جمعیتها در هوش مصنوعی

مقدمه

تا حالا به بازیهای استراتژی با دقت توجه کرده اید؟ در قسمتهایی از این نوع بازیها لازم است تا یک کار گروهی انجام گیرد. به عنوان مثال گروهی از سربازان به طرف محل خاصی حرکت می کنند. برای آنکه حرکت این گروه از مکانی به مکان دیگر طبیعی به نظر برسد باید اصول جمعیتها (در مورد حیوانات گله ها) رعایت شوند. این مسئله در هوش مصنوعی (AI) به مسئله جمعیتها (Flock) معروف است.

اصول جمعیتها بدین شرح هستند:

۱- افراد جمعیت باید همیشه سعی کنند به سمت مرکز جمعیت حرکت کنند.
۲- افراد جمعیت باید فاصله حداقل را بین خودشان و افراد دیگر جمعیت رعایت کنند.
۳- افراد جمعیت باید سرعت متناسبی با سرعت جمعیت داشته باشند.

این سه عامل باعث می شوند که افراد یک جمعیت بصورت پراکنده نباشند. در یک بازی و یا یک محیط شبیه سازی شده فردی که فاصله اش از جمعیت زیاد می شود میزان آسیب پذیری بیشتری خواهد داشت. به عنوان مثال اگر بخواهیم محیط زیر آب را شبیه سازی کنیم، ماهیهایی که از گروه خودشان عقب می افتند بیشتر در معرض حمله کوسه قرار خواهند گرفت و برای واقع گرایی هر چه بیشتر باید این مسئله را در محیط شبیه سازی شده نیز رعایت کنیم. در اینجاست که به راه حلی برای مسئله جمعیتها در هوش مصنوعی نیاز پیدا می شود.

برای حل این مسئله باید سه قانون ذکر شده برای جمعیتها را پیاده سازی کنیم. با یک مثال در مورد ماهیها راه حل این مسئله را بررسی می کنیم. ماهی تترا یک ماهی گروهی است و از جمله ماهیهای مورد علاقه من است بنابراین فرض می کنیم که جمعیت مورد نظر ما از تعدادی ماهی تترا تشکیل شده! ;)

این مثال با استفاده از زبان C# برنامه نویسی شده و در آن روش ساده ای برای حل مسئله Flocking در نظر گرفته شده است. البته باید توجه داشته باشید که برای پیاده سازی Flocking در بازیهای واقعی بیشتر از الگوریتمهای ژنتیک استفاده می کنند.

این مقاله به صورتی نوشته شده است که ضمن بررسی مسئله جمعیت در هوش مصنوعی با قسمتهای مختلف یک برنامه در C# و نکاتی در مورد این زبان برنامه نویسی آشنا می شویم.

ابتدا یک پروژه جدید در VS.NET ایجاد کرده و در قسمت Project Type زبان C# ، در قسمت Templates گزینه Windows Application را انتخاب می کنیم. در قسمت Name نام پروژه و در قسمت Location محل ذخیره فایلهای پروژه را تعیین می کنیم.

Form اصلی

پس از ایجاد یک پروژه جدید، یک Form بطور پیش فرض در اختیار شما قرار داده می شود که در مثال ما، نام آن را frmMain گذاشته ایم. با استفاده از منوی Project\Add Class یک کلاس به نام TetraFish به پروژه اضافه می کنیم (جزییات این کلاس را در ادامه مقاله خواهید دید). به قسمت Code مربوط به Form رفته و در ابتدای تعریف کلاس Form ، یک آرایه از نوع TetraFishبه نام mudtFish تعریف می کنیم. تعداد اعضای این آرایه با استفاده از ثابت CountOfFish تعیین شده است. در این قسمت متغیرهای عمومی دیگری نیز تعریف شده است که برای بررسی آنها می توانید به کد همراه مقاله مراجعه کنید. یکی از این متغیرها mblnRunning است که به عنوان متغیر کنترل حلقه بازی بکار رفته است. ثابتهایی نیز به عنوان ثابتهای عمومی در این قسمت تعریف شده اند. دو متد GetDistance و TetraFishAI را به کلاس Form اضافه کرده ایم که شرح آنها بدین صورت است:

- GetDistance : پارامترهای آن دو مقدار int به عنوان اندیس دو ماهی هستند که در این متد فاصله بین آنها محاسبه می شود. فاصله بین دو ماهی بر اساس رابطه √((X2-X1)²+(Y2-Y1)²) محاسبه شده است. پارامتر برگشتی این متد فاصله محاسبه شده بین دو ماهی است.
- TetraFishAI : کار شبیه سازی جمعیت در این متد انجام می شود. در این متد با استفاده از یک حلقه for متدهای NearestNeighbour و FindCenter برای هر کدام از ماهیها (هر کدام از اعضای آرایه mudtFish) فراخوانی می شوند. دقت کنید که به عنوان پارامتر دوم متد FindCenter ، مقدار this را ارسال کرده ایم که ارجاعی است به کلاسی که در آن است (در اینجا کلاس Form اصلی برنامه).

در Form اصلی برنامه علاوه بر این دو متد سه گرداننده رویداد (Event Handler) برای رویدادهای (FrmMain_Load) Load و (frmMain_Closing) Closing و (frmMain_KeyDown) KeyDown اضافه کرده ایم که شرح آنها بدین صورت است:

- frmMain_Load : در این گرداننده رویداد ابتدا با استفاده از متد CreateGraphics شی ای از نوع Graphics برای ترسیم روی آن می سازیم. با توجه به اینکه متد فراخوانی شده متعلق به this است بنابراین ترسیمات روی فرم اصلی برنامه صورت می گیرد. پس از آن شی ای از نوع Random برای ایجاد مقادیر تصادفی تعریف کرده ایم. در ادامه اشیای TetraFish ساخته شده و بطور تصادفی مکان آنها تعیین می شود. مقدار متغیر کنترل حلقه بازی (mblnRunning) به True تنظیم شده و سپس وارد حلقه بازی می شویم. در حلقه بازی با استفاده از خاصیت TickCount از شی Environment و مقداری که به عنوان تعداد فریم در هر ثانیه تعیین شده است (ثابت MS_PER_FRAME) بررسی می کنیم تا این تعداد فریم رعایت شود. سپس متد TetraFishAI را فراخوانی می کنیم تا کار شبیه سازی جمعیت ماهیها را انجام دهد و پس از آن متد ShowTetraFish را برای هر کدام را ماهیها فراخوانی می کنیم تا جمعیت ماهیها با مشخصات جدید نمایش داده شود. در انتهای حلقه بازی از متد Application.DoEvents استفاده می کنیم. فراخوانی این متد باعث می شود که به پیغامهای دیگری که ویندوز برای این برنامه ارسال می کند، پاسخ داده شود. در صورتی که این خط از کد را حذف کنیم دیگر به رویدادهایی مانند KeyDown و Closing و ... پاسخ داده نمی شود.
- frmMain_Closing : این گرداننده رویداد برای پایان دادن به حلقه بازی است (رویداد Closing در زمان بستن Form اتفاق می افتد). در اینجا تنها کاری که می کنیم این است که مقدار متغیر کنترل حلقه بازی را به false تنظیم کنیم تا در اجرای بعدی، شرط حلقه false شده و از حلقه خارج شود و در نتیجه برنامه به اتمام برسد. در صورت انجام ندادن این کار با بستن پنجره برنامه، فقط پنجره بسته می شود و برنامه به اجرای خود ادامه خواهد داد، در نتیجه باعث هدر رفتن منابع سیستم خواهد شد.
- frmMain_KeyDown : در این گرداننده رویداد کلید فشار داده شده با استفاده از پارامتر KeyEventArgs بدست می آید (رویداد KeyDown در زمان فشار دادن کلیدها اتفاق می افتد) و آن را با کلیدهای مورد نظر از مجموعه System.Windows.Forms.Keys مقایسه می کنیم و عملیات لازم برای هر کلید را انجام می دهیم.

توجه داشته باشید که باید این گرداننده های رویداد را به برنامه معرفی کنیم. برای این کار در قسمت " Windows Form Designer generated code " و در انتهای متد InitializeComponent کدهای زیر را اضافه می کنیم:

```
this.KeyDown += new System.Windows.Forms.KeyEventHandler(this.frmMain_KeyDown);
this.Closing += new System.ComponentModel.CancelEventHandler(this.frmMain_Closing);
this.Load += new System.EventHandler(this.frmMain_Load);
```

کلاس TetraFish

کلاسی با نام TetraFish می سازیم و برای آن خواص زیر را در نظر می گیریم:

- X : موقیت TetraFish روی محور X
- Y: موقعیت TetraFish روی محور Y
- Xspeed : مولفه سرعت در راستای محور X
- Yspeed: مولفه سرعت در راستای محور Y

این کلاس دارای ۳ متد به شرح زیر است:

- ShowTetraFish : کار این متد رسم ماهی تترا روی شی Graphic است. تنها پارامتر ورودی آن شی ای از نوع System.Drawing.Graphics است.
- NearestNeighbour : این متد ابتدا مقدار فاصله و مولفه های سرعت نزدیکترین ماهی تترا به تترای مورد نظر ما را پیدا می کند. سپس سرعت تترای مورد نظر ما را با ماهی تترای مجاور متناسب می سازد. تنها پارامتر این متد یک مقدار int است که اندیس ماهی تترایی است که می خواهیم مشخصات آن را تنظیم کنیم.
- FindCenter : این متد ابتدا با استفاده از میانگین مختصات ماهی ها، مرکز جمعیت را بدست می آورد. برای واقع گرایی بیشتر مقداری به عنوان Noise به آن اضافه می شود تا ماهی ها در حال جنبش به نظر برسند. سپس ماهی را با سرعت مشخص شده به سمت مرکز جمعیت حرکت می دهیم و در انتها فاصله تعیین شده

به عنوان فاصله بین ماهیها را بررسی می کنیم. در انتها موقعیت ماهی را بررسی می کنیم و در صورتی که از حدود Form خارج شده باشد موقعیت آن را دوباره تنظیم می کنیم.

نتیجه

در اینجا کار کدنویسی برنامه Flocking به پایان رسیده است. کد این برنامه به دلیل راحتی خواندن مقاله، در این مقاله نوشته نشده است. این کد به همراه مقاله است و پیشنهاد می کنم که با خواندن هر قسمت از مقاله کد معادل آن را هم مطالعه کنید. برنامه flocking در VS.NET 2003 نوشته شده است و در صورتی که شما از VS.NET 2002 استفاده می کنید باید یکی از راههای زیر را برای استفاده از این کد انتخاب کنید:

۱- ارتقای VS.NET به نسخه 2003
۲- مطالعه مقاله "تبدیل یک پروژه VS.NET 2003 به VS.NET 2002" در وبلاگ http://vblog.Persianblog.com

حالا برنامه را اجرا کرده و از اجرای آن لذت ببرید.

اگر سوال و یا نظر و انتقادی در زمینه این برنامه و یا مسئله Flock داشتید می توانید در آدرس زیر مطرح کنید:

http://www.persian-designers.com/forum/viewforum.php?f=9

نویسنده: محمد صافدل
نسخه: 1.0
اگر سوال و یا نظر و انتقادی در زمینه این برنامه و یا مسئله Flock داشتید می توانید در آدرس زیر مطرح کنید:
http://www.persian-designers.com/forum/viewforum.php?f=9

منابع:
در نوشتن این مقاله و برنامه نویسی آن از سایتهای زیر به عنوان منبع استفاده شده است:
http://www.generation5.org
http://www.gamedev.net
http://www.rookscape.com

و همینطور کتاب زیر:
An Introduction to Genetic Algorithms- By:Melanie Mitchell (Publisher: Prentice Hall of India)