به نام خدا

با سلام

امیدوارم بتونم اونچه رو که می خوام بگم بدون پیچیدگی های اضافی و حرفای مبهم بیان کنم.

خب یه سوال: بنظر شما اگه بخوایم یه تصویر متحرک بسازیم باید چیکار کنیم؟ همون طور که می دونید باید توی هر ثانیه چندین تصویر رو پشت سر هم نشون بدیم و حالا هرچه تعداد تصاویر بیشتر باشه کیفیت فیلم ما بهتر می شه(البته تایه حدی باید تعداد تصاویر رو بیشتر کنیم).

حالا فرض کنیم می خوایم یه توپ بکشیم که در راستای افقی صفحه مختصات حرکت کنه. تصمیم می گیریم که برای این کار توی هر ثانیه 25 تصویر رو نشون بدیم. پس بین هر تصویر باید 0.04 ثانیه صبر کنیم . از طرفی برای اینکه توپ از چپ به راست حرکت کنه باید به مختصه X اون یه مقداری اضافه شه. بیاید یه شبه کد ساده برای این توپ بپیچیم!

ابتدای برنامه

یه توپ در مختصات (x,y) رسم کن

0.04 ثانیه صبر کن

توپ رو پاک کن

به مقدار x ده واحد اضافه کن

برگرد به ابتدا

موتور تمام برنامه های گرافیکی همینه.

حالا بیایید وارد بحث اصلی بشیم. ما می خوایم حرکت یه سری آونگ رو مدل سازی کنیم. از دبیرستان که یادتون هست فرمول حرکت آونگ اینه:

$X=A \sin(wt)$ & $w=\sqrt{g/L}$

$Y=\sqrt{L^2-X^2}$

پس اگه ما این فرمولا رو به کامپیوتر بدیم و زمان (t) رو براش مشخص کنیم، مختصات (x,y) آونگا رو توی لحظه t بدست میاریم.

پیشنهادتون چیه اگه بخوایم حرکت آونگا رو بصورت یه فیلم ببینیم؟ خب مثل همون روشی که برای توپ بکار بردیم اینجا هم استفاده می کنیم اما اینجا هر بار زمان رو 0.04 ثانیه جلو می کشیم و مختصات جدید رو حساب می کنیم و بعد یه آونگ اونجا می کشیم:

ابتدای برنامه

با استفاده از فرمولا مختصات (x,y) آونگ رو توی زمان t حساب کن

یه آونگ توی (x,y) رسم کن

0.04 ثانیه صبر کن

آونگ رو پاک کن

به مقدار t 0.04 ثانیه اضافه کن

برگرد به ابتدا

حالا چی داریم؟ یه اونگ که توی صفحه حرکت می کنه. یکمی بریم جلوتر ، باید ده آونگ رسم کنیما! شما رو نمیدونم اما خودم بدجور دلم واسه حلقه لوپ تنگ شده! اون قسمتی بود که x , y روحساب می کرد و یه آونگ توی این مختصات رسم می کرد، اون رو باید بذاریم تویه لوپ تا ده بار تکرار بشه:

ابتدای برنامه

ابتدای لوپ(برای ده بار تکرار کن)

با استفاده از فرمولا مختصات (x,y) آونگ رو توی زمان t حساب کن

یه آونگ توی (x,y) رسم کن

برگرد به ابتدای لوپ

0.04 ثانیه صبر کن

آونگ رو پاک کن

به مقدار t 0.04 ثانیه اضافه کن

برگرد به ابتدا

توجه کنید که مقدار w برای هر آونگ متفاوته پس مختصاتی که برای هر آونگ بدست میاد متفاوت خواهد بود.

تا اینجا خیلی ساده و ابتدایی بود حالا خیلی جدی وارد خود برنامه می شیم ولی فقط موتور اصلی برنامه بدون هیچ گونه کارای اضافه و مخلفات:

برنامش اینه:

```
1.Int  xp,yp;
2.Double x,y,L,t;
3.initwindow(800,600);
4.Do
5.{
6.        For(int  count=1 ; count<=10 ; count ++)
7.        {
8.                L=1.025- count*0.025;
9.                W=sqrt(9.8/L);
10.               X= L*tan(teta)*sin(w*t);
```

```
11.             Y=sqrt(L*L-x*x);
12.             Xp=400+x*400/1;
13.             Yp=y*400/1;
14.             Line(400,0,xp,yp);
15.             fillellipse(xp,yp,20,20);
16.     }
17.     T+=0.04;
18.     Delay(40);
19.     clearviewport();
20.} while(t<=100);
```

3.این دستور پنجره ای که توی اون نمایس داده میشه رو باز می کنه. اندازه این پنجره رو خودمون بهش می دیم که من800در 600 رو دادم. میشه گفت این پنجره همون سیستم مختصاته اما مبدا اون بالا سمت چپ هست.

6. یه حلقه for هست که باعث می شه تویه هر تصویر ده آونگ قرار داشته باشه.

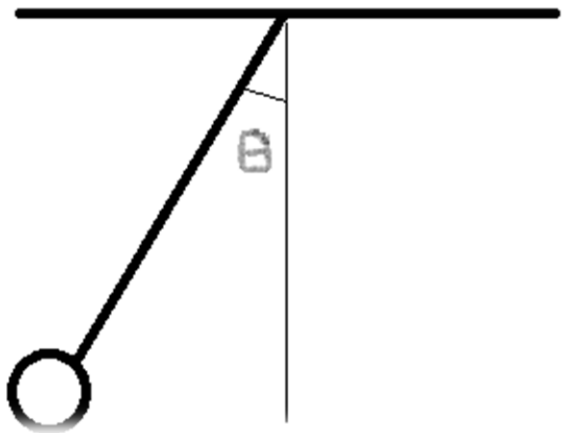
8. یادتونه که قرار بود طول آونگا متفاوت باشه به خاطر همین از مقدار طول بلند ترین آونگ یعنی یک متر0.025 مقدار count کم می کنه که این خودش باعث میشه آونگ هایی با طول های 1 ، 0.975 ، 0.950 ، ... ،0.750 داشته باشیم.

9. مقدار w رو برای هر آونگ حساب می کنه

10.توجه کنین که این همون فرمول x=A sin(wt) هست فقط بجای A از یه فرمول ساده استفاده کردم:

اگه بخوایید با توجه به شکل زیر طول تصویر L رو روی محور xها پیدا کنید چکار می کنید؟

آره میشه : A=L sin(teta)



12 و13. توابع رسم گرافیکی مقادیر صحیح می پذیرند پس مقادیر اعشاری را باید در یک متغیر صحیح قرار دهیم. یه نکته ظریف. گفتم که مبدا مختصات bgi بالا سمت چپه. اما اگه x آونگ ما صفر باشه ما انتظار داریم که اونگ وسط صفحه باشه یعنی x اون باید 400 باشه که نصف طول صفحه یعنی 800 هست. پس توی خط 12 علاوه بر اون که مقدار اعشاری رو به صحیح تبدیل کردم یه انتقال محور هم دادم.(از ریاضی دبیرستان که به یاد دارین). از طرف دیگه شما باید طولتون رو به تعداد پیکسل تبدیل کنین یعنی مثلا یه متر تبدیل شه به 400 پیکسل بخاطر همین مقادیر x و y در 400 پیکسل ضرب و بر یک متر تقسیم شدن.

14. بند آونگ را از نقطه (400,0) به نقطه (x,y) می کشد

دستور کشیدن خط به شکل مقابل است: line(xi,yi,xf,yf)

15. بعنوان گلوله آونگ یه دایره به مرکز (x,y) رسم می کند که شعاع آن 20 است.

17. به t که متغیر زمان ماست 0.04 ثانیه اضافه می کند.

18.این دستور باعث میشه برنامه به مدت 40 ms متوقف بشه. مدت زمان توقف در پرانتز به میلی ثانیه نوشته می شه.

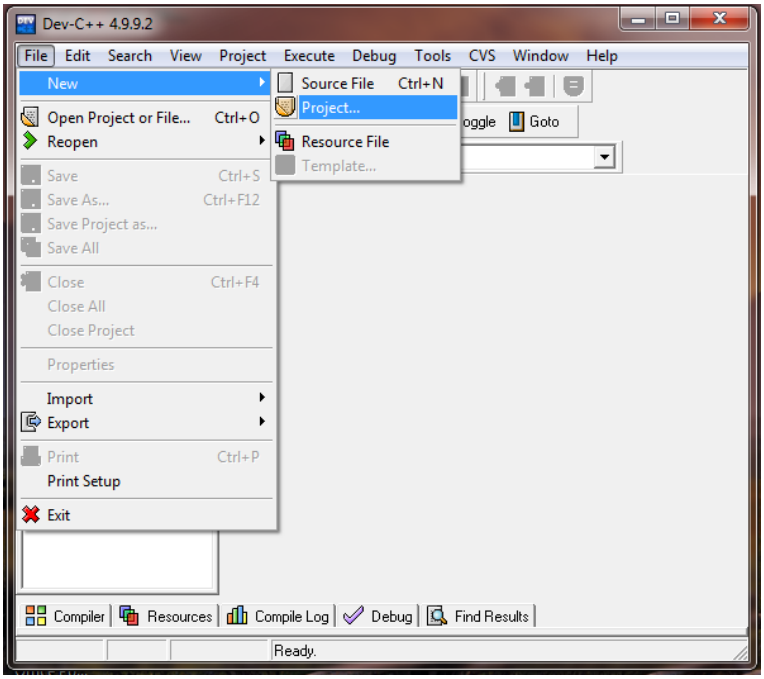
19.این دستور تصویری رو که به مدت 40 ms نمایش داده شده رو پاک می کنه.

20. بررسی می کنه که اگر زمان به 100 ثانیه نرسیده باشه حلقه do ادامه پیدا کنه و دوباره عکس جدید بسازه و نمایش بده و خلاصه فیلم ادامه پیدا کنه.

خب تاحالا با موتور اصلی رسم ده آونگ با طول 1 تا 0.75 آشنا شدیم امیدوارم نکته مبهمی تا اینجا وجود نداشته باشه

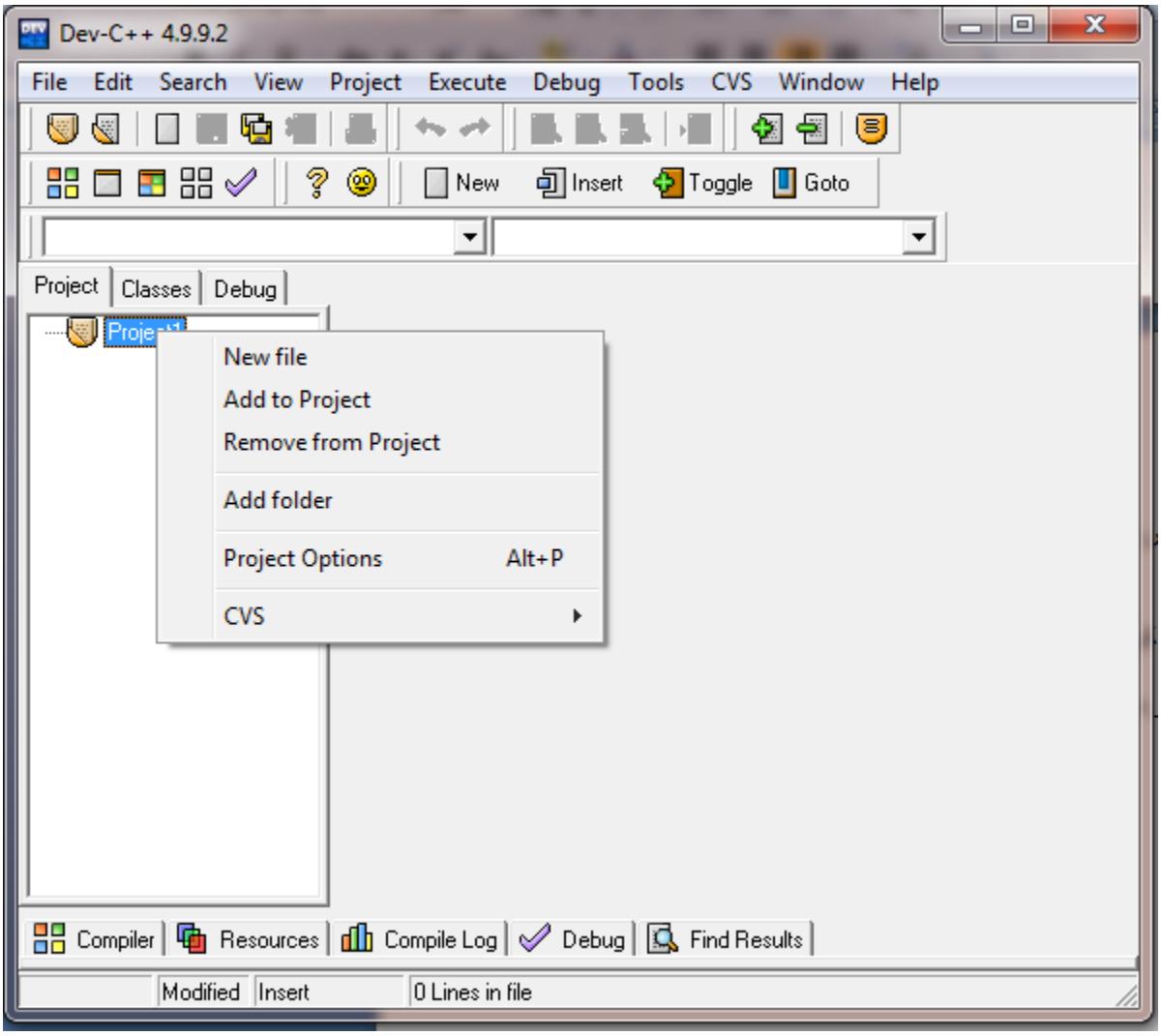
اما واقعا کجای کاریم؟ اگه این برنامه رو به یه کامپایلر بدیم اجرا میشه؟ نه چون باید تابع graphics.h رو به کامپیوتر اضافه کنیم. من تقریبا یه هفته توی اینترنت دنبال این مطلب بودم تا بالاخره کامپایلر dev رو تونستم تنظیم کنم.

برای این کار برنامه dev رو از اینترنت بگیرین و اون رو نصب کنین. اون رو باز کنین و از file توی new ، project رو انتخاب کنین



حالا empty project رو انتخاب کنین و یه اسم برای برنامتون وارد کنین و اونجایی که میخوایید برنامتون ذخیره بشه رو وارد کنین.

حالا روی اون قسمت که آبی رنگه و project نوشته راست کلیک کنین و new file رو انتخاب کنین تا صفحه ای برای نوشتن برنامه باز بشه:



فایل های libbgi.a و graphics.h و winbgim.h رو نمی دونم چطوری ولی یه طوری پیدا کنین و توی همون فایلی که برنامتون رو ذخیره کردین قرار بدین.

دوباره روی project راست کلیک کنین و اینبار روی add to project کلیک کنین و با پنجره باز شده فایل های libbgi.a و graphics.h و winbgim.h رو add کنین.

حالا از اون قسمت بالایی که file و edit و ... رو داره که اسمش یادم رفته (شاید نوار منو) برید توی project و بعد project option . توی پنجره جدید سرپنجره parameters رو انتخاب کنین و در قسمت linker اون کلمات زیر رو عینا بنویسید و بعد ok رو بزنید.

-L.

-lbgi

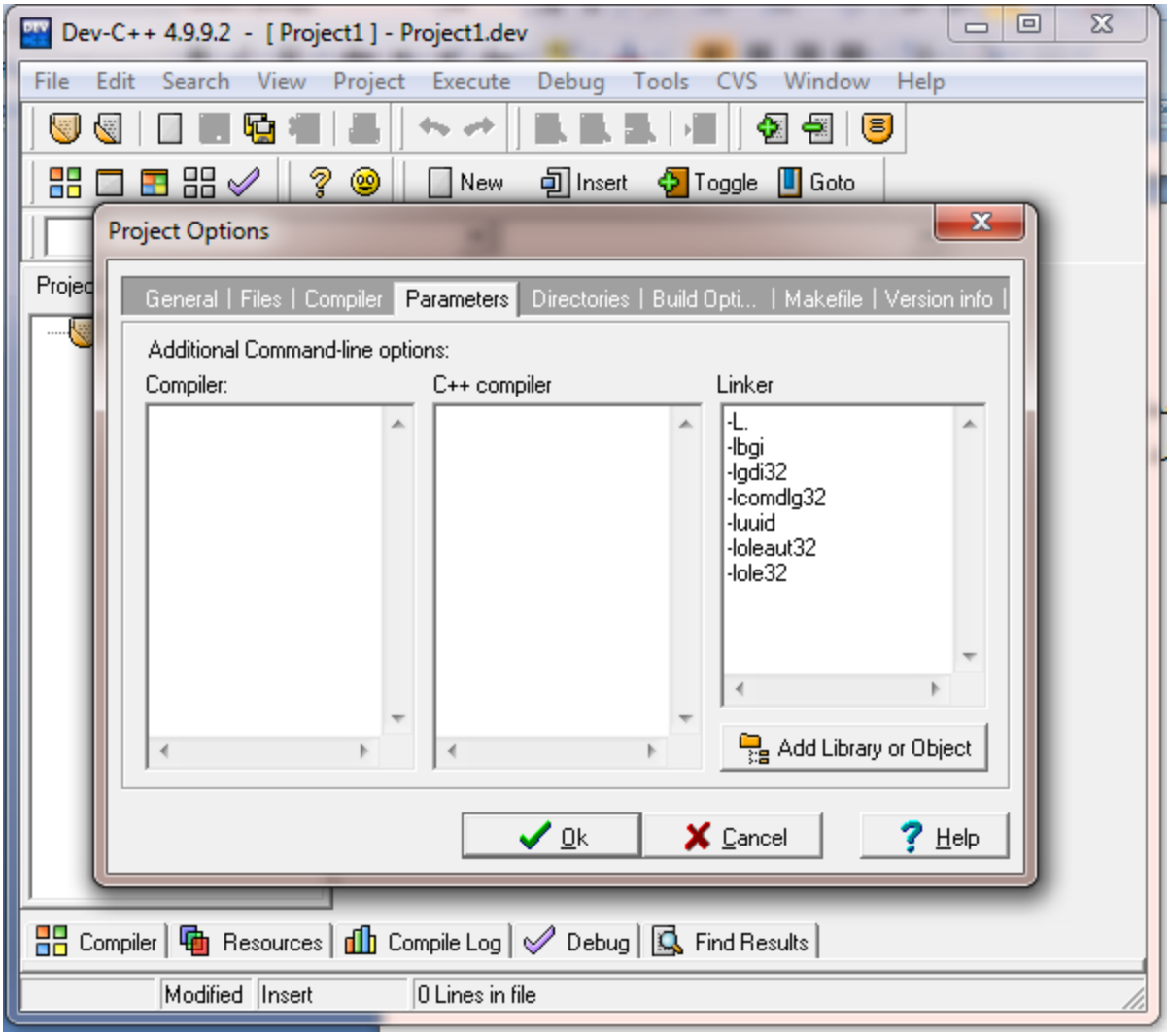
-lgdi32

-lcomdlg32

-luuid

-loleaut32

-lole32



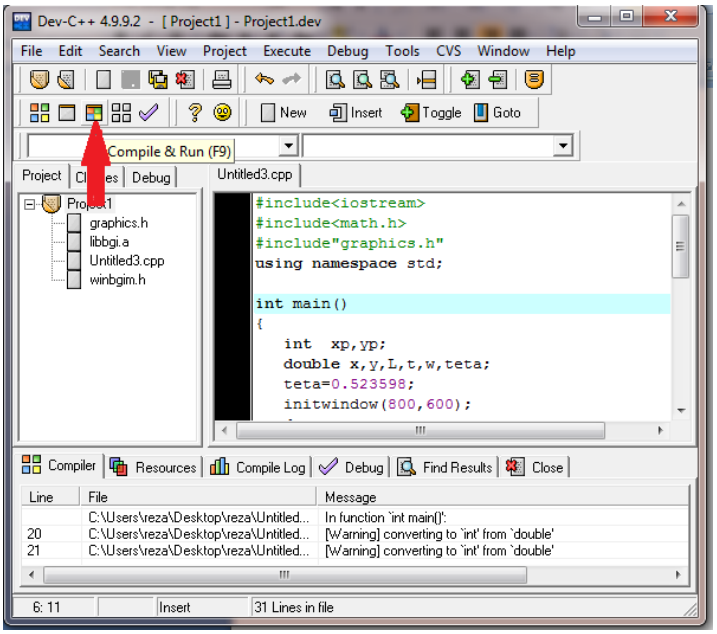
امیدوارم تا اینجا موفق باشین و تونسته باشین تابع graphics.h رو با موفقیت add کرده باشین.

خب اگه برنامه ی زیر رو که درواقع کامل شده ی گفته های بالا هست اجرا کنین آونگا رو می بینید که دارن حرکت می کنن فقط یه توضیح درباره مقدار عجیب و غریب teta=0.523598 بدم، این همون 30 درجه هست که باید با رادیان به برنامه بدیم.

```
#include<iostream>
#include<math.h>
#include"graphics.h"
using namespace std;

int main()
{
  int  xp,yp;
  double x,y,L,t,w,teta;
  teta=0.523598;
  initwindow(800,600);
     do
     {
          for(int  count=1 ; count<=10 ; count ++)
                {
                 L=1.025- count*0.025;
                 w=sqrt(9.8/L);
                 x= L*tan(teta)*sin(w*t);
                 y=sqrt(L*L-x*x)*400;
                 xp=400+x*400;
                 yp=y;
                 line(400,0,xp,yp);
                 fillellipse(xp,yp,20,20);
           }
           t+=0.04;
           delay(40);
           clearviewport();
     }while(t<=100);
  return 0;
}
```

یادتون باشه برای اجرای برنامه توی dev باید دکمه زیر رو بزنید:



خب فکر می کنم وقت اون رسیده که تک تک خط های برنامه اصلی رو بررسی کنیم.

001.این خط معرفی یک کتابخانه خاصه. همانطور که متوجه شدید اون رو میون " " قرار دادم چون این کتابخونه از ابتدا در کامپایلر وجود نداشته به همین دلیل برای تمایز دادن اون با بقیه ی کتاب خونه ها میون " " قرار گرفته.

010. تا 022. این دستورا مقادیر اولیه ی طول و زمان و اصطکاک و زاویه انحراف رو از کاربر می گیره.

024. می دونید که مقادیر ورودی توابع ... sin & cos & tan در C++ از نوع رادیانه. کاربر هم teta رو بر حسب درجه وارد می کنه. وظیفه این خط اینه که درجه رو به رادیان تبدیل کنه.

025.برنامه طول آونگ ابتدایی و انتهایی رو از کاربر می گیره حالا اگر کاربر این طول ها رو برعکس وارد کرده باشه این خط جای اونا رو تعویض می کنه تا در ادامه برنامه مشکلی پیش نیاد.

027.پنجره نمایش رو در ابعاد 600*800 باز میکنه.

030. & 031. w مربوط به آونگ ابتدایی و انتهایی را با دقت بالایی حساب می کند.

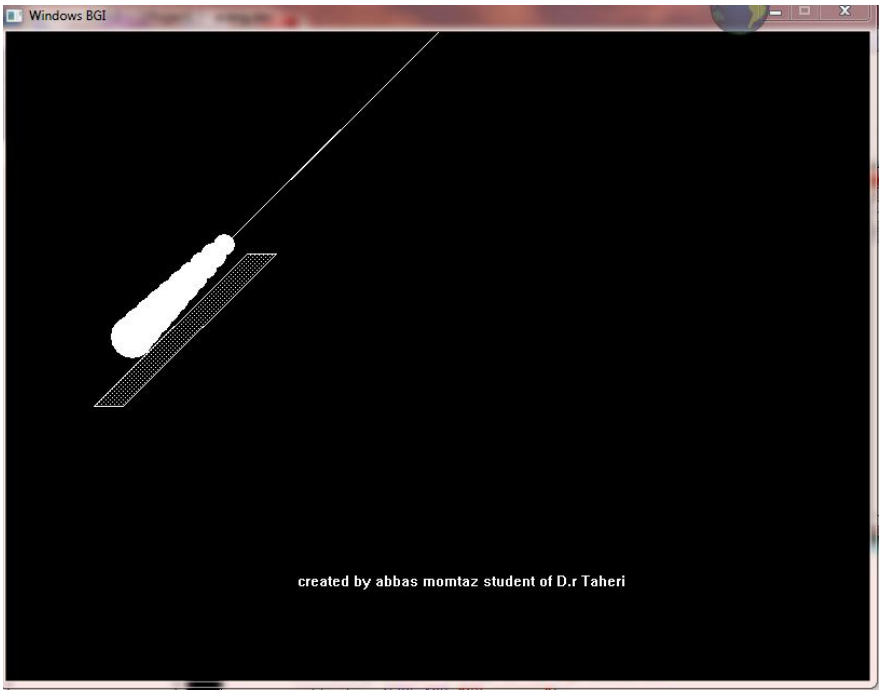
033.& 034. با توجه به w آونگ ابتدایی و انتهایی w بقیه آونگ ها را بصورت خطی پیدا می کند.

038. تا 048. یک خط کش با توجه به زاویه انحراف که خودمون بهش دادیم رسم می کنه.این قسمت برنامه تنها جنبه تزییناتی داره و واقعا توضیح خط به خط اون مشکله چون برای رسم هر ضلع این خط کش باید یه فرمول خط پیدا کنیم و بعد مشخص کنیم که از کجا تا کجای این فرمول باید چاپ بشه. همونطور که گفتم این قصمت فقط جنبه تزییناتی داره اما بهرحال اگه خیلی مشتاق بودین که این فرمولا از کجا اومده روی این فکر کنین که شیب دوضلع بزرگ با شیب نخ اولین آونگ برابره و می تونیم دامنه ی این خطوط رو از سه چهارم تا پنج چهارم مختصه X بلند ترین آونگ بگیریم.

048. درست که این خط توی قسمت بالا بود اما یه دستور گرافیکی خاص هست و باعث میشه سطح داخل یه چند ضلعی به شکل نقطه نقطه پر بشه. که من با اون خط کش رو رنگ کردم!

050. این دستور مثل دستور قبل هست با این تفاوت که سطح ها رو به صورت یک دست و با رنگ سفید پر می کنه.

052. تا 062. این حلقه for آونگ های ما رو روی خط کش رسم می کنه. یه نکته ظریف اینجا هست. من برای اینکه خاصیت پرسپکتیو توی برنامم وجود داشته باشه و برنامه طبیعی تر بشه کوتاهترین آونگ رو با گوی کوچکتری رسم کردم. در حالت عادی اگه آونگ ها رو تحت زاویه انحراف اولیه چاپ کنم آخرین گوی روی خط کش قرار نمی گیره و شکل ضایع زیر بدست می یاد!



برای رفع این مشکل یه فرمول بدست اوردم که در خطوط058. و 059. جنابتان اونا رو ملاحظه می فرماید! این فرمولا هم فقط جنبه تزییناتی داره دیگه اما برای علاقه مندا میگم اگه از مرکز گوی ها یه پاره خط عمود بر خط کش رسم کنیم و مختصاتX & Y اون پاره خط رو پیدا کنیم ، می تونیم با جمع و منها کردن از مختصات اصلی آونگ ها اونا رو روی خط کش بچسبونیم.

064. اسم من و آقای دکتر رو نمایش می ده

065. برنامه رو به مدت 4 ثانیه متوقف می کنه تا کاربر خوب با اسم من و آقای دکتر آشنا بشه!

066. صفحه رو پاک میکنه و برنامه رو آماده ورود به موتور اصلی می کنه.

070. موتور اصلی برنامه تویه یه حلقه for قرار داره و شرط اون کمتر بودن زمان از زمانی هست که کاربر به عنوان زمان پایان برنامه وارد کرده. زمان جاری برنامه رو متغیر t مشخص می کنه که همون طور که متجه شدین هر بار 0.04 ثانیه اون رو جلو می بریم و مختصات آونگا براساس اون مشخس میشه

072. تا 076. عبارت های "D.T" & "T" & "top screen" & "3D screen" رو سر جای خودشون چاپ می کنه.

078. تا 081. این قسمت نمودار زمان و زمان مرگ رو چاپ میکنه. روش کار برای بدست اوردن طول این نمودارا اینه که نسبت T به زمان کلی که کاربر وارد کرده رو بدست میاریم و ضرب در طول کل صفحه یعنی600 (البته به دلایلی ساده کمتر از 600 چون نمودار ما باید توی صفحه جا بشه) می کنیم.

084. یه حلقه for هست دیگه! اگه به برنامه کاملا دقت کرده باشن و سال ها رو اون تفکر نموده باشین متوجه می شین که آونگ ها سه جای مختلف نمایش داده می شن. یکی مرکز صفحه که تصویر اصلی ما هست و دوتا پایین سمت راست. ما توی این حلقهfor مختصات اصلیه هر آونگ رو با توجه به زمان بدست اوردیم و بعد از اون با انتقال محور ها و... اون رو تو سه جا با دستورات خطوط094. و 099. و 104. چاپ کردیم.

086. این خط باعث میشه طول آونگا بطور خطی بین دو مقدار طول اولیه و ثانویه تخییر کنه.

087. این خط مکان X آونگا رو باتوجه به زمان و طول هر آونگ پیدا میکنه.

088. & 089.این دستور اصطکاک رو وارد برنامه می کنه و ر نهایت در t=deathtime حرکت آونگا متوقف میشه.

090. مختصه Y آونگا رو با توجه به X بدست اومده حساب میکنه.

091. & 092. این قسمت سه وظیفه برعهده داره . یکی اینکه انتقال محور میده و دوم اینکه با نسبت گرفتن طول بلند ترین آونگ رو به 400 پیکسل تبدیل میکنه و سوم اینکه مقدار اعشاری رو به مقدار صحیح تبدیل میکنه تا برای توابع گرافیکی آماده بشه.

093. گفتم که برای اینکه فضای برنامه زیباتر و به واقعیت نزدیک تر بشه دورترین آونگ رو با سایز کوچکتری رسم کردم. این خط یه فرمول با توجه به i که یه شمارندهه هست که سایز رو برای هر آونگ به دست میاره.در واقع size همون شعاع آونگ هست.

094. & 095. یه آونگ که متشکل از یه دایره و یه خط هست رو رسم میکنه.

097. این چهار خط وظیفه رسم قسمت Top screen رو دارن . ما که مختصات x&y آونگا رو داریم. اگه یه بار دیگه انتقال محور بردیم و سایز و ... رو دوباره تعریف کنیم یه تصویر دیگه تویه یه جای دیگه از صفحه به وجود میاد. به همین سادگی.

102. این چهار خط هم 3D screen روچاپ می کنن. دقیقا مثل قسمت قبل یعنی Top screen .(توضیح این قسمتا یکمی سخته. منظورم شیوه انتقال محور و رسم دوباره آونگا اینبار با یه سایز متفاوته. مطمئنم تعداد افراد کمی هستن که تا اینجای این متن رو می خونن. اگه خودم بودم نمی خوندم! الآن که دارم این توضیحات رو می نویسم گاه گاهی با خودم می گم مگه0.2 نمره می خواد واسم

چیکار کنه که باید اینقد براش بنویسم که تازه شاید گیرم بیاد که بازم چشم آب نمی خوره که بتونم بهتر از سرکار خانم کربلایی توضیح بدم و نمره کامل بگیرم!!!)

110. این قسمت یه ویژگی برتر برنامه ی من هست. در واقع قسمت هوشمند! این حلقهfor به همراه if بعدی قدرت این رو دارن که حالت های خاصی که ضمن اجرای برنامه پیش میاد رو تشخیص بدن و به کاربر اعلام کنن. یه سوال ، مگه حالت خاصی که ما توی برنامه می بینیم و کلی ذوق می کنیم چیه؟ وقتیه که اختلاف W چنتا آونگ از یه مقدار مشخصی کمتر بشه و این آونگا در چند دسته قرار بگیرن. خب این for تو در تو نگاه میکنه که چنتا آونگ هست که اختلاف w اونا کمتر از یه مقدار خاصی هست. If بعدی بررسی میکنه که اگر این تعداد بیشتر از یه مقدار خاصی بود تشخیص بده که حالت خاص اتفاق اقتاده. فقط یه نکته در باره تعدادی هست که تعداد آونگ های هم w باید بیشتر از اون باشن. اگه 100 آونگ داشته باشیم این تعداد متفاوت خواهد بود با زمانی که ما 10 آونگ داریم.پس باید این تعداد رو با یه فرمولی وابسته به تعداد آونگا پیدا کنیم.

120. 25 میلی ثانیه در اجرای برنامه تاخیر ایجاد میکنه. اگه روند برنامه رو خوب متوجه شده باشید یعنی خیلی خیلی خوب متوجه شده باشید براتون یه سوال آزار دهنده پیش میاد و اونم اینکه مگه ما زمان رو هر بار 40 میلی ثانیه جلو نبردیم و قرار نشد که به همین مقدار در اجرای برنامه تاخیر ایجاد کنیم ؟ پس چی شد؟ جواب اینجاس : اجرای خود دستورات برنامه یه تاخیری ایجاد می کنن. طبق محاسبات من و آزمایش و خطا تمام دستورات برنامه در موتور اصلی برنامه15 میلی ثانیه تاخیر ایجاد می کنه پس اگه ما به طور مصنوعی 25 میلی ثانیه دیگه تاخیر ایجاد کنیم میشه سر جمع 40 میلی ثانیه.

121. این دستور صفحه را پاک میکند و برنامه رو برای رسم آونگا در 40 میلی ثانیه جلوتر آماده میکنه.

دو سه خط آخر برنامه کاملا الکی و واضح هست! امیدوارم تونسته باشم حق مطلب رو ادا کرده باشم. البته می دونم که جاهایی بود که برنامه رو کاملا توضیح ندادم. اگه سوالی در باره برنامه بود خوشحال میشم ازم بپرسید.

خداحافظ

ابوالفتحی 90/9/4 ساعت 13:15