development, Python offers an array of tools and libraries for game development. Would you believe, Battlefield 2 – one of the most popular shooting games of the early 2000s, was developed with the use of Python.

Python's 2D and 3D game development libraries are PyGame, Pycap, Panda#D, Construct, PySoy, and PyOpenGL.

Python has been used to develop popular games, including Sims 4, World of Tanks, Eve Online, Mount & Blade, DokiDoki Literature Club, and Disney's Toontown Online, to name a few.

Scientific and Numeric Computing

The Python ecosystem offers numerous tools and libraries that help scientists and researchers in scientific and numeric computing.

- SciPy is a set of packages for mathematics, science, and engineering
- Pandas is a library used for data analysis and modeling

- IPython is a strong interactive shell that provides hassle-free editing and recording of a work session and aid in visualizing and parallel computing.
- FreeCAD and Abaqus are real-life numerical and scientific applications built with Python

Artificial Intelligence and Machine Learning

The hottest buzzwords of the decade – Artificial Intelligence (AI) and Machine Learning are mostly about algorithms, code, and logic. Python, along with a few other programming languages, is increasingly being used for developing AI and ML-powered solutions. The scope and power of Python, along with its stability and security, make it ideal for running AI and ML systems.

Some important libraries for the job are:

- Scikit-Learn – for building various machine learning models
- SciPy – for scientific and technical computing
- TensorFlow – for state-of-the-art neural networks
- Keras – for artificial neural networks
- Pandas – for data analysis and manipulation

Desktop GUI

Python is an excellent choice for desktop GUI (Graphical User Interface) programming. The language offers numerous options for developers to build a fully functional GUI. The comprehensive syntax and modular programming approach of the Python framework help create a super-fast and responsive GUI.

Some prominent applications of Python tools for GUI development are PyQt, Tkinter, wxWidgets, Python GTK+, and Kivy. Standard applications like Dropbox and BitTorrent are primarily written in Python.

Business Applications

Applications of Python also include building ERP and ecommerce systems. Business applications are different from typical consumer software because they offer a set of specific features instead of a variety of features. Besides, they target a very tight-knit user group, usually an organization.

Python is perfect for delivering best-performance custom solutions for business applications as well as consumer applications.

Odoo is a well-rounded management software that offers numerous business applications that constitute a complete set of enterprise management applications

Tryton is a three-tier high-level application platform designed for general purposes

TYPES OF OPERATORS USED IN PYTHON

1. Arithmetic Operators

Operators	Name	Example
+	Addition	X+y
-	Subtraction	x-y
*	Multiplication	X*y
/	Division	x/y
%	Modulus	x%y
**	Exponentiation	x**y
//	Floor division	x//y

2. Python Assignment Operators

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3

**=	x **= 3	x = x ** 3
&=	x &= 3	x = x & 3
|=	x |= 3	x = x | 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3

3. Comparison Operators

Operator	Name	Example
==	Equal	x == y
!=	Not equal	x != y
>	Greater than	x > y

<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

4. Logical Operators

operators	description	Example
And	Returns True if both statements are true	x < 5 and x < 10
Or	Returns True if one of the statements is true	x < 5 or x < 4
not	Reverse the result, returns False if the result is true	not(x<5 and x<10)

5. Identity Operators

Operators	Description	Example
Is	Return True if both variables are same object	X is y
Is not	Return True if both variables are not same object	X is not y

6. Bitwise Operators

Operators	Name	Description
&	And	Sets each bit to 1 if both bits are 1
|	Or	Sets each bit to 1 if one of two bits are 1
^	Xor	Sets each bit to 1 if only one of two bits are 1
~	Not	Inverts all the bits
<<	Zero fill with left shift	Shift left by pushing zeros in form the right and let the leftmost bit falloff
>>	Signed right shift	Shift right by pushing copies in form the

		right and let the leftmost bit falloff, and let the rightmost bit fall off

Program:

```
M = 9

def puzzle(a):
    for i in range(M):
        for j in range(M):
            print(a[i][j],end = "
")
        print()
defsolve(grid, row, col, num):
    for x in range(9):
        if grid[row][x] == num:
            return False

    for x in range(9):
        if grid[x][col] == num:
            return False


    startRow = row - row % 3
    startCol = col - col % 3
```

```
    for i in range(3):
        for j in range(3):
            if grid[i + startRow][j
+ startCol] == num:
                return False
    return True

defSuduko(grid, row, col):

    if (row == M - 1 and col == M):
        return True
    if col == M:
        row += 1
        col = 0
    if grid[row][col] > 0:
        return Suduko(grid, row,
col + 1)
    for num in range(1, M + 1, 1):

        if solve(grid, row, col,
num):

            grid[row][col] = num
            if Suduko(grid, row,
col + 1):
                return True
```

```
            grid[row][col] = 0
        return False

'''0 means the cells where no value
is assigned'''
grid = [[2, 5, 0, 0, 3, 0, 9, 0,
1],
            [0, 1, 0, 0, 0, 4, 0, 0,
0],
        [4, 0, 7, 0, 0, 0, 2, 0, 8],
        [0, 0, 5, 2, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 9, 8, 1, 0, 0],
        [0, 4, 0, 0, 0, 3, 0, 0, 0],
        [0, 0, 0, 3, 6, 0, 0, 7, 2],
        [0, 7, 0, 0, 0, 0, 0, 0, 3],
        [9, 0, 3, 0, 0, 0, 6, 0, 4]]

if (Suduko(grid, 0, 0)):
    puzzle(grid)
else:
    print("Solution does not
exist:(")
```

Output:

py

2 5 8 7 3 6 9 4 1

6 1 9 8 2 4 3 5 7

4 3 7 9 1 5 2 6 8

3 9 5 2 7 1 4 8 6

7 6 2 4 9 8 1 3 5

8 4 1 6 5 3 7 2 9

1 8 4 3 6 9 5 7 2

5 7 6 1 4 2 8 9 3

9 2 3 5 8 7 6 1 4