

Technical university of Cluj-Napoca
-ETTI-

Money&Gold
Exchange and pawn shop

Student: Giurgiu Alexandru CORNEL

Teacher: Cirlugea Mihaela

Year: 2023-2024

Content:

- Introduction
- MatLab History
- Theoretical Presentation
- Experimental Presentation
- Picture of the interface
- Conclusion
- Bibliography
- The code

Introduction

MATLAB, short for MATrix LABoratory, is a high-performance programming language and environment primarily used for numerical computing, data analysis, and visualization. Developed by MathWorks, MATLAB has a rich history dating back to the late 1970s when it was initially created by Cleve Moler. Over the years, it has evolved into a versatile tool widely employed in various fields, including engineering, finance, and scientific research.

My project, "Money & Gold," uses MATLAB to make a user-friendly computer program. It is a graphical user interface that has buttons and windows that help people with money exchange and pawnshop activities. It's like a helpful tool on your computer screen.

In the money world of computers, many programs do similar things. My GUI/ is a bit special because we focused on making it easy to use and nice to look at. We know there are other programs, but we tried to make ours the friendliest.

My project stands within the context of existing financial software solutions. While we acknowledge the multitude of projects addressing similar functionalities, our emphasis lies in delivering an accessible and visually appealing interface for users engaging in currency exchange and pawnshop transactions.

Theoretical Presentation

This GUI combines theory and practicality in financial technology using MATLAB. I've created a user-friendly Graphical User Interface (GUI) for currency exchange and pawnshop transactions, focusing on simplicity and visual appeal.

For creating this GUI I needed a function in MatLab called GUI.m and a class of methods named "conversionFunction.m". For the main window, the title and the buttons I've used this formulas.

```

mainFig = figure('Name', 'Money & Gold', 'NumberTitle', 'off', 'Position', [100, 100, 800, 600], 'Color', [0.8, 0.9, 1]); |

mainAxes = axes('Parent', mainFig, 'Position', [0, -0.20, 1.2, 2]);
backgroundImage = imread('money.jpg');
imshow(backgroundImage, 'Parent', mainAxes);

exchangeButton = uicontrol('Style', 'pushbutton', 'String', 'Exchange', 'Position', [200, 400, 400, 100], 'Callback', ...
    @exchangeButton_Callback, 'FontSize', 18, 'BackgroundColor', [0.4, 0.7, 1], 'ForegroundColor', [1, 1, 1]);

amanetButton = uicontrol('Style', 'pushbutton', 'String', 'Amanet', 'Position', [200, 250, 400, 100], 'Callback', ...
    @amanetButton_Callback, 'FontSize', 18, 'BackgroundColor', [1, 0.9, 0.5], 'ForegroundColor', [0, 0, 0]);

docButton = uicontrol('Style', 'pushbutton', 'String', 'Documentation', 'Position', [200, 100, 400, 50], ...
    'Callback', @openDocumentation, 'FontSize', 16, 'BackgroundColor', [0.7, 0.7, 0.7], 'ForegroundColor', [1, 1, 1]);

titleText = uicontrol('Style', 'text', 'String', 'Money & Gold', 'Position', [200, 550, 400, 40], ...
    'FontSize', 24, 'FontWeight', 'bold', 'HorizontalAlignment', 'center', ...
    'BackgroundColor', [0.8, 0.8, 0.8, 0.8], 'ForegroundColor', [0, 0, 0]);

```

In the “Exchange” feature the user can convert his money through a range of currencies, including Euro (EUR), US Dollar (USD), Romanian Leu (RON), British Pound (GBP), Japanese Yen (JPY), Swiss Franc (CHF), Mexican Peso (PESO), Danish Krone (DKK), and United Arab Emirates Dirham (AED) and also can convert in three ways (Standard, Priority and Express). For the conversions, the formulas are found in the “conversionFunctions.m”.

```

exchangeFees = [0.02, 0.01, 0.05, 0.03, 0.04, 0.02, 0.01, 0.02, 0.05];
transactionTypes = {'Standard', 'Express', 'Priority'};
marketVolatility = [0.01, 0.02, 0.03];
function result = exchangeConversion(fromCurrency, toCurrency, amount, transactionType)
    exchangeRates = [1, 1.18, 4.87, 0.84, 130.73, 1.05, 23.65, 7.44, 4.02];
    volatilityFactor = 1 + conversionFunctions.marketVolatility(randi([1,
        numel(conversionFunctions.marketVolatility)]));
    result = amount * exchangeRates(toCurrency) / exchangeRates(fromCurrency) * volatilityFactor;
    switch transactionType
        case 'Express'
            result = result - (result * conversionFunctions.exchangeFees(toCurrency) * 1.5);
        case 'Priority'
            result = result - (result * conversionFunctions.exchangeFees(toCurrency) * 2);
        otherwise
            result = result - (result * conversionFunctions.exchangeFees(toCurrency));
    end
end
function symbol = getCurrencySymbol(currency)
    symbols = {'EUR', 'USD', 'RON', 'GBP', 'JPY', 'CHF', 'PESO', 'DKK', 'AED'};
    symbol = symbols{currency};
end

```

This code facilitates currency exchange with predefined rates, fees, and transaction types. The exchangeConversion function converts amounts considering rates, market volatility, and transaction fees. getCurrencySymbol retrieves currency symbols. Key elements include exchangeFees, transactionTypes, marketVolatility and exchangeRates. Volatility introduces randomness and variability into exchange rate simulations.

In the “Exchange” window I also have 2 plots “Conversion Rate (RON to EUR)” and “Conversion Rate (USD to EUR)”

```

historicalDatesRON = datetime('2023-01-01') + caldays(1:30);
historicalRatesRON = rand(1, 30) * 0.1 + 4.8;
historicalDatesUSD = datetime('2023-01-01') + caldays(1:30);
historicalRatesUSD = rand(1, 30) * 0.1 + 1.2;
% Add a subplot for the RON to EUR conversion plot
subplot('Position', [0.65, 0.55, 0.3, 0.3]);
plot(historicalDatesRON, historicalRatesRON, '-*', 'LineWidth', 2, 'MarkerSize', 8);
title('RON to EUR Conversion Over Time');
xlabel('Date');
ylabel('Conversion Rate (RON to EUR)');
grid on;
% Add a subplot for the USD to EUR conversion plot
subplot('Position', [0.65, 0.1, 0.3, 0.3]);
plot(historicalDatesUSD, historicalRatesUSD, '-*', 'LineWidth', 2, 'MarkerSize', 8);
title('USD to EUR Conversion Over Time');
xlabel('Date');
ylabel('Conversion Rate (USD to EUR)');
grid on;

```

The other feature of my GUI is the “Amanet” one calculates the value of gold in euros based on weight, current gold value, and simulated fluctuations. Additionally, it allows for extra evaluation depending on the chosen method (purity or craftsmanship). The weight refers to the physical mass of the gold item, purity relates to the gold's composition in terms of pure gold content versus impurities or alloying metals, 24-karat gold is considered pure gold and the craftsmanship takes into account the quality of craftsmanship in shaping or designing the gold item, this factor recognizes the artistic and aesthetic aspects of the gold piece.. The result represents the estimated value of the gold item in euros for pawnshop transactions.

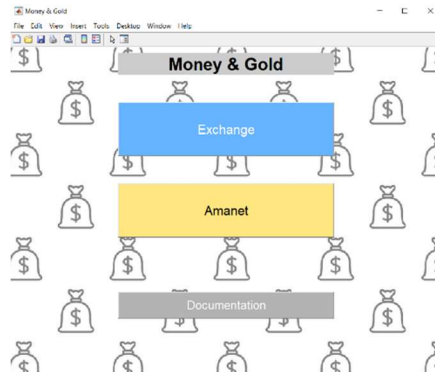
```

goldValuePerOunce = 1800;
goldEvaluationMethods = {'Weight', 'Purity', 'Craftsmanship'};
goldFluctuationRange = [0.95, 1.05];
function result = amanetConversion(goldWeight, evaluationMethod)
    gramToOunceConversion = 0.03527396;
    goldValueFactor = conversionFunctions.goldFluctuationRange(1) + rand() *
    (conversionFunctions.goldFluctuationRange(2) - conversionFunctions.goldFluctuationRange(1));
    result = goldWeight * gramToOunceConversion * conversionFunctions.goldValuePerOunce * goldValueFactor;
witch evaluationMethod
    case 'Purity'
        result = result * 1.2; % Example of additional evaluation for purity
    case 'Craftsmanship'
        result = result * 1.5; % Example of additional evaluation for
craftsmanship
    otherwise

```

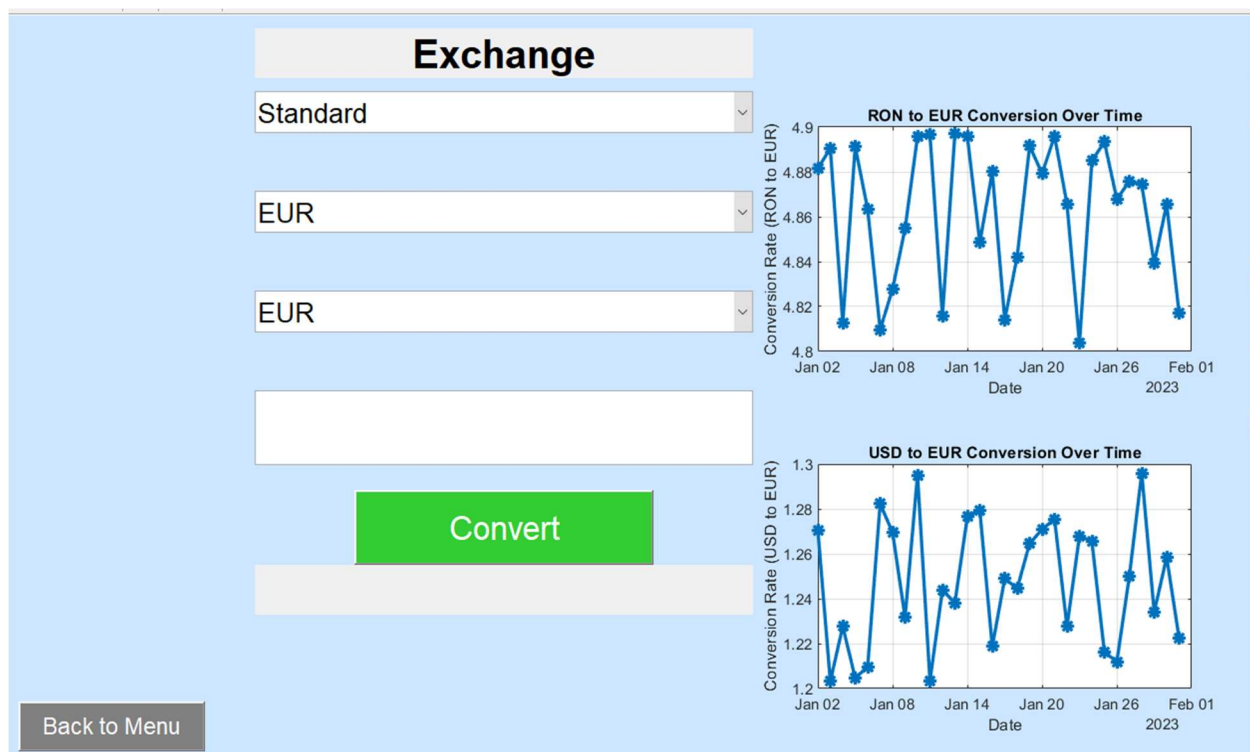
Experimental part

When you run the code , the main window has a light blue font with a image as background,creating a welcome interface.I've added here a title”Money&Gold” and three buttons “Exchange” “Amanet” and “Documentation”,for easy navigation,ensuring users can access the features.



The main window

The”Exchange”feature opens a new window where users can convert currencies(RON,EUR,USD,GBP,AED,PESO,JPY,CHF,DKK) in three ways(Standard,Express,Priority).

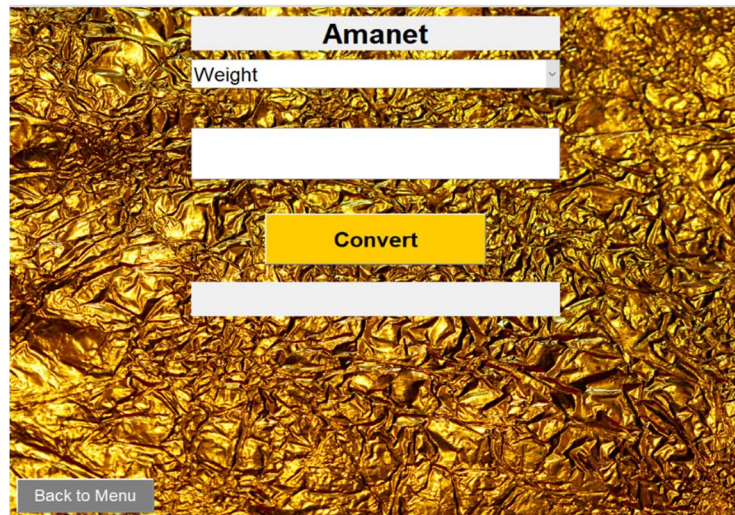


Let's try some examples on this:

1. We convert 10 Ron into EUR in a Standard way and the result is:2.07EUR(If we convert in Priority and Express way the result is 2.01 EUR which is closer to the value from the internet)
2. We have 10 GBP and we want to convert into USD in a Priority way.The result is 14.18 USD(in standard and express the result is 14.05,respectively,14.11)

It's important to note that the conversions may have a slight error compared to online rates due to the dynamic nature of exchange rates influenced by fluctuations and volatility.

The “Amanet” feature is the second feature of my GUI,that opens a window where users can convert Gold depending on the “Weight”,”Purity”,”Craftsmanship”.The window has a background image. The feature aims to offer users a quick evaluation for pawnshop transactions, but the actual value may vary based on market conditions and specific characteristics of the gold item.



Examples:

23 grams of gold is 1520.31 EUR,23 grams of gold but now is pure is 1696.66 EUR,and in the craftsmanship way 23 grams of gold is 2138.77 EUR

The differences between weight, purity, and craftsmanship significantly influence the assessed value of gold items. However, it's important to note that the values obtained in this project may have slight variations compared to those found on the internet.

Fluctuations in rates due to market volatility introduce minor errors, contributing to deviations from online values.

Both features “Exchange” and “Amanet” has a “Back to menu” button and a “Convert” Button. The “Back to Menu” button returns the user to the main menu, providing a way to navigate back to the initial screen of the application. The “Convert” button executes the conversion process based on the user's input. In the exchange feature, it calculates the converted amount after considering exchange rates, fees, and market volatility. In the amanet feature, it computes the value of gold, incorporating weight, gold value, and fluctuations, as well as any additional evaluation based on the chosen method (purity or craftsmanship).

The functions for these buttons:

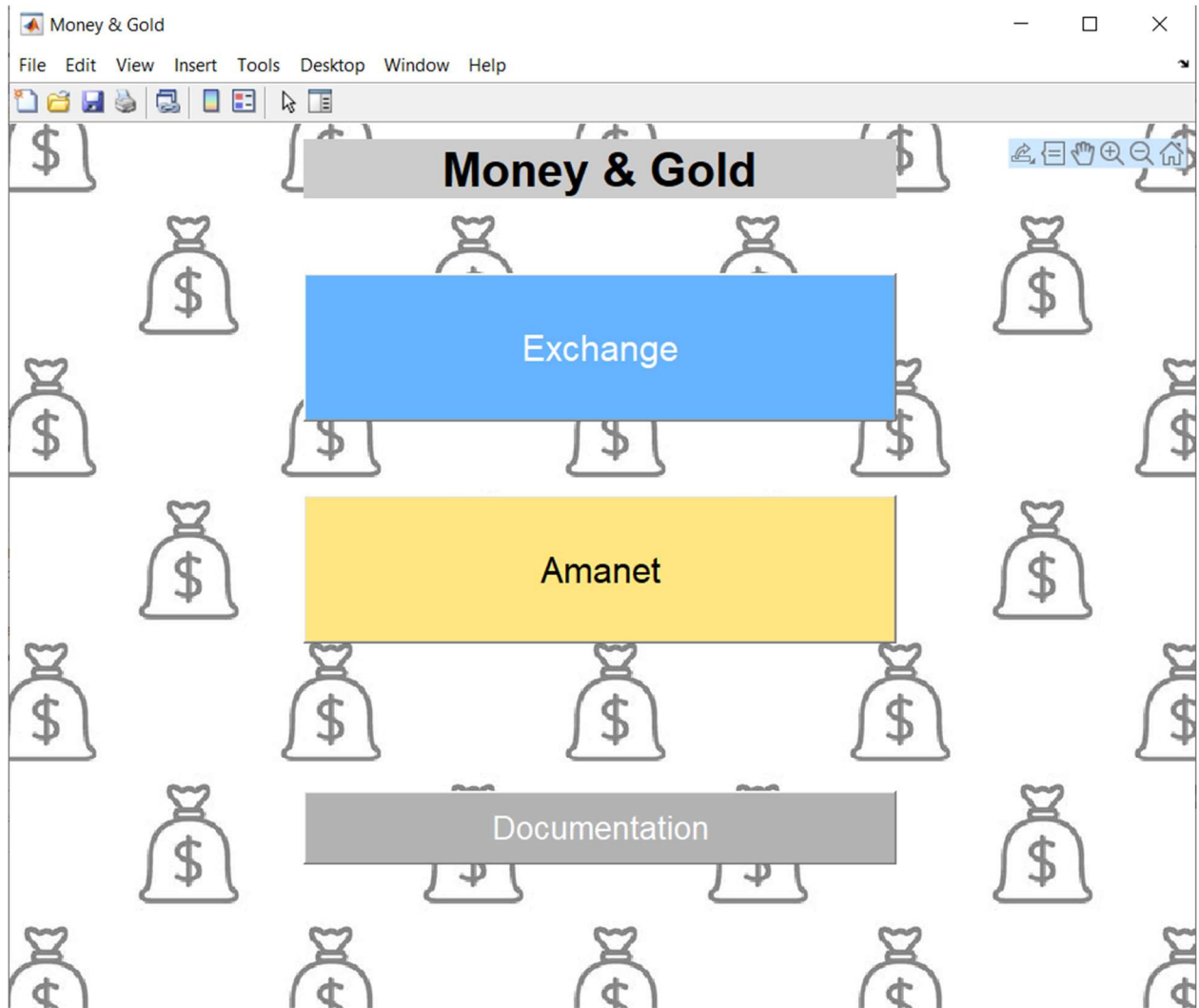
```
convertButton = uicontrol('Style', 'pushbutton', 'String', 'Convert', 'Position',  
[280, 160, 240, 60], 'Callback' ...  
    , @convertButton_Callback, 'FontSize', 20, 'BackgroundColor', [0.2, 0.8,  
0.2], 'ForegroundColor', [1, 1, 1]);  
backToMenuButton = uicontrol('Style', 'pushbutton', 'String', 'Back to Menu',  
'Position', [10, 10, 150, 40], 'Callback' ...  
    , @backToMenu, 'FontSize', 14, 'BackgroundColor', [0.5, 0.5, 0.5],  
'ForegroundColor', [1, 1, 1]);
```

The third feature on my GUI, is placed on the main interface and is the “Documentation” button. This button opens the word/pdf documentation about my project/GUI. If the file does not exist it displays an error message.

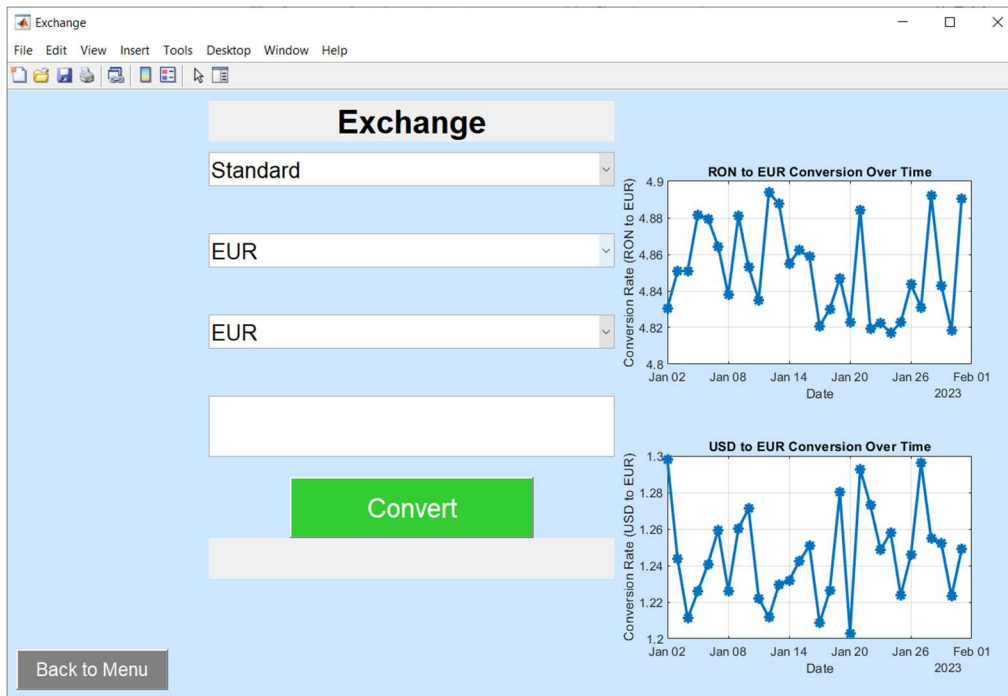
The function for this button:

```
docButton = uicontrol('Style', 'pushbutton', 'String', 'Documentation', 'Position',  
[200, 100, 400, 50], ...  
    'Callback', @openDocumentation, 'FontSize', 16, 'BackgroundColor', [0.7, 0.7,  
0.7], 'ForegroundColor', [1, 1, 1]);  
% Callback for the "Documentation" button  
function openDocumentation(~, ~)  
  
    docFilePath = 'Documentation GUI.docx';  
  
    % Check if the file exists  
    if exist(docFilePath, 'file') == 2  
        % open the Word document  
        winopen(docFilePath);  
    else  
        % Display a message if the file is not found  
        msgbox('Documentation file not found.', 'File Not Found', 'warn');  
    end  
end
```

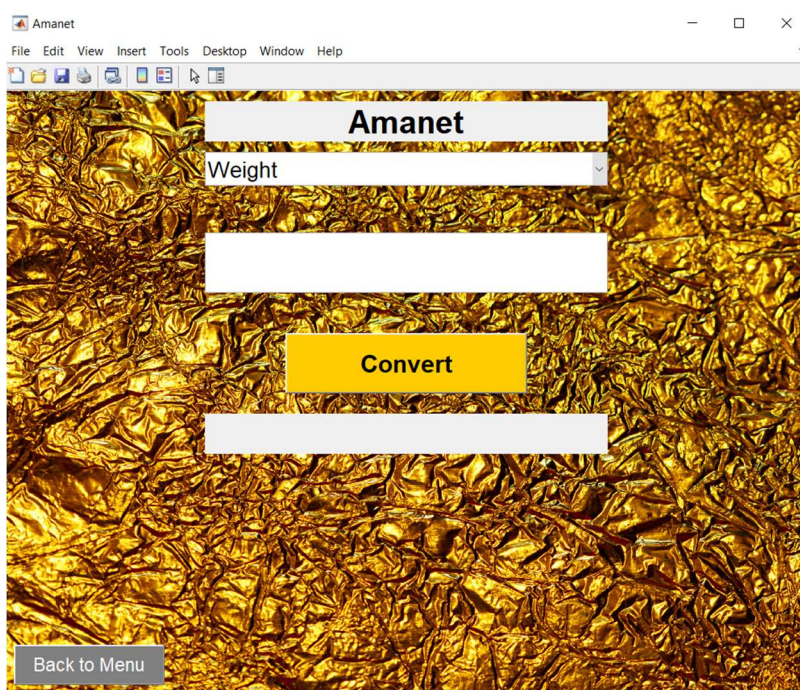

Picture of the Interface:



THE MAIN INTERFACE



The exchange window



The Amanet window

Conclusion

All in all, the "Money & Gold" MATLAB project successfully combines theoretical finance concepts with practical GUI implementation. Results aligned with expectations, and potential areas for improvement were identified. Future research could focus on refining exchange rate models, enhancing GUI aesthetics, or integrating additional financial features.

Bibliography

<https://en.wikipedia.org/wiki/MATLAB>

https://en.wikipedia.org/wiki/Exchange_rate

https://ro.wikipedia.org/wiki/Casă_de_amanet

<https://www.cursbnr.ro/grafic-valute>

The code

GUI.m

```
function GUI

    mainFig = figure('Name', 'Money & Gold', 'NumberTitle', 'off', 'Position', [100, 100, 800, 600], 'Color', [0.8, 0.9, 1]);

    mainAxes = axes('Parent', mainFig, 'Position', [0, -0.20, 1.2, 2]);
    backgroundImage = imread('money.jpg');
    imshow(backgroundImage, 'Parent', mainAxes);

    exchangeButton = uicontrol('Style', 'pushbutton', 'String', 'Exchange', 'Position', [200, 400, 400, 100], 'Callback', ...
        @exchangeButton_Callback, 'FontSize', 18, 'BackgroundColor', [0.4, 0.7, 1], 'ForegroundColor', [1, 1, 1]);

    amanetButton = uicontrol('Style', 'pushbutton', 'String', 'Amanet', 'Position', [200, 250, 400, 100], 'Callback', ...
```

```

@amanetButton_Callback, 'FontSize', 18, 'BackgroundColor', [1, 0.9, 0.5],
'ForegroundColor', [0, 0, 0]);

docButton = uicontrol('Style', 'pushbutton', 'String', 'Documentation',
'Position', [200, 100, 400, 50], ...
'Callback', @openDocumentation, 'FontSize', 16, 'BackgroundColor', [0.7, 0.7,
0.7], 'ForegroundColor', [1, 1, 1]);

titleText = uicontrol('Style', 'text', 'String', 'Money & Gold', 'Position', [200,
550, 400, 40], ...
'FontSize', 24, 'FontWeight', 'bold', 'HorizontalAlignment', 'center', ...
'BackgroundColor', [0.8, 0.8, 0.8, 0.8], 'ForegroundColor', [0, 0, 0]);

function exchangeButton_Callback(~, ~)

delete(mainFig);

historicalDatesRON = datetime('2023-01-01') + caldays(1:30);
historicalRatesRON = rand(1, 30) * 0.1 + 4.8;
historicalDatesUSD = datetime('2023-01-01') + caldays(1:30);
historicalRatesUSD = rand(1, 30) * 0.1 + 1.2;

exchangeFig = figure('Name', 'Exchange', 'NumberTitle', 'off', 'Position', [100,
100, 1000, 600], 'Color', [0.8, 0.9, 1]);

titleText = uicontrol('Style', 'text', 'String', 'Exchange', 'Position', [200,
550, 400, 40], 'FontSize', 24, 'FontWeight', 'bold', 'HorizontalAlignment', 'center',
'ForegroundColor', [0, 0, 0]);

transactionTypeDropDown = uicontrol('Style', 'popupmenu', 'String',
conversionFunctions.transactionTypes, 'Position', [200, 480, 400, 60], 'FontSize',
16);
fromCurrencyDropDown = uicontrol('Style', 'popupmenu', 'String', {'EUR', 'USD',
'RON', 'GBP', 'JPY', 'CHF', 'PESO', 'DKK', 'AED'}, 'Position', [200, 400, 400, 60],
'FontSize', 16);
toCurrencyDropDown = uicontrol('Style', 'popupmenu', 'String', {'EUR', 'USD',
'RON', 'GBP', 'JPY', 'CHF', 'PESO', 'DKK', 'AED'}, 'Position', [200, 320, 400, 60],
'FontSize', 16);

amountEdit = uicontrol('Style', 'edit', 'Position', [200, 240, 400, 60],
'FontSize', 16);
convertButton = uicontrol('Style', 'pushbutton', 'String', 'Convert', 'Position',
[280, 160, 240, 60], 'Callback' ...
, @convertButton_Callback, 'FontSize', 20, 'BackgroundColor', [0.2, 0.8,
0.2], 'ForegroundColor', [1, 1, 1]);
backToMenuButton = uicontrol('Style', 'pushbutton', 'String', 'Back to Menu',
'Position', [10, 10, 150, 40], 'Callback' ...
, @backToMenu, 'FontSize', 14, 'BackgroundColor', [0.5, 0.5, 0.5],
'ForegroundColor', [1, 1, 1]);
resultLabel = uicontrol('Style', 'text', 'Position', [200, 120, 400, 40],
'HorizontalAlignment', 'center', 'FontSize', 16);

subplot('Position', [0.65, 0.55, 0.3, 0.3]);

```

```

8);
plot(historicalDatesRON, historicalRatesRON, '-*', 'LineWidth', 2, 'MarkerSize',
title('RON to EUR Conversion Over Time');
xlabel('Date');
ylabel('Conversion Rate (RON to EUR)');
grid on;

subplot('Position', [0.65, 0.1, 0.3, 0.3]);
8);
plot(historicalDatesUSD, historicalRatesUSD, '-*', 'LineWidth', 2, 'MarkerSize',
title('USD to EUR Conversion Over Time');
xlabel('Date');
ylabel('Conversion Rate (USD to EUR)');
grid on;

function convertButton_Callback(~, ~)
    fromCurrency = get(fromCurrencyDropDown, 'Value');
    toCurrency = get(toCurrencyDropDown, 'Value');
    amount = str2double(get(amountEdit, 'String'));
    transactionType =
conversionFunctions.transactionTypes{get(transactionTypeDropDown, 'Value')};
    if isnan(amount)
        set(resultLabel, 'String', 'Enter a valid amount');
    else
        result = conversionFunctions.exchangeConversion(fromCurrency, toCurrency,
amount, transactionType);
        set(resultLabel, 'String', sprintf(' %.2f %s', result,
conversionFunctions.getCurrencySymbol(toCurrency)));
    end
end
end
function amanetButton_Callback(~, ~)

    delete(mainFig);

    amanetFig = figure('Name', 'Amanet', 'NumberTitle', 'off', 'Position', [100, 100,
800, 600], 'Color', [1, 0.95, 0.8]);

    amanetAxes = axes('Parent', amanetFig, 'Position', [0, -0.1, 1.2, 1.2]);
    amanetImage = imread('gold2.jpg');
    imshow(amanetImage, 'Parent', amanetAxes);

    titleText = uicontrol('Style', 'text', 'String', 'Amanet', 'Position', [200, 550,
400, 40], 'FontSize', 24, 'FontWeight', 'bold', 'HorizontalAlignment', 'center',
'ForegroundColor', [0, 0, 0]);

    evaluationMethodDropDown = uicontrol('Style', 'popupmenu', 'String',
conversionFunctions.goldEvaluationMethods, 'Position', [200, 480, 400, 60],
'FontSize', 16);

    goldAmountEdit = uicontrol('Style', 'edit', 'Position', [200, 400, 400, 60],
'FontSize', 16);

```

```

    % Add "Convert" button with an improved appearance
    convertButton = uicontrol('Style', 'pushbutton', 'String', 'Convert', 'Position',
[280, 300, 240, 60], 'Callback', @convertButton_Callback, 'BackgroundColor', [1, 0.8,
0], 'ForegroundColor', [0, 0, 0], 'FontSize', 18, 'FontWeight', 'bold');

    % Add "Back to Menu" button with an improved appearance
    backToMenuButton = uicontrol('Style', 'pushbutton', 'String', 'Back to Menu',
'Position', [10, 10, 150, 40], 'Callback', @backToMenu, 'FontSize', 14,
'BackgroundColor', [0.5, 0.5, 0.5], 'ForegroundColor', [1, 1, 1]);

    resultLabel = uicontrol('Style', 'text', 'Position', [200, 240, 400, 40],
'HorizontalAlignment', 'center', 'FontSize', 16);

    function convertButton_Callback(~, ~)
        goldAmount = str2double(get(goldAmountEdit, 'String'));
        evaluationMethod =
conversionFunctions.goldEvaluationMethods{get(evaluationMethodDropDown, 'Value')};

        if isnan(goldAmount)
            set(resultLabel, 'String', 'Enter a valid amount');
        else
            result = conversionFunctions.amanetConversion(goldAmount,
evaluationMethod);
            set(resultLabel, 'String', sprintf(' %.2f EUR', result));
        end
    end
end

    function openDocumentation(~, ~)
        docFilePath = 'Documentation GUI.docx';

        if exist(docFilePath, 'file') == 2
            winopen(docFilePath);
        else
            msgbox('Documentation file not found.', 'File Not Found', 'warn');
        end
    end

    function backToMenu(~, ~)
        close(gcf);
        GUI
    end
end
end

```

conversionsFunctions.m

```

classdef conversionFunctions
    properties(Constant)
        goldValuePerOunce = 1800
        exchangeFees = [0.02, 0.01, 0.05, 0.03, 0.04, 0.02, 0.01, 0.02, 0.05];
        transactionTypes = {'Standard', 'Express', 'Priority'};
        goldEvaluationMethods = {'Weight', 'Purity', 'Craftsmanship'}
        marketVolatility = [0.01, 0.02, 0.03

```

```

        goldFluctuationRange = [0.95, 1.05];
    end

    methods(Static)
        function result = exchangeConversion(fromCurrency, toCurrency, amount,
transactionType)
            exchangeRates = [1, 1.18, 4.87, 0.84, 130.73, 1.05, 23.65, 7.44, 4.02];

            volatilityFactor = 1 + conversionFunctions.marketVolatility(randi([1,
numel(conversionFunctions.marketVolatility)]));
            result = amount * exchangeRates(toCurrency) / exchangeRates(fromCurrency)
* volatilityFactor;
            switch transactionType
                case 'Express'
                    result = result - (result *
conversionFunctions.exchangeFees(toCurrency) * 1.5);
                case 'Priority'
                    result = result - (result *
conversionFunctions.exchangeFees(toCurrency) * 2);
                otherwise
                    result = result - (result *
conversionFunctions.exchangeFees(toCurrency));
            end
        end

        function result = amanetConversion(goldWeight, evaluationMethod)
            gramToOunceConversion = 0.03527396;

            goldValueFactor = conversionFunctions.goldFluctuationRange(1) + rand() *
(conversionFunctions.goldFluctuationRange(2) -
conversionFunctions.goldFluctuationRange(1));

            result = goldWeight * gramToOunceConversion *
conversionFunctions.goldValuePerOunce * goldValueFactor;

            switch evaluationMethod
                case 'Purity'
                    result = result * 1.2
                case 'Craftsmanship'
                    result = result * 1.5;
                otherwise
            end
        end

        function symbol = getCurrencySymbol(currency)
            symbols = {'EUR', 'USD', 'RON', 'GBP', 'JPY', 'CHF', 'PESO', 'DKK',
'AED'};
            symbol = symbols{currency};
        end

    end
end

```

