" مستندات پروژه ی Fruit Ninja "

معرفی و عملکرد :

پروژه ی بازی فروت نینجا، نسخه ای ساده تر از بازی محبوب اندروید fruit ninja است. این بازی با زدن دکمه ی Space شروع و چندین نوع میوه مانند: سیب، هندوانه و... از بالای صفحه به پایین پرتاب می شود که در میان این میوه ها ممکن است بمب نیز وجود داشته باشد .

بازیکن با کلیک (Mouse Click) بر روی هر میوه ، یک امتیاز دریافت می کند و آن میوه از بین می رود. در قسمت بالا وسط صفحه ، امتیاز بازیکن نمایش داده می شود .

بازیکن برای بازی، 3 جان دارد که این جان ها به شکل قلب در قسمت بالا سمت چپ برنامه نمایان است. با کلیک بر روی هر بمب ، یک جان از جان های بازیکن کم شده و اگر هر 3 جان بازیکن از بین برود، Game Over خواهد شد .

بازیکن هم چنین می تواند از منوی واقع در بالا سمت راست برنامه، اقدام به انتخاب و تعویض بک گراند بازی کند .

مباحث و تمارین استفاده شده در پروژه :

- اجسام ThreeJS
- نور
- صدا
- دوربین
- بافت
- مدل های سه بعدی
- استفاده از کیبورد
- سایه ها
- دخالت در صحنه (Interaction)
- اعمال فیزیک بر روی Object ها

توضیحی کوتاه بر قسمت هایی از کد :

استفاده از Editor سایت threejs ، ساختن یک Plane و استفاده از پنل ها برای به دست آوردن اندازه های دقیق و تنظیمات : Material، Camera، SpotLight

دوربین : استفاده از PerspectiveCamera

نور : نور محیط (AmbientLight) به رنگ سفید ملایم ، قرار دادن SpotLight با مشخصات به دست آمده از طریق ادیتور ، قرار دادن 4 نور PointLight در 4 گوشه ی PlaneGeometry به همراه ایجاد سایه ، ایجاد رقص نور با تغییر سینوسی مقدار intensity

مدل ها و اجسام : ساختن دو کره و قرار دادن بافت های هندوانه و اسکلت ، اضافه کردن سه مدل آناناس، سیب و موز با فرمت gltf که این مدل ها برای ایجاد شدن در بالای صحنه (خارج از محدوده ی دوربین) ، در یک آرایه به نام Objects اضافه شده اند.

صدا : 1- ساختن صدای اصلی با استفاده از AudioLoader که شنونده ی این صدا Scene است. این صدا در کد لود شده و بلافاصله play نشد . 2- لود کردن sound effects (صدای شمشیر ، بمب و game over)

بافت : علاوه بر استفاده از بافت هندوانه و اسکلت بر روی مدل ها ، چندین تصویر پس زمینه بر روی PlaneGeometry ، به عنوان بک گراند های مختلف بازی نیز قرار داده شده است.

استفاده از کیبورد : برای شروع شدن بازی ، از Space استفاده شده است که پس از زدن space توسط بازیکن، بازی start خورده و آهنگ اصلی (Main Sound) ، play می شود.

سایه : در این برنامه،کتابخانه ی سایه اضافه شده و renderer از PCFSoftShadow استفاده می کند. نور هایی که ایجاد سایه می کنند شامل PointLight ها و SpotLight است و AmbientLight بدون سایه می باشد.

بک گراند PlaneGeometry ایجاد سایه نمی کند اما دریافت سایه را انجام می دهد. هم چنین، روی تمام Object های داخل بازی (میوه و...) علاوه بر ایجاد سایه، ممکن است سایه بیافتد.

برخورد : برخورد موس با اجسام بازی ؛ در هنگام کلیک موس بر نقطه ای ، موقعیت موس را به دست می آوریم که یک vector2 می باشد. سپس با استفاده از آن vector ، raycast و تنظیمات لازم، تشخیص می دهیم که آیا در این نقطه، موس با چیزی برخورد داشته یا خیر . در صورت برخورد، اندیس صفر در آرایه برخورد ها ، اولین برخورد است و بررسی می شود که در این اندیس، میوه یا بمب و... وجود دارد.

فیزیک : 1- استفاده از کتابخانه ی Cannon.js و ساخت یک دنیای فیزیکی به نام World که جاذبه ی آن کم تر از جاذبه ی عادی باشد (مقدار جاذبه 1- و در خلاف جهت محور y ها یعنی به سمت پایین درنظر گرفته شده است.) . علت کم کردن جاذبه آن است که به جای پایین آمدن طبیعی اجسام ، جاذبه روی آن ها اعمال شود و آرام تر به سمت پایین حرکت کنند تا بازیکن فرصت کلیک کردن بر روی اجسام و کسب امتیاز را داشته باشد.

2- نحوه ی فیزیک دار شدن اجسام: مدل های ساخته شده در ابتدا به آرایه ای از object ها، push شده است. در تابعی به نام creator ، به صورت رندوم یک clone از یکی از این object ها ساخته و شی ای نیز از کلاس PhysicalObject می سازیم.

حالت نمایشی PhysicalObject (چیزی که ما از PhysicalObject در بازی می بینیم) همان clone ای است که از یکی از اندیس های آرایه ی objects گرفته شده است و برای حالت فیزیکی آن، یک body جدید با جرم (mass) 1 ایجاد می کنیم. بدین صورت هرگاه یکی از مدل های بازی در بالای صحنه ایجاد شود بلافاصله دارای فیزیک شده و می افتد.

دخالت در صحنه (interaction) : با استفاده از کتابخانه ی dat.gui.min.js ، در بازی یک drop box تعبیه شده که به وسیله ی آن می توان بک گراند بازی را به دلخواه عوض کرد.

تقسیم کار در پروژه :

نازنین زیبایی :

طراحی صحنه و منوی بک گراند - نمایش تعداد جان های بازیکن - نور پردازی - بارگذاری مدل های gltf - صدا گذاری - فیزیک بازی

ماهان زیاری :

پیاده سازی صحنه ی پایه، renderer و دوربین – تنظیم اندازه ها و فاصله ها – نورپردازی – بارگذاری مدل ها و اجسام – ساخت نمایشگر امتیاز – برخورد موس با اجسام – رفع برخی ایرادات