

بنام خدا



استفاده از Auto Path Finding در Unity3D

ترجمه : حمزه شعبانی

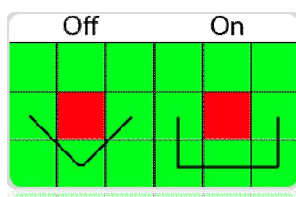
[www.Persian-Designers.com](http://www.Persian-Designers.com)

[www.Unity3D.mihanblog.com](http://www.Unity3D.mihanblog.com)

## Static Settings (تنظیمات ثابت) :

**Show Grid** (نمایش خطوط شبکه ای) : با فعالسازی این کلید خطوط شبکه ای در سکانس نشان داده میشوند.

**Dont Cut Corners** (غیر فعال کردن حرکت به صورت ضربدری) : در صورتی که این قابلیت فعال باشد مسیر عبور به شکل مربع خواهد بود ولی در غیر این صورت حرکت به شکل لوزی نیز امکانپذیر است، یعنی داخل هر بلوک شبکه یک مسیر ضربدری هم برای عبور بوجود می آید (مانند شکل زیر)



## Grid (خطوط شبکه ای) :

**Grid Width** : مقدار عرض خطوط شبکه ای را مشخص میکند.

**Grid Depth** : مقدار طول خطوط شبکه ای را مشخص میکند.

**Node Size** : اندازه نقاط اتصال شبکه را مشخص میکند.

**Height** : مقدار ارتفاع خطوط شبکه ای را مشخص میکند.

**Offset** : مکان قرار گیری شبکه ها در محیط بازی را مشخص میکند.

## Y Position (محور عمودی) :

**Flat** (سطح صاف) : در خطوط شبکه ای مسطح تمامی نقاط اتصال شبکه (Nodes) روی یک سطح صاف (مثلا یک Plan) تشکیل میشوند.

**Terrain** (زمین) : خطوط شبکه را براساس تغییرات ارتفاع فایل Terrain (زمین) انجام میدهد.

**Raycast** (نقاط برخورد) : خطوط شبکه را بر اساس Raycast های مشخص شده تنظیم میکند.

## Walkable Check (چک کردن حالت حرکت آبجکت) :

**Overlap** (اصطکاک) : با استفاده از فیزیک محاسبه میکند که آیا دایره محیط آبجکت در حال راه رفتن بر روی نقاط اتصال شبکه است یا خیر.

**Touch** (لمس کردن) : با استفاده از فیزیک دایره محیط آبجکت را چک میکند.

**Capsule** (کپسول) : (پیشنهاد شده) تمام آبجکت های درون کپسول چک میشوند، محیط کیسول را میتوانید با تنظیمات ارتفاع و قطر آن تغییر دهید.

**Raycast** (برخورد) : برخورد از پایین یا بالای آبجکت را محاسبه میکند.

**Neighbours** (نقاط مجاور هم) :

تعداد نقاط برخورد شبکه در هر بلوک را مشخص میکند.

**Eight** (هشت) : با انتخاب این گزینه در هر بلوک تعداد ۸ نقطه برخورد بوجود می آید.

**Four** (چهار) : با انتخاب این گزینه در هر بلوک تعداد ۴ نقطه برخورد بوجود می آید.

**RunTime Settings** (تنظیمات زمان اجرا) :

**Simplify Paths** (ساده سازی مسیر) : اگر این اسکریپت فعال باشد ساده کردن مسیر فقط شامل نقاطی میشود که در مسیر تعریف شده باشند.

**Max Frame Time** (بیشترین زمان در هر فریم) : میزان حداکثر زمانی که اسکریپت از هر فریم را مصرف می کند، البته ممکن است زمان بیشتری نیز مصرف شود اما برای مسیر های کوتاه مشخص کردن این مقدار مهم است.

**Formula** (فرمول) :

**H** : با انتخاب این گزینه تنها مقدار **H** محاسبه میشود. **H** مقدار فاصله تا پایان است، محاسبه این فرمول سریع است اما این فرمول توانایی تشخیص دقیق کنج آبجکت های دیگر را ندارد.

**G** : با انتخاب این گزینه تنها مقدار **G** محاسبه میشود. **G** مقدار فاصله شروع تا نقطه است، این فرمول از فرمول **H** دقیقتر است اما سرعت محاسبه این فرمول کمتر از **H** است. این الگوریتم مانند الگوریتم **Dijkstra's** است.

**H+G** : با انتخاب این گزینه هر دو فرمول **H** و **G** محاسبه میشود. این فرمول بطور عادی مانند الگوریتم **A\*algorithm** است، این فرمول برای محاسبه از هر دو الگوریتم **G** و **H** استفاده میکند.

**Node Links** (پیوند نقاط) :

**Node Links** (پیوند نقاط) : از ابزار **Node Links** برای مشخص کردن مختصات نقطه آغاز و مکان پایان حرکت آبجکت استفاده میشود یعنی شما مختصات مکان شروع حرکت و مکان پایان حرکت را مشخص میکنید و آبجکت در این مسیر حرکت میکند.

### چگونگی استفاده از **Auto Path Finding** برای آبجکت خود :

- شما قبل از هر چیز باید اطمینان داشته باشید که یک فایل **AstarPath.cs** در سکانس خود دارید.
- پس از این کار فایل اسکریپت **Seeker** را به آبجکتی که قرار است تحت **PathFinding** عمل کند نسبت دهید.
- در کد اسکریپت اصلی خود در زمانی که میخواهید حرکت با عمل **PathFinding** انجام شود (برای مثال در بازی های سبک **RTS** وقتی که بازیاز روی یک نقطه از زمین کلیک کرد و قرار است آبجکت به سمت آن نقطه برود) کد زیر را وارد کنید:

```
GetComponent ("Seeker").StartPath (fromHere, toHere);
```

- این تابع چیزی را برنمیگرداند و نمیداند چطور باید حرکت کند، این تابع را به کدهایتان اضافه کنید :

```
SetPoints (points : Vector3[]) {  
//Start follow the path  
}
```

زمانی که شما نیاز به **PathFinding** دارید اسکریپت **Seeker** این تابع را صدا میزند، دقت کنید که این اسکریپت را به آبجکت هایی مانند دشمن نسبت دهید نه آبجکت های ثابت بازی، همچنین یک مثال از این موضوع را میتوانید در فایل اسکریپت **AI** در پروژه های **PathFinding** مشاهده کنید.

**:API**

متغیر ها :

**totalNodeAmount** : تعداد کل نقاط خطوط شبکه ای در سانس.

**staticNodes** : یک آرایه از تمام نقاط خطوط شبکه، گرفتن یک نقطه از تابع **GetNode(x,y,z)**.

**grids** : یک آرایه از تمام خطوط شبکه ها.

**links** : یک آرایه از تمام پیوندهای نقاط خطوط شبکه در سکانس.

**levelCost** : برای محاسبه فرمول  $H$  در خطوط شبکه ها استفاده میشود، این متغیر مقدار مکان محور  $Y$  در هنگام کار با فرمول  $H$  را تنظیم می کند.

**activePath** : مشخص میکند که کدام  $Path$  ها باید محاسبه شوند.

**costs** : یک آرایه برای ارزش دادن به حرکت در جهات مختلف.

**area** : مقدار محیط قابل حرکت در سکانس.

**areaColors** : یک آرایه که شامل رنگ نواحی مختلف خطوط شبکه ای است.

توابع :

**ToLocal** : تغییر شکل دادن مختصات مکان آبجکت از جهانی به محلی، یک مختصات نقاط محلی از ۳ عدد صحیح تشکیل شده، شما توسط متغیر **GetNode** میتوانید به آن دست پیدا کنید. اگر یک نقطه از خطوط شبکه در خارج از خطوط شبکه باشد این تابع مقدار  $(-1,-1,-1)$  را برمیگرداند.

**GetNode** : مختصات محلی نقاط خطوط شبکه را بر میگرداند  $(x,y,z)$ ،  $Y$  مکان نقاط خطوط شبکه را برمیگرداند.

**SetNodes** : مکانی را برای هر خط شبکه فراهم میکند که وجود داشته باشد یا خیر. در صورتی که تنظیمات **fullPhysicsCheck** فعال باشد مسیر حرکت اولیه جسم یکبار محاسبه میشود. در غیر این صورت جسم یکبار میانبری را با توجه به نقاط برخورد برای حرکت محاسبه میکند.

**SetNodesWorld** : مانند بالا عمل میکند با این تفاوت که این محاسبه با توجه به مکان جهانی و مکان های خطوط شبکه در جهان بازی و بدون توجه به نقاط برخورد اجسام انجام میشود.

**SetNodesLocal** : مانند بالا عمل میکند اما بدون استفاده از **fullPhysicsCheck**.

**RecalculateArea** : محاسبه دوباره مکان محیط قابل حرکت.

**FullPhysicsCheck** : تعیین میکند که یک نقطه در حال راه رفتن است یا خیر، این تابع همچنان مقدار ارتفاع را نیز محاسبه می نماید.

**RecalcNeighbours** : نقاط همسایه نقطه خطوط شبکه جاری را محاسبه میکند.

**FloodFill** : شروع پر کردن نقاط خطوط شبکه، زمانی که حرکت در مسیر محاسبه نمیشود اسکرپت از **FloodFill** استفاده میکند.

**Scan :** این تابع خطوط شبکه ای را دوباره محاسبه میکند. این تابع در آغاز کار یک بار اجرا میشود. اگر تمام محاسبات غیر فعال شود فایل اسکریپت مکانهایی تعیین شده را محاسبه میکند. پیشنهاد نمیشود که این تابع بشکل ریل تایم اجرا شود زیرا حدود نصف ثانیه برای محاسبه آن زمان لازم است.



ترجمه : حمزه شعبانی

[www.Persian-Designers.com](http://www.Persian-Designers.com)

[www.Unity3D.mihanblog.com](http://www.Unity3D.mihanblog.com)