# آموزش استفاده از GKinect v1.3

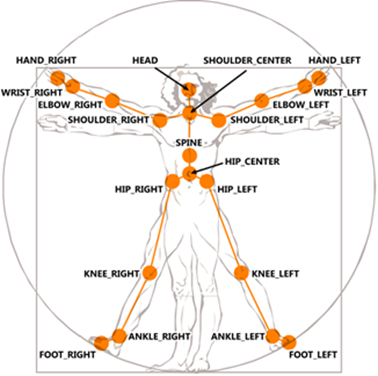
برای شروع کار تمامی برنامه هایی که قرار است از کینکت استفاده کنند ابتدا باید دستور زیر را وارد کنید.

GKinect.initkinect(ColorFormat.Rgb640x480Fps30, DepthFormat.R640x480Fps30);

این دستور باعث می شود که دستگاه کینکت روشن شود و همچنین باید مدل دوربین رنگی را با انتخاب   
RGB و YUV و اندازه تصویر هر دوربین را مشخص کرد.

در این مثال مقدار پیش فرض و استاندارد قرار داده شده است.

# دریافت مکان یک مفصل (دست، سر، پا و ...)

برای دریافت مکان x و y و z یک مفصل باید دستور زیر وارد شود.

GKinect.getjoint(int Person,

MapToCamera.Color,

BodyParts.Head,

out float x,

out float y,

out short z, true);

* Person به معنی شماره فرد مورد نظر است. زیرا تا 6 نفر قابل شناسایی هستند و اولین نفر 0 است.
* MapToCamera به معنی مقیاس اعداد است که بر اساس کدامین دوربین اندازه گیری شوند.

|  |  |
| --- | --- |
| MapToCamera.Color | یعنی مفصل دقیقا سر همان مکانی باشد که در تصویر رنگی قابل مشاهده است.  محور x از 0 تا 640 و محور y از 0 تا 480 (مقدار مشخص شده در initkinect) |
| MapToCamera.Depth | یعنی مفصل دقیقا سر همان مکانی باشد که در تصویر عمقی قابل مشاهده است.  محور x از 0 تا 640 و محور y از 0 تا 480 (مقدار مشخص شده در initkinect) |
| MapToCamera.Metric | یعنی مقیاس اعداد بر اساس میلیمتر نسبت به دوربین محاسبه شوند.  محور x از 2200- تا 2200+ و محور y از از 1600- تا 1600+ و محور z از 0 تا 4000 |

* BodyParts به معنی مفصل درخواستی (از بین مفصل های داخل عکس) می باشد.
* x و y و z متغیر های خروجی هستند که پس از اجرای دستور، نتایج در آن ها قرار می گیرند.  
  در صورت خاموش بودن یا آماده نبودن نتایج عدد 1- در هر سه قرار خواهد گرفت.

# مثال برای دریافت و نمایش سر

1. Graph.initgraph(640, 480);
2. GKinect.initkinect(ColorFormat.Rgb640x480Fps30, DepthFormat.R640x480Fps30);
3. while (true)
4. {
5. float x0;
6. float y0;
7. short z0;
8. GKinect.getjoint(0, MapToCamera.Color, BodyParts.Head, out x0, out y0, out z0);
9. int x1 = Convert.ToInt32(x0);
10. int y1 = Convert.ToInt32(y0);
11. Graph.cleardevice();
12. Graph.fillcircle(x1,y1, 10);
13. Graph.delay(0);
14. }

* در خط 1 پنجره ی GraphicBox با اندازه ی 640 در 480 باز می شود.
* در خط 2 کینکت با تنظیمات 640 در 480 روشن می شود.
* در خط 3 وارد یک حلقه ی بی نهایت شده تا بتوان مداوم مکان سر را حتی با تکان خوردنش دید.
* در خط 8 دستور دریافت مفصل اجرا شده و مفصل سر را بر اساس دوربین رنگی درخواست کرده و در جواب نتایج در x0 ، y0 و z0 قرار داده می شوند.
* در خط 9 و 10 مقادیر تبدیل به int می شوند تا بتوان آن ها را با رسم دایره نشان داد (تبدیل نوع)
* در خط 11 صفحه پاک شده تا رسم های قبلی دیده نشوند.
* در خط 12 یک دایره تو پر رسم می شود.
* در خط 13 نتیجه ی رسم روی مانیتور نشان داده خواهد شد. و سپس به بالای حلقه باز خواهد گشت.

# دریافت و نمایش تصویر دوربین رنگی

دریافت تصویر دوربین رنگی به صورت پیکسل به پیکسل امکان پذیر است. به نحوی که مکان هر پیکسل دلخواه را به دستور GetColorPixel داده و در نتیجه یک رنگ دریافت می شود.

1. Graph.initgraph(640, 480);
2. GKinect.initkinect(ColorFormat.Rgb640x480Fps30, DepthFormat.R640x480Fps30);
3. while (true)
4. {
5. for (int i = 0; i < 640; i++)
6. {
7. for (int j = 0; j < 480; j++)
8. {
9. Color c = GKinect.GetColorPixel(i, j);
10. Graph.putpixel(i, j, c);
11. }
12. }
13. Graph.delay(0);
14. }

* در خط 1 پنجره ی GraphicBox با اندازه ی 640 در 480 باز می شود.
* در خط 2 کینکت با تنظیمات 640 در 480 روشن می شود.
* در خط 3 وارد یک حلقه ی بی نهایت شده تا بتوان مداوم تصاویر صادر شده از کینکت را دید.
* در خط 5 و 7 برای پیمایش تمام رنگ ها وارد دو حلقه می شود.
* در خط 9 از مختصات خواسته شده رنگ را (مشکی در صورت آماده نبودن) دریافت می کند.
* در خط 10 رنگ را بر روی مکان مشخص شده قرار می دهد.
* در خط 13 نتیجه ی این رنگ آمیزی ها را روی مانیتور نشان می دهد.



# دریافت و نمایش تصویر دوربین عمقی (فاصله اجسام از کینکت)

دریافت تصویر دوربین عمقی به صورت پیکسل به پیکسل امکان پذیر است. به نحوی که مکان هر پیکسل دلخواه را به دستور GetDepthPixel داده و در نتیجه یک عدد short دریافت می شود.

1. Graph.initgraph(640, 480);
2. GKinect.initkinect(ColorFormat.Rgb640x480Fps30, DepthFormat.R640x480Fps30);
3. while (true)
4. {
5. for (int i = 0; i < 640; i++)
6. {
7. for (int j = 0; j < 480; j++)
8. {
9. short a = GKinect.GetDepthPixel(i, j, MapToCamera.Depth);
10. byte b = 255 - Convert.ToByte(a / 200);
11. Color c = Color.FromRGB(b, b, b);
12. Graph.putpixel(i, j, c);
13. }
14. }
15. Graph.delay(0);
16. }

* در خط 1 پنجره ی GraphicBox با اندازه ی 640 در 480 باز می شود.
* در خط 2 کینکت با تنظیمات 640 در 480 روشن می شود.
* در خط 3 وارد یک حلقه ی بی نهایت شده تا بتوان مداوم تصاویر صادر شده از کینکت را دید.
* در خط 5 و 7 برای پیمایش تمام رنگ ها وارد دو حلقه می شود.
* در خط 9 در مختصات خواسته شده فاصله آن پیکسل را (1- در صورت آماده نبودن) دریافت می کند. این فاصله می تواند منطبق بر تصویر دوربین رنگی و یا مقیاص میلیمتری تنظیم شود.
* در خط 10 آن عدد را تغییر می دهد و برای مرحله بعد (تبدیل به یک رنگ) تبدیل به بایت می کند.
* در خط 11 آن بایت را به صورت قرمز و آبی و سبز وارد کرده و تبدیل به رنگ می کند.
* در خط 12 رنگ را بر روی مکان مشخص شده قرار می دهد.
* در خط 15 نتیجه ی این رنگ آمیزی ها را روی مانیتور نشان می دهد.

در این مثال تبدیل به رنگ به گونه ای انجام شده که اجسام نزدیک تر روشن تر و اجسام دور تر تیره تر باشند. شما می توانید این نوع تبدیل را تغییر دهید.

# نکات حرفه ای

* با مقدار دهی به متغیر زیر می توانید زاویه دوربین را به صورت عمودی تغییر دهید.

GKinect.CameraVerticalAngle = 0;

این مقدار می تواند بین 27- درجه تا 27+ درجه باشد.

این نکته توجه شود که با اجرای دستور initkinect زاویه به صفر باز خواهد گشت. پس در صورتی که این مقدار دهی بعد از initkinect باشد دوربین یک بار صفر شده و سپس به زاویه تغییر درجه خواهد داد. اما اگر این مقدار دهی قبل از آن انجام شود دوربین مستقیما به آن درجه خواهد رفت.

* می توان تعلق هر پیکسل عمقی را با دستور زیر تعیین کرد:

int a = GetPersonID(int x, int y, MapToCamera camera);

نتیجه یک عدد بین 0 تا 6 است که 1- به معنی هیچ کس می باشد.

* برای نرم تر شدن حرکت مفصل ها دستور زیر وجود دارد که باید قبل از initkinect مقدار دهی شود.

GKinect.EnableSmoothSetting(int Correction, int Prediction, int Smoothing,   
 int JitterRadius, int MaxDeviationRadius);

که میزان هر کدام از این int ها بین صفر تا صد هستند.

* برای پردازش سطح پایین که مناسب هوش مصنوعی است آرایه های دریافتی از کینکت را قابل دسترسی قرار داده ایم که شامل ColorMap ، DepthMap،DepthMappedToColor و DepthMappedToMetric می شوند.