

وزارت علوم، تحقیقات و فناوری



بسم الله الرحمن الرحيم

پروژه درس گرافیک کامپیوتری

اساتید : آقایان مینوفام و سامعی

تهیه کنندگان:

احسان علی حسینی ۹۵۱۱۱۳۰۶۳

حسین سیفی ۹۵۱۱۱۳۰۵۱

محمد صادق نیک اندیش ۹۵۱۱۱۳۰۸۷

تیرماه ۱۳۹۹

شرح پروژه

نام این بازی شوتبال (Shootball) است. در این بازی کاربر یک توپ در اختیار دارد و باید با هدف گیری مناسب به گونه ای شوت کند که توپ درون دروازه برود.

قوانین بازی

در این بازی توپ روی محور x ها حرکت میکند و کاربر باید با نگه داشتن Enter قدرت ضربه زدن به توپ را تعیین کند و با هدف گیری مناسب و یا جهت دهی به حرکت با کلید های چپ و راست آن را به درون دروازه هدایت کند و امتیاز کسب کند .

با هر گل شدن یک امتیاز به کاربر می دهد و آن را به کاربر نشان می دهد. اما اگر توپ به تیرک ها برخورد یا دروازه بان آن را بگیرد یا از کناره ها خارج شود ، برای کاربر امتیاز محسوب نمیشود.

توابع اصلی

در این بازی ۳ تابع اصلی داریم که بازی با آنها اجرا میشود که در ادامه به اختصار توضیح داده می شود :

init()

initCannon()

Animate()

تابع Init()

در تابع init تمام متغیرهای پایه threejs یعنی scene و camera و renderer تعریف شده است. همچنین تمام اجزا و عناصر صحنه در این تابع تعریف و نوشته میشوند از جمله :

کف صحنه، دروازه ، توپ ، دروازه بان ، نورهایی که برای دنیای خود درنظر میگیریم ، سایه اجسام ، موقعیت اجسام ، مه و... .

```

71 |
72 | init()
73 |
74 | initCannon()
75 |
76 | animate()
77 |
78 | async function init() {
79 |     scene = new THREE.Scene()
80 |

```

تابع InitCannon()

برای نوشتن این تابع نیاز به کتابخانه ی cannon.js داریم. در این تابع اول متغیر world را مقداردهی کرده ایم سپس برای جهان خود یک جاذبه به سمت y های منفی نوشته ایم تا دنیای بازی ما مانند دنیای واقعی دارای نیروی جاذبه زمین باشد.

سپس در این تابع به تعریف فیزیک اجسام پرداخته ایم و برای هر جسمی که در init تعریف کرده ایم و میدانیم که به فیزیک نیاز دارد ، در اینجا فیزیک مورد نیازش را نوشته ایم. برای مثال تعریف فیزیک توسط لوپ ها برای توپ و دروازه و... انجام می شود.

```

315 |
316 | function initCannon() {
317 |     //? World Setup
318 |     world = new CANNON.World()
319 |     world.gravity.set(0, -26, 0)
320 |     world.broadphase = new CANNON.NaiveBroadphase()
321 |     world.solver.iterations = 20
322 |     world.defaultContactMaterial.contactEquationStiffness = 1e9
323 |     world.defaultContactMaterial.contactEquationRelaxation = 4
324 |

```

پس از تعریف فیزیک اجسام، برای مطابقت دادن هر جسم تعریف شده در init به فیزیک مربوط به آن ،تابع updatePhysics توسط تابع animate صدا زده می شود.

تابع goalkeepermove()

این تابع حرکت دروازه بان و گرفتن توپ را کنترل می کند که با کم و زیاد کردن پارامتر آن می توان سرعت حرکت آن را کنترل کرد.

تابع `onDocumentKeyDown`

این تابع برای تعیین قدرت توپ مورد استفاده است به این به صورت که با نگه داشتن Enter قدرت پرتاپ افزایش می یابد. همچنین با کلید های جهت چپ و راست می توان حرکت توپ را کنترل کرد و آن را هدایت کرد.

تابع `onDocumentKeyUp`

این تابع برای هنگامی استفاده می شود که بازیکن دست خود را از روی Enter برمی دارد و توپ شوت می شود.

تابع `checked`

این تابع برای امتیاز دهی به بازیکن استفاده می شود به این صورت که در پایان شوت ، وضعیت توپ بررسی می شود اگر داخل دروازه بود ، گل محسوب می شود (مدتی طول می کشد تا امتیاز و گل شدن نمایش داده شود).

تابع `rel`

برای reload استفاده می شود.

تابع `ballMove`

برای حرکت توپ بر روی محور x ها است.

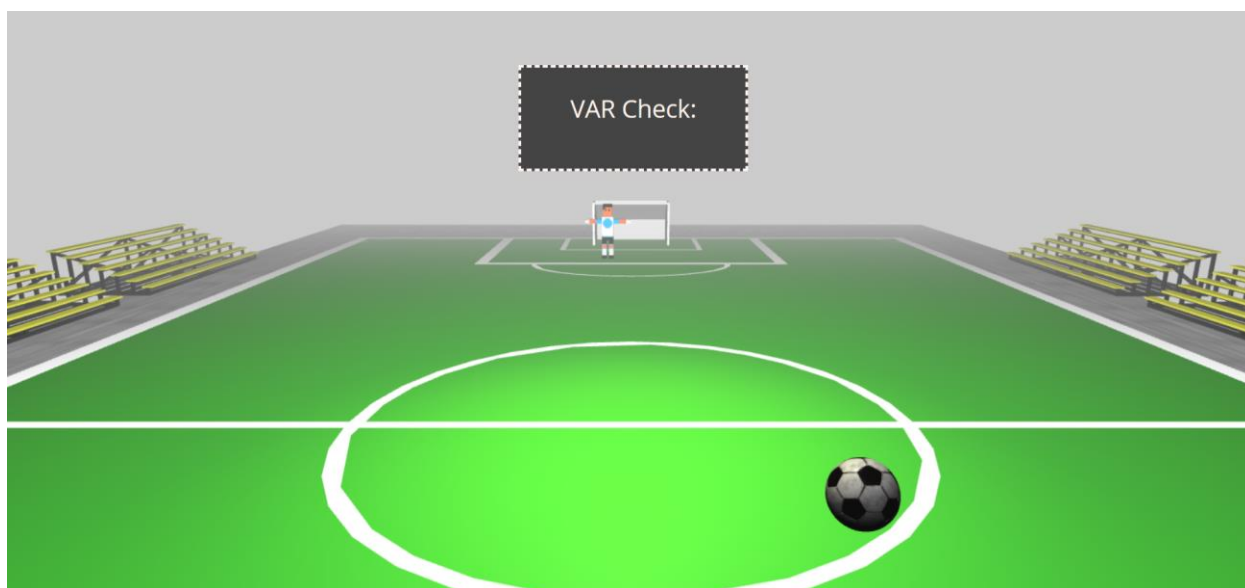
تابع `updatePhysics`

پس از تعریف فیزیک اجسام ، برای مطابقت دادن هر جسم تعریف شده در init به فیزیک مربوط به آن ، تابع `updatePhysics` توسط تابع `animate` صدا زده میشود .

تابع `onWindowResize`

این تابع برای موقعی است که کاربر اندازه صفحه مرورگر خود را تغییر میدهد.

در ادامه تصاویری از محیط بازی را مشاهده می کنید :



تمرین های انجام شده در این پروژه (اجباری کامل انجام شدند و اختیاری):

۱. استفاده از اجسام threejs

۲. نور

۳. حرکت و تغییر شکل اجسام

۴. صدا

۵. دوربین

۶. بافت

۷. مدل سه بعدی

۸. استفاده از کیبورد

۹. کنترل ها

۱۰. مه

۱۱. برخورد

۱۲. فیزیک

درصد مشارکت و تقسیم کار

تمامی اعضا به نسبت مساوی در تهیه این بازی همکاری کردند همچنین این پروژه در محیط گیت هاب (github) قرار داده شده و کامیت ها نیز براین مسئله گواه است.

با تشکر و احترام