**UNIVERZITET U BIHAĆU**

**TEHNIČKI FAKULTET**

**BIHAĆ**

**PREDMET: MATLAB**

**TEMA: Matlab kao platforma za razvoj mini igrica**

**Profesor:** dr. Adnan Ramakić **Studentica:** Sumea Kličić

**Asistentica:** Una Drakulić **Broj indexa:** 1285

Bihać, 2025

Sadržaj

[**UVOD:** 1](#_Toc199768898)

[**1.UVODNI PROZOR** 2](#_Toc199768899)

[**1.1.Funkcionalnost buttona** 3](#_Toc199768900)

[**2.PRVA IGRICA: Matematička zamka** 4](#_Toc199768901)

[**2.1. Kod za igricu: Matematička zamka** 5](#_Toc199768902)

[**3.DRUGA IGRICA: Papir kamen makaze** 10](#_Toc199768903)

[**3.1. Kod za drugu igricu: Papir kamen makaze** 12](#_Toc199768904)

[**4. TREĆA IGRICA: Pogađanje broja od 1 do 100** 16](#_Toc199768905)

[**4.1. Kod za igricu podađanja zamišljenog broja** 18](#_Toc199768906)

[**ZAKLJUČAK:** 22](#_Toc199768907)

# **UVOD:**

Pravljenje prvih igrica je uvijek cilj svakog programera. Korištenjem Matlab GUI kreirala sam 3 jednostavne igrice koje mogu koristiti svi. Cilj je da igrice budu jednostavne za korištenje, ali da budu zanimljive i korisne. Korištenjem osnovnih matematičkih operacija ili samo igrom na sreću svaka od igrica se može jednostavno igrati.

Detaljnijom analizom došla sam do toga da jasno pišu upute na prozorima, te da se kraj igrice ispiše te da korisnik zna o čemu se radi i zašto više igrica nije aktivna.

Smatram da je kreiranje igrica dobar način za razvijanje logike, ali i učenje prvih pravila u dizajnu. Postoji nekoliko osnovnih komponenti koje sam koristila u svakom prozoru i bez kojih nije moguće napraviti neku zanimljivu interaktivnost. Jedna od najčešćih komponenti je Label, a nakon toga button.

Dodatno me ovo naučilo kako da organizujem kod, te da problem podijelim na više dijelova tako da bude razumljiviji i jasniji kod, ali i da se korisnik više zabavi jer u jednom momentu vidi samo jednu aktivnost koju treba raditi.

# **1.UVODNI PROZOR**

Nakon što se pokrene igrica pojavi se prvi prozor gdje korisnik vidi sve igrice koje imamo u ponudi. Korisnik može da izabere željenu igricu. Da bi se pokrenula igrica koju želi dovoljno je da klikne na dugme gdje je naziv njegove željene igrice.



Slika : Uvodni prozor

Prozor ima tri komponente sa slikama, a nalaze se iznad dugmadi. Ispod svake slike ima odgovarajući gumb koji nas vodi na idući prozor. Iza svakog dugmeta stoji ista funkcionalnost.

# **1.1.Funkcionalnost buttona**

function MatematikazamkaButtonPushed(app, event) %Automatski se pojavi funkcija

MatematickaZamka; %Ime prozora gdje je prva igrica je takav i ovo simbolizira da

%se klikom na taj button treba preci na novi prozor

end

function PapirkamenmakazeButtonPushed(app, event)

pkm; %naziv prozora druge aplikacije

end

function PogodizamisljenibrojButtonPushed(app, event)

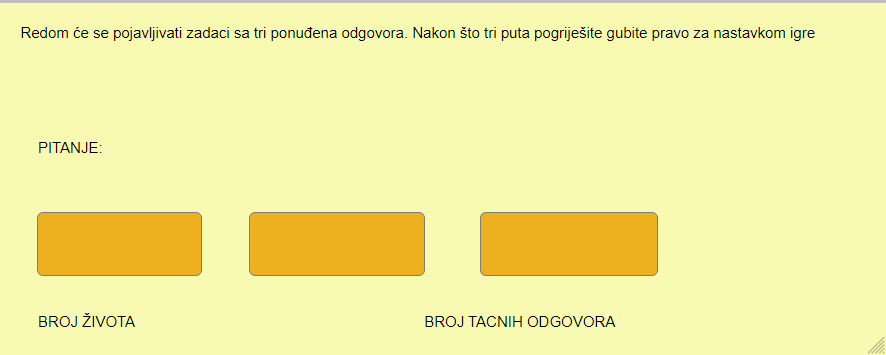
PogodiBroj; %prozor za trecu igricu

end

Funkcionalnost svakog buttona je ista zato što je cilj samo da se pređe na anredni prozor. Ovdje imamo samo ponuđene sve igrice koje imamo u aplikaciji da se korisnik može odlučiti za jednu i pokrenuti je

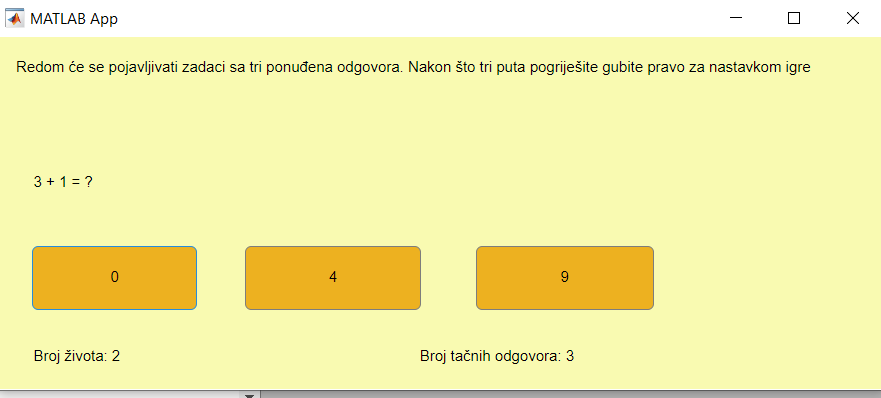
# **2.PRVA IGRICA: Matematička zamka**

Igrica se zove matematička zamka. Kroz program random se pojavljuju jednocifreni brojevi te osnovne matematičke operacije. Korisnik ima 3 života, a sa svakim pogrešnim odgovorom gubi jedan život. Korisnik odgovara na pitanja tako što klikne na dugme sa odgovorom za kojeg smatra da je ispravan. Sa desne strane mijenja se broj tačnih odgovora kako korisnik odgovara tačno. Cilj igrice je učenje osnovne matematike te brzo razmišljanje. Dizajn je jednostavan, ali efektan jer nudi sve potrebne informacije. Na prozoru korištene su komponente Label i Button.



Slika 2: Prikaz kreiranog prozora za matematičku zamku

Nakon što je prozor pokrene unutar buttona i dva donja labela pojavi se tekst, a takođe pitanje postane vidljivo. Tada je izgled prozora idući:



Slika 3: Pokrenuti prozor igrice matematička zamka

Nakon nekoliko odgovorenih pitanja promijenio se broj tačnih odgovora sa 0 na 3, a izgubljen je jedan život tako da je i to promijenjeno. Buttoni imaju jednostanu funkcionalnost a to je da samo pozovu funkciju za provjeru tacnih odgovora. Dok se u funkciji za provjeru gleda da li je tacan odgovor jednak onom sto smo mi izabrali. Takođe imamo operaciju inicijalizacije igre, te postavljanje pitanja. Pitanja se generišu slučajno tako što se koriste funkcie za random brojeve od 1 do 10 i niz operatora.

# **2.1. Kod za igricu: Matematička zamka**

classdef MatematickaZamka < matlab.apps.AppBase

%MATEMATICKAZAMKA konstruktor

% Properties that correspond to app components

properties (Access = public)

UIFigure matlab.ui.Figure

BROJTACNIHODGOVORALabel matlab.ui.control.Label

BROJIVOTALabel matlab.ui.control.Label

Button\_3 matlab.ui.control.Button

Button\_2 matlab.ui.control.Button

Button matlab.ui.control.Button

PITANJELabel matlab.ui.control.Label

Label matlab.ui.control.Label

end

% Public properties that correspond to the Simulink model

properties (Access = public, Transient)

Simulation simulink.Simulation

end

%U ovom dijelu koda inicijalizujemo naše varijable sa početnim

%vrijednostima

properties (Access = private)

zivot double = 3; % Broj preostalih zivota, počinje se sa 3

tacniOdgovori double = 0; % Broj tacnih odgovora, kreće od 0

trenutnoTacno char % Trenutno tačnan odgovor da bi mogli porediti

end

methods (Access = private)

function initializeGame(app)

% Resetuje stanje igre, to su varijable koje su gore definisane

app.zivot = 3;

app.tacniOdgovori = 0;

% Ažurira tekstualne oznake

app.BROJIVOTALabel.Text = ['Broj života: ' num2str(app.zivot)];

app.BROJTACNIHODGOVORALabel.Text = ['Broj tačnih odgovora: ' num2str(app.tacniOdgovori)];

% Postavi prvo pitanje

postaviPitanje(app);

end

%Funkcija koja sluzi da se postavi naredno matematicko pitanje

function postaviPitanje(app)

% Generiši dva nasumična broja i nasumičnu operaciju

a = randi([1 10]); %bilo koji broj od 1 do 10

b = randi([1 10]);

operacije = {'+', '-', '\*'}; %vektor sa operacijama

op = operacije{randi(3)}; %rendom znak iz naseg niza svih operacija

% Izračunaj tačan odgovor

switch op %u zavisnosti od operacije koja je spremljena u varijablu op gledamo koji je tacan odgvor

case '+'

tacan = a + b;

case '-'

tacan = a - b;

case '\*'

tacan = a \* b;

end

% Sacuvan tacni odgvor

app.trenutnoTacno = num2str(tacan);

% Generiše 2 pogrešna odgovora na nasumican nacin

netacni = [tacan + randi([1 5]), tacan - randi([1 5])]; %ovo je algoritam na koji se generise novi broj

%s tim da nije nuzno da se odradi ovako vec mozemo i na bilo koji

%zeljeni nacin

while netacni(1) == tacan %ako je ipak tacan nastavlja se dodavanjem

netacni(1) = tacan + randi([1 5]);

end

while netacni(2) == tacan || netacni(2) == netacni(1) %odraduje se isto za drugi netecan odgvoor

netacni(2) = tacan - randi([1 5]);

end

% Mijesanje tačnih i netacnih odgovora da ne bi uvijek tacni bili na

% istom mjestu

svi = [tacan, netacni];

svi = svi(randperm(3)); % nasumičan raspored

% Prikaz na ekranu, kupe se sad sredeni podaci

app.PITANJELabel.Text = sprintf('%d %s %d = ?', a, op, b);

app.Button.Text = num2str(svi(1));

app.Button\_2.Text = num2str(svi(2));

app.Button\_3.Text = num2str(svi(3));

end

%Provjerava klikuno i ono sto je tacno

function provjeriOdgovor(app, odgovor)

if strcmp(odgovor, app.trenutnoTacno) %poredi

app.tacniOdgovori = app.tacniOdgovori + 1; %ako je tacno povecava se broj tacnih odgovora

app.BROJTACNIHODGOVORALabel.Text = ['Broj tačnih odgovora: ' num2str(app.tacniOdgovori)]; %U labelu se mijenja broj

else

app.zivot = app.zivot - 1; %ako odgovor nije tacan oduzima se jedan zivot

app.BROJIVOTALabel.Text = ['Broj života: ' num2str(app.zivot)]; %mijenja se vrijednost u labelu gdje pise broj zivota

end

if app.zivot == 0 %nakon sto tri puta izgubilo broj zivota ce doci na 0, a to je kraj igre

uialert(app.UIFigure, 'Izgubili ste igru!', 'Kraj igre');

igrica; %vracanje na pocetni prozor

app.Button.Enable = 'off';

app.Button\_2.Enable = 'off';

app.Button\_3.Enable = 'off';

return;

end

% Novi zadatak

app.postaviPitanje(); %ako se ne ude u prethodni uslov znaci da nije 0 broj zivota pa se postavlja novo pitanje

end

end

% Callbacks that handle component events

methods (Access = private)

% Code that executes after component creation

function startupFcn(app)

initializeGame(app); %posto ne mogu pristupiti konstruktoru na ovaj nacin pokrecem program, to je

%callbacks metoda

end

% Button pushed function: Button

function ButtonPushed(app, event)

app.provjeriOdgovor(app.Button.Text); % provjeri odgovor na osnovu teksta dugmeta

%Kad se klikne na odredeno dugme cilj je provjeriti da li je

%odgovor tacan tako da pozivam gore napisanu funkciju

end

% Button pushed function: Button\_2

function Button\_2Pushed(app, event)

app.provjeriOdgovor(app.Button\_2.Text); % provjeri odgovor na osnovu teksta dugmeta

end

% Button pushed function: Button\_3

function Button\_3Pushed(app, event)

app.provjeriOdgovor(app.Button\_3.Text); % provjeri odgovor na osnovu teksta dugmeta

end

end

%DONJI DIO KODA JE KOD KOJI SE SAM GENERIŠE

% Component initialization

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Create UIFigure and hide until all components are created

app.UIFigure = uifigure('Visible', 'off');

app.UIFigure.Color = [0.9765 0.9804 0.6941];

app.UIFigure.Position = [100 100 709 282];

app.UIFigure.Name = 'MATLAB App';

% Create Label

app.Label = uilabel(app.UIFigure);

app.Label.Position = [17 248 640 22];

app.Label.Text = 'Redom će se pojavljivati zadaci sa tri ponuđena odgovora. Nakon što tri puta pogriješite gubite pravo za nastavkom igre';

% Create PITANJELabel

app.PITANJELabel = uilabel(app.UIFigure);

app.PITANJELabel.Position = [31 149 192 36];

app.PITANJELabel.Text = 'PITANJE:';

% Create Button

app.Button = uibutton(app.UIFigure, 'push');

app.Button.ButtonPushedFcn = createCallbackFcn(app, @ButtonPushed, true);

app.Button.BackgroundColor = [0.9294 0.6941 0.1255];

app.Button.Position = [30 65 132 51];

app.Button.Text = '';

% Create Button\_2

app.Button\_2 = uibutton(app.UIFigure, 'push');

app.Button\_2.ButtonPushedFcn = createCallbackFcn(app, @Button\_2Pushed, true);

app.Button\_2.BackgroundColor = [0.9294 0.6941 0.1255];

app.Button\_2.Position = [200 65 141 51];

app.Button\_2.Text = '';

% Create Button\_3

app.Button\_3 = uibutton(app.UIFigure, 'push');

app.Button\_3.ButtonPushedFcn = createCallbackFcn(app, @Button\_3Pushed, true);

app.Button\_3.BackgroundColor = [0.9294 0.6941 0.1255];

app.Button\_3.Position = [385 65 142 51];

app.Button\_3.Text = '';

% Create BROJIVOTALabel

app.BROJIVOTALabel = uilabel(app.UIFigure);

app.BROJIVOTALabel.Position = [31 17 83 22];

app.BROJIVOTALabel.Text = 'BROJ ŽIVOTA';

% Create BROJTACNIHODGOVORALabel

app.BROJTACNIHODGOVORALabel = uilabel(app.UIFigure);

app.BROJTACNIHODGOVORALabel.Position = [340 17 158 22];

app.BROJTACNIHODGOVORALabel.Text = 'BROJ TACNIH ODGOVORA';

% 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 = MatematickaZamka

% 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)

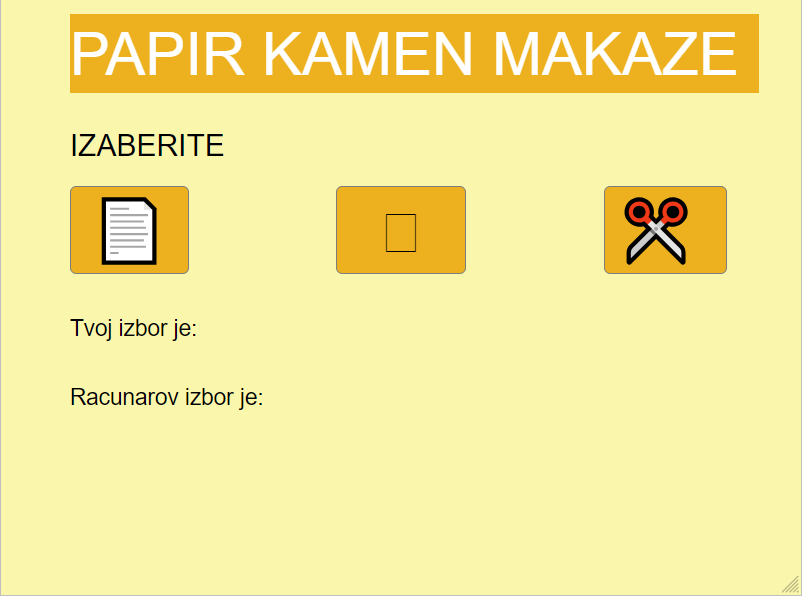
end

end

end

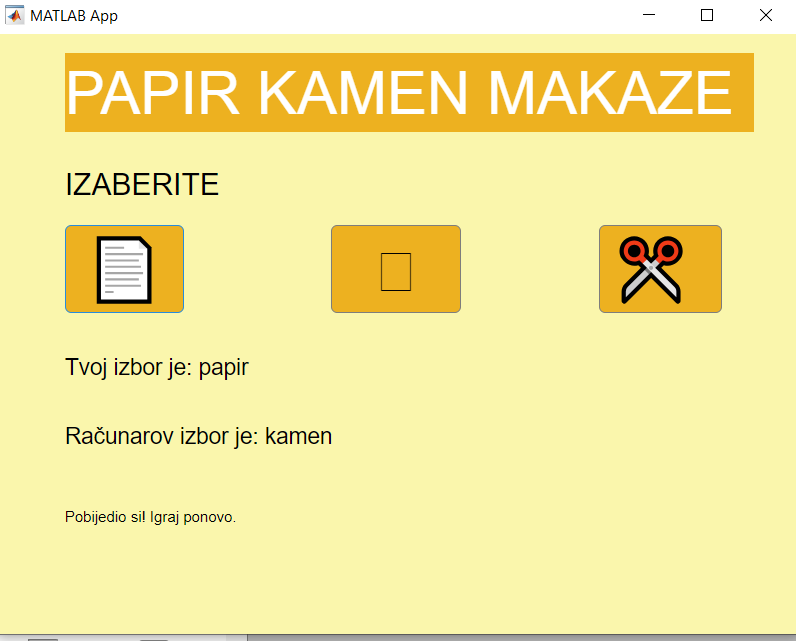
# **3.DRUGA IGRICA: Papir kamen makaze**

Cilj igrice jeste da korisnik igra popularnu igricu papir kamen makaze sa računarom. Korisniku se nudi da izabere koji od elementa on želi, a računar sam random bira jedan od elemenata. Ispisuje se ko je izabrao šta te se bira pobjednik. Onog momenta kad računar pobijedi korisnik gubi, ali se pojavi prozor gdje može pokušati ponovo.



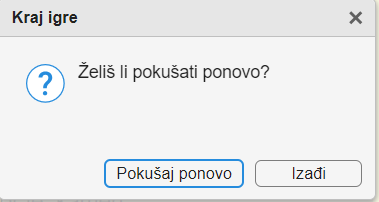
Slika 4: Prikaz igrice papir kamen makaze

Nakon što se igrica pokrene Izbori se popunjavaju te se ispod pojavljuje poruka o tome ko je u rundi pobijedio.



Slika 5: Prikaz nakon odrađene runde

Igra se sve dok korisnik pobjeđuje. Onog momenta kad računarov izbor nadjača korisnikov tada korisnik gubi i pojavljuje se poruka o tome. Korisnik ima dalje pravo na svoj izbor



Slika 6: Prozor nakon što korisnik izgubi

Korisnik može izabrati da li želi izaći iz ove igrice ili želi pokučati ponovo. U odnosu na njegov izbor tako se dalje stvari nastavljaju odvijati.

# **3.1. Kod za drugu igricu: Papir kamen makaze**

classdef pkm < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

UIFigure matlab.ui.Figure

Label matlab.ui.control.Label

RacunarovizborjeLabel matlab.ui.control.Label

TvojizborjeLabel matlab.ui.control.Label

IZABERITELabel matlab.ui.control.Label

Button\_3 matlab.ui.control.Button

Button\_2 matlab.ui.control.Button

Button matlab.ui.control.Button

PAPIRKAMENMAKAZELabel matlab.ui.control.Label

end

properties (Access = private)

BrojPobjeda = 0; %varijabla za brojanje pobjeda

end

methods (Access = private)

function obradiPotez(app, korisnik)

opcije = {'papir', 'kamen', 'makaze'}; %niz od svih mogucih opcija koje racunar moze izabrati

racunar = opcije{randi(3)}; % Računar nasumično bira jedno od 3

% Prikaz izbora korisnika i računara

app.TvojizborjeLabel.Text = ['Tvoj izbor je: ', korisnik];

app.RacunarovizborjeLabel.Text = ['Računarov izbor je: ', racunar];

% Provjera odabrano, u ova tri slucaja pobjeduje korisnik

if (strcmp(korisnik, 'papir') && strcmp(racunar, 'kamen')) || ...

(strcmp(korisnik, 'kamen') && strcmp(racunar, 'makaze')) || ...

(strcmp(korisnik, 'makaze') && strcmp(racunar, 'papir'))

%ispisuje se poruka da je pobijedio korisnik

app.Label.Text = "Pobijedio si! Igraj ponovo.";

app.BrojPobjeda = app.BrojPobjeda + 1; %povecava se broj pobjeda

elseif strcmp(korisnik, racunar) %poredi se izbor korisnika i racunara, ako je isto situacija je nerijesena

app.Label.Text = "Neriješeno!";

else %ako korisnik nije pobijedio, nisu isti izbori znaci da je korisnik izubio

app.Label.Text = ['Izgubio si. Ukupno pobjeda: ', num2str(app.BrojPobjeda)];

izbor = uiconfirm(app.UIFigure, ... %skocni prozor koji se pojavi kada korisnik izgubi

'Želiš li pokušati ponovo?', ...

'Kraj igre', ...

'Options', {'Pokušaj ponovo', 'Izađi'}, ... %dvije opcije u slucaju gubitka

'DefaultOption', 1, ...

'CancelOption', 2);

if strcmp(izbor, 'Pokušaj ponovo') %znaci da se ponovo igra

app.BrojPobjeda = 0; %resetuje se broj pobjeda na 0

app.Label.Text = 'Igra počinje ponovo!';

else

delete(app.UIFigure); % Zatvori aplikaciju ako korisnik izabe da zeli izaći

end

end

end

end

% Callbacks that handle component events

methods (Access = private)

% Button pushed function: Button\_3

function Button\_3Pushed(app, event)

korisnik = 'makaze'; %ako klikne na 3 button njegova opcija su makaze

obradiPotez(app, korisnik); %odraduje se potez shodno tome koju opciju je korisnik unio

end

% Button pushed function: Button\_2

function Button\_2Pushed(app, event)

korisnik = 'kamen';

obradiPotez(app, korisnik);

end

% Button pushed function: Button

function ButtonPushed(app, event)

korisnik = 'papir';

obradiPotez(app, korisnik);

end

end

%DIO KOJI SE SAM GENERIŠE

% Component initialization

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Create UIFigure and hide until all components are created

app.UIFigure = uifigure('Visible', 'off');

app.UIFigure.Color = [0.9804 0.9647 0.6745];

app.UIFigure.Position = [100 100 640 480];

app.UIFigure.Name = 'MATLAB App';

% Create PAPIRKAMENMAKAZELabel

app.PAPIRKAMENMAKAZELabel = uilabel(app.UIFigure);

app.PAPIRKAMENMAKAZELabel.BackgroundColor = [0.9294 0.6941 0.1255];

app.PAPIRKAMENMAKAZELabel.FontSize = 48;

app.PAPIRKAMENMAKAZELabel.FontColor = [1 1 1];

app.PAPIRKAMENMAKAZELabel.Position = [56 403 551 63];

app.PAPIRKAMENMAKAZELabel.Text = 'PAPIR KAMEN MAKAZE';

% Create Button

app.Button = uibutton(app.UIFigure, 'push');

app.Button.ButtonPushedFcn = createCallbackFcn(app, @ButtonPushed, true);

app.Button.BackgroundColor = [0.9294 0.6941 0.1255];

app.Button.FontSize = 48;

app.Button.Position = [56 258 95 70];

app.Button.Text = '📄';

% Create Button\_2

app.Button\_2 = uibutton(app.UIFigure, 'push');

app.Button\_2.ButtonPushedFcn = createCallbackFcn(app, @Button\_2Pushed, true);

app.Button\_2.BackgroundColor = [0.9294 0.6941 0.1255];

app.Button\_2.FontSize = 48;

app.Button\_2.Position = [269 258 104 70];

app.Button\_2.Text = '🪨';

% Create Button\_3

app.Button\_3 = uibutton(app.UIFigure, 'push');

app.Button\_3.ButtonPushedFcn = createCallbackFcn(app, @Button\_3Pushed, true);

app.Button\_3.BackgroundColor = [0.9294 0.6941 0.1255];

app.Button\_3.FontSize = 48;

app.Button\_3.Position = [483 258 99 70];

app.Button\_3.Text = '✂️ ';

% Create IZABERITELabel

app.IZABERITELabel = uilabel(app.UIFigure);

app.IZABERITELabel.FontSize = 24;

app.IZABERITELabel.Position = [56 345 129 31];

app.IZABERITELabel.Text = 'IZABERITE';

% Create TvojizborjeLabel

app.TvojizborjeLabel = uilabel(app.UIFigure);

app.TvojizborjeLabel.FontSize = 18;

app.TvojizborjeLabel.Position = [56 203 470 23];

app.TvojizborjeLabel.Text = 'Tvoj izbor je: ';

% Create RacunarovizborjeLabel

app.RacunarovizborjeLabel = uilabel(app.UIFigure);

app.RacunarovizborjeLabel.FontSize = 18;

app.RacunarovizborjeLabel.Position = [56 148 504 23];

app.RacunarovizborjeLabel.Text = 'Racunarov izbor je:';

% Create Label

app.Label = uilabel(app.UIFigure);

app.Label.Position = [56 84 561 22];

app.Label.Text = '';

% 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 = pkm

% Create UIFigure and components

createComponents(app)

% Register the app with App Designer

registerApp(app, app.UIFigure)

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)

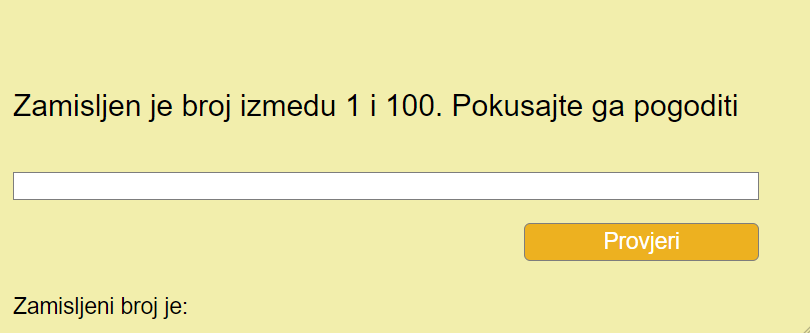
end

end

end

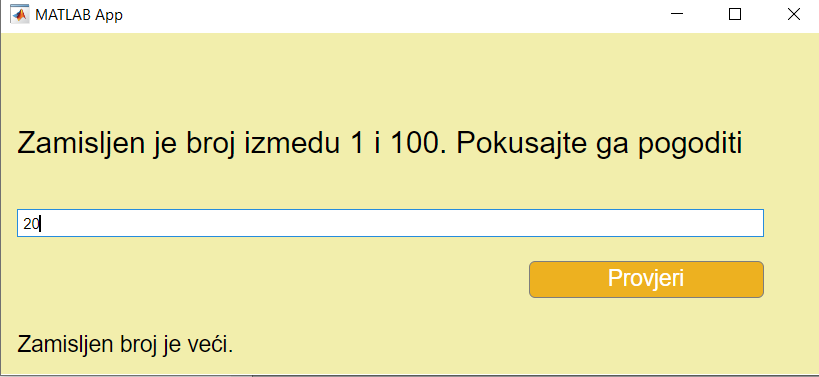
# **4. TREĆA IGRICA: Pogađanje broja od 1 do 100**

Sam prikaz je jednostavan a zanimljiv za korištenje. Imamo jedan label gdje je korisniku objašnjeno šta se od njega traži. Ispod imamo polje za unos teksta. Upravo to je tekst kojeg treba provjeriti da vidimo da li je isti kao generisan od strane računara. On provjerava da li je isti, manji i veći te nakon provjere biva ispisano stanje ispod dijela za unos tekst.



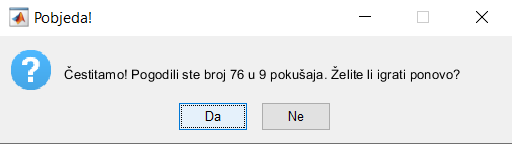
Slika 7: Izgled prozora za igricu pogađanja zamišljenog broja

Klikom na button provjerava se da li je unijeti broj isti kao generisani. Prilikom pokretanja prozora izgled je idući:



Slika 8: Opisuje u kojem smjeru trebamo ići nakon svakog pokušaja

Da bi korisnik lakše razumio sve mu je ispod ispisano. Korisnik može pogađati sve dok zamišljeni i unijeti broj nisu isti. Nakon što se unese tačan broj ispiše se poruka. To ne mora značiti kraj programa zato što se može izabrati opcija da se igra nastavi. U zavisnosti od želje korisnika dalje se nastavlja odvijati rad aplikacije.



Slika 9: Prozor nakon pogotka

Kao što možemo vidjeti na slici dva su buttona. Odabirom gumba ne izlazi se iz igrice. Ovako korisnik može jednostavno pratiti u kom smjeru da se kreće. Ukoliko ne bismo imali ispisivanje kakav je traženi broj traganje za njim bi bilo dugo i vremenom bi postalo dosadno. Ovako se održava dinamičnost, ali isto tako se lakšava put do pobjede.

# **4.1. Kod za igricu podađanja zamišljenog broja**

classdef PogodiBroj < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

UIFigure matlab.ui.Figure

ProvjeriButton matlab.ui.control.Button

ZamisljenibrojjeLabel matlab.ui.control.Label

EditField matlab.ui.control.EditField

Zamisljenjebrojizmedu1i100PokusajtegapogoditiLabel matlab.ui.control.Label

end

properties (Access = private)

ZamisljenBroj %onaj kojeg racunar izabere

Pokusaji = 0 %da bi se na kraju ispisalo nakon koliko pokusaja je pogoden broj

end

methods (Access = private)

%nakon sto se izabere da ne zelimo nastavak vracami se na glavni

%prozor

function povratakNaGlavni(app)

% Zatvori trenutni prozor

delete(app.UIFigure);

igrica; %ime glavnog dijela

end

end

% Callbacks that handle component events

methods (Access = private)

% Code that executes after component creation

function startupFcn(app)

app.ZamisljenBroj = randi([1 100]); %ovo je kao konstruktor za nasu klasu za aplikaciju i odmah tu je potrebno

%da se generise broj da bi korisnik mogao pogadati

app.ZamisljenibrojjeLabel.Text = 'Unesite broj i kliknite Provjeri.';

app.Pokusaji = 0; %stavi se da je 0 na pocetki

end

% Button pushed function: ProvjeriButton

function ProvjeriButtonPushed(app, event)

unosStr = app.EditField.Value;

broj = str2double(unosStr); %pretvaranje stringa u double

if isnan(broj) || broj < 1 || broj > 100 %provjera da li je broj u odgovarajucem opsegu od 1 do 100

uialert(app.UIFigure, 'Molim unesite cijeli broj između 1 i 100!', 'Nevažeći unos');

return

end

app.Pokusaji = app.Pokusaji + 1; %povecava se broj pokusaja

if broj == app.ZamisljenBroj %ako je unijeti broj jednak zamislenom

app.ZamisljenibrojjeLabel.Text = ['Tačno! Pogodili ste broj u ', num2str(app.Pokusaji), ' pokušaja.'];

%Ovaj tekst se ispise u labelu ispod

% Prikaz opcije za nastavak ili izlaz

izbor = questdlg(['Čestitamo! Pogodili ste broj ', num2str(app.ZamisljenBroj), ...%gornji dio prozora

' u ', num2str(app.Pokusaji), ' pokušaja. Želite li igrati ponovo?'], ...

'Pobjeda!', ...

'Da', 'Ne', 'Da');

switch izbor %zavisi od izbora na prozoru

case 'Da'

% Pokreni novu igru - resetuj broj i pokušaje

app.ZamisljenBroj = randi([1 100]);

app.Pokusaji = 0;

app.ZamisljenibrojjeLabel.Text = 'Unesite broj i kliknite Provjeri.';

case 'Ne'

%Zatvara se prozor gdje je bila igrica i otvara glavni

%prozor

delete(app.UIFigure);

igrica();

end

elseif broj < app.ZamisljenBroj

app.ZamisljenibrojjeLabel.Text = 'Zamisljen broj je veći.'; %ispis u zavisnosti da li je zamisljeni veci

%ili manji od unijetog

else

app.ZamisljenibrojjeLabel.Text = 'Zamisljen broj je manji.';

end

app.EditField.Value = ''; % prazno polje za novi unos

end

end

% Component initialization

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Create UIFigure and hide until all components are created

app.UIFigure = uifigure('Visible', 'off');

app.UIFigure.Color = [0.949 0.9333 0.6745];

app.UIFigure.Position = [100 100 658 273];

app.UIFigure.Name = 'MATLAB App';

% Create Zamisljenjebrojizmedu1i100PokusajtegapogoditiLabel

app.Zamisljenjebrojizmedu1i100PokusajtegapogoditiLabel = uilabel(app.UIFigure);

app.Zamisljenjebrojizmedu1i100PokusajtegapogoditiLabel.FontSize = 24;

app.Zamisljenjebrojizmedu1i100PokusajtegapogoditiLabel.Position = [14 172 597 31];

app.Zamisljenjebrojizmedu1i100PokusajtegapogoditiLabel.Text = 'Zamisljen je broj izmedu 1 i 100. Pokusajte ga pogoditi';

% Create EditField

app.EditField = uieditfield(app.UIFigure, 'text');

app.EditField.Position = [14 112 597 22];

% Create ZamisljenibrojjeLabel

app.ZamisljenibrojjeLabel = uilabel(app.UIFigure);

app.ZamisljenibrojjeLabel.FontSize = 18;

app.ZamisljenibrojjeLabel.Position = [14 15 411 23];

app.ZamisljenibrojjeLabel.Text = 'Zamisljeni broj je: ';

% Create ProvjeriButton

app.ProvjeriButton = uibutton(app.UIFigure, 'push');

app.ProvjeriButton.ButtonPushedFcn = createCallbackFcn(app, @ProvjeriButtonPushed, true);

app.ProvjeriButton.BackgroundColor = [0.9294 0.6941 0.1255];

app.ProvjeriButton.FontSize = 18;

app.ProvjeriButton.FontColor = [1 1 1];

app.ProvjeriButton.Position = [423 63 188 30];

app.ProvjeriButton.Text = 'Provjeri';

% 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 = PogodiBroj

% 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

Poenta je pratiti kakav je broj u odnosu na zamišljeni, a druga bitna stvar je da se mijenja broj pokušaja. Ako se to prati ispis na kraju je jednostavan za to koji broj je zamišljen i koliko puta je korisnik pogađao.

# **ZAKLJUČAK:**

Kreiranje ove aplikacije me naučilo kreativnosti i pravljenju dizajna u Matlabu. Uz nekoliko jednostavnih komponenti igrice su funkcionalne i rade onako kako je prvobitno i zamišljeno. Korištenjem kreativnosti i još par komponenti u Matlabu se može napraviti veliki niz različitih igrica. Iako njegova namjena nije takva, za početnike je zanimljivo iskustvo i super način za učenje logike pojedinih, jednostavnih igrica.

Ovaj projekat me naučio kako se piše kod iza prozora kojeg korisnik vidi te kako u Matlabu organizirati kod korištenjem dijela za deklarisanje i inicijaliziranje varijabli, korištenje privatnih funkcija koje su potrebne samo programeru da logika bude dobra iza prozora. Buttoni imaju svoj kod, međutim to je većinom kraći kod koji poziva određenu funkciju koja je prije deklarisana ili eventualno nekoliko dodatnih linija koda. Cilj je da akcija bude jednostavna, čitka ako neko kasnije bude gledao naš kod.

Ovo kodiranje mi je pomoglo da razmišljam i kao korisnik, ali i kao programer. Htjela sam da za korisnika sve bude razumljivo i stavila sam se u poziciju da sam ja osoba koja će koristiti aplikaciju. Ali da bi sve bilo funkcionalno morala sam razmišljati i kao programer te unaprijediti sve što bi potencijalno korisniku stvaralo problem ili nerazumijevanje.