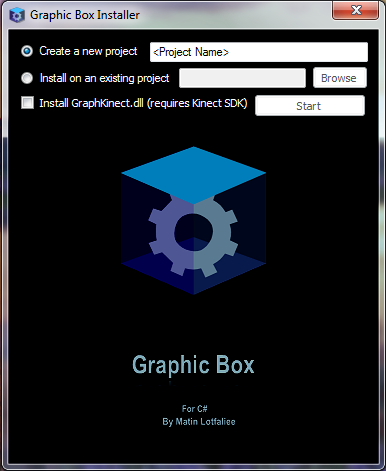
# نصب کتابخانه Graphic Box

در پوشه ای که این فایل آموزشی قرار دارد، یک فایل اجرایی به نام GraphInstaller   
وجود دارد که برای انجام تمام کارهای مورد نیاز برای نصب این کتابخانه و طراحی شده است.

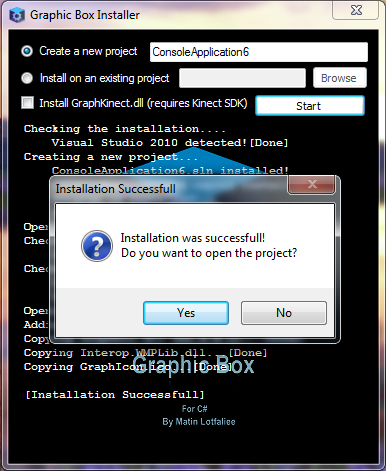
با باز کردن آن پنجره ای مشابه پنجره زیر مشاهده می شود.



همانطور که مشاهده می کنید نصب این کتابخانه به دو صورت امکان پذیر است. نوع اول با ایجاد  
یک پروژه کاملا تازه و نوع دوم نصب بر روی پروژه ای که از قبل ساخته بوده اید. پس از انتخاب نوع نصب و تمایل یا عدم تمایل به نصب کتابخانه GKinect، بر روی دکمه Start کلیک کنید تا نصب انجام شود.

لازم به ذکر است نام پروژه نباید شامل فاصله باشد. همچنین نباید با عدد شروع شود.

در صورت نصب موفقیت آمیز، پیغامی مشابه پیغام زیر نمایش داده میشود که می پرسد آیا مایلید وارد پروژه شوید یا خیر؟



پس از باز کردن Visual Studio از داخل Solution Explorer فایل Program.cs را باز کنید.

# برنامه نویسی با کتابخانه Graphic Box

## قانون شماره 1:

برای باز شدن پنجره Graphic Box **باید** در متد main دستور زیر اولین دستور از دستورات کتابخانه باشد:

Graph.initgraph(width, height);

* که در آن width طول و height عرض پنجره است.

## قانون شماره 2:

تمامی رسم هایی که انجام می دهید بر روی فضایی مجازی انجام می شود. برای آنکه بر روی پنجره دیده شوند، **باید** از یکی از دستورات زیر استفاده کنید:

1. در صورتی عکسی را نمایش می دهید و می خواهید منتظر زده شدن یک دکمه شوید از دستور زیر استفاده کنید:

Graph.getch();

1. در صورتی که انیمیشنی را می سازید و می خواهید مدتی (مثلا 1 ثانیه) توقف برای دیده شدن داشته باشید از دستور زیر استفاده کنید:

Graph.delay(1000);

## نکته شماره 1:

بعد از اتمام برنامه، پنجره Graphic Box باز می ماند. برای بسته شدن آن می توانید از دستور زیر استفاده کنید:

Graph.closegraph();

## نکته شماره 2:

برای نمایش برنامه به صورت تمام صفحه، می توانید از دستور زیر چه قبل و چه بعد از initgraph استفاده کنید:

Graph.fullscreen();

همچنین با فشردن دکمه های Alt+Enter میتوانید در وسط اجرا آن را تمام صفحه کنید.

البته اندازه پنجره باید برای نمایش تمام صفحه استاندارد باشد.

# نمونه برنامه نمایش دو عکس

1. static void Main(string[] args)
2. {
3. Graph.initgraph(400, 400);
4. Graph.setcolor(Color.YellowGreen);
5. Graph.line(0, 200, 400, 200);
6. Graph.floodfill(0, 220, Color.YellowGreen);
7. Graph.setcolor(Color.LightBlue);
8. Graph.floodfill(0, 180, Color.YellowGreen);
9. Graph.getch();
10. Graph.setcolor(Color.Blue);
11. Graph.fillcircle(200, 200, 50);
12. Graph.setcolor(Color.Red);
13. Graph.fillellipse(200, 200, 35, 50);
14. Graph.setcolor(Color.Yellow);
15. Graph.fillellipse(200, 200, 20, 50);
16. Graph.getch();
17. Graph.closegraph();
18. }

در خط 3: وارد محیط Graphic Box با پنجره ای به اندازه 400 در 400 می شود.

در خط 4: رنگ قلم را از پیش فرض (سفید) به سبز مایل به زرد تغییر می دهد.

در خط 5: خطی افقی در مختصات های داده شده روی صفحه مجازی 400 در 400 رسم می کند.

در خط 6: فضای پایین خط سبز را تا زمان رسیدن به این خط سبز، به رنگ قلم رنگ آمیزی می کند.

در خط 7: رنگ قلم را به آبی روشن تغییر می دهد.

در خط 8: فضای بالای خط سبز را تا زمان رسیدن به این خط سبز، به رنگ قلم رنگ آمیزی می کند.

در خط 9: نتیجه تمام این رسم ها را بر روی پنجره نمایش می دهد و منتظر یک دکمه از کاربر می ماند.

در خط 10: رنگ قلم را به آبی تغییر می دهد.

در خط 11: یک دایره تو پر به شعاع 50 در مرکز صفحه مجازی رسم می کند.

در خط 12: رنگ قلم را به قرمز تغییر می دهد.

در خط 13: یک بیضی تو پر به شعاع عمودی 50 و شعاع افقی 35 در مرکز صفحه مجازی رسم می کند.

در خط 14: رنگ قلم را به زرد تغییر می دهد.

در خط 15: یک بیضی تو پر به شعاع عمودی 50 و شعاع افقی 20 در مرکز صفحه مجازی رسم می کند.

در خط 16: نتیجه تمام این رسم ها را بر روی پنجره نمایش می دهد و منتظر یک دکمه از کاربر می ماند.

در خط 17: پنجره Graphic Box بسته می شود.

# نمونه برنامه یک انیمیشن

1. static void Main(string[] args)
2. {
3. Graph.initgraph(400, 400);
4. for (int i = 100; i < 300; i++)
5. {
6. Graph.cleardevice();
7. Graph.setcolor(Color.Blue);
8. Graph.fillcircle(i, 200, 50);
9. Graph.setcolor(Color.Red);
10. Graph.fillellipse(i, 200, 35, 50);
11. Graph.setcolor(Color.Yellow);
12. Graph.fillellipse(i, 200, 20, 50);
13. Graph.delay(100);
14. }
15. Graph.closegraph();
16. }

در خط 3: وارد محیط Graphic Box با پنجره ای به اندازه 400 در 400 می شود.

در خط 4: وارد حلقه ای که مقدار متغیر i را از 100 تا 300 تغییر می دهد.

در خط 6: کل صفحه مجازی به رنگ مشکی پاک می کند.

در خط 7: رنگ قلم را به آبی تغییر می دهد.

در خط 8: یک دایره تو پر به شعاع 50 در مختصات داده شده روی صفحه مجازی رسم می کند.

در خط 9: رنگ قلم را به قرمز تغییر می دهد.

در خط 10: یک بیضی تو پر به شعاع عمودی 50 و شعاع افقی 35 در مختصات داده شده رسم می کند.

در خط 11: رنگ قلم را به زرد تغییر می دهد.

در خط 12: یک بیضی تو پر به شعاع عمودی 50 و شعاع افقی 20 در مختصات داده شده رسم می کند.

در خط 13: نتیجه تمام این رسم ها را بر روی پنجره نمایش می دهد و به 100 میلی ثانیه صبر می کند.

در خط 15: پنجره Graphic Box بسته می شود.

## نکته:

در صورتی که بخواهید توپ با تمام سرعت ممکن حرکت کند می توانید به جای 100 میلی ثانیه، صفر قرار دهید تا هیچ مکثی صورت نگیرد.

# نمونه برنامه کار با کیبرد

1. static void Main(string[] args)
2. {
3. Graph.initgraph(400, 400);
4. int i = 200;
5. while (!Graph.keydown(Keys.Escape))
6. {
7. Graph.cleardevice();
8. Graph.fillcircle(i, 200, 50);
9. Graph.delay(0);
10. if(Graph.keydown(Keys.Left)) i--;
11. if(Graph.keydown(Keys.Right)) i++;
12. }
13. Graph.closegraph();
14. }

در خط 3: وارد محیط Graphic Box با پنجره ای به اندازه 400 در 400 می شود.

در خط 4: مقدار متغیر i را به عنوان مکان افقی دایره در مرکز قرار می دهد.

در خط 5: تا زمانی که دکمه Escape زده نشده باشد در حلقه می چرخد.

در خط 7: کل صفحه مجازی به رنگ مشکی پاک می کند.

در خط 8: یک دایره تو پر به شعاع 50 در مختصات داده شده روی صفحه مجازی رسم می کند.

در خط 9: نتیجه ی این رسم را بر روی پنجره نمایش می دهد و اصلا صبر نمی کند.

در خط 10: اگر دکمه چپ زده شده بود از مقدار i کم می کند.

در خط 11: اگر دکمه راست زده شده بود به مقدار i اضافه می کند.

در خط 13: پنجره Graphic Box بسته می شود.

# دستورات رسم دو بعدی

|  |  |
| --- | --- |
| شرح | دستور |
| پاک کردن کل صفحه | Graph.cleardevice();  کل صفحه را مشکی می کند. |
| رسم خط | Graph.line(x1,y1,x2,y2);  که دو پارامتر اول مربوط به نقطه شروع  و دو پارامتر دوم مربوط به نقطه پایان خط است. |
| رسم کمان | Graph.arc(x, y, start, end, radius);  که دو پارامتر اول مربوط به نقطه مرکز کمان،  دو پارامتر بعدی مربوط به زاویه شروع و پایان کمان بر حسب درجه و خلاف جهت دایره مثلثاتی  و پارامتر آخر مربوط به شعاع کمان نسبت به مرکز کمان است. |
| رسم دایره | Graph.circle(x, y, radius);  که دو پارامتر اول مربوط به نقطه مرکز دایره  و پارامتر آخر مربوط به شعاع دایره است. |
| رسم بیضی | Graph.ellipse)x, y, start, end, xr, yr);  که دو پارامتر اول مربوط به مرکز بیضی،  دو پارامتر دوم مربوط به زاویه شروع و پایان بیضی بر حسب درجه و خلاف جهت دایره مثلثاتی (مانند رسم کمان)، پارامتر چهارم مربوط به شعاع عمودی و پارامتر آخر مربوط به شعاع افقی است. |
| رسم بیضی توپر | Graph.fillellipse(x, y, xr, yr);  که دو پارامتر اول مربوط به مرکز بیضی، پارامتر سوم مربوط به شعاع عمودی و پارامتر آخر مربوط به شعاع افقی است. |
| رسم سکتور  )پیتزا) | Graph.sector)x, y, start, end, xr, yr);  پارامتر ها دقیقا مثل بیضی است. |
| رسم سکتور کروی | Graph.pieslice(x, y, start, end, radius);  پارامتر ها دقیقا مثل کمان است. |
| رسم مستطیل | Graph.rectangle(x1, y1, x2, y2);  پارامترها دقیقا مثل خط است به عنوان قطر مستطیل،  یعنی دو پارامتر اول مربوط به نقطه بالا سمت چپ مستطیل  و دو پارامتر دوم مربوط به نقطه پایین سمت راست مستطیل است. |
| رسم مستطیل توپر | Graph.bar(x1, y1, x2, y2);  پارامترها دقیقا مثل مستطیل است. |
| رنگ آمیزی محیط بسته | Graph.floodfill(x, y, Color.Blue);  که دو نقطه اول نقطه شروع رنگ زدن و پارامتر آخر مربوط به رنگ مرز بسته است که رنگ آمیزی از آن خارج نشود. |
| نقطه | Graph.putpixel(x, y, Color.Blue);  که دو پارامتر اول مختصات و پارامتر سوم رنگ نقطه است. |
| دریافت رنگ یک نقطه | Color a = Graph.getpixel(x, y);  که دو پارامتر مختصات دریافت  و یک رنگ به عنوان جواب خارج می کند. |
| تعیین رنگ قلم | Graph.setcolor(Color.Blue,255);  که پارامتر ورودی رنگ جدید و پارامتر دوم میزان شفافیت از 0 تا 255 است. |

|  |  |
| --- | --- |
| تعیین رنگ قلم با اعداد | Color a = Color.FromRGB(255, 255, 255);  Graph.setcolor(a);  که ابتدا باید یک رنگ بسازیم، متدی را اجرا میکنیم که 3 پارامتر از جنس byte دریافت و به عنوان خروجی، رنگ جدید را خارج می کند.  پارامتر اول شدت قرمز، پارامتر دوم شدت سبز و پارامتر سوم شدت آبی است. |
| رسم دایره توپر | Graph.fillcircle(x, y, r);  پارامترها دقیقا مثل دایره است. |
| مقایسه دو رنگ | if(Graph.getpixel(x, y) == Color.Blue)  چون خروجی متد getpixel یک رنگ است، پس میتوان آن را مقایسه کرد.  همچنین این کار را می توان با Color.FromRGB انجام داد. |
| نوشتن متن | Graph.outtextxy(x, y, ”hello”);  که دو پارامتر اول مربوط به گوشه بالا سمت چپ متن  و پارامتر سوم رشته می باشد. |
| نوشتن متن با زاویه | Graph.outtextxy(x, y, ”hello”, angle);  که پارامتر ها مانند نوشتن متن است  اما در انتها یک پارامتر به عنوان زاویه بر حسب درجه قرار دارد. |
| تغییر قلم متن | Graph.setfont(“Tahoma”, FontStyle.Bold, 20);  که پارامتر اول اسم قلم، پارامتر بعدی حالت قلم  و پارامتر آخر اندازه قلم بر حسب pixel است. |
| ذخیره عکس در حافظه موقت | Image p = Graph.getimage(x1, y1, x2, y2);  که پارامترهای ورودی مانند پارامتر های مستطیل است  و به یک عکس خارج می کند. |
| رسم عکس از  حافظه موقت | Graph.putimage(p, x, y);  که پارامتر اول یک عکس  و دو پارامتر بعدی نقطه بالا سمت چپ در عکس است. |
| رسم عکس از حافظه موقت با تعیین اندازه | Graph.putimage(p, x1, y1, x2, y2);  که اول یک عکس  و سه پارامتر بعدی مانند پارامتر های مستطیل است.  نکته: پارامتر دوم و سوم باید کوچکتر از پارامتر چهارم و پنجم باشد. |
| رسم عکس از هارد | Graph.image(“C:\\a.jpg”, x, y);  که پارامتر ها مانند رسم عکس از حافظه موقت است  اما به جای عکس، آدرس آن را دریافت می کند. |
| رسم عکس از هارد  با تعیین اندازه | Graph.image(“C:\\a.jpg, x1, y1, x2, y2);  که پارامتر ها مانند رسم عکس از حافظه موقت با تعیین اندازه است  اما به جای عکس، آدرس آن را دریافت می کند. |
| رسم عکس از هارد با زاویه | Graph.image(“C:\\a.jpg, x1, y1, x2, y2, angle);  که پارامتر ها مانند رسم عکس از هارد با تعیین اندازه است  اما در انتها یک پارامتر به عنوان زاویه بر حسب درجه قرار دارد. |
| ذخیره عکس از حافظه موقت به هارد | p.Save(“D:\\a.jpg”);  که p به معنی نام خود عکس در حافظه موقت  و پارامتر ورودی آدرس محل ذخیره سازی است. |
| ذخیره پیش زمینه | Graph.copybackground();  همانند دستور getimage است اما تمام صفحه و با سرعت بیشتر. |
| قرار دادن پیش زمینه | Graph.pastebackground();  همانند دستور image است اما تمام صفحه و با سرعت بیشتر. |

# دستورات کار با صفحه کلید و موس و صدا

|  |  |
| --- | --- |
| شرح دستور | معادل در Graphic Box |
| دریافت یک دکمه | char ch = Graph.getch();  با اجرای این دستور برنامه متوقف شده و منتظر فشرده شدن یک دکمه می ماند. سپس دکمه زده شده را به عنوان char خارج میکند. |
| دریافت یک رشته | string s = Graph.intextxy(x, y, false);  با اجرای این دستور برنامه متوقف شده و منتظر فشرده شدن حروف و دکمه اینتر می ماند. که در آن دو پارامتر اول مختصات محل نمایش ورودی و مکان نما، و پارامتر آخر یک بولین برای چپ پین یا راست چین است. |
| بررسی فشرده بودن هر دکمه ای | if(Graph.kbhit())  در صورت فشرده بودن هر دکمه ای وارد شرط می شود. |
| بررسی فشرده بودن دکمه ای خاص  (در این مثال کلید راست) | if(Graph.keydown(Keys.Right))  که در آن یک پارامتر از نوع Key دریافت و در صورت فشرده بودن آن وارد شرط می شود. |
| دریافت وضعیت موس | int x, y, click;  Graph.getmouse(out x, out y, out click);  که در آن دو پارامتر اول مکان نقطه ای که موس به آن اشاره دارد را تحویل می دهد و پارامتر سوم اگر 1 باشد یعنی کلیک چپ، اگر 2 باشد یعنی کلیک راست و اگر 0 باشد یعنی کلیک نشده است. |
| پنهان کردن موس | Graph.hidemouse();  این دستور صرفا برای آن است که خودتان شکل موس بسازید. |
| نمایش موس | Graph.showmouse(); |
| دریافت عدد تصادفی | Random r = new Random();  int a = r.Next(100);  که با وارد کردن عدد 100 به این دستور، یک عدد از میان 0 تا 99 به صورت تصادفی خارج می کند. |
| پخش نت موسیقی | Graph.sound("A4");  Graph.delay(1000);  Graph.nosound("A4");  که در دستور اول نت را به صورت رشته وارد می کنیم.  در دستور دوم به میزان لازم مکث میکنیم. در دستور سوم نواختن نت قطع می شود. (این دستور ممکن است در پیانو لازم نباشد.) |
| قطع تمام صداها | Graph.nosound();  این دستور هم برای موزیک و هم برای تمام نت ها کاربرد دارد. |
| تعیین آلت موسیقی | Graph.setinstrument(MIDIInstruments.Violin);  که به عنوان پارامتر ورودی نوع آلت موسیقی دریافت می شود.  بدون این دستور پیش فرض پیانو استفاده می شود. |
| پخش موزیک | Graph.playsound(“C:\\a.mp3”);  که به عنوان پارامتر ورودی آدرس فایل صوتی از هارد دریافت می شود. این دستور همه نوع فایلی را می تواند پخش کند اما با اجرای مجدد این دستور موزیک قبلی قطع میشود. |
| بررسی در حال پخش بودن موزیک | if(Graph.isplaying) |
| پخش صدا از نوع wave | Graph.playwave(“C:\\a.wav”);  این دستور همانند دستور playsound است با این تفاوت که هر چند بار قابل اجراست و اجرای جدید باعث قطع شدن اجرای قبلی نمی شود. اما فقط فایل های با پسوند wav را می تواند پخش کند. |

# دستورات رسم سه بعدی و آناگلیف و مقایسه آن ها با graph3d.h

* برای توضیحات بیشتر به PDF آموزشی استاد هاشمیان واقع در همین پوشه مراجعه کنید.

|  |  |  |
| --- | --- | --- |
| شرح دستور | معادل در graph3d.h | معادل در Graphic Box |
| رسم خط | line3d(x1,y1,z1,x2,y2,z2); | Graph3d.line(x1,y1,z1,x2,y2,z2); |
| رسم دایره | circle3d(x,y,z,r); | Graph3d.circle(x,y,z,r); |
| رسم بیضی | ellipse3d(x, y, z, st, end, r1, f1, t1, r2, f2, t2); | Graph3d.ellipse(x, y, z, st, end, r1, f1, t1, r2, f2, t2); |
| رسم بیضی پر | fillellipse3d(x, y, z, st, end, r1, f1, t1, r2, f2, t2); | Graph3d.fillellipse(x, y, z, st, end, r1, f1, t1, r2, f2, t2); |
| رسم مستطیل | rectangle3d(x, y, z, r1, f1, t1, r2, f2, t2); | Graph3d.rectangle(x, y, z, r1, f1, t1, r2, f2, t2);  یا  Graph3d.rectangle(x,y,z,r1,r2,Surface.XY); |
| رسم مستطیل پر | fillrectangle3d(x, y, z, r1, f1, t1, r2, f2, t2); | Graph3d.fillrectangle(x, y, z, r1, f1, t1, r2, f2, t2);  یا  Graph3d.fillrectangle(x, y, z, r1, r2, Surface.XY); |
| نقطه گذاری | putpixel3d(x, y, z, BLUE); | Graph3d.putpixel(x, y, z, Color.Blue); |
| نوشتن متن | outtextxy3d(x, y, z, “hello”); | Graph3d.outtextxy(x, y, z, “hello”); |
| تنظیم میزان پر کردن | setfillstyle3d(x); | Graph3d.setfillstyle(x); |
| دریافت میزان پر کردن | int a = getfillstyle3d(); | int a = Graph3d.getfillstyle(); |
| رسم آناگلیف | line3da(x1,y1,z1,x2,y2,z2);  circle3da(x,y,z,r); | Graph3d.set3DGlass(Glass.Anaglyph);  Graph3d.line(x1,y1,z1,x2,y2,z2);  Graph3d.circle(x,y,z,r); |
| رسم برای تلویزیون های 3 بعدی | ---- | Graph3d.set3DGlass(Glass.TV3D);  Graph3d.line(x1,y1,z1,x2,y2,z2);  Graph3d.circle(x,y,z,r); |
| فعال سازی موس 3 بعدی | initmouse3da(); | Graph3d.initmouse(); |
| دریافت مکان موس 3 بعدی | int x,y,z,click;  getmouse3da(&x,&y,&z,&click); | int x,y,z;  bool click=Graph3d.getmouse(out x,out y, out z);  برای جابه جایی در محور z باید دکمه راست موس نگاه داشته شود. |
| نمایش موس | showmouse3da(); | Graph3d.showmouse(); |
| پنهان کردن | hidemouse3da(); | Graph3d.hidemouse(); |
| تغییر نشانگر موس | int x,y,z,click;  getmouse3da(&x,&y,&z,&click);  hidemouse3da();  circle3da(x,y,z,10); | Graph3d.mouseCursor = new Bitmap("a.jpg"); |
| رسم متحرک | while(true)  {  animatecircle3da(x1,y1,z1,r1);  animatecircle3da(x2,y2,z2,r2);  delay(1000);  clearcircle3da(x2,y2,z2,r2);  clearcircle3da(x1,y1,z1,r1);  } | Graph.copybackground();  while(true)  {  Graph3d.circle(x1,y1,z1,r1);  Graph3d.circle(x2,y2,z2,r2);  Graph.delay(1000);  Graph.pastebackground();  } |
| رسم عکس | ---- | Graph3d.image(x, y, z, r1, f1, t1, r2, f2, t2, bitmap);  یا  Graph3d.image(x, y, z, r1, r2, Surface.XY, bitmap); |
| رسم عکس استریو | ---- | Graph3d.stereoImage(x, y, z, r1, f1, t1, r2, f2, t2, bitmap);  یا  Graph3d.stereoImage(x, y, z, r1, r2, Surface.XY, bitmap); |

# دستورات ارتباط با شبکه (آزمایشی)

* قبل از استفاده از این دستورات بر روی شبکه های کامپیوتری و IP مطالعه داشته باشید.

|  |  |
| --- | --- |
| معادل در GraphicBox | شرح دستور |
| Graph.TCPinitSender("127.0.0.1", 3055); | شروع برای ارسال |
| Graph.TCPSend(data); | ارسال یک آرایه از بایت |
| Graph.TCPinitListener(3055); | شروع برای دریافت (شنونده) |
| byte[] data = Graph.TCPReceive();  در صورتی که اطلاعاتی وجود نداشته باشد مقدار null قرار می دهد. | دریافت اطلاعات (در صورت وجود) |

# نکاتی جالب

* شما می توانید به صورت هم زمان، هم از Console و هم از Graphic Box در برنامه   
  استفاده کنید و این دو را با هم ترکیب کنید.
* علت رسم بر روی صفحه مجازی، کنترل پرش های پی در پی در Turbo C++ بود. اما نقطه ضعف آن این است که کنترل خط به خط برنامه دیگر ممکن نیست.

برای رفع آن، دو دستور وجود دارد:

Graph.trace();

که باعث می شود دستورات بعد از آن دیگر روی صفحه مجازی قرار نگیرند و بلافاصله دیده شوند.

Graph.notrace();

که باعث پایان این روند خواهد شد و به حالت رسم مجازی باز خواهد گشت.

* برای کار های انیمیشن دو متغیر وجود دارد که به عنوان گزارش دهنده از قدرت سیستم می توانید از آن ها استفاده کنید.

double a = Graph.MSPF;

که به معنی میلی ثانیه به ازای هر فریم است. می توانید ببینید در یک حلقه هر بار نمایش چه مدت طول کشیده است.

double a = Graph.FPS;

که به معنی تعداد فریم در هر ثانیه است. می توانید ببینید در یک حلقه هر ثانیه چند نمایش صورت گرفته است. در فیلم ها این عدد 25 فریم در ثانیه است.

* برای حذف لوگوی ابتدایی می‌توان دستور زیر را قبل از اجرای initgraph نوشت.

Graph.skiplogo();

# تاریخچه تغییرات اعمال شده

## در نسخه 4.5

1. پشتیبانی GraphInstaller از ویژوال استودیو 2013
2. بهینه سازی معماری کتابخانه بر روی دات نت 4
3. استفاده از پردازش های چند هسته ای برای رسم های سه بعدی

## در نسخه 4.4

1. پیاده سازی مجدد تابع image زاویه دار با سرعت بیشتر و بدون خطا
2. پیاده سازی مجدد تابع outtextxy زاویه دار با سرعت بیشتر و بدون خطا
3. اصلاح تابع nosound برای قطع نت های در حال پخش بدون ایجاد وقفه و بدون تغییر ساز

## در نسخه 4.3

1. ایجاد تابع جدید intextxy به عنوان معادل Console.ReadLine گرافیکی
2. اضافه شدن نوع جدیدی از تابع image همراه با ورودی زاویه (آزمایشی)
3. اضافه شدن نوع جدیدی از تابع outtextxy همراه با ورودی زاویه (آزمایشی)
4. اصلاح تابع delay برای افزایش دقت زمان توقف
5. تطبیق کتابخانه GKinect بر روی SDK v1.8
6. اصلاح قسمت هایی از دستورات PDF آموزشی که به اشتباه نوشته شده بود.

## در نسخه 4.2

1. حل مشکل تابع playsound در نیمی از کامپیوترها
2. بازنویسی getch و getkey برای افزایش دقت توابع
3. تصحیح تابع trace برای تسهیل بیشتر کار با آن
4. محدودیت جدید در GraphInstaller برای نامگذاری پروژه ها
5. اصلاح قسمت هایی از PDF آموزشی با توجه به سوالات دانشجویان استاد هاشمیان

## در نسخه 4.1

1. بازنویسی تابع setfont برای استفاده از قلم های جدیدی که کنار فایل exe قرار دارند. بدون نیاز به کپی کردن قلم در ویندوز و درگیر کردن سیستم عامل.
2. اصلاح توابع image و playsound برای پیغام عدم وجود فایل به جای رد کردن دستور.
3. اضافه شدن transparency به تابع setcolor به منظور تنظیم شفافیت.

## در نسخه 4

1. پیاده سازی مجدد fullscreen با تغییر رزولوشن کامپیوتر که در نتیجه نسبت به طراحی قبل سرعت برنامه های تمام صفحه بین 2 تا 4 برابر شد.
2. پیاده سازی دکمه Alt+Enter در زمان اجرا برای نمایش تمام صفحه یا پنجره.
3. پیاده سازی دو مشخصه MonitorWidthInMillimeter و MonitorHeight-InMillimeter برای دریافت اندازه مانیتور به میلیمتر با دقت بسیار بالا.

## در نسخه 3.9

1. ارتقاع مدیریت حافظه با نگه داری تصاویر تغییر اندازه داده شده و تصاویر وارد شده از هارد
2. ارتقاع و اصلاح کتابخانه GKinect منطبق بر SDK v1.7
3. تغییر متد getjoint و اضافه شدن پارامتر sort برای تشخیص جلوترین فرد به دوربین.
4. توابع جدید عکس برداری سه بعدی با کینکت و تبدیل عکس استدیو به آناگلیف (آزمایشی)
5. باز نویسی مجدد توابع شبکه TCP با استفاده از socket (آزمایشی)
6. اصلاح مشکل GraphInstaller در اولویت بندی ورژن های مختلف Visual Studio
7. افزایش سرعت تابع outtextxy سه بعدی با استفاده از مدیریت حافظه جدید
8. اصلاح PDF آموزشی در توضیحات مربوط به دستورات و حذف موضوع Turbo C++

## در نسخه 3.6

1. تغییر ساختار بنیادین سه بعدی با توابع set3DGlass و SetAnaglyphMethod برای کار با تلویزون های سه بعدی(عینک های پولاریزه) و انواع عینک های آناگلیف
2. ایجاد توابع image و stereoimage برای رسم عکس سه بعدی و عکس استریو
3. پیاده سازی توابع شبکهTCP به صورت آزمایشی
4. ساده سازی توابعی که نیاز به یاد گیری مختصات قطبی داشتند. نظیر rectangle
5. استفاده از MIDI برای پخش نوت های موسیقی
6. توضیع دومین نسخه از کتابخانه GKinect منطبق بر SDK v1.6
7. پیاده سازی توابع محاسبه مکان انگشت با Kinect به صورت آزمایشی
8. ایجاد تابع GetPersonID به جای تقسیم و باقیمانده گیری در Kinect
9. استفاده از overloading در بعضی از توابع مشابه نظیر image و imagescaled
10. اصلاح تابع delay برای رفع وابستگی آن به initgraph مناسب برای موسیقی
11. اصلاح GraphInstaller برای Visual Studio 2012

## در نسخه 2.3.0.1

1. قرار گرفتن لوگو و آیکون منتخب برای Graphic Box
2. پیاده سازی Color و FontStyle مستقل از System.Drawing
3. حذف متد colorequal و پیاده سازی مقایسه کننده دو رنگ توسط عملگر == و !=
4. توضیع اولین نسخه از کتابخانه GKinect برای استفاده از Kinect
5. اصلاح GraphInstaller

## در نسخه 2.2.0.0

1. تغییر نام پروژه از Graph Emulator به Graphic Box
2. نوشته ی آغازین به عنوان مکان قرار گیری لوگو
3. اضافه شدن دستور skiplogo برای پرش از روی لوگو
4. ایجاد صف فایل ها برای کاهش I/O در دسترسی به عکس ها و فایل های صوتی
5. اصلاح موس سه بعدی آناگلیف

## در نسخه 2.1.0.0

1. استفاده از پردازش موازی برای سرعت بیشتر.
2. اصلاح مشکلی بزرگ در تابع floodfill.
3. اصلاح تابع getch و فعال شدن پنجره در صورت اجرای آن.
4. اصلاح تمامی توابع مخصوص موس و افزایش سرعت چرخ موس.
5. قرار گرفتن چند نمونه برنامه در کنار فایل.
6. بهینه سازی کد های کتابخانه توسط کامپایلر (optimization)

## در نسخه 2.0.4.4

1. طراحی توابع موس آناگلیف
2. طراحی توابع copybackground و pastebackground به جای animate ها
3. طراحی نصب کننده GraphDLL
4. اصلاح بسته شدن ناگهانی پنجره به دلیل دست رسی مستقیم به بیتمپ
5. اصلاح ساختار جدید Graph3da با استفاده از صف (برای آینده و کینکت)

## در نسخه 2.0.2.1

1. اصلاح متد های outtextxy3d و outtextxy3da
2. تابع جدید isplaying برای بررسی در حال پخش بودن موزیک
3. تغییر ساختار Graph3da و تنظیم دستورات با استفاده از صف (برای آینده و کینکت)

## در نسخه 1.6

1. دسترسی بایت به بایت به بیتمپ که باعث افزایش سرعتgetpixel و putpixel و متد های وابسته مثل floodfill و تمام دستورات graph3da شد.
2. اصلاح مجدد متد floodfill و استفاده از الگوریتم بر اساس صف
3. اصلاح مشکل متد getmouse
4. حذف بررسی وجود یک راس در همه متد ها

## در نسخه 1.5

1. قرار گرفتن refresh در متد getch
2. تغییر نام و اصلاح بسیاری از توابع به علت تطبیق با Turbo C++
3. اصلاح مشکل زاویه در متد های arc و ellipse و pieslice
4. اصلاح مشکل متد floodfill و استفاده از الگوریتم پوبا

## در نسخه 1.4

1. اضافه شدن توابع سه بعدی و آناگلیف در Graph3d و Graph3da
2. اضافه شدن گزارشگر های MSPF و FPS
3. اضافه شدن تابع colorequal برای مقایسه برابری دو رنگ
4. بهینه سازی متد delay با کنترل مدت زمان آخرین عمل رسم
5. کنترل وجود نقطه ها در داخل محیط
6. تغییر مکان پارامتر های متد outtextxy برای تطبیق با Turbo C++
7. اصلاح مشکل initgraph در اندازه های بزرگتر از اندازه محیط ویندوز
8. اصلاح مشکل جلو بودن تمام وقت و غیر قابل کنترل پنجره

## در نسخه 1.3

1. افزایش سرعت متد setcolor
2. تغییر پارامتر های ورودی متد putpixel و اضافه کردن پارامتر رنگ به آن

## در نسخه 1.2

1. حذف صف دستورات ارسالی و استفاده از روش اجرای مستقیم.
2. نمایش فضای مجازی فقط در صورت اجرا شدن توابع refresh و delay
3. اصلاح تابع trace برای اجرای خط به خط برنامه.

## در نسخه 1.1

1. DLL مربوط به پخش موزیک به Interop.WMPLib تغییر یافت.
2. اضافه شدن تابع playwave برای پخش فایل های wav
3. اضافه شدن توابع trace و notrace برای کنترل خط به خط برنامه
4. توانایی تغییر اندازه پنجره با تغییر میزان بزرگنمایی
5. حذف محدودیت 100 تایی صف دستورات ارسالی

با تشکر فراوان از:

استاد سید ابراهیم هاشمیان  
علیرضا رحیمیان

احسان احسانی

محسن ضیاآبادی

* در صورت مشاهده هر گونه مشکل در **Graphic Box** به این ایمیل اطلاع دهید تا سریعا اصلاح شود: [matinlotfali@gmail.com](mailto:matinlotfali@gmail.com)
* در صورت تمایل به دنبال کردن اخبار این کتابخانه در این گروه عضو شوید:

<https://www.facebook.com/groups/graphicbox2/>

موفق باشید

ایده پرداز و طراح

متین لطفعلی ئی