

# مستندسازی بازی درس گرافیک - فوتبال دستی

## فضای بازی

### فضای کلی

فضای کلی بازی به گونه ای است که نقطه مرکزی بازی بر روی مرکز تخته بازی قرار دارد.

تمامی توابعی که با توجه به کیبرد یا حرکت توپ باید محاسبه شوند در تابع رندrer قرار می گیرند تا در هر ۶۰ فریم بر ثانیه محاسبه شوند.

### تخته بازی

تخته بازی شامل سطحی صاف با تکسچر زمین بازی است، و چند دیواره که با مکعب ساخته شده اند. دیواره ها شامل دو دیواره بزرگ که در کنار محل های اوت قرار دارند و در کنار دروازه های بازی هر کدام ۳ مکعب قرار داده شده اند که فقط قسمت دروازه خالی مانده و قسمت دروازه به ۳ قسمت مساوی تقسیم شده که محل دروازه خالی و دوتای دیگر پر هستند و قسمت بالایی دروازه یک دیواره (مکعب) سراسری از محل کررها است. این دیواره ها بدلیل اینکه نسبت به نقطه مرکزی قرینه هستند فقط نقطه ایکس شان با هم قرینه هستند.

### دسته های بازی

دسته های بازی هر کدام در یک آبجکت ۳ بعدی قرار دارند تا بتوان آن ها را در یک گروه قرار داد و برای حرکت چرخشی راحت تر نمود. پس دسته بازی شامل چند شکل است. یک آدمک که یک مکعب و یک کره است. میله که یک استوانه است. دسته بازی هم شامل دو استوانه است، یکی بزرگ و یکی کوچک با شعاع بیشتر. برای حریف هم به همین صورت است.

یک مشکل، در این بازی چون مشکلی در برخورد توپ با حریف ها به وجود می آید، یک آدمک هم به صورت مخفی ساخته شده که در گروه قرار داده نشده است و با حرکت دسته ها، آن هم حرکت می کند.

## قانون بازی

### امتیاز دهی

امتیاز دهی در این بازی بدین صورت است که هر وقت توپ از محل دروازه به بیرون تر رفت یک امتیاز به حریف افزوده می شود. یعنی زمانی که نقطه مرکز توپ که همان X توپ است اگر در محل نقطه وسط باشد صفر است. ولی اگر از نصف

زمین بازی به بیشتر شد. یک امتیاز محسوب می شود و همین طور اگر به صورت منفی بیشتر شود باعث امتیاز به حریف می شود. سمت بازیکن نقطه  $X$  منفی و سمت حریف مثبت است.

اگر تعداد گل های یک کاربر به تعداد پیش فرض ۵ تا برسد، که این تابع هر بار پس از گل زدن چک می شود که اگر فردی به این میزان برسد، بازی متوقف می شود.

## حرکت بازیکن ها

### حرکت بازیکن

حرکت بازیکن با استفاده از کیبرد انجام می گیرد. دکمه ها چپ و بالا دسته را یک طرف و دکمه های راست و پایین هم به طرف دیگر می برد. برای این کار از `eventlistener` های جاوااسکریپت استفاده می کنیم. زمانی که دکمه در حالت پایین قرار می گیرد، باید همین طور حرکت انجام گیرد. پس زمانی که دکمه کیبرد بالا آمد، باید `eventlistener` را حذف نماییم. تا تابع حرکت غیر فعال شود.

با فشار دادن دکمه های کیبرد،  $Y$  دسته و آدمک تغییر می کنند و اگر بیشتر از حالت معمول نباشد (زمانی که آدمک در زمین باشد) این کار انجام می گیرد.

### حرکت حریف

حرکت حریف با استفاده هوش مصنوعی است. خیلی ساده.

زمانی که  $Y$  توپ تغییر می کند،  $Y$  آدمک و دسته هم تغییر می کند اما با توجه به سرعت بازی که از قبل تعیین شده. هر بار با استفاده از سرعت بازی دسته حریف هم تغییر می کند و نقطه  $Y$  توپ را تعقیب می کند. اگر بتواند به توپ برسد و توپ به آن برخورد کند، برگشت داده می شود.

## حرکت توپ

به دلیل اینکه هر وقت توپ از نصف زمین بازی بیشتر می شود و امتیاز محسوب می شود. یک حلت دیگر می ماند که اگر دیواره ها وجود داشته باشند، هم باید امتیاز محسوب شود. اما اینگونه نیست. زمانی که نقطه  $Y$  توپ بین نقطه ابتدایی دروازه و نقطه انتهایی دروازه باشد. امتیاز محسوب می شود. وقتی که توپ به دیواره ها برخورد باید توپ برگردد. برنامه نویسی این حالت بسیار ساده است. برای حالتی که توپ به قسمت های اوت برخورد می کند یعنی زمانی است که قدرمطلق  $Y$  توپ اگر بیشتر از نصف پهنای زمین شده است. که در این موقع  $Y$  توپ منفی می شود. برای حالت برخورد توپ به قسمت های کرنز نزدیک به دروازه هم باید اینگونه در نظر بگیریم: اگر  $X$  توپ بیش از درازای زمین باشد و  $Y$  آن بین

مقادیر ابتدایی تا انتهای دروازه نباشد، در این صورت به این دیواره برخورد کرده که باید این مورد را هر بار در رندر کردن محاسبه کرد و اگر مقدار صحیح برگردانده شد، باید  $X$  آن در منفی ضرب شود.

برای مواقعی که توپ به دسته بازی برخورد می کند هم باید  $X$  توپ را تغییر دهیم. خوب این حالت همان طور که گفتیم، اگر بخواهیم آبجکت ۳ بعدی را حرکت دهیم و مقدار  $X$  یا  $Y$  آن را بخواهیم، همان مقدار قبلی را برای ما برمی گرداند که این طور صحیح نیست. بنابراین باید آدمک را جدا بسازیم و در گروه قرار ندهیم. خوب، وقتی که دسته بازی حرکت می کند و آدمک هم حرکت می کند. نقطه  $Y$  آدمک هم جابه جا می شود. در این حالت باید نقطه ابتدایی دسته و نقطه انتهایی دسته را به  $Y$  توپ مقایسه کنیم که اگر بین این دو حالت بود و همچنین مقدار  $X$  توپ اگر بین ابتدای آدمک تا وسط آدمک بود (چون ممکن است مقدار  $X$  توپ دقیقاً با مقدار  $X$  آدمک در هنگام حرکت یکی نباشد، پس مقداری فرضی از ابتدای آدمک تا میان آدمک در نظر می گیریم) پس توپ به آدمک برخورد کرده. و در این حالت  $X$  توپ در منفی ضرب و  $Y$  توپ هم کمی منحرف می شود.

## ویژگی های دیگر بازی

### انیمیشن دوربین

در این بازی ۳ حالت برای دوربین قرار داده شده که یکی برای حالت استراتژیک از بالا و یکی به صورت اینتراکتور و یکی هم برای نشان دادن حالت کاربر از نزدیک. اگر ما بتوانیم تغییرات محل و چرخش دوربین را با توجه به رندر تغییر دهیم، در این صورت دوربین متحرک می شود. برای این کار موقعیت قبلی دوربین را به محل دیگر قدم قدم تغییر می دهیم و تابع آن را در تابع رندر قرار می دهیم تا کم کم دوربین تغییر مکان و چرخش دهد تا به حالت متحرک قرار گیرد.

دکمه C در این بازی این کار را انجام می دهد. ولی اگر با همان تابعی که دستورات کیبرد را نوشته بودیم، این کار یک مشکل ایجاد میکند، و مشکل اینجاست که اگر دکمه را تا چند مدت بگیریم، همین طور تابع پشت سرهم تکرار می شود. برای جلوگیری از این مشکل باید تابع دیگری برای کیبرد بنویسیم. زمانی که کلید بالا آمد این تابع را اجرا نماید. پس `eventlistener` را به حالت `keyup` قرار می دهیم. و زمانی که دکمه C فشار داده شد، دوربین را به حالت بعدی می برد.

### قابلیت ایست

قابلیت ایست یکی از ویژگی هایی است که برای برخی از بازی های فکری و ... قرار می دهند، این بازی هم مستثنا از این امر نیست. برای اینکار، مثل حالت تغییر دوربین ها در `eventlistener` زمانی که کلید `Enter` فشار داده شد، سرعت توپ صفر و حرکت های دسته کاربر و حریف با استفاده از یک دستور `if` متوقف می شود.

### پخش صدا

زمانی که فردی گل زد، یک دستور پخش آهنگ با جاوااسکریپت و یک تگ `html5` اجرا می‌شود. که باعث پخش شدن آهنگ در مرورگر می‌شود.

### چرخش دسته توسط بازیکن

از دیگر ویژگی‌های این بازی می‌توان به چرخش دسته بازی در حالتی که کاربر با فشار دادن دکمه `space` انجام می‌دهد، نام برد. زمانی که این دکمه فشار داده می‌شود. تا هر زمانی چرخش با سرعت خاصی ادامه می‌یابد، ولی باید این چرخش به حالت اولیه درآید. حالت صفر درجه برای نقطه معمول این آدمک تعیین شده است. زمانی که قدر مطلق باقیمانده چرخش دسته نسبت به حالت اولیه و زاویه  $180^\circ$  درجه (چرخش آدمک بین  $0^\circ$  تا  $180^\circ$  درجه) باشد، و دکمه رها شود، چرخش آدمک به حالت منفی ادامه پیدا می‌کند تا زمانی که به نزدیک صفر درجه برسد. در این حالت یک متغیر بولین برای در حالت چرخش بودن آن `true` می‌شود تا دیگر این چرخش متوقف شود. حالت چرخش آن به حالت اولیه قرار داده می‌شود. چون ممکن است دسته بیش از چند دور بزند، دیگر نیازی نیست چند بار هم دسته بچرخد، باید یک بار تا نزدیک قدر مطلق زاویه چرخش نسبت به باقیمانده چرخش به حالت اولیه محاسبه شود تا عددی بین  $0^\circ$  تا  $180^\circ$  درجه یا عددی بین  $180^\circ$  تا  $360^\circ$  درجه بدست آید تا به نسبت این دو چرخش به راست یا چپ ادامه پیدا کند. یعنی باید بهترین حالت نسبت به صفر درجه تعیین شود،