

DICE GAME

A PROJECT REPORT

Submitted by

HEMANG SHARMA [Reg No: RA1811003030054]

Under the guidance of

Mr. Nishant Anand

(Assistant Professor, Department of Computer Science & Engineering)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY



SRM INSTITUTE OF SCIENCE & TECHNOLOGY, NCR CAMPUS

May 2022

SRM INSTITUTE OF SCIENCE & TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled “**DICE GAME**” is the bonafide work of “**HEMANG SHARMA [Reg No: RA1811003030054]**, , ”, who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Mr. Nishant Anand
GUIDE
Assistant Professor
Dept. of Computer Science & Engineering

Signature of the Internal Examiner

SIGNATURE

Dr. R. P. Mahapatra
HEAD OF THE DEPARTMENT
Dept. of Computer Science & Engineering

Signature of the External Examiner

ABSTRACT

The proposed Dice Game, is a standalone multiplayer game created using MATLAB App. Each player will roll the dice and player having bigger number in each roll will get a chance to perform action of choice (heal action or attack action). GUI and animations are created using MATLAB and machine learning model is built using Python programming language. The objective of machine learning model is to predict the winning probability of each layer. The variables are exchange between MATLAB and python language with the help of Application Programming Interface (API), which is built using Flask (Python library). The model will be using Random Forest Regressor to predict the outcome based on the variables passed. Due to the nature of the game, the graphics will be in 2D and offer a layout and feels like that of a board game.

ACKNOWLEDGEMENTS

I would like to convey my heartfelt thanks to my guide, Mr. Nishant Anand for his invaluable advice, constant encouragement, prompt assistance and provision of an ideal research environment. Throughout the project, despite his hectic schedule, he has been a cordial supporter of my efforts to complete this re-search task.

Hemang Sharma [RA1811003030054]

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	vii
LIST OF FIGURES	viii
ABBREVIATIONS	ix
1 INTRODUCTION	1
2 LITERATURE SURVEY	4
3 SYSTEM ANALYSIS	7
3.1 Supervised Learning problem	7
3.2 Model specific details	8
3.2.1 MIN-MAX Function	8
4 SYSTEM DESIGN	9
4.1 Architecture	9
4.2 The GUI	10
4.3 Random Forest Regression Model	11
5 EXPERIMENT RESULTS	13
5.1 Code	13
5.1.1 Machine Learning (M.L) Model Code	13

5.1.2	M.L Model Code Workflow	14
5.1.3	MATLAB Code	15
5.1.4	MATLAB Code Explanation	50
5.2	Output	50
5.2.1	Graphical User Interface (GUI) Result	50
6	CONCLUSION	52
7	FUTURE ENHANCEMENTS	53

List of Tables

1.1 Comparison between different ML Algorithms	3
--	---

LIST OF FIGURES

1.1	M.L Classification	2
3.1	Min-Max Formula	8
3.2	Re-scaled Min-Max Formula	8
3.3	Mean-Normalized Formula	8
4.1	Architecture of the Game	10
4.2	Random Forest Regressor (RFR) 1	11
4.3	Random Forest Regression 2	12
5.1	Game-play 1	50
5.2	Game-play 2	51
5.3	Battle Log	51

ABBREVIATIONS

GUI Graphical User Interface

RFR Random Forest Regressor

RF Random Forest

M.L Machine Learning

SL Supervised Learning

A.I Artificial Intelligent

CART Classification and Regression tree

CHAPTER 1

INTRODUCTION

Many sports activities take place every day and everywhere. Sports Prediction is making a difference to some people, websites, organizations and others across the globe. Predicting the outcome/outcomes of the match has become a significant fad among sports fans, especially in big league games like football and basketball. In many sports, commentators make comments on who will win or lose or whether the sport could be a draw. People are very curious lately to know the consequences in advance. Thus, Sports Prediction was born.

M.L is an application of Artificial Intelligent (A.I) that provides structures with the ability to regularly study and improvise from exams without being explicitly programmed. In other words, to learn and improvise from past experiences without being explicitly programmed. Event-focused M.L of bots will access the information and use it to discover for them selves. Predicting sports outcomes is often thought of as a classification problem, with predicting a win, loss or tie.

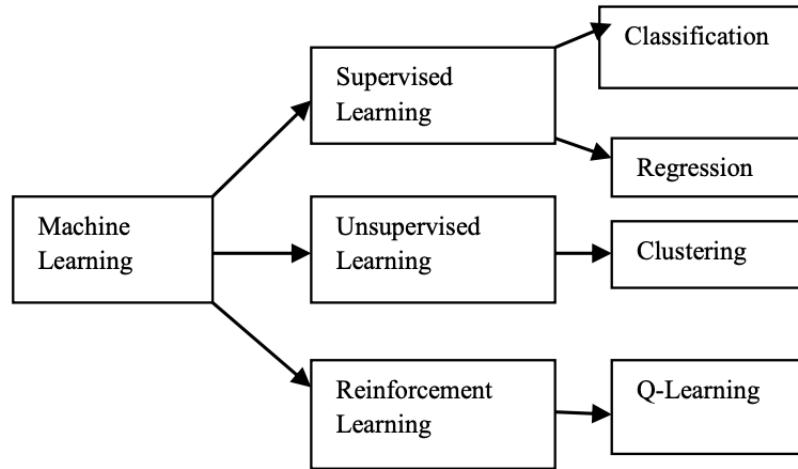


Figure 1.1: M.L Classification

Machine learning is classified into 3 kinds which are unsupervised, supervised and reinforcement learning.

Supervised learning allows models to predict future outcomes after they have been trained based on past data. To train the model, an input and output group is given first. It is further classified in classification and regression.

Unsupervised learning can be a type of M.L technique for finding patterns of information. The information provided for the unsupervised rule appears to be unlabelled, implying that only input variables (X) are provided without corresponding output variables. In unsupervised learning, algorithms are left to their own devices to find interesting patterns of information. It is more classified in clustering.

Reinforcement learning allows machines and software package agents to automatically check for perfect behavior in a particular context, to optimize its performance.

I have created a GUI game using MATLAB. The goal of the game is the player having higher number on the dice will carry out action of choice. The game can be played on same device as well as on separate machines using Thingspeak over the internet. Furthermore, I have created and implemented a M.L model to predict the game's outcome as the game progress. The machine learning model will be taking each dice roll as an input, process it and predict the future roll. The output will be probability (between 0 and 1).

The relaxation of the paper is prepared as follows:

In Chapter two, a short evaluate of the literature is presented. Chapter four covers methodology. The consequences and their dialogue are depicted in Chapter five and Chapter six, observed with the aid of using the realization and guidelines for destiny research in Chapter seven.

Algorithm	Training Speed	Prediction Speed
KNN	Quick	Fast
Linear Regression	Spontaneous	Quick
Logical Regression	Spontaneous	Fast
Naive Bayes	Slow	Quick
Decision Trees	Quick	Quicker
Random Forest	Spontaneous	Spontaneous

Table 1.1: Comparison between different ML Algorithms

CHAPTER 2

LITERATURE SURVEY

1) MachineLearning-as-a-Service for Consumer Electronics Fault Diagnosis: A Comparison between MATLAB and Azure M.L Prist et al. (2020)

The authors draws a comparison between M.L algorithm performances between Azure by Microsoft and MATLAB. MATLAB out shined Azure by Microsoft in terms of accuracy but the authors suggested that Microsoft Azure was more practical.

2) MATLAB vs Open-CV: A Comparative study of different machine learning algorithms Elsayed and Yousef (2019)

The paper revolves around likeness using twenty different data-sets between the speed of MATLAB and Open-CV for some M.L algos. The paper resulted that MATLAB is advantageous in developing and data presentation. Open-CV had speed factor.

3) DataCenter Predictions using MATLAB: M.L Toolbox Levy et al. (2019)

The paper talks about predictive model using MATLAB M.L mechanism for different M.L algorithms. Predictive analytic qualifies evidence-based decision making in the case of uncertainty and risk. It also appraises the areas of improvement in Data Center.

4) Application of Game-Based Learning in MATLAB Perutka and Vy-mazal (2021)

This paper focuses on game-based learning as a modern method of education, which is used by many experts from practice and confirms the benefits of its use in teaching. Furthermore, practical part deals with the design of the original game, designed primarily to teach MATLAB. The game was created using object-oriented programming in MATLAB.

5) Study and Analysis of Decision Tree Based Classification Algo-rithms Patel and Prajapati (2018)

This paper talks about three different algorithms of Decision Tree. These algorithms includes which ID3, C4.5 and Classification and Regression tree (CART).

6) Cricket Team Prediction using M.L Techniques Patil et al. (2020)

In this paper, Random Forest (RF) algo and decision tree classifiers are used to construct the prediction models. The conclusion drawn is RF classifier is well grounded for the proposed problem.

7) New Prospective on MultipleDiceRollingGame and its statistical implications Claver et al. (2017)

This paper uses a version of the ChapmanKolmogorov equations. This version is used to create a state transition of the probability mass function for each side of dice. This function creates the state during the game-play.

8) Using Machine Learning to Interpret Dice Rolls Campbell et al.

(2021)

The authors developed a M.L and computer vision blend for the analysing the dice rolls. When merged with an automated dice roller, it would ease the study of dice unfairness.

9) A Survey on Sports Prediction using Machine Learning Kumar and Gandhi (2019)

The paper discusses various factors that impact an outcome of a sporting event and what techniques/methods we can use to predict the outcome. At the end of the paper, there is a comparison of between all methods used for Sports prediction.

10) Machine Learning: A Review on Binary-Classification Kumari and Srivastava (2017)

Binary-classification is the process of categorising given data on the basis of predefined classes. This research amalgamates the various approaches used for binary classification. Also this paper discusses various application of Binary-Classification method.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Supervised Learning problem

Supervised Learning (SL) is a method to develop A.I, wherein an algorithm is trained on labeled input data that has been. The model is trained till it could come across the underlying styles and relationships among the enter information and the output labels, allowing it to yield correct labeling consequences of provided with new information.

SL is ideal for classification and regression problems, consisting of figuring out what class an information article belongs to or predicting the quantity of income for a given destiny date. SL is primarily based on information. During its training phase, the machine is fed with categorized information sets, which educate the machine what output is associated with every unique enter value. The skilled version is then provided with test data: This is information has been categorized and labeled, however the labels have not been revealed to the algorithm. The aim of testing data is to analysis how accurately the algorithm will carry out on unlabeled information.

Ground truth values are known, making this a SL problem. This is a RFR problem. Each output label is independent and acts as an independent binary classification problem.

3.2 Model specific details

3.2.1 MIN-MAX Function

Re-scaling

Re-scaling also known as min-max scaling. Re-scaling is a method in which the range of features are adjusted between the range of either 0 to 1 or -1 to 1. The formula for re-scaling of [0, 1] is given as:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Figure 3.1: Min-Max Formula

(x is original value and x' is normalized value)

To re-scale a range between the set of values $[a, b]$, the formula becomes:

$$x' = a + \frac{(x - \min(x))(b - a)}{\max(x) - \min(x)}$$

Figure 3.2: Re-scaled Min-Max Formula

Mean-Normalization

$$x' = \frac{x - \text{average}(x)}{\max(x) - \min(x)}$$

Figure 3.3: Mean-Normalized Formula

CHAPTER 4

SYSTEM DESIGN

4.1 Architecture

The MATLAB game will produce three variables for each player. These variables will store the values of health, dice roll and mode of each player. These variables are then passed to the python M.L model with the help of FLASK.

The model will process the incoming variables and will predict the winning probability for the players. MATLAB will adjust the prediction bar according to the prediction by the model. 4.1 shows the working of the game and how variables are passed.

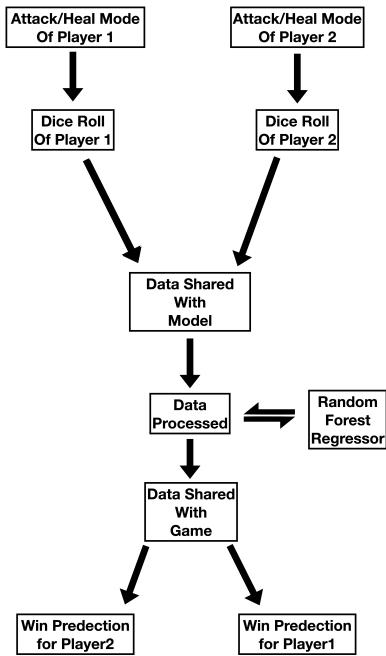


Figure 4.1: Architecture of the Game

4.2 The GUI

MATLAB stands for "MATrix LABoratory". It is a multi-paradigm programming language evolved with the aid of using MathWorks. MATLAB permits matrix manipulations, plotting of data, implementation of algorithms, and combining packages written in other programming languages.

Although MATLAB is supposed in most cases for numeric computing, but due to GUI driven MATLAB package gets the right of entry to symbolic computing abilities. Simulink, provides graphical multi-area simulation. The GUI and animations are created using MATLAB.

4.3 Random Forest Regression Model

RFR is SL set of rules that makes use of ensemble gaining knowledge of approach for regression. Ensemble learning is a way that mixes predictions from a couple of M.L algo. to make a greater correct prediction than a single model.

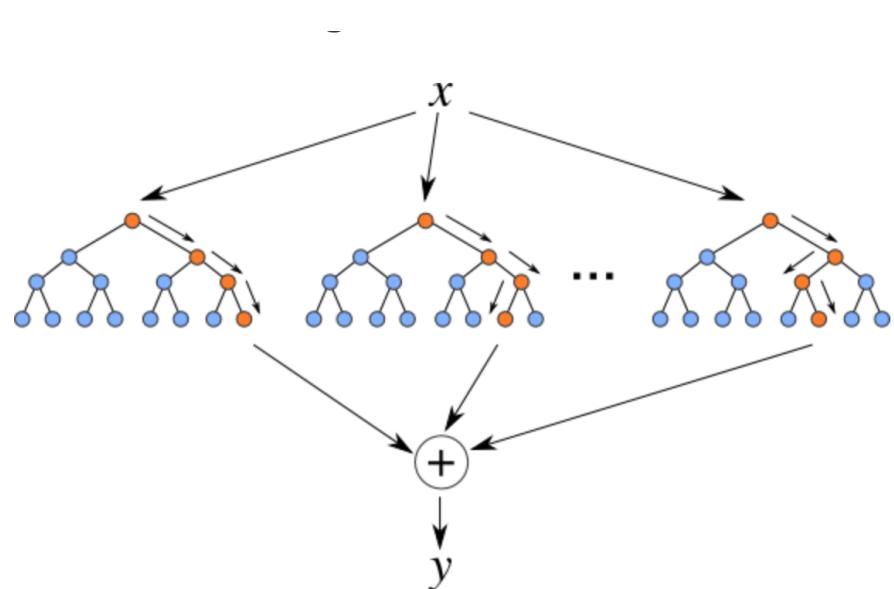


Figure 4.2: RFR 1

Pseudo code for Random Forest:

1. Pick at random k data points from the training set.
2. A decision tree is created.
3. Select N number of trees to rerun steps i and ii.
4. For a brand new information factor, make every N-tree tree expect the price of y for the information factor in query and assign the brand new information factor to the average throughout all the expected y values.

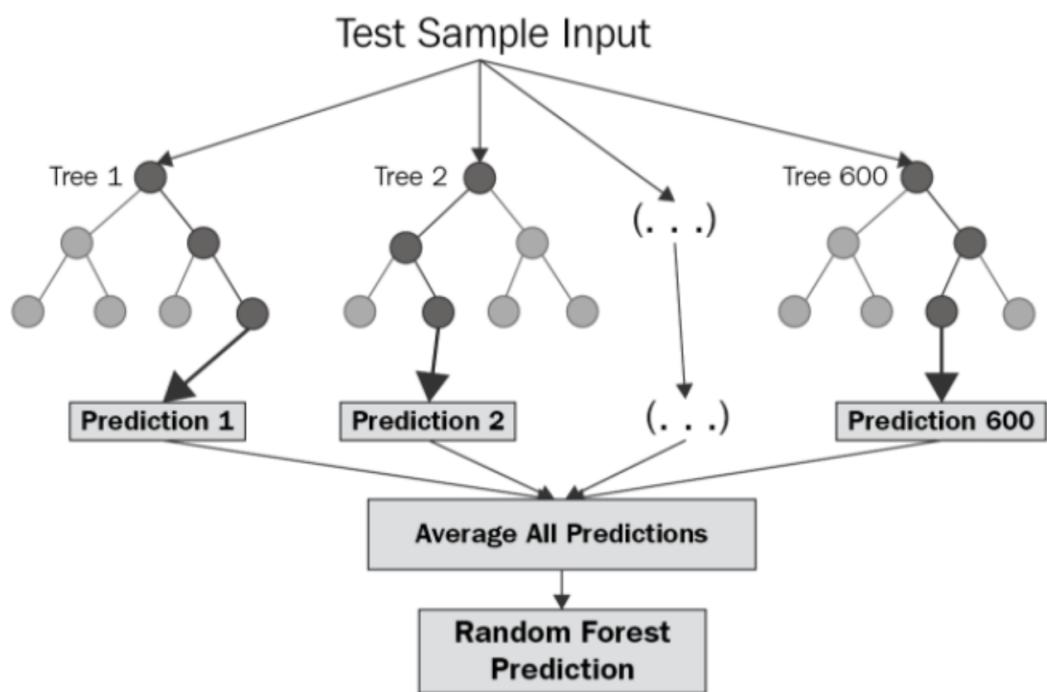


Figure 4.3: Random Forest Regression 2

A RFR model usually carry out great on numerous problems, which including features with non-linear relationships.

CHAPTER 5

EXPERIMENT RESULTS

5.1 Code

5.1.1 M.L Model Code

Model compilation code:

```
from flask import Flask, request, jsonify
import numpy as np
import random
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
import pandas as pd
import time

# complete training code

start = time.time()

df = pd.read_csv("test2.csv")
df_new=df.dropna()
X, Y = df_new[[list(df_new)[:2]+list(df_new)[4:6]], df_new[['p1_win_or_loose']]]
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=13, stratify=Y)
model = RandomForestRegressor()
model.fit(X_train, Y_train.values.ravel())

stop = time.time()

print(f"It took {round((stop-start)/1000, 5)} seconds to load the model")

app = Flask(__name__)

@app.route('/api/v1/winpred', methods=["GET"])
def result():

    h1 = request.args.get('h1')
    h2 = request.args.get('h2')
    m1 = request.args.get('m1')
    m2 = request.args.get('m2')

    req = np.array([[h1, h2, m1, m2]])

    op = np.squeeze(model.predict(req)).tolist()

    print(op)

    return jsonify(float(op))

if __name__ == '__main__':
    app.run()
```

5.1.2 M.L Model Code Workflow

Importing Modules

- **numpy** is imported for removing axes of length one from passed value.
- **flask** is used to get and receive the values from MATLAB.
- **pandas** is used to manipulate data present in the data-set "test2.csv".
- **sklearn** is used to divide my data set into two parts, i.e. training and testing set. It is also used to implement RandomForestRegressor.

Data-Set

My data-set contains unbiased 1,000 game-plays. Its has 8 columns, namely:

- healthcond1: This column stores the value of health of player one after each roll
- healthcond2: This column stores the value of health of player two after each roll
- roll_of_p1: This column stores the value of each roll of player 1
- roll_of_p2: This column stores the value of each roll of player 2
- mode_of_p1: This column stores whether player 1 is attacking or healing
- mode_of_p2: This column stores whether player 2 is attacking or healing
- abs_diff: This column stores absolute difference of the rolls

- p1_win_or_lose: This column stores 1 if player 1 wins otherwise stores 0

Before this data-set is passed to the model for testing and training, I have used *dropna()* command to drop any row counting null or empty value.

5.1.3 MATLAB Code

```

classdef dicegame_exported < matlab.apps.AppBase|


    % Properties that correspond to app components
properties (Access = public)
    UIFigure                      matlab.ui.Figure
    GridLayout                     matlab.ui.container.GridLayout
    Player1                        matlab.ui.container.Panel
    ROLLButton_online              matlab.ui.control.Button
    Image                          matlab.ui.control.Image
    ReadyLabel                     matlab.ui.control.Label
    STARTButton                    matlab.ui.control.Button
    ScoreValue                     matlab.ui.control.Label
    ScoreLabel                     matlab.ui.control.Label
    SELECTSwitch                  matlab.ui.control.RockerSwitch
    SELECTSwitchLabel              matlab.ui.control.Label
    ROLLButton                     matlab.ui.control.Button
    CenterPanel                    matlab.ui.container.Panel
    Probabilityofplayer2winningLabel matlab.ui.control.Label
    prob_value_2                   matlab.ui.control.Label
    prob_value                     matlab.ui.control.Label
    Probabilityofplayer1winningLabel matlab.ui.control.Label
    prob                           matlab.ui.control.Image
    SnakeButton                    matlab.ui.control.Button
    PickOneLabel                  matlab.ui.control.Label
    DogButton                      matlab.ui.control.Button
    ONLINEButton                  matlab.ui.control.Button
    OFFLINEButton                 matlab.ui.control.Button
    help                           matlab.ui.control.Image
    HelpButton                     matlab.ui.control.Button
    BattleLog                      matlab.ui.control.Table
    BattleLogButton               matlab.ui.control.Button
    WinLabel                       matlab.ui.control.Label
    Player2Label                  matlab.ui.control.Label
    Player1Label                  matlab.ui.control.Label
    Label_2                        matlab.ui.control.Label
    Label                          matlab.ui.control.Label
    HealthValue                   matlab.ui.control.Label
    HealthValue_2                 matlab.ui.control.Label
    HealthLabel_2                 matlab.ui.control.Label
    HealthLabel                   matlab.ui.control.Label
    HealthBar2                     matlab.ui.control.Image
    HealthBar1                     matlab.ui.control.Image
    Background                     matlab.ui.control.Image
    Player2                        matlab.ui.container.Panel
    ROLLButton_2_online            matlab.ui.control.Button
    -                               -

```

```

ROLLButton_2_online matlab.ui.control.Button
Image2 matlab.ui.control.Image
ReadyLabel_2 matlab.ui.control.Label
STARTButton_2 matlab.ui.control.Button
ScoreValue_2 matlab.ui.control.Label
ScoreLabel_2 matlab.ui.control.Label
SELECTSwitch_2Label matlab.ui.control.Label
SELECTSwitch_2 matlab.ui.control.RockerSwitch
ROLLButton_2 matlab.ui.control.Button
ContextMenu matlab.ui.container.ContextMenu
Menu matlab.ui.container.Menu
Menu2 matlab.ui.container.Menu
end

% Properties that correspond to apps with auto-reflow
properties (Access = private)
    onePanelWidth = 576;
    twoPanelWidth = 768;
end

properties (Access = public)
    channelID = 1670178;
    writeAPIKey='GVV3NUK6VQ1D8WU9';
    readAPIKey='CD8PG73I0LLX80Z6';
    userAPIKey = '0164YBHQAA0B4DS6';
end

% Callbacks that handle component events
methods (Access = private)

    % Code that executes after component creation
    function startupFcn(app)
        set(app.Background, 'Visible', 'off');
        set(app.Player1, 'Visible', 'off');
        set(app.Player2, 'Visible', 'off');
        set(app.BattleLog, 'Visible', 'off');
        set(app.HealthBar1, 'Visible', 'off');
        set(app.HealthBar2, 'Visible', 'off');
        set(app.HealthValue, 'Visible', 'off');
        set(app.HealthValue_2, 'Visible', 'off');
        set(app.HealthLabel, 'Visible', 'off');
    end

```

```

% Code that executes after component creation
function startupFcn(app)
    set(app.Background, 'Visible', 'off');
    set(app.Player1, 'Visible', 'off');
    set(app.Player2, 'Visible', 'off');
    set(app.BattleLog, 'Visible', 'off');
    set(app.HealthBar1, 'Visible', 'off');
    set(app.HealthBar2, 'Visible', 'off');
    set(app.HealthValue, 'Visible', 'off');
    set(app.HealthValue_2, 'Visible', 'off');
    set(app.HealthLabel, 'Visible', 'off');
    set(app.HealthLabel_2, 'Visible', 'off');
    set(app.Label, 'Visible', 'off');
    set(app.Label_2, 'Visible', 'off');
    set(app.Player1Label, 'Visible', 'off');
    set(app.Player2Label, 'Visible', 'off');
    set(app.PickOneLabel, 'Visible', 'off');
    set(app.prob,'Visible', 'off');
    set(app.prob_value,'Visible', 'off');
    set(app.prob_value_2,'Visible', 'off');
    set(app.Probabilityofplayer2winningLabel,'Visible', 'off');
    set(app.Probabilityofplayer1winningLabel,'Visible', 'off');
    set(app.ONLINEButton,'Visible','off');

end

% Button pushed function: ROLLButton
function ROLLButtonPushed(app, event)
    global number1
    number1 = randi(6);

    global number2

    set(app.ROLLButton, 'Enable', 'off');
    app.help.ImageSource = imread("help.png");

    % Player 1's Dice Roll
    switch number1
        case 1
            app.Image.ImageSource = imread("1.jpg");
        case 2
            app.Image.ImageSource = imread("2.jpg");
        case 3
            app.Image.ImageSource = imread("3.jpg");
        ----
    end

```

```

        app.Image.ImageSource = imread("3.jpg");
case 4
    app.Image.ImageSource = imread("4.jpg");
case 5
    app.Image.ImageSource = imread("5.jpg");
otherwise
    app.Image.ImageSource = imread("6.jpg");
end

% Health Bar Display function for Player 1
function DISPLAYPlayer1HealthBar(health)
switch health
case 0
    app.HealthBar1.ImageSource = imread("player1hp0.png");
case 1
    app.HealthBar1.ImageSource = imread("player1hp1.png");
case 2
    app.HealthBar1.ImageSource = imread("player1hp2.png");
case 3
    app.HealthBar1.ImageSource = imread("player1hp3.png");
case 4
    app.HealthBar1.ImageSource = imread("player1hp4.png");
case 5
    app.HealthBar1.ImageSource = imread("player1hp5.png");
case 6
    app.HealthBar1.ImageSource = imread("player1hp6.png");
case 7
    app.HealthBar1.ImageSource = imread("player1hp7.png");
case 8
    app.HealthBar1.ImageSource = imread("player1hp8.png");
case 9
    app.HealthBar1.ImageSource = imread("player1hp9.png");
otherwise
    app.HealthBar1.ImageSource = imread("player1hp10.png");
end
end

% Health Bar Display function for Player 2
function DISPLAYPlayer2HealthBar(health)
switch health
case 0
    app.HealthBar2.ImageSource = imread("hp0.png");

```

```

function DISPLAYPlayer2HealthBar(health)
    switch health
        case 0
            app.HealthBar2.ImageSource = imread("hp0.png");
        case 1
            app.HealthBar2.ImageSource = imread("hp1.png");
        case 2
            app.HealthBar2.ImageSource = imread("hp2.png");
        case 3
            app.HealthBar2.ImageSource = imread("hp3.png");
        case 4
            app.HealthBar2.ImageSource = imread("hp4.png");
        case 5
            app.HealthBar2.ImageSource = imread("hp5.png");
        case 6
            app.HealthBar2.ImageSource = imread("hp6.png");
        case 7
            app.HealthBar2.ImageSource = imread("hp7.png");
        case 8
            app.HealthBar2.ImageSource = imread("hp8.png");
        case 9
            app.HealthBar2.ImageSource = imread("hp9.png");
        otherwise
            app.HealthBar2.ImageSource = imread("hp10.png");
    end
end

function DispProb(pro)
    pro2=round(pro*100);
    pro3=100-pro2;
    app.prob_value.Text = string(pro2)+" %";
    app.prob_value_2.Text = string(pro3)+" %";
    xRound = floor(pro2/5)*5;
    switch xRound
        case 50
            app.prob.ImageSource = imread("p.001.png");
        case 55
            app.prob.ImageSource = imread("p.011.png");
        case 60
            app.prob.ImageSource = imread("p.012.png");
        case 65
            app.prob.ImageSource = imread("p.013.png");
        case 70
            app.prob.ImageSource = imread("p.014.png");
        case 75

```

```

case 60
    app.prob.ImageSource = imread("p.012.png");
case 65
    app.prob.ImageSource = imread("p.013.png");
case 70
    app.prob.ImageSource = imread("p.014.png");
case 75
    app.prob.ImageSource = imread("p.015.png");
case 80
    app.prob.ImageSource = imread("p.016.png");
case 85
    app.prob.ImageSource = imread("p.017.png");
case 90
    app.prob.ImageSource = imread("p.018.png");
case 95
    app.prob.ImageSource = imread("p.019.png");
case 100
    app.prob.ImageSource = imread("p.021.png");
case 45
    app.prob.ImageSource = imread("p.002.png");
case 40
    app.prob.ImageSource = imread("p.003.png");
case 35
    app.prob.ImageSource = imread("p.004.png");
case 30
    app.prob.ImageSource = imread("p.005.png");
case 25
    app.prob.ImageSource = imread("p.006.png");
case 20
    app.prob.ImageSource = imread("p.007.png");
case 15
    app.prob.ImageSource = imread("p.008.png");
case 10
    app.prob.ImageSource = imread("p.009.png");
case 5
    app.prob.ImageSource = imread("p.009.png");
case 0
    app.prob.ImageSource = imread("p.010.png");
otherwise
    app.prob.ImageSource = imread("p.001.png");
end

```

% Play Sound File function

```

function PLAYSoundFile(filename)
    % Your code here
end

```

```

        otherwise
            app.prob.ImageSource = imread("p.001.png");
        end
    end

% Play Sound File function
function PLAYSoundFile(filename)
    [y,Fs] = audioread(filename);
    sound(y,Fs);
    clear y Fs;
end

% WARNING: IF YOU ALTER THE BELOW CODE, MAKE SURE THAT PLAYER 2'S CODE
% IS ALSO CHANGED!

if number1 > 0 & number2 > 0
    % Player 1 has the highest roll:
    s = 'ATTACK';
    if strcmp(app.SELECTSwitch.Value,s)
        m1=0;
    else
        m1=1;
    end

    if strcmp(app.SELECTSwitch.Value,s)
        m2=0;
    else
        m2=1;
    end
    h1=str2double(app.HealthValue.Text);
    h2=str2double(app.HealthValue_2.Text);
    we=webread('http://127.0.0.1:5000/api/v1/winpred','h1',h1,'h2',h2,'m1',m1,'m2',m2);
    DispProb(we);
    if number1 > number2

        switch app.SELECTSwitch.Value
            % If Player 1 is attacking:
            case 'ATTACK'

                conv2num = str2double(app.HealthValue_2.Text);
                newValue = conv2num - (number1 - number2);
                DISPLAYPlayer2HealthBar(newValue);
                DispProb(we);

```

```

DISPLAYPlayer2HealthBar(newValue);
DispProb(we);
conv2char = string(newValue);
app.HealthValue_2.Text = conv2char;
%PLAYSoundFile("hiss.mp3");

% To prevent health from dropping below zero:
if str2double(app.HealthValue_2.Text) < 0
    app.HealthValue_2.Text = "0";
end

% Update Battle Log
app.BattleLog.Data = [app.BattleLog.Data;{number1 number2}];

% Reset board:
number1 = 0;
number2 = 0;
pause(3);
app.Image.ImageSource = imread("blank.jpg");
app.Image2.ImageSource = imread("blank.jpg");
set(app.ROLLButton, 'Enable', 'on');
set(app.ROLLButton_2, 'Enable', 'on');

% If Player 1 is healing:
case 'HEAL'
    %PLAYSoundFile("bark.mp3");
    conv2num = str2double(app.HealthValue.Text);
    newValue = conv2num + (number1 - number2);
    DISPLAYPlayer1HealthBar(newValue);
    DispProb(we);
    conv2char = string(newValue);
    app.HealthValue.Text = conv2char;

    % To prevent health from going above ten:
    if str2double(app.HealthValue.Text) > 10
        app.HealthValue.Text = "10";
    end

    % Update Battle Log
    app.BattleLog.Data = [app.BattleLog.Data;{number1 number2}];

    % Reset board:
    number1 = 0;

```

```

% Update Battle Log
app.BattleLog.Data = [app.BattleLog.Data;{number1 number2}];

% Reset board:
number1 = 0;
number2 = 0;
pause(3);
app.Image.ImageSource = imread("blank.jpg");
app.Image2.ImageSource = imread("blank.jpg");
set(app.ROLLButton, 'Enable', 'on');
set(app.ROLLButton_2, 'Enable', 'on');

end

% Player 2 has the highest roll:
else
switch app.SELECTSwitch_2.Value
% If Player 2 is attacking:
case 'ATTACK'
    conv2num = str2double(app.HealthValue.Text);
    newValue = conv2num - (number2 - number1);
    DISPLAYPlayer1HealthBar(newValue);
    DispProb(we);
    conv2char = string(newValue);
    app.HealthValue.Text = conv2char;
    %PLAYSoundFile("whimper.wav");

% To prevent health from dropping below zero:
if str2double(app.HealthValue.Text) < 0
    app.HealthValue.Text = "0";
end

% Update Battle Log
app.BattleLog.Data = [app.BattleLog.Data;{number1 number2}];

% Reset board:
number1 = 0;
number2 = 0;
pause(3);
app.Image.ImageSource = imread("blank.jpg");
app.Image2.ImageSource = imread("blank.jpg");
set(app.ROLLButton, 'Enable', 'on');
set(app.ROLLButton_2, 'Enable', 'on');

% If Player 2 is healing:

```

```

% If Player 2 is healing:
case 'HEAL'
    %PLAYSoundFile("lunge.mp3");
    conv2num = str2double(app.HealthValue_2.Text);
    newValue = conv2num + (number2 - number1);
    DISPLAYPlayer2HealthBar(newValue);
    DispProb(we);
    conv2char = string(newValue);
    app.HealthValue_2.Text = conv2char;

% To prevent health from going above ten:
if str2double(app.HealthValue_2.Text) > 10
    app.HealthValue_2.Text = "10";
end

% Update Battle Log
app.BattleLog.Data = [app.BattleLog.Data;{number1 number2}];

% Reset board:
number1 = 0;
number2 = 0;
pause(3);
app.Image.ImageSource = imread("blank.jpg");
app.Image2.ImageSource = imread("blank.jpg");
set(app.ROLLButton, 'Enable', 'on');
set(app.ROLLButton_2, 'Enable', 'on');
end
end
end

% Player 1 loses:
if str2double(app.HealthValue.Text) <= 0
    app.HealthValue.Text = "0";
    DISPLAYPlayer1HealthBar(0);
    PLAYSoundFile("victory.mp3");
    score2num = str2double(app.ScoreValue_2.Text);
    newScore = score2num + 1;
    score2string = string(newScore);
    app.ScoreValue_2.Text = score2string;
    app.WinLabel.Text(8) = "2";
    set(app.WinLabel, 'Visible', 'on');

% Reset Game:
number1 = 0;

```

```

% Reset Game:
number1 = 0;
number2 = 0;
set(app.ROLLButton, 'Enable', 'off');
set(app.ROLLButton_2, 'Enable', 'off');
pause(3);
app.Image.ImageSource = imread("blank.jpg");
app.Image2.ImageSource = imread("blank.jpg");
DISPLAYPlayer1HealthBar(10);
DISPLAYPlayer2HealthBar(10);
app.HealthValue.Text = "10";
app.HealthValue_2.Text = "10";
set(app.ROLLButton, 'Enable', 'on');
set(app.ROLLButton_2, 'Enable', 'on');
set(app.WinLabel, 'Visible', 'off');
app.BattleLog.Data = {};
app.prob_value.Text = "50 %";
app.prob_value_2.Text = "50 %";
app.prob.ImageSource = imread("p.001.png");

end

% Player 2 loses:
if str2double(app.HealthValue_2.Text) <= 0
app.HealthValue_2.Text = "0";
DISPLAYPlayer2HealthBar(0);
PLAYSoundFile("victory.mp3");
score2num = str2double(app.ScoreValue.Text);
newScore = score2num + 1;
score2string = string(newScore);
app.ScoreValue.Text = score2string;
app.WinLabel.Text(8) = "1";
set(app.WinLabel, 'Visible', 'on');

% Reset Game:
number1 = 0;
number2 = 0;
set(app.ROLLButton, 'Enable', 'off');
set(app.ROLLButton_2, 'Enable', 'off');
pause(3);
app.Image.ImageSource = imread("blank.jpg");
app.Image2.ImageSource = imread("blank.jpg");
DISPLAYPlayer1HealthBar(10);
DISPLAYPlayer2HealthBar(10);

```

```

DISPLAYPlayer1HealthBar(10);
DISPLAYPlayer2HealthBar(10);
app.HealthValue.Text = "10";
app.HealthValue_2.Text = "10";
set(app.ROLLButton, 'Enable', 'on');
set(app.ROLLButton_2, 'Enable', 'on');
set(app.WinLabel, 'Visible', 'off');
app.BattleLog.Data = {};
app.prob_value.Text = "50 %";
app.prob_value_2.Text = "50 %";
app.prob.ImageSource = imread("p.001.png");

end
end

% Button pushed function: ROLLButton_2
function ROLLButton_2Pushed(app, event)
global number2
number2 = randi(6);

global number1

set(app.ROLLButton_2, 'Enable', 'off');

%Player 2's Dice Roll:
switch number2
case 1
    app.Image2.ImageSource = imread("1.jpg");
case 2
    app.Image2.ImageSource = imread("2.jpg");
case 3
    app.Image2.ImageSource = imread("3.jpg");
case 4
    app.Image2.ImageSource = imread("4.jpg");
case 5
    app.Image2.ImageSource = imread("5.jpg");
otherwise
    app.Image2.ImageSource = imread("6.jpg");
end

% Health Bar Display function for Player 1
function DISPLAYPlayer1HealthBar(health)
switch health

```

```

% Health Bar Display function for Player 1
function DISPLAYPlayer1HealthBar(health)
    switch health
        case 0
            app.HealthBar1.ImageSource = imread("player1hp0.png");
        case 1
            app.HealthBar1.ImageSource = imread("player1hp1.png");
        case 2
            app.HealthBar1.ImageSource = imread("player1hp2.png");
        case 3
            app.HealthBar1.ImageSource = imread("player1hp3.png");
        case 4
            app.HealthBar1.ImageSource = imread("player1hp4.png");
        case 5
            app.HealthBar1.ImageSource = imread("player1hp5.png");
        case 6
            app.HealthBar1.ImageSource = imread("player1hp6.png");
        case 7
            app.HealthBar1.ImageSource = imread("player1hp7.png");
        case 8
            app.HealthBar1.ImageSource = imread("player1hp8.png");
        case 9
            app.HealthBar1.ImageSource = imread("player1hp9.png");
        otherwise
            app.HealthBar1.ImageSource = imread("player1hp10.png");
    end
end

% Health Bar Display function for Player 2
function DISPLAYPlayer2HealthBar(health)
    switch health
        case 0
            app.HealthBar2.ImageSource = imread("hp0.png");
        case 1
            app.HealthBar2.ImageSource = imread("hp1.png");
        case 2
            app.HealthBar2.ImageSource = imread("hp2.png");
        case 3
            app.HealthBar2.ImageSource = imread("hp3.png");
        case 4
            app.HealthBar2.ImageSource = imread("hp4.png");
        case 5
            app.HealthBar2.ImageSource = imread("hp5.png");
        case 6
            app.HealthBar2.ImageSource = imread("hp6.png");
        case 7

```

```

switch health
case 0
    app.HealthBar2.ImageSource = imread("hp0.png");
case 1
    app.HealthBar2.ImageSource = imread("hp1.png");
case 2
    app.HealthBar2.ImageSource = imread("hp2.png");
case 3
    app.HealthBar2.ImageSource = imread("hp3.png");
case 4
    app.HealthBar2.ImageSource = imread("hp4.png");
case 5
    app.HealthBar2.ImageSource = imread("hp5.png");
case 6
    app.HealthBar2.ImageSource = imread("hp6.png");
case 7
    app.HealthBar2.ImageSource = imread("hp7.png");
case 8
    app.HealthBar2.ImageSource = imread("hp8.png");
case 9
    app.HealthBar2.ImageSource = imread("hp9.png");
otherwise
    app.HealthBar2.ImageSource = imread("hp10.png");
end
end

function DispProb(pro)
pro2=round(pro*100);
pro3=100-pro2;
app.prob_value.Text = string(pro2)+" %";
app.prob_value_2.Text = string(pro3)+" %";
xRound = floor(pro2/5)*5;
switch xRound
case 50
    app.prob.ImageSource = imread("p.001.png");
case 55
    app.prob.ImageSource = imread("p.011.png");
case 60
    app.prob.ImageSource = imread("p.012.png");
case 65
    app.prob.ImageSource = imread("p.013.png");
case 70
    app.prob.ImageSource = imread("p.014.png");
case 75
    app.prob.ImageSource = imread("p.015.png");

```

```

        app.prob.ImageSource = imread("p.015.png");
case 80
        app.prob.ImageSource = imread("p.016.png");
case 85
        app.prob.ImageSource = imread("p.017.png");
case 90
        app.prob.ImageSource = imread("p.018.png");
case 95
        app.prob.ImageSource = imread("p.019.png");
case 100
        app.prob.ImageSource = imread("p.021.png");
case 45
        app.prob.ImageSource = imread("p.002.png");
case 40
        app.prob.ImageSource = imread("p.003.png");
case 35
        app.prob.ImageSource = imread("p.004.png");
case 30
        app.prob.ImageSource = imread("p.005.png");
case 25
        app.prob.ImageSource = imread("p.006.png");
case 20
        app.prob.ImageSource = imread("p.007.png");
case 15
        app.prob.ImageSource = imread("p.008.png");
case 10
        app.prob.ImageSource = imread("p.009.png");
case 5
        app.prob.ImageSource = imread("p.009.png");
case 0
        app.prob.ImageSource = imread("p.010.png");
otherwise
        app.prob.ImageSource = imread("p.001.png");
end
end

% Play Sound File function
function PLAYSoundFile(filename)
[y,Fs] = audioread(filename);
sound(y,Fs);
clear y Fs;
end

% The rest of the code is the same as Player 1's, so that way it doesn't
% matter which person rolls first.

```

```

% WARNING: IF YOU ALTER THE BELOW CODE, MAKE SURE THAT PLAYER 1'S CODE
% IS ALSO CHANGED!

if number1 > 0 & number2 > 0
    s = 'ATTACK';
    if strcmp(app.SELECTSwitch.Value,s)
        m1=0;
    else
        m1=1;
    end

    if strcmp(app.SELECTSwitch.Value,s)
        m2=0;
    else
        m2=1;
    end

    h1=str2double(app.HealthValue.Text);
    h2=str2double(app.HealthValue_2.Text);
    we=webread('http://127.0.0.1:5000/api/v1/winpred','h1',h1,'h2',h2,'m1',m1,'m2',m2);
    DispProb(we);

    % Player 1 has the highest roll:
    if number1 > number2
        switch app.SELECTSwitch.Value
            % If Player 1 is attacking:
            case 'ATTACK'
                conv2num = str2double(app.HealthValue_2.Text);
                newValue = conv2num - (number1 - number2);
                DISPLAYPlayer2HealthBar(newValue);
                DispProb(we);
                conv2char = string(newValue);
                app.HealthValue_2.Text = conv2char;
                %PLAYSoundFile("hiss.mp3");

                % To prevent health from dropping below zero:
                if str2double(app.HealthValue_2.Text) < 0
                    app.HealthValue_2.Text = "0";
                end

                % Update Battle Log
                app.BattleLog.Data = [app.BattleLog.Data;{number1 number2}];

            % Reset board:
            number1 = 0;
        end
    end
end

```

```

% Reset board:
number1 = 0;
number2 = 0;
pause(3);
app.Image.ImageSource = imread("blank.jpg");
app.Image2.ImageSource = imread("blank.jpg");
set(app.ROLLButton, 'Enable', 'on');
set(app.ROLLButton_2, 'Enable', 'on');

% If Player 1 is healing:
case 'HEAL'
    %PLAYSoundFile("bark.mp3");
    conv2num = str2double(app.HealthValue.Text);
    newValue = conv2num + (number1 - number2);
    DISPLAYPlayer1HealthBar(newValue);
    DispProb(we);
    conv2char = string(newValue);
    app.HealthValue.Text = conv2char;

% To prevent health from going above ten:
if str2double(app.HealthValue.Text) > 10
    app.HealthValue.Text = "10";
end

% Update Battle Log
app.BattleLog.Data = [app.BattleLog.Data;{number1 number2}];

% Reset board:
number1 = 0;
number2 = 0;
pause(3);
app.Image.ImageSource = imread("blank.jpg");
app.Image2.ImageSource = imread("blank.jpg");
set(app.ROLLButton, 'Enable', 'on');
set(app.ROLLButton_2, 'Enable', 'on');
end

% Player 2 has the highest roll:
else
    switch app.SELECTSwitch_2.Value
        % If Player 2 is attacking:
        case 'ATTACK'
            conv2num = str2double(app.HealthValue.Text);
            newValue = conv2num - (number2 - number1);
            DISPLAYPlayer1HealthBar(newValue);

```

```

newValue = conv2num - (number2 - number1);
DISPLAYPlayer1HealthBar(newValue);
DispProb(we);
conv2char = string(newValue);
app.HealthValue.Text = conv2char;
%PLAYSoundFile("whimper.wav");

% To prevent health from dropping below zero:
if str2double(app.HealthValue.Text) < 0
    app.HealthValue.Text = "0";
end

% Update Battle Log
app.BattleLog.Data = [app.BattleLog.Data;{number1 number2}];

% Reset board:
number1 = 0;
number2 = 0;
pause(3);
app.Image.ImageSource = imread("blank.jpg");
app.Image2.ImageSource = imread("blank.jpg");
set(app.ROLLButton, 'Enable', 'on');
set(app.ROLLButton_2, 'Enable', 'on');

% If Player 2 is healing:
case 'HEAL'
    %PLAYSoundFile("lunge.mp3");
    conv2num = str2double(app.HealthValue_2.Text);
    newValue = conv2num + (number2 - number1);
    DISPLAYPlayer2HealthBar(newValue);
    DispProb(we);
    conv2char = string(newValue);
    app.HealthValue_2.Text = conv2char;

    % To prevent health from going above ten:
    if str2double(app.HealthValue_2.Text) > 10
        app.HealthValue_2.Text = "10";
    end

    % Update Battle Log
    app.BattleLog.Data = [app.BattleLog.Data;{number1 number2}];

```

```

% Reset board:
    number1 = 0;
    number2 = 0;
    pause(3);
    app.Image.ImageSource = imread("blank.jpg");
    app.Image2.ImageSource = imread("blank.jpg");
    set(app.ROLLButton, 'Enable', 'on');
    set(app.ROLLButton_2, 'Enable', 'on');
end
end
end

% Player 1 loses:
if str2double(app.HealthValue.Text) <= 0
    app.HealthValue.Text = "0";
    DISPLAYPlayer1HealthBar(0);
    %PLAYSoundFile("victory.mp3");
    score2num = str2double(app.ScoreValue_2.Text);
    newScore = score2num + 1;
    score2string = string(newScore);
    app.ScoreValue_2.Text = score2string;
    app.WinLabel.Text(8) = "2";
    set(app.WinLabel, 'Visible', 'on');
    app.prob.ImageSource = imread("p.010.png");
    app.prob_value.Text="0 %";
    app.prob_value_2.Text="100 %";

% Reset Game:
    number1 = 0;
    number2 = 0;
    set(app.ROLLButton, 'Enable', 'off');
    set(app.ROLLButton_2, 'Enable', 'off');
    pause(5);
    app.Image.ImageSource = imread("blank.jpg");
    app.Image2.ImageSource = imread("blank.jpg");
    DISPLAYPlayer1HealthBar(10);
    DISPLAYPlayer2HealthBar(10);
    app.HealthValue.Text = "10";
    app.HealthValue_2.Text = "10";
    app.prob.ImageSource = imread("p.001.png");
    set(app.ROLLButton, 'Enable', 'on');
    set(app.ROLLButton_2, 'Enable', 'on');
    set(app.WinLabel, 'Visible', 'off');
    app.BattleLog.Data = {};

```

```

    app.BattleLog.Data = {};
    app.prob_value.Text = "50 %";
    app.prob_value_2.Text = "50 %";

end

% Player 2 loses:
if str2double(app.HealthValue_2.Text) <= 0
    app.HealthValue_2.Text = "0";
    DISPLAYPlayer2HealthBar(0);
    %PLAYSoundFile("victory.mp3");
    score2num = str2double(app.ScoreValue.Text);
    newScore = score2num + 1;
    score2string = string(newScore);
    app.ScoreValue.Text = score2string;
    app.WinLabel.Text(8) = "1";
    set(app.WinLabel, 'Visible', 'on');
    app.prob.ImageSource = imread("p.021.png");
    app.prob_value.Text="100 %";
    app.prob_value_2.Text="0 %";

% Reset Game:
number1 = 0;
number2 = 0;
set(app.ROLLButton, 'Enable', 'off');
set(app.ROLLButton_2, 'Enable', 'off');
pause(5);
app.Image.ImageSource = imread("blank.jpg");
app.Image2.ImageSource = imread("blank.jpg");
DISPLAYPlayer1HealthBar(10);
DISPLAYPlayer2HealthBar(10);
app.HealthValue.Text = "10";
app.HealthValue_2.Text = "10";
set(app.ROLLButton, 'Enable', 'on');
set(app.ROLLButton_2, 'Enable', 'on');
set(app.WinLabel, 'Visible', 'off');
app.BattleLog.Data = {};
app.prob.ImageSource = imread("p.001.png");
app.prob_value.Text = "50 %";
app.prob_value_2.Text = "50 %";

end
end

```

```

% Button pushed function: STARTButton
function STARTButtonPushed(app, event)
    set(app.STARTButton, 'Enable', 'off');

    if app.STARTButton.Enable == 'off'
        set(app.STARTButton, 'Visible', 'off');
        set(app.STARTButton_2, 'Visible', 'off');
        set(app.ReadyLabel, 'Visible', 'off');
        set(app.ReadyLabel_2, 'Visible', 'off');

        set(app.ROLLButton, 'Visible', 'on');
        set(app.SELECTSwitch, 'Visible', 'on');
        set(app.SELECTSwitchLabel, 'Visible', 'on');
        set(app.HealthValue, 'Visible', 'on');
        set(app.Player1Label, 'Visible', 'on');
        set(app.Player2Label, 'Visible', 'on');

        set(app.ROLLButton_2, 'Visible', 'on');
        set(app.SELECTSwitch_2, 'Visible', 'on');
        set(app.SELECTSwitch_2Label, 'Visible', 'on');
        set(app.HealthValue_2, 'Visible', 'on');

        app.Image.ImageSource = imread("blank.jpg");
        app.Image2.ImageSource = imread("blank.jpg");
        set(app.Image, 'Visible', 'on');
        set(app.Image2, 'Visible', 'on');
    end
end

% Button pushed function: STARTButton_2
function STARTButton_2Pushed(app, event)
    set(app.STARTButton_2, 'Enable', 'off');

    if app.STARTButton_2.Enable == 'off'
        set(app.STARTButton, 'Visible', 'off');
        set(app.STARTButton_2, 'Visible', 'off');
        set(app.ReadyLabel, 'Visible', 'off');
        set(app.ReadyLabel_2, 'Visible', 'off');

        set(app.ROLLButton, 'Visible', 'on');
        set(app.SELECTSwitch, 'Visible', 'on');
        set(app.SELECTSwitchLabel, 'Visible', 'on');
        set(app.HealthValue, 'Visible', 'on');

```

```

        set(app.HealthLabel, 'Visible', 'on');
        set(app.ScoreValue, 'Visible', 'on');
        set(app.ScoreLabel, 'Visible', 'on');

        set(app.ROLLButton_2, 'Visible', 'on');
        set(app.SELECTSwitch_2, 'Visible', 'on');
        set(app.SELECTSwitch_2Label, 'Visible', 'on');
        set(app.HealthValue_2, 'Visible', 'on');
        set(app.HealthLabel_2, 'Visible', 'on');
        set(app.ScoreValue_2, 'Visible', 'on');
        set(app.ScoreLabel_2, 'Visible', 'on');

        app.Image.ImageSource = imread("blank.jpg");
        app.Image2.ImageSource = imread("blank.jpg");
        set(app.Image, 'Visible', 'on');
        set(app.Image2, 'Visible', 'on');
    end
end

% Button pushed function: BattleLogButton
function BattleLogButtonPushed(app, event)
    if app.BattleLog.Visible == false
        set(app.BattleLog, 'Visible', 'on');

    else
        set(app.BattleLog, 'Visible', 'off');
    end
end

% Button pushed function: HelpButton
function HelpButtonPushed(app, event)
    if app.help.Visible == false
        set(app.help, 'Visible', 'on');
    else
        set(app.help, 'Visible', 'off');
    end
end

% Button pushed function: ONLINEButton
function ONLINEButtonPushed(app, event)
    set(app.OFFLINEButton, 'Visible', 'off');
    set(app.ONLINEButton, 'Visible', 'off');
    set(app.DogButton, 'Visible', 'on');
    set(app.SnakeButton, 'Visible', 'on');
    set(app.PickOneLabel, 'Visible', 'on');

```

```

app.Image2.ImageSource = imread("blank.jpg");
DISPLAYPlayer1HealthBar(10);
DISPLAYPlayer2HealthBar(10);
app.HealthValue.Text = "10";
app.HealthValue_2.Text = "10";
%set(app.ROLLButton, 'Enable', 'on');
set(app.ROLLButton_2_online, 'Enable', 'on');
set(app.WinLabel, 'Visible', 'off');
app.BattleLog.Data = {};
webwrite(url, weboptions('RequestMethod','delete'));
app.prob_value.Text = "50 %";
app.prob_value_2.Text = "50 %";
app.prob.ImageSource = imread("p.001.png");
app.prob_value.Text = "50 %";
app.prob_value_2.Text = "50 %";
app.prob.ImageSource = imread("p.001.png");
end
end

% Changes arrangement of the app based on UIFigure width
function updateAppLayout(app, event)
currentFigureWidth = app.UIFigure.Position(3);
if(currentFigureWidth <= app.onePanelWidth)
    % Change to a 3x1 grid
    app.GridLayout.RowHeight = {487, 487, 487};
    app.GridLayout.ColumnWidth = {'1x'};
    app.CenterPanel.Layout.Row = 1;
    app.CenterPanel.Layout.Column = 1;
    app.Player1.Layout.Row = 2;
    app.Player1.Layout.Column = 1;
    app.Player2.Layout.Row = 3;
    app.Player2.Layout.Column = 1;
elseif (currentFigureWidth > app.onePanelWidth && currentFigureWidth <= app.twoPanelWidth)
    % Change to a 2x2 grid
    app.GridLayout.RowHeight = {487, 487};
    app.GridLayout.ColumnWidth = {'1x', '1x'};
    app.CenterPanel.Layout.Row = 1;
    app.CenterPanel.Layout.Column = [1,2];
    app.Player1.Layout.Row = 2;
    app.Player1.Layout.Column = 1;
    app.Player2.Layout.Row = 2;
    app.Player2.Layout.Column = 2;
else
    % Change to a 1x3 grid
    app.GridLayout.RowHeight = {'1x'};

```

```

% Change to a 1x3 grid
app.GridLayout.RowHeight = {'1x'};
app.GridLayout.ColumnWidth = {163, '1x', 177};
app.Player1.Layout.Row = 1;
app.Player1.Layout.Column = 1;
app.CenterPanel.Layout.Row = 1;
app.CenterPanel.Layout.Column = 2;
app.Player2.Layout.Row = 1;
app.Player2.Layout.Column = 3;
end
end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Get the file path for locating images
pathToMLAPP = fileparts(fullfile('fullpath'));

% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.AutoScaleChildren = 'off';
app.UIFigure.Position = [100 100 1089 487];
app.UIFigure.Name = 'MATLAB App';
app.UIFigure.SizeChangedFcn = createCallbackFcn(app, @updateAppLayout, true);

% Create GridLayout
app.GridLayout = uigridlayout(app.UIFigure);
app.GridLayout.ColumnWidth = {163, '1x', 177};
app.GridLayout.RowHeight = {'1x'};
app.GridLayout.ColumnSpacing = 0;
app.GridLayout.RowSpacing = 0;
app.GridLayout.Padding = [0 0 0 0];
app.GridLayout.Scrollable = 'on';

% Create Player1
app.Player1 = uipanel(app.GridLayout);
app.Player1.Layout.Row = 1;
app.Player1.Layout.Column = 1;

% Create ROLLButton
app.ROLLButton = uibutton(app.Player1, 'push');

```

```

% Create ROLLButton
app.ROLLButton = uibutton(app.Player1, 'push');
app.ROLLButton.ButtonPushedFcn = createCallbackFcn(app, @ROLLButtonPushed, true);
app.ROLLButton.FontName = 'Bookman Old Style';
app.ROLLButton.FontSize = 20;
app.ROLLButton.FontWeight = 'bold';
app.ROLLButton.Visible = 'off';
app.ROLLButton.Position = [19 141 124 35];
app.ROLLButton.Text = 'ROLL';

% Create SELECTSwitchLabel
app.SELECTSwitchLabel = uilabel(app.Player1);
app.SELECTSwitchLabel.HorizontalAlignment = 'center';
app.SELECTSwitchLabel.FontAngle = 'italic';
app.SELECTSwitchLabel.Visible = 'off';
app.SELECTSwitchLabel.Position = [54 319 54 23];
app.SELECTSwitchLabel.Text = 'SELECT:';

% Create SELECTSwitch
app.SELECTSwitch = uiswitch(app.Player1, 'rocker');
app.SELECTSwitch.Items = {'HEAL', 'ATTACK'};
app.SELECTSwitch.Visible = 'off';
app.SELECTSwitch.FontSize = 20;
app.SELECTSwitch.FontWeight = 'bold';
app.SELECTSwitch.Position = [67 225 27 62];
app.SELECTSwitch.Value = 'ATTACK';

% Create ScoreLabel
app.ScoreLabel = uilabel(app.Player1);
app.ScoreLabel.FontName = 'Bookman Old Style';
app.ScoreLabel.FontSize = 25;
app.ScoreLabel.Position = [27 69 81 32];
app.ScoreLabel.Text = 'Score:';

% Create ScoreValue
app.ScoreValue = uilabel(app.Player1);
app.ScoreValue.FontName = 'Bookman Old Style';
app.ScoreValue.FontSize = 25;
app.ScoreValue.Position = [118 69 25 32];
app.ScoreValue.Text = '0';

% Create STARTButton
app.STARTButton = uibutton(app.Player1, 'push');
app.STARTButton.ButtonPushedFcn = createCallbackFcn(app, @STARTButtonPushed, true);
app.STARTButton.FontName = 'Bookman Old Style';

```

```

% Create STARTButton
app.STARTButton = uibutton(app.Player1, 'push');
app.STARTButton.ButtonPushedFcn = createCallbackFcn(app, @STARTButtonPushed, true);
app.STARTButton.FontName = 'Bookman Old Style';
app.STARTButton.FontSize = 20;
app.STARTButton.FontWeight = 'bold';
app.STARTButton.Position = [20 391 124 35];
app.STARTButton.Text = 'START';

% Create ReadyLabel
app.ReadyLabel = uilabel(app.Player1);
app.ReadyLabel.HorizontalAlignment = 'center';
app.ReadyLabel.FontSize = 15;
app.ReadyLabel.FontAngle = 'italic';
app.ReadyLabel.Position = [41 435 93 22];
app.ReadyLabel.Text = 'Ready?';

% Create Image
app.Image = uiimage(app.Player1);
app.Image.Visible = 'off';
app.Image.Position = [36 354 91 90];

% Create ROLLButton_online
app.ROLLButton_online = uibutton(app.Player1, 'push');
app.ROLLButton_online.ButtonPushedFcn = createCallbackFcn(app, @ROLLButton_onlinePushed, true);
app.ROLLButton_online.FontName = 'Bookman Old Style';
app.ROLLButton_online.FontSize = 20;
app.ROLLButton_online.FontWeight = 'bold';
app.ROLLButton_online.Visible = 'off';
app.ROLLButton_online.Position = [20 141 124 35];
app.ROLLButton_online.Text = 'ROLL';

% Create CenterPanel
app.CenterPanel = uipanel(app.GridLayout);
app.CenterPanel.Layout.Row = 1;
app.CenterPanel.Layout.Column = 2;

% Create Background
app.Background = uiimage(app.CenterPanel);
app.Background.Position = [4 58 742 426];
app.Background.ImageSource = 'back.png';

```

```
% Create HealthBar1
app.HealthBar1 = uiimage(app.CenterPanel);
app.HealthBar1.ImageClickedFcn = createCallbackFcn(app, @updateAppLayout, true);
app.HealthBar1.HorizontalAlignment = 'left';
app.HealthBar1.VerticalAlignment = 'top';
app.HealthBar1.Position = [1 273 340 207];
app.HealthBar1.ImageSource = 'player1hp10.png';

% Create HealthBar2
app.HealthBar2 = uiimage(app.CenterPanel);
app.HealthBar2.Position = [405 280 343 210];
app.HealthBar2.ImageSource = 'hp10.png';

% Create HealthLabel
app.HealthLabel = uilabel(app.CenterPanel);
app.HealthLabel.FontName = 'KaiTi';
app.HealthLabel.FontSize = 20;
app.HealthLabel.Position = [125 391 75 26];
app.HealthLabel.Text = 'Health:';

% Create HealthLabel_2
app.HealthLabel_2 = uilabel(app.CenterPanel);
app.HealthLabel_2.FontName = 'KaiTi';
app.HealthLabel_2.FontSize = 20;
app.HealthLabel_2.Position = [527 391 75 26];
app.HealthLabel_2.Text = 'Health:';

% Create HealthValue_2
app.HealthValue_2 = uilabel(app.CenterPanel);
app.HealthValue_2.FontName = 'KaiTi';
app.HealthValue_2.FontSize = 20;
app.HealthValue_2.Visible = 'off';
app.HealthValue_2.Position = [601 390 25 26];
app.HealthValue_2.Text = '10';

% Create HealthValue
app.HealthValue = uilabel(app.CenterPanel);
app.HealthValue.FontName = 'KaiTi';
app.HealthValue.FontSize = 20;
app.HealthValue.Visible = 'off';
app.HealthValue.Position = [199 390 25 26];
app.HealthValue.Text = '10';
```

```
% Create Label
app.Label = uilabel(app.CenterPanel);
app.Label.FontName = 'KaiTi';
app.Label.FontSize = 20;
app.Label.Position = [223 390 35 26];
app.Label.Text = '/10';

% Create Label_2
app.Label_2 = uilabel(app.CenterPanel);
app.Label_2.FontName = 'KaiTi';
app.Label_2.FontSize = 20;
app.Label_2.Position = [625 390 35 26];
app.Label_2.Text = '/10';

% Create Player1Label
app.Player1Label = uilabel(app.CenterPanel);
app.Player1Label.FontName = 'KaiTi';
app.Player1Label.FontSize = 20;
app.Player1Label.Position = [139 422 85 26];
app.Player1Label.Text = 'Player 1';

% Create Player2Label
app.Player2Label = uilabel(app.CenterPanel);
app.Player2Label.FontName = 'KaiTi';
app.Player2Label.FontSize = 20;
app.Player2Label.Position = [551 420 85 26];
app.Player2Label.Text = 'Player 2';

% Create WinLabel
app.WinLabel = uilabel(app.CenterPanel);
app.WinLabel.HorizontalAlignment = 'center';
app.WinLabel.FontSize = 50;
app.WinLabel.FontWeight = 'bold';
app.WinLabel.Visible = 'off';
app.WinLabel.Position = [171 297 408 67];
app.WinLabel.Text = 'PLAYER _ WINS!';
```

```

% Create BattleLogButton
app.BattleLogButton = uibutton(app.CenterPanel, 'push');
app.BattleLogButton.ButtonPushedFcn = createCallbackFcn(app, @BattleLogButtonPushed, true);
app.BattleLogButton.VerticalAlignment = 'top';
app.BattleLogButton.BackgroundColor = [0.8745 0.8863 0.9608];
app.BattleLogButton.FontName = 'FangSong';
app.BattleLogButton.FontSize = 13;
app.BattleLogButton.Position = [14 20 74 23];
app.BattleLogButton.Text = 'Battle Log';

% Create BattleLog
app.BattleLog = uitable(app.CenterPanel);
app.BattleLog.ColumnName = {'Player 1 Rolls'; 'Player 2 Rolls'};
app.BattleLog.RowName = {};
app.BattleLog.Visible = 'off';
app.BattleLog.Position = [276 116 198 248];

% Create HelpButton
app.HelpButton = uibutton(app.CenterPanel, 'push');
app.HelpButton.ButtonPushedFcn = createCallbackFcn(app, @HelpButtonPushed, true);
app.HelpButton.VerticalAlignment = 'top';
app.HelpButton.BackgroundColor = [0.8745 0.8863 0.9608];
app.HelpButton.FontName = 'FangSong';
app.HelpButton.FontSize = 13;
app.HelpButton.Position = [659 20 74 23];
app.HelpButton.Text = 'Help';

% Create help
app.help = uiimage(app.CenterPanel);
app.help.Visible = 'off';
app.help.Position = [174 43 401 422];
app.help.ImageSource = 'back.png';

% Create OFFLINEButton
app.OFFLINEButton = uibutton(app.CenterPanel, 'push');
app.OFFLINEButton.ButtonPushedFcn = createCallbackFcn(app, @OFFLINEButtonPushed, true);
app.OFFLINEButton.Position = [283 209 183 37];
app.OFFLINEButton.Text = 'OFFLINE';

```

```

% Create ONLINEButton
app.ONLINEButton = uibutton(app.CenterPanel, 'push');
app.ONLINEButton.ButtonPushedFcn = createCallbackFcn(app, @ONLINEButtonPushed, true);
app.ONLINEButton.Position = [283 283 183 37];
app.ONLINEButton.Text = 'ONLINE';

% Create DogButton
app.DogButton = uibutton(app.CenterPanel, 'push');
app.DogButton.ButtonPushedFcn = createCallbackFcn(app, @DogButtonPushed, true);
app.DogButton.FontSize = 20;
app.DogButton.Visible = 'off';
app.DogButton.Position = [250 319 250 45];
app.DogButton.Text = 'Dog';

% Create PickOneLabel
app.PickOneLabel = uilabel(app.CenterPanel);
app.PickOneLabel.FontSize = 20;
app.PickOneLabel.Position = [341 386 87 26];
app.PickOneLabel.Text = 'Pick One';

% Create SnakeButton
app.SnakeButton = uibutton(app.CenterPanel, 'push');
app.SnakeButton.ButtonPushedFcn = createCallbackFcn(app, @SnakeButtonPushed, true);
app.SnakeButton.FontSize = 20;
app.SnakeButton.Visible = 'off';
app.SnakeButton.Position = [250 246 250 45];
app.SnakeButton.Text = 'Snake';

% Create prob
app.prob = uiimage(app.CenterPanel);
app.prob.ScaleMethod = 'stretch';
app.prob.Position = [109 14 527 34];
app.prob.ImageSource = fullfile(pathToMLAPP, 'p.001.png');

% Create Probabilityofplayer1winningLabel
app.Probabilityofplayer1winningLabel = uilabel(app.CenterPanel);
app.Probabilityofplayer1winningLabel.Position = [139 20 174 22];
app.Probabilityofplayer1winningLabel.Text = 'Probability of player 1 winning: ';

```

```

% Create prob_value
app.prob_value = uilabel(app.CenterPanel);
app.prob_value.HorizontalAlignment = 'center';
app.prob_value.Position = [314 20 42 22];
app.prob_value.Text = '50';

% Create prob_value_2
app.prob_value_2 = uilabel(app.CenterPanel);
app.prob_value_2.HorizontalAlignment = 'center';
app.prob_value_2.Position = [560 20 42 22];
app.prob_value_2.Text = '50';

% Create Probabilityofplayer2winningLabel
app.Probabilityofplayer2winningLabel = uilabel(app.CenterPanel);
app.Probabilityofplayer2winningLabel.Position = [378 20 174 22];
app.Probabilityofplayer2winningLabel.Text = 'Probability of player 2 winning: ';

% Create Player2
app.Player2 = uipanel(app.GridLayout);
app.Player2.Layout.Row = 1;
app.Player2.Layout.Column = 3;

% Create ROLLButton_2
app.ROLLButton_2 = uibutton(app.Player2, 'push');
app.ROLLButton_2.ButtonPushedFcn = createCallbackFcn(app, @ROLLButton_2Pushed, true);
app.ROLLButton_2.FontName = 'Bookman Old Style';
app.ROLLButton_2.FontSize = 20;
app.ROLLButton_2.FontWeight = 'bold';
app.ROLLButton_2.Visible = 'off';
app.ROLLButton_2.Position = [19 141 124 35];
app.ROLLButton_2.Text = 'ROLL';

% Create SELECTSwitch_2
app.SELECTSwitch_2 = uiswitch(app.Player2, 'rocker');
app.SELECTSwitch_2.Items = {'HEAL', 'ATTACK'};
app.SELECTSwitch_2.Visible = 'off';
app.SELECTSwitch_2.FontSize = 20;
app.SELECTSwitch_2.FontWeight = 'bold';
app.SELECTSwitch_2.Position = [67 225 27 62];
app.SELECTSwitch_2.Value = 'ATTACK';

% Create SELECTSwitch_2Label
app.SELECTSwitch_2Label = uilabel(app.Player2);
app.SELECTSwitch_2Label.HorizontalAlignment = 'center';

```

```

app.SELECTSwitch_2Label.Visible = 'off';
app.SELECTSwitch_2Label.Position = [54 319 54 23];
app.SELECTSwitch_2Label.Text = 'SELECT';

% Create ScoreLabel_2
app.ScoreLabel_2 = uilabel(app.Player2);
app.ScoreLabel_2.FontName = 'Bookman Old Style';
app.ScoreLabel_2.FontSize = 25;
app.ScoreLabel_2.Position = [27 69 81 32];
app.ScoreLabel_2.Text = 'Score:';

% Create ScoreValue_2
app.ScoreValue_2 = uilabel(app.Player2);
app.ScoreValue_2.FontName = 'Bookman Old Style';
app.ScoreValue_2.FontSize = 25;
app.ScoreValue_2.Position = [118 69 25 32];
app.ScoreValue_2.Text = '0';

% Create STARTButton_2
app.STARTButton_2 = uibutton(app.Player2, 'push');
app.STARTButton_2.ButtonPushedFcn = createCallbackFcn(app, @STARTButton_2Pushed, true);
app.STARTButton_2.FontName = 'Bookman Old Style';
app.STARTButton_2.FontSize = 20;
app.STARTButton_2.FontWeight = 'bold';
app.STARTButton_2.Position = [19 391 124 35];
app.STARTButton_2.Text = 'START';

% Create ReadyLabel_2
app.ReadyLabel_2 = uilabel(app.Player2);
app.ReadyLabel_2.HorizontalAlignment = 'center';
app.ReadyLabel_2.FontSize = 15;
app.ReadyLabel_2.FontAngle = 'italic';
app.ReadyLabel_2.Position = [34 435 93 22];
app.ReadyLabel_2.Text = 'Ready?';

% Create Image2
app.Image2 = uiimage(app.Player2);
app.Image2.Visible = 'off';
app.Image2.Position = [36 356 91 90];

% Create ROLLButton_2_online
app.ROLLButton_2_online = uibutton(app.Player2, 'push');
app.ROLLButton_2_online.ButtonPushedFcn = createCallbackFcn(app, @ROLLButton_2_onlinePushed, true);
app.ROLLButton_2_online.FontName = 'Bookman Old Style';

```

```

    app.ROLLButton_2_online.FontSize = 20;
    app.ROLLButton_2_online.FontWeight = 'bold';
    app.ROLLButton_2_online.Visible = 'off';
    app.ROLLButton_2_online.Position = [20 141 124 35];
    app.ROLLButton_2_online.Text = 'ROLL';

    % Create ContextMenu
    app.ContextMenu = uicontextmenu(app.UIFigure);

    % Create Menu
    app.Menu = uimenu(app.ContextMenu);
    app.Menu.Text = 'Menu';

    % Create Menu2
    app.Menu2 = uimenu(app.ContextMenu);
    app.Menu2.Text = 'Menu2';

    % Show the figure after all components are created
    app.UIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

    % Construct app
    function app = dicegame_exported

        % Create UIFigure and components
        createComponents(app)

        % Register the app with App Designer
        registerApp(app, app.UIFigure)

        % Execute the startup function
        runStartupFcn(app, @startupFcn)

        if nargout == 0
            clear app
        end
    end

    % Code that executes before app deletion
    function delete(app)

```

```

% Create ContextMenu
app.ContextMenu = uicontextmenu(app.UIFigure);

% Create Menu
app.Menu = uimenu(app.ContextMenu);
app.Menu.Text = 'Menu';

% Create Menu2
app.Menu2 = uimenu(app.ContextMenu);
app.Menu2.Text = 'Menu2';

% Show the figure after all components are created
app.UIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

    % Construct app
    function app = dicegame_exported

        % Create UIFigure and components
        createComponents(app)

        % Register the app with App Designer
        registerApp(app, app.UIFigure)

        % Execute the startup function
        runStartupFcn(app, @startupFcn)

        if nargout == 0
            clear app
        end
    end

    % Code that executes before app deletion
    function delete(app)

        % Delete UIFigure when app is deleted
        delete(app.UIFigure)
    end
end
end

```

5.1.4 MATLAB Code Explanation

- DISPLAYPlayer2HealthBar(): to display health of 2nd player.
- DISPLAYPlayerHealthBar(): to display health of player 1.
- ROLLButton_Pushed(app, event): to roll the dice.
- ROLLButtonPushed(): This function compares the dice value of both players and update their respective health.
- BattleLog(): to create and display battle log of each round.
- Each button is created using MATLAB App visualizer and function for each button is at the end of MATLAB code.
- A round ends when a player health reaches 0, the game updates the score.

5.2 Output

5.2.1 GUI Result

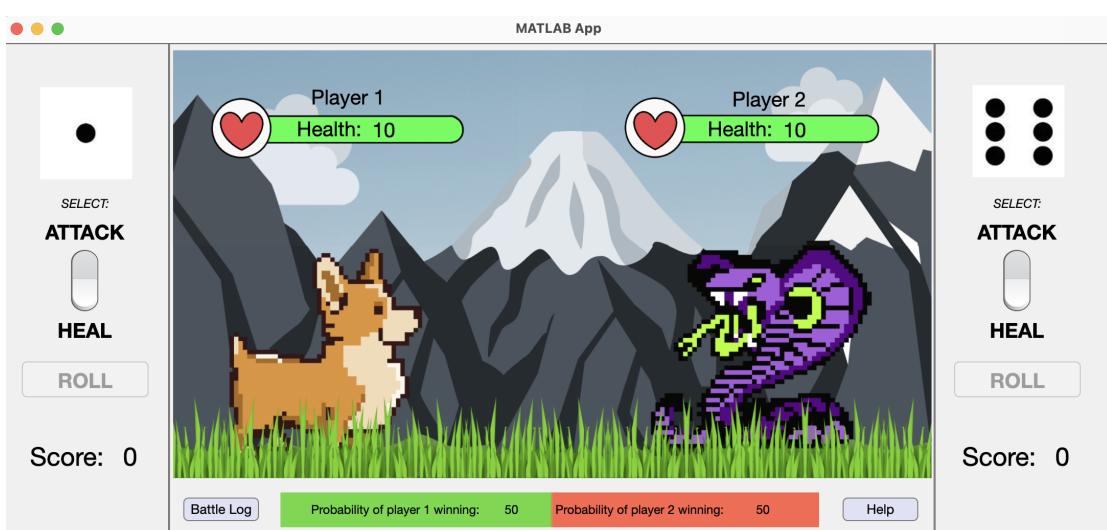


Figure 5.1: Game-play 1



Figure 5.2: Game-play 2



Figure 5.3: Battle Log

Figure 5.1 and 5.2 shows the game at different stages. In both the figures the probability bars moves as the game progress. The probability is calculated using the model, after `health1`, `health2`, `mode1`, `mode2`, `roll1`, `roll2` are passed to the model. After a player wins, he/she will be awarded with a score of one point and the game will reset itself for a new session.5.3 shows a log file for dice rolls of each player in tabular format.

CHAPTER 6

CONCLUSION

Random Forest Regression Method seems to be the appropriate methodology for training and making prediction in my project. RFR has high sensitivity and accuracy.

Future studies related to computer game prediction using M.L will be benefited by this game and model. This model can be incorporated into prediction games. It aims at using Artificial Intelligence for helping radiologist in making quick and accurate decisions while making an assessment.

CHAPTER 7

FUTURE ENHANCEMENTS

While this game and model is efficient, there is a scope of enhancements in the game. The enhancements are discussed below:-

1. In the current implementation of the project, the model is trained in python programming language. The model can be created using MATLAB and trained within the MATLAB app. This can eliminate the requirement of running a separate program and use of API for communication.
2. Additionally to above mentioned enhancement, the trained model can be used to create and train an Artificial Intelligent bot. A.I bot then can be added to the game as an additional feature, in order to make game single-player oriented.
3. We can also add ThingSpeak to the game in-order to add analysis on each player.

Furthermore the trained model of this game can be used by other dice based games and can be used to create prediction model for other games too.

REFERENCES

1. Bansal, S. “Rest api (introduction).” *URL: https://www.geeksforgeeks.org /rest-api-introduction/.*
2. Bharadwaj, A. “Build a rest api using flask.” *URL: https://www.geeksforgeeks.org/ python-build-a-rest-api-using-flask /.*
3. Biau, G. and Scornet, E. (2016). “A random forest guided tour.” *Test*, 25(2), 197–227.
4. Campbell, W., Wimsatt, H., Panzade, A., and Shankar, K. (2021). “Using machine learning to interpret dice rolls.
5. Claver, J. H., Azimi, J., and Suzuki, T. (2017). “New prospective on multiple dice rolling game and its statistical implications.” *American Journal of Applied Mathematics and Statistics*, 5(5), 169–174.
6. Elsayed, A. A. and Yousef, W. A. (2019). “Matlab vs. opencv: A comparative study of different machine learning algorithms.” *arXiv preprint arXiv:1905.01213*.
7. Etter, D. M., Kuncicky, D. C., and Hull, D. W. (2002). *Introduction to MATLAB*. Prentice Hall Hoboken, NJ, USA.
8. Gilat, A. (2004). *MATLAB: an introduction with applications*. John Wiley & Sons.
9. Higham, D. J. and Higham, N. J. (2016). *MATLAB guide*. SIAM.

10. Kuhn, M., Johnson, K., et al. (2013). *Applied predictive modeling*, Vol. 26. Springer.
11. Kumar, A. and Gandhi, J. (2019). “A survey on sports prediction using machine learning.” *JASC: Journal of Applied Science and Computations*, VI(IV).
12. Kumari, R. and Srivastava, S. K. (2017). “Machine learning: A review on binary classification.” *International Journal of Computer Applications*, 160(7).
13. Levy, M., Raviv, D., and Baker, J. (2019). “Data center predictions using matlab machine learning toolbox.” *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*. 0458–0464.
14. Lokhande, P., Aslam, F., Hawa, N., Munir, J., and Gulamgaus, M. (2015). “Efficient way of web development using python and flask.
15. MATHWORKS, I. (2018). “Mastering machine learning: A step-by-step guide with matlab.” *Mathworks Inc.*
16. Mwiti, D. (2020). “Random forest regression: When does it fail and why?” URL: <https://neptune.ai/blog/random-forest-regression-when-does-it-fail-and-why>.
17. Pasha, S. (2016). “Thingspeak based sensing and monitoring system for iot with matlab analysis.” *International Journal of New Technology and Research (IJNTR)*, 2(6), 19–23.
18. Patel, H. H. and Prajapati, P. (2018). “Study and analysis of decision tree based classification algorithms.” *International Journal of Computer Sciences and Engineering*, 6(10), 74–78.

19. Patil, N., Sequeira, B. H., Gonsalves, N. N., and Singh, A. A. (2020). “Cricket team prediction using machine learning techniques.” *Available at SSRN 3572740*.
20. Perutka, K. and Vymazal, M. (2021). “Application of game-based learning in matlab using object-oriented programming..” *Annals of DAAAM & Proceedings*, 10(2).
21. Prist, M., Monteriù, A., Freddi, A., Pallotta, E., Ciabattoni, L., Cicconi, P., Giuggioloni, F., Caizer, E., Verdini, C., and Longhi, S. (2020). “Machine learning-as-a-service for consumer electronics fault diagnosis: a comparison between matlab and azure ml.” *2020 IEEE International Conference on Consumer Electronics (ICCE)*. 1–5.
22. Relan, K. (2019). “Building rest apis with flask.” *Building REST APIs with Flask*.
23. Smith, S. T. (2006). *MATLAB: advanced GUI development*. Dog ear publishing.
24. Zhu, X. and Goldberg, A. B. (2009). “Introduction to semi-supervised learning.” *Synthesis lectures on artificial intelligence and machine learning*, 3(1), 1–130.
25. Zhu, X. J. (2005). “Semi-supervised learning literature survey”.

ANNEXURE-I



Date: March 9, 2022

To Whomsoever It May Concern

This is to certify that Mr. Hemang Sharma (920396331) is undertaking Dice Game as his final project for ENG006:Engineering Problem Solving, WQ 2022. The project will build on all he has learned during the Winter Quarter at UC Davis.

A handwritten signature in black ink that appears to read "D. Yankelevich".

Professor Diego Yankelevich
Department of Electrical and Computer Engineering
One Shields Avenue, Kemper Hall
University of California
Davis, CA 95616

ANNEXURE-2

DiceGame_22MPA165_Hemang_Sharma

ORIGINALITY REPORT

2%	2%	2%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ijcseonline.org Internet Source	1%
2	M. Prist, A. Monteriu, A. Freddi, E. Pallotta, L. Ciabattoni, P. Cicconi, F. Giuggioloni, E. Caizer, C. Verdini, S. Longhi. "Machine Learning-as-a-Service for Consumer Electronics Fault Diagnosis: a Comparison between Matlab and Azure ML", 2020 IEEE International Conference on Consumer Electronics (ICCE), 2020 Publication	1%
3	pubs.sciepub.com Internet Source	1%

Exclude quotes On
Exclude bibliography On

Exclude matches Off