< به نام خدا >

نام و نام خانوادگی: محمدرضا عزیزپور **نام استاد: استاد کارگر**

موضوع: اپلیکشن اندروید و IOS **پروژه: ساخت رباتی که بازی Angry Birds را اجرا کند**

کلاس: طراحی نرم افزار ، پنجشنبه ساعت 17:30 **افراد پروژه: یک نفر**

مقدمه:

امروز قرار است با هم رباتی طراحی کنیم که میتواند مراحل بازی رایانه ای "انگری بردز" را یک به یک به بهترین شکل ممکن پشت سر بگذارد.

برای ساخت این ربات ، از Python ورژن 3.11 در محیط VS IDE استفاده شده است؛ منابع این پروژه از YouTube و Python Community بهره گرفته شده.

نکات اولیه:

قبل از هر چیز باید بدانیم که پروژه مورد نظر ما یعنی "انگری بردز" چگونه کار میکند؛ در این بازی ، پرنده ها همیشه در حال حرکت اند و خوک ها ساکنند، البته از روی گیمپلی بازی میتوان متوجه شد که خوک ها پس از برخورد با پرنده ها به حرکت در میآیند، دلیل اینکه باید طریقه ی بازی را بدانیم این است که بتوانیم قیزیک بازی را درست به پایتون بفهمانیم.

برای طی کردن مراحل بازی نیاز است که بازیکن ، با حرکت دادن یک طناب و پرتاب کردن پرنده ها ، تمامی خوک ها را با بیشترین امتیاز ممکن بزند.

شروع:

در ابتدا برای خود با استفاده از # ، چهار اولویت را برای ربات مشخص میکنیم، ابتدا detect pigs ، سپس get current bird و aim ، در انتها هم shot؛ دلیل اصلی این کار این است که هم خودمان بتوانیم کلاس بندی بهتری در کد داشته باشیم و هم دیگران بهتر متوجه کد های ما شوند.

سپس هر چهار اولویت ساخته شده را زیر while True قرار میدهیم ؛ این کار بسیار مهم است چون ما میخواهیم ربات ما به هیچ عنوان از این چهار اولویت خارج نشود، دقت کنید که اهمیتی ندارد که برای ربات یک پرنده باقی مانده یا لول چند است یا مثلا در کدام مپ بازی است؛ در هر حالتی باید در چارچوب این اولویت ها باشد.

یک نکته ی مهم تر این است که ما در این ربات فقط میخواهیم پرنده ی قرمز و یک خوک سایز متوسط ساده استفاده کنیم؛ همانطور که میدانید در بازی های انگری بردز، ما پرنده ها و خوک های متفاوت با ویژگی های متفاوت داریم که در صورت ساخت همه ی آنها ، سورس کد بسیار طولانی و سنگین میشود؛ پس ما ربات را ساده تر میسازیم تا قابل فهم تر باشد؛ دقت کنید که در صورت درخواست ساخت مابقی ، به همین روش امکان پذیر است.

قبل از ساخت کلاس ها ، من پیشنهاد میکنم که بازی انگری بردزی که ما میخواهیم از روی آن کد بزنیم را عکس برداری کنیم، مثلا در بخش resources ، تصاویری از شخصیت های بازی و مپ های بازی را مشاهده میکنید ، دلیل این تصویر برداری ها را بعدا خواهیم فهمید. (بخشی از این تصاویر در گوگل قابل یافتن است ولی خب برای Background ها چیزی پیدا نکردم ، پس از python community یک سری تصاویر شبیه رو یافتم.)

برای شروع while ابتدا میخواهیم از detect pigs شروع کنیم که اولین اولویت بود، برای خوک ها model_path ای تعریف میکنیم که با استفاده از آن ربات بتواند خوک را تشخیص دهد.

برای این کار برای زیرپوشه ی **models** ، دو بخش اسم و **yolo** تعریف میکنیم، در "یولو" میخواهیم مکان های قیزیکی که خوک ها در آن قرار دارند را تعریف کنیم؛ شاید بپرسید چگونه ممکن است؟؟!

همان طور که میدانیم در بازی انگری بردز ما به عنوان بازیکن همیشه در سمت چپ مپ قرار داریم و خوک ها در سمت راست هستند، کافیست با استفاده از قراردهی تصاویر **background** در نمودار مربعی ، حدود عددی قرار گیری خوک ها را بدست آوریم. (دقت فراوان داشته باشید که این حدود اعداد قرار گیری است نه عدد دقیق چون ممکن است که مپ تغییر کند و جای خوک ها عوض شود.)

در قسمت بعد ، تصویر خوک را به ربات میفهمانیم (اینجا یکی از جاهاییست که به تصاویر نیاز داریم)؛ نیاز جدی به انجام این کار نیست ولی همانطور که گفتیم میخواهیم ربات به بهترین شکل ممکن ، این کار را انجام دهد پس داشتن تصویر کمک بزرگی برای فهم بهتر ربات است.

در این قسمت از **pyautogui** استفاده کنیم که با استفاده از آن میتوانیم از صفحه ی خود یک اسکرین شات گرفته و مستقیم وارد کد کنیم، ما انگری بردز را باز کرده و از صفحه ی مرحله ی اول آن عکس میگیرم.(تا زمانی که **background** مرحله ها تغییر نکند، بازی را ادامه میدهد.)

حال اگر در **main** ، اجرا را بزنید ، یک فریم از بازی را اجرا میکند که شروع خوبی است.

اما شاید بعضی ها بپرسید که الان کل صفحه ی دسکتاپ ما عکس برداری شده، حال چه کنیم؟ الان باید تصویر انگری بردز را **crop** کرده و برای آن موقعیت های عددی خود را تعریف کنیم، یعنی در قسمت **y** ، از **0** تا **600** به شکل بالا به پایین و در **x** ، از چپ به راست **0** تا **1200** و جایگزین میکنیم. (این اعداد از یک ویدیو در **youtube** گرفتم که بر حسب **aspect ratio** ی **3:4** ساخته شده.)

در قسمت بعدی، میخواهیم که پایتون آبجکت ما یعنی همان خوک را پیدا کند؛برای اینکه ربات راحت تر خوک ها را بیابد ، پیشنهاد این است که از **for** استفاده کنیم. (در لیست بگذارید تا یافتن بعدا آن راحت شود!)

برای اینکه متوجه شویم که ربات توانسته خوک ها را درست پیدا کند یا خیر، با فانکشن پرینت ، کار را ساده تر میکنیم؛ برای اطمینان یک بار دیگر پس از نوشتن **for** ، کد را اجرا کنید تا رفتار کد را ببنید، اگر ربات توانست خوک ها را نمایش دهد و بنویسد **pig** ، نصف کار تمام شده.

حال میرویم سراغ بخش بعدی، یعنی پرنده ها؛ این بخش ساده تر است چرا که پرنده ها همیشه یک جای ثابت ایستاده اند، پس برای آن مکان دقیق تعریف میکنیم ، مکانی که هر بار هم بازی را اجرا کند باز هم همان جاست.

در اینجا عین کاری که با خوک ها کردیم، برای پرنده ها هم **pyautogui** تعریف میکنیم با این تفاوت که مکان حرکتی برای ما اهمیت دارد، چون قرار است که پرنده پرتاب شود، برای مسیر اول که همیشه ثابت است به عنوان مکان **0** و مکانی که قرار است پرت شود را **1** در نظر میگیریم؛ دقت کنید که ما نمیدانیم ربات قرار است چگونه پرنده را پرت کند پس نباید مکان ثابتی به آن داده شود وگرنه هوش مصنوعی فقط کار از پیش تعریف شده را اجرا میکند و خودش هیچ تغییری در آن ایجاد نمیکند.

اینجا، فقط **mousedown** را تعریف میکنیم، دلیل این کار اینست که اگر انگری بردز را بازی کرده باشید، ما در **95** درصد اوقات موقع گرفتن پرنده، به سمت پایین حرکت میکنیم تا آن را پرت کنیم، اگر به سمت بالا را هم نعریف کنید باز هم متوجه میشود که به خوک نخورده و ادامه میدهد اما پرنده های زیادی را موقع آموزش حروم میکند و لول را میبازد...البته این کار درستی است که همه ی جهات تعریف شود چون در لول های بالاتر ممکن است خوک ها به سمت پایین باشند اما ما برای سادگی بیشتر فقط طرف اصلی را تعریف کردیم.

حال اگر کد را مجدد اجرا کنید میبینید که **mouse cursor** شما روی پرنده قفل میشود.

این بخش مهم ترین بخش پروژه است، زیرا باید به ربات آموزش دهیم تا هدف گیری کند؛ برای راحت تر شدن کار، از تعریف فاصله برای پرنده ها استفاده میکنیم.

در این قسمت هم به وسیله ی **for** سعی میکنیم تا ربات خوک ها را هدف بگیرد ، دقت داشته باشیم که در انتهای شبیه سازی مورد نظر، باید محل خوک را رندوم انتخاب کنیم؛ دلیل این کار اینست که خوک ها ممکن است بعد از ضربه جای خود را عوض کنند یا حتی تعداد خوک ها بیش از یک عدد باشد؛ اگر رندوم انتخاب نکنیم ، ربات با زدن اولین خوک متوقف میشود و دیگر ادامه نمیدهد.

همانطور که میبینید لیست **pig** که قبلا ساخته بودیم را در این قسمت فراخوانی میکنیم. برای پایان کار هم بهتر است به وسیله ی **if** کد را ببندیم؛ این کار اجبار خاصی ندارد اما در صورت زدن خوک ها و بردن مرحله و داشتن تعدادی پرنده ی پرتاب نشده، ربات با اینکه از مرحله عبور کرده ولی باز هم هدف میگیرد که کمی غیر عقلانی به نظر مخاطب میرسد، پس بهتر است که حتما در انتها از **break** استفاده کنیم.

در صورت اجرا کردن کد متوجه میشویم که تعداد فریم ها افزایش پیدا میکند، پایتون به طور اتوماتیک بین **35** تا **50** فریم را انتخاب میکند که به فریم استاندارد فیلم معروف است، البته شاید بگویید که **50** فریم خیلی کم است اما حتی بهترین فیلم های سینمایی دنیا هم در همین رنج ساخته میشوند، حالا اینکه چرا پایتون فکر میکند که انگری بردز ما ، یک فیلم است رو والا خودم هم نمیدونم 😉

همراه با افزایش فریم، میبینمیم که ربات ، کرسر موس را به سمت پرنده ها برده ولی این بار هدف گیری میکند، البته هنوز بلد نیست که چگونه پرتاب کند؛ برای بهتر متوجه شدن این قضیه ، باز هم میتوانید از فانکشن ساده ی print کمک بگیرید.

و سرانجام میرسیم به بخش آخر، در اینجا باید پرنده ها را پرتاب کند، پس با دو تعریف ریاضی شروع میکنیم که یکی عمودی و یکی افقی است.

دقت فراوان داشته باشیم که حتما dx- و dy- را وارد کنیم و نه برعکس؛ دلیل این کار این است که ما در هنگام گرفتن و پرتاب کردن پرنده ها، آنها را با انگشت به سمت عقب میکشیم، پس باید حتما مختصات را برعکس وارد کنیم؛ در اینجا برای آن یک زمان هم مشخص کنیم، مثلا 2 ثانیه...این کار فقط برای سرعت بخشیدن به ربات است و دلیل خاصی ندارد.

کار کاملا تمام است، اگر تمام مراحل را پی در پی انجام دهیم و تصویر اولیه را درست وارد کرده باشیم، میبینید که مرحله را به سادگی تمام میکند، البته ممکن است در ابندا یکم گیج بزند ولی درست خواهد شد، نگران نباشید!

همانطور که مرحله عبور میکند متوجه یک مشکل ریز میشویم و آن هم اینست که پس از عبور از مرحله، ربات کاملا خشک میشود و باید دستی مرحله بعدی را برای آن بیاورید و ران کنید که کمی تو ذوغ میزند.

کاری که میکنیم این است که از python community سورس کد عبور از مرحله ها را میگیریم و با def برای آن تعریف میکنیم، با این کار مرحله ها را خودش رد میکند، میتوانید برای آن ، زمان هم مشخص کنید مثلا 4 یا 5 ثانیه تا ربات حالت خشک پیدا نکند.

< پایان >