

بنام خدا



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیووتر

پروژه پایانی درس گرافیک کامپیووتری

(ویرایش اول)

استاد درس

دکتر علیرضا باقری

نگارش

تیم تدریسیاری درس گرافیک کامپیووتری

خرداد ۹۹

مقدمه

هدف از انجام این پروژه بکار گیری و تثبیت مفاهیم اولیه گرافیک ۳-بعدی، جبر خطی ۳-بعدی و مبانی OpenGL است. خروجی این پروژه یک نرمافزار با قابلیت شبیه‌سازی یک فضای ۳-بعدی است که در آن دوربین به مانند روح در صفحه حرکت می‌کند و می‌تواند با اشیاء ۳-بعدی موجود در فضا تعامل داشته باشد و اشیاء جدید به فضا اضافه کند.

این پروژه شامل بخش‌های اجباری و **امتیازی** است. در بخش اجباری این پروژه به مباحث: تنظیمات اولیه OpenGL و پنجره نمایش، کار با بافر رئوس^۱، کار با بافر المان‌ها^۲، کار با آرایه و صفات رئوس^۳، سایه‌زن‌ها^۴، بافت‌ها^۵، ماتریس دگرگونی^۶، ماتریس دید^۷ و ماتریس تجسم^۸ پرداخته می‌شود. در بخش **امتیازی** این پروژه به کارهای پیچیده‌تر با مباحث مطرح شده و کار با مباحث جدید پرداخته می‌شود.

منابع

برای پیاده‌سازی همه بخش‌های مربوط به بخش گرافیک این پروژه می‌توانید از منابع مختلف اینترنتی بدون هیچ محدودیتی استفاده کنید. برای پیاده‌سازی بخش‌های منطق این پروژه می‌توانید از منابع مختلف اینترنتی، با ذکر منبع در فایل گزارش، بدون هیچ محدودیتی استفاده کنید.

تماشای ویدئوهای آموزشی قرار داده شده در سایت درس اکیدا توصیه می‌شود.

در [این](#) مخزن گیت‌هاب پروژه‌ای برای شما با استفاده از مباحثی که با آن آشنا هستید ایجاد شده و می‌توانید با مراجعه به آن با یک نمونه معماری مناسب برای این پروژه آشنا شوید. توجه داشته باشید پروژه موجود در مخزن معرفی شده ناقص است و نمی‌توان از آن به عنوان یک پروژه کامل با ساختاری کامل یاد کرد.

VBO ^۱

EBO ^۲

Vertex Attributes ^۳ و VAO ^۴

Shaders ^۵

Textures ^۶

Transformation Matrix ^۷

View Matrix ^۸

Projection Matrix ^۹

نکات مهم در انجام پروژه

- حتما قبل از شروع به پیاده‌سازی این پروژه، تمام صورت پروژه را مطالعه کنید.
- برای ساخت تصاویر می‌بایست حتما از کتابخانه OpenGL استفاده شود. استفاده از کتابخانه‌های غیر مرتبط و حتی کتابخانه‌های مشتق شده از OpenGL مجاز نمی‌باشد. نسخه OpenGL مورد استفاده می‌بایست 3.0 به بالا باشد و استفاده از نسخه 2.x یا پایین‌تر مجاز نمی‌باشد.
- برای استفاده از OpenGL می‌بایست از حالت Core-Profile استفاده شود و استفاده از حالت Immediate مجاز نیست.
- برای پیاده‌سازی این پروژه می‌بایست از زبان‌های C، C++, Java، Visual Basic، C#، Visual C++/CX استفاده کنید و استفاده از زبان‌های دیگر و حتی زبان‌های مشتق شده از این زبان‌ها مجاز نمی‌باشد.
- ساختار فایل‌های پروژه می‌بایست به گونه‌ای باشد که فایل همه کتابخانه‌های خارجی (به غیر از OpenGL و کتابخانه‌های موجود در محیط سیستم‌عامل) و تمامی فایل‌های مورد استفاده در پروژه، در داخل پوشه پروژه قرار گرفته و همه آدرس‌ها به صورت نسبی وارد شده باشد.
- در زمان ارسال پروژه خود می‌بایست کل پوشه پروژه را به طور کامل ارسال کنید (با حفظ ساختار فایل‌ها و پوشه‌های درون آن) و اگر از قالب‌کار^۹ خاصی برای پیاده‌سازی اجرای پروژه خود استفاده می‌کنید، نحوه اجرا شدن برنامه خود را روی فایل گزارش توضیح دهید.
- برای انجام این پروژه به غیر از موارد مطرح شده در صورت پروژه نیازی به نوشتن گزارش نیست.
- صورت این پروژه ممکن است کمی طولانی بنظر برسد که آن هم بدلیل توضیحات کامل هر بخش می‌باشد تا هیچ ابهامی باقی نماند.
- هر تغییری روی صورت پروژه به منظور رفع ابهام یا توضیحات تکمیلی، به صورت نسخه جدید از صورت پروژه در اختیار شما قرار داده می‌شود و ایجاد نسخه جدید به شما اطلاع‌رسانی خواهد شد. هر گونه توضیحات اضافه در هر محیط دیگر، هیچ گونه رسمیتی ندارد و شما در زمان پیاده‌سازی کافی است به آخرین ویرایش صورت پروژه استناد کنید.
- همان‌طور که پیشتر نیز مطرح شد، این پروژه شامل دو بخش **امتیازی** و اجباری است. نمره هر بخش **امتیازی** تنها در صورتی به شخص تعلق می‌گیرد که تمامی بخش‌های اجباری را پیاده‌سازی کرده باشد (در صدی قابل قبول از خطأ در بخش‌های اجباری هم مورد تایید است). بنابراین توصیه می‌شود اول به پیاده‌سازی قسمت‌های اجباری بپردازید.
- بخش‌هایی که به صورت **امتیازی** مشخص شده‌اند جزو بخش‌های اجباری نیستند.
- برای گرفتن نمره بخش‌های **امتیازی** که ذیل بخش‌های **امتیازی** دیگر تعریف شده‌اند می‌بایست ابتدا بخش **امتیازی** "شامل" را قبل از بخش "مشمول" پیاده‌سازی کنید.
- قسمت **امتیازی** به بخش‌های **امتیازی**، **امتیازی ویژه** و **امتیازی طلایی** تقسیم می‌شوند که بسته به حجم کار آن بخش، نوع امتیازشان تعیین می‌شود. نمره **امتیازی** این بخش‌های مطرح شده به ترتیب از کم‌تر به بیشتر است.

- شما می‌بایست به ازای هر بخش **امتیازی** (از هر سه نوع) که پیاده‌سازی می‌کنید، آن را در گزارش خود مطرح کرده و توضیح مختصری از نحوه پیاده‌سازی و استفاده از آن ذکر کنید.
- در بخش‌های **امتیازی** ممکن است با اصطلاحاتی مواجه شوید که با آن‌ها آشنا نیستید. شما می‌توانید به اختیار خود از آن‌ها بگذرید یا با آن‌ها آشنا شوید و آن‌ها را پیاده‌سازی کنید.

صورت پروژه

۱. راهاندازی

در ابتدا می‌بایست پنجره‌ای ایجاد شود تا محیط ۳-بعدی از داخل آن قابل نمایش باشد. صفحه‌ای که محیط ۳-بعدی را نمایش می‌دهد و داخل این پنجره قرار گرفته است، می‌بایست همواره هم اندازه خود پنجره باشد (بدون در نظر گرفتن نوار بالا و حاشیه‌های پنجره). نرم‌افزار می‌بایست بتواند به خوبی به تغییرات اندازه پنجره نمایش رسیدگی کند و تغییرات اندازه پنجره نمایش نباید باعث اختلال در تصاویر ایجاد شده یا از فرم خارج شدن آن بشود.

تصویری که پنجره نمایش نشان می‌دهد می‌بایست یک محیط ۳-بعدی با تجسم پرسپکتیو^۱ باشد. دید در این محیط ۳-بعدی از نقطه نظر یک دوربین انجام می‌شود (که در ادامه به توضیح آن خواهیم پرداخت).

تصاویر می‌بایست حداقل ویژگی‌های یک تصویر مناسب را داشته باشند و قوانین طبیعی را نقض نکند. مثلاً اجسام دورتر در فضا نباید در تصویر جلوی اجسام نزدیک‌تر قرار بگیرند.

- **امتیازی:** برای پیاده‌سازی برنامه از برنامه‌نویسی شیء‌گرا استفاده شود. ساختار فایل‌های کلاس و تصمیم استفاده از یک یا چند فایل برای پیاده‌سازی کلاس به عهده فرد می‌باشد.

۲. اجسام هندسی ابتدایی

در این پروژه می‌بایست تعدادی جسم ۳-بعدی ابتدایی پیاده‌سازی شوند. این اجسام ابتدایی شامل مکعب، هرم و منشور می‌شود. برای سهولت پیاده‌سازی می‌توانید مقاطع منشور و هرم را مثلث در نظر بگیرید.

برنامه می‌بایست قابلیت این را داشته باشد که قبل از زمان اجرا و در داخل کد به راحتی بتوان هر کدام از این اجسام را با چرخش^{۱۱}، مقیاس^{۱۲} و موقعیت^{۱۳} دلخواه در محیط ۳-بعدی قرار داد. چرخش، مقیاس و موقعیت یک جسم می‌بایست با **Transform** اعمال شود.

Perspective^{۱۰}

Rotation^{۱۱}

Scale^{۱۲}

Position^{۱۳}

اجسام ۳-بعدی ابتدایی (مکعب، هرم و منشور) می‌باشد دارای بافت روی همه سطوح خود باشند.

- امتیازی: پیاده‌سازی استوانه
 - امتیازی: ایجاد بافت روی سطح استوانه
 - امتیازی: پیاده‌سازی کره
 - امتیازی: اینجاد بافت روی سطح کره
 - امتیازی: استفاده از Tessellation برای پیاده‌سازی کره یا استوانه
 - امتیازی: ایجاد مدل ۳-بعدی یک شکل پیچیده مانند یک انسان یا هر شیء پیچیده دیگر از روی فایل با استفاده از کتابخانه‌های مناسب
 - امتیازی: ایجاد بافت روی مدل ۳-بعدی پیچیده به صورت که با آن منطبق باشد
- برای ساخت هر کدام از اجسام مطرح شده می‌توانید از منابع موجود در اینترنت (مثلاً استفاده از لیست موقعیت‌های رؤوس یک کره موجود در اینترنت) استفاده کنید.
- امتیازی: پیاده‌سازی حالت ثانویه رنگ شدن سطح همه اجسام پیاده‌سازی شده. این رنگ‌آمیزی می‌تواند رنگ یکدست، رنگ ترکیبی یا بافت باشد. (در ادامه نحوه و زمان تغییر رنگ‌آمیزی جسم به حالت ثانویه و به حالت بافت اصلی مطرح خواهد شد)
 - امتیازی: پیاده‌سازی بافت برای همه اجسام در حالت رنگ‌آمیزی ثانویه
 - امتیازی: پیاده‌سازی حداقل یک Material برای سطح حداقل یک جسم (شامل نور محیطی، نور پخشی، نور بازتابی و درخشش)
 - امتیازی: پیاده‌سازی منبع نور^{۱۴}
 - امتیازی: پیاده‌سازی منبع نور جهت‌دار^{۱۵}
 - امتیازی: پیاده‌سازی منبع نور نقطه‌ای^{۱۶}
 - امتیازی: پیاده‌سازی منبع نور نورافکن^{۱۷}
 - امتیازی: پیاده‌سازی سایه
 - امتیازی: پیاده‌سازی نقشه نور پخشی
 - امتیازی: پیاده‌سازی نقشه نور بازتابی

Light Caster^{۱۴}

Directional Light^{۱۵}

Point Light^{۱۶}

Spotlight^{۱۷}

۲/۱. راهنمایی بخش

برای پیاده‌سازی این بخش توصیه می‌شود کلاسی در نظر گرفته شود که دارای خواص و قابلیت‌های اصلی یک جسم باشد و برای اجسام مختلف کلاس‌های دیگری درست ایجاد شود و از این کلاس به ارت برزن.

بهتر است برای هر جسم یک VBO، لیست رئوس، EBO، لیست المان‌ها، Texture، Transform و VAO مختص خود در نظر گرفته شود. این موارد می‌توانند در کلاس پدر قرار بگیرند زیرا همه‌ی اجسام دارای این موارد هستند. این موارد می‌توانند در سازنده^{۱۸} کلاس‌های خود مقادیر اولیه را خود را دریافت کنند.

همه‌ی اجسام می‌توانند در یک Vector (ساختمن داده‌ای شبیه به لیست با طول نامتناهی) قرار بگیرند و در پردازش هر فریم، ابتدا تابع تصویرسازی^{۱۹} همه، سپس تابع بروزرسانی^{۲۰} همه‌ی اجسام فراخوانی شود. وظیفه تابع تصویرسازی، کشیدن همان جسم روی صفحه و وظیفه تابع بروزرسانی یک جسم، انجام منطق آن جسم در آن فریم است. توجه داشته باشید که همیشه بهتر است همه‌ی منطق مربوط به یک جسم در کلاس خود آن تعریف شود و با صدا شدن تابع بروزرسانی آن، تمام تغییرات مورد نظر روی آن جسم در آن فریم اعمال شود.

هنگامی که تابع تصویرسازی یک جسم فراخوانی شد، می‌بایست همه‌ی دارایی‌های^{۲۱} مورد نیاز آن جسم برای تصویرسازی (مانند VBO و VAO در OpenGL) در حال bind شده درآید و متغیرهای مورد نیاز Shader برای آن جسم (مانند Transform) نیز به صورت Uniform برای Shader فرستاده شود. در این صورت بعد از فراخوانی تابع glDrawElements یا glDrawArrays که تابع تصویرسازی OpenGL هستند، آن جسم روی پنجره کشیده می‌شود. بعد از صدا شدن این تابع برای همه اجسام در یک فریم، همه اجسام به همان گونه که انتظار داریم روی صفحه نمایش داده می‌شوند.

۳. دوربین

محیط ۳-بعدی برنامه از نقطه نظر یک دوربین دیده می‌شود که خود در این محیط ۳-بعدی قرار دارد. دوربین دارای دو دارایی اصلی است:

۱. موقعیت دوربین
۲. جهت دوربین

موقعیت دوربین همان مکان آن در محیط ۳-بعدی است و مشخص می‌کند نگاه کردن به محیط ۳-بعدی از کدام نقطه انجام می‌شود.

جهت دوربین مشخص می‌کند که دوربین از مکانی که در آن قرار گرفته، به کدام جهت نگاه می‌کند و اشیاء کدام سمت خود را می‌بیند می‌کند. دو روش متدائل برای پیاده‌سازی جهت دوربین وجود دارد:

۱. نگهداری چرخش دوربین

Constructor^{۱۸}

Render^{۱۹}

Update^{۲۰}

Property^{۲۱}

۲. نگهداری بردار روبه‌رو و بردار بالا

چرخش به صورت مستقیم می‌تواند بردار بالا، راست و مستقیم دوربین را مشخص کند. برای این کار کافی است ماتریس چرخش را در بردارهای $(0,1,0)$, $(1,0,0)$ و $(0,0,-1)$ ضرب کنیم تا به ترتیب بردارهای بالا، راست و مستقیم دوربین بدست بیاید.

بردار بالا و روبه‌رو در مرحله اول می‌تواند بردار راست را مشخص کند (با ضرب خارجی بردار روبه‌رو و بالا) و در مرحله دوم می‌تواند با عملیات مثلثاتی، زوایای چرخش و ماتریس چرخش را تعیین کند.

این دوربین قابلیت حرکت در فضای ۳-بعدی را به گونه‌ای دارد که به آن حالت روح گفته می‌شود. نمونه آن را می‌توان در بازی‌هایی مثل بازی Counter-Strike (حالت free look بعد از مرگ) مشاهده کرد. در این حالت دوربین قابلیت حرکت با توجه به جهت قرارگیری خود و قابلیت چرخش با توجه به حرکت نشانگر ماوس را دارد.

۳/۱. حرکت دوربین

دوربین با ۴ دکمه قابلیت حرکت دارد:

- حرکت به سمت جلو (در راستای بردار روبه‌روی دوربین) با فشردن دکمه W
- حرکت به سمت عقب (در خلاف راستای بردار روبه‌روی دوربین) با فشردن دکمه S
- حرکت به سمت راست (در راستای بردار راست دوربین) با فشردن دکمه D
- حرکت به سمت چپ (در خلاف راستای بردار راست دوربین) با فشردن دکمه A
- امتیازی: حرکت به سمت بالا (در راستای بردار بالای دوربین) با فشردن دکمه E
- امتیازی: حرکت به سمت پایین (در خلاف راستای بردار بالای دوربین) با فشردن دکمه Q

توجه داشته باشید که بردارهای جهت مطرح شده (روبه‌رو، راست و بالا)، بردارهای نسبی هستند و با توجه به جهت دوربین مشخص می‌شوند.

توجه داشته باشید که مادامی که هر کدام از این دکمه‌ها در حالت فشرده است، دوربین به حرکت خود در راستای تعیین شده ادامه می‌دهد و در هنگامی که فشردن دکمه متوقف می‌شود، دوربین از حرکت در آن راستا می‌ایستد. این دکمه‌ها می‌توانند به صورت ترکیبی و همزمان فشرده شوند و حرکت دوربین در این حالت از قائدۀ جمع برداری تبعیت می‌کند.

- امتیازی: سرعت حرکت دوربین در هر راستا باید همواره ثابت باشد و نرخ فریم برنامه نباید تاثیری روی سریع و کند شدن سرعت حرکت داشته باشد.

۳/۲. چرخش دوربین

دوربین دو چرخش اصلی به صورت Yaw و Pitch دارد.

چرخش Yaw همیشه حول بردار بالای جهان $(0,1,0)$ انجام می‌شود. این چرخش با حرکت نشانگر ماوس در راستای X صفحه ایجاد می‌شود. چرخش اصلی به این صورت است که با حرکت ماوس در راستای مثبت X صفحه، دوربین به سمت راست خود و با حرکت ماوس در راستای منفی صفحه، دوربین به سمت چپ خود می‌چرخد.

- امتیازی: پیاده‌سازی تغییر حالت چرخش **Yaw** به صورت معکوس. در این حالت جهت چرخش دوربین به ازای حرکت ماوس، بر عکس چرخش اصلی تعریف شده در بالا می‌باشد (با حرکت ماوس در راستای مثبت **X** به سمت چپ می‌چرخد). برای این تغییر حالت می‌بایست دکمه‌ای در نظر گرفته شود که با فشردن آن دوربین از چرخش اصلی به چرخش معکوس و از چرخش معکوس به چرخش اصلی تغییر وضعیت دهد.

چرخش **Pitch** همیشه حول بردار سمت راست دوربین (بردار نسبی) انجام می‌شود. این چرخش با حرکت نشانگر ماوس در راستای **Z** صفحه ایجاد می‌شود. چرخش اصلی به این صورت است که با حرکت ماوس در راستای مثبت **Z** صفحه، دوربین به سمت بالا خود و با حرکت ماوس در راستای منفی صفحه، دوربین به سمت چپ خود می‌چرخد.

- امتیازی: پیاده‌سازی تغییر حالت چرخش **Pitch** به صورت معکوس. در این حالت جهت چرخش دوربین به ازای حرکت ماوس، بر عکس چرخش اصلی تعریف شده در بالا می‌باشد (با حرکت ماوس در راستای مثبت **Z** به سمت پایین می‌چرخد). برای این تغییر حالت می‌بایست دکمه‌ای در نظر گرفته شود که با فشردن آن دوربین از چرخش اصلی به چرخش معکوس و از چرخش معکوس به چرخش اصلی تغییر وضعیت دهد.

۴. عملیات اضافه و تغییر اجسام

این بخش شامل دو عمل اصلی اضافه کردن و تغییر اجسام فضای ۳-بعدی است. برای پیاده‌سازی این قسمت دو وضعیت^{۲۲} برای برنامه در نظر گرفته می‌شود. این دو وضعیت از دو عمل اصلی متفاوت‌اند و در ادامه به تعریف و رابطه آن‌ها می‌پردازیم.

۴/۱. وضعیت عادی

در وضعیت عادی همه اجسام فضا بدون تحرک در جای خود به صورت ثابت قرار گرفته‌اند و تغییری روی آن‌ها نسبت به جهان صورت نمی‌گیرد. در این وضعیت دوربین به صورت روح در فضا حرکت می‌کند.

۴/۲. وضعیت تغییر

• تغییرات مکان جسم انتخاب شده

در این وضعیت جسم انتخاب شده (در مورد نحوه انتخاب آن در ادامه توضیحاتی ارائه می‌شود) روبروی دوربین قرار می‌گیری و با حرکت دوربین در جهات مختلف و چرخش دوربین، مکان آن جسم نسبت به جلوی دوربین حفظ می‌شود. جسم انتخاب شده نباید چرخشی داشته باشد. برای مثال اگر مکعبی انتخاب شود و یکی از وجهه آن روبروی دوربین قرار گیرد، از آنجا که چرخش مکعب در جهان تغییر نمی‌کند با چرخش دوربین، ممکن است وجهی از آن مکعب که روبروی دوربین است تغییر کنید. برای تفهیم بیشتر، فرض کنید در اتاقی **۴** دیواری (با دیوارهای موازی و عمود بر هم) قرار دارید و مکعبی را دور سر خود می‌چرخانید به صورتی که وجهه این مکعب همواره موازی دیوارهای این اتاق باقی بماند. در این صورت آن وجه از مکعب که به سمت شما قرار دارد با هر **۹۰** درجه چرخش دور سر شما عوض می‌شود.

راهنمایی: کافیست برای پیاده‌سازی چرخش جسم انتخاب شده هیچ کاری انجام ندهید(!) و صرفاً مکان جسم انتخاب شده را نسبت به دوربین ثابت نگهدارید.

راهنمایی: برای حفظ مکان جسم انتخاب شده نسبت به دوربین، کافی است بردار روبروی دوربین را با بردار مکان دوربین جمع کرده و مکان جسم انتخاب شده را برابر حاصل قرار دهید. در این صورت مکان جسم انتخاب شده، در هر فریم، برابر یک واحد به سمت روبروی دوربین است.

تغییر مکان جسم می‌بایست با استفاده از **Transform** پیاده‌سازی شود.

• تغییرات چرخش جسم انتخاب شده

زمانی که در وضعیت تغییر هستیم، با فشردن بعضی از کلیدها جسم انتخاب شده می‌بایست حول محورهای مختصات جهان (($1,0,0$) و ($0,0,1$) و ($0,1,0$) دوران انجام دهد.

- مادامی که کلید H فشار داده می‌شود، جسم در جهت مثبت چرخش Yaw انجام دهد
- مادامی که کلید F فشار داده می‌شود، جسم در جهت منفی چرخش Yaw انجام دهد
- امتیازی: مادامی که کلید T فشار داده می‌شود، جسم در جهت مثبت چرخش Pitch انجام دهد
- امتیازی: مادامی که کلید B فشار داده می‌شود، جسم در جهت منفی چرخش Pitch انجام دهد
- امتیازی: مادامی که کلید Z فشار داده می‌شود، جسم در جهت مثبت چرخش Roll انجام دهد
- امتیازی: مادامی که کلید R فشار داده می‌شود، جسم در جهت منفی چرخش Roll انجام دهد

توجه داشته باشید که کلیدها می‌توانند به صورت همزمان فشرده شوند.

این تغییر زوایا باید به صورت همگن انجام شود. مثلاً در هر فریم، π درجه دوران در راستای متناظر با کلید فشرده شده.

تغییرات چرخش جسم می‌بایست با استفاده از **Transform** پیاده‌سازی شود.

• تغییرات مقیاس جسم انتخاب شده

زمانی که در وضعیت تغییر هستیم، با فشردن بعضی از کلیدها، جسم انتخاب شده می‌بایست در راستای محورهای مختصات محلی خود (راست، بالا، پشت/جلو) تغییر مقیاس دهد.

- مادامی که کلید L فشار داده می‌شود، جسم در جهت راست/چپ خود افزایش طول دهد
- مادامی که کلید L فشار داده می‌شود، جسم در جهت راست/چپ خود کاهش طول دهد (وقتی به صفر رسید متوقف شود)
- امتیازی: مادامی که کلید I فشار داده می‌شود، جسم در جهت بالا/پایین خود افزایش طول دهد
- امتیازی: مادامی که کلید M فشار داده می‌شود، جسم در جهت بالا/پایین خود کاهش طول دهد (وقتی به صفر رسید متوقف شود)
- امتیازی: مادامی که کلید O فشار داده می‌شود، جسم در جهت پشت/جلو خود افزایش طول دهد
- امتیازی: مادامی که کلید L فشار داده می‌شود، جسم در جهت پشت/جلو خود کاهش طول دهد (وقتی به صفر رسید متوقف شود)

این تغییر مختصات باید به صورت همگن پیاده‌سازی شود. مثلاً در هر فریم، π واحد در راستای متناظر با کلید فشرده شده.

تغییرات مقیاس هر جسم می‌بایست با استفاده از **Transform** پیاده‌سازی شود.

۴/۳ عملیات روی اجسام

در حالت شروع برنامه، برنامه در وضعیت عادی قرار دارد. یعنی حرکت و چرخش دوربین.

• ایجاد اجسام

برنامه می‌بایست این قابلیت را داشته باشد که زمانی که در وضعیت عادی قرار دارد، با فشردن کلیدهایی یکی اجسام هندسی پیاده‌سازی شده به محیط ۳-بعدی اضافه شود. انتخاب این کلیدها به عهده فرد می‌باشد و می‌بایست به ازای هر کدام از اجسام پیاده‌سازی شده، یک کلید اختصاص داده شود. بعد از فشردن کلید متناظر با یک جسم (مثلاً مکعب) آن جسم در جلو دوربین ایجاد می‌شود و به حالت انتخاب شده در می‌آید. در این هنگام برنامه از وضعیت عادی به وضعیت تغییر می‌رود. در این وضعیت تمام عملیات تعریف شده در بالا روی جسم انتخاب شده (جسم تازه ایجاد شده) قابل انجام است.

هر گاه دکمه **Space** فشرده شد، در صورتی که برنامه در حالت تغییر باشد، ابتدا برنامه به حالت عادی می‌رود و سپس جسم انتخاب شده، از حالت انتخاب شده خارج می‌شود و در همان مکانی که قرار دارد (جلوی دوربین) با همان چرخش و با همان مقیاس رها می‌شود. با توجه به این که جسم از حالت انتخاب شده خارج شده است دیگر با حرکت دوربین حرکت نمی‌کند و به کلیدها واکنشی نشان نمی‌دهد. به این صورت جای جدید، چرخش جدید و مقیاس جدید شکل مشخص می‌شود.

• امتیازی: تغییر اجسام

برنامه می‌بایست قابلیت این را داشته باشد که زمانی که در وضعیت عادی قرار دارد، وقتی با حرکت و چرخش دوربین، جسمی روبروی آن با فاصله مناسب قرار می‌گیرد با فشردن دکمه **Space**. ابتدا آن جسم انتخاب شده و سپس برنامه وارد وضعیت تغییر شود. مانند حالت قبل نیز برنامه با فشردن **Space** از وضعیت تغییر به وضعیت عادی برگرد و جسم از حالت انتخاب شده خارج شود.

- امتیازی: در صورتی که ۲ نحوه رنگ‌آمیزی برای سطح جسم پیاده‌سازی شده باشد، هر جسم زمانی که به حالت انتخاب شده در می‌آید، می‌بایست با نحوه رنگ‌آمیزی ثانویه خود نمایش داده شود.

۵. قسمت‌های امتیازی اضافه

- امتیازی: برنامه قابلیت تصویرسازی از مثلث‌های مدل‌های ۳-بعدی فقط یا یک جهت را داشته باشد. برای مثلاً اگر یک مکعب از بیرون آن دیده می‌شود، وقتی دوربین به داخل آن حرکت می‌کند نباید هیچ قسمی از مکعب نمایش داده شود. (**Culling**)
- امتیازی: برنامه قابلیت فعل و غیرفعال کردن **Culling** را در زمان اجرا با فشردن یک دکمه داشته باشد.
- امتیازی: نرم کردن لبه‌ها (Anti-Aliasing): برنامه قابلیت نرم کردن لبه‌های مدل‌های ۳-بعدی را با الگوریتم **MSAA** داشته باشد.
- امتیازی: برنامه علاوه بر **MSAA** قابلیت استفاده از الگوریتم دیگری را برای نرم کردن لبه‌ها داشته باشد.
- امتیازی: برنامه قابلیت عوض کردن الگوریتم نرم کردن لبه را بین الگوریتم‌های پیاده‌سازی شده (حداقل **MSAA**) و حالت غیر فعال را داشته باشد.

- امتیازی: پیاده‌سازی برخورد^{۳۳}: دوربین قابلیت تشخیص برخورد با اجسام ابتدایی موجود در فضای داشته باشد و نتواند داخل حجم اجسام ابتدایی شود. این قابلیت باید چرخش و مقیاس اجسام را نیز در نظر بگیرد.
- امتیازی: برنامه قابلیت چاپ کردن نرخ فریم روی کنسول را داشته باشد.
- امتیازی: پیاده‌سازی HUD: برنامه قابلیت نمایش مختصات دوربین، وضعیت برنامه (عادی، تغییر) را روی صفحه نمایش به صورت ۲بعدی داشته باشد.
- امتیازی: برنامه قابلیت چاپ کردن نرخ فریم روی صفحه را داشته باشد.
- امتیازی: استفاده از عماری‌های استاندارد و متداول برای برنامه‌های گرافیکی.
- امتیازی: پیاده‌سازی برنامه به روش برنامه‌نویسی داده گرا^{۴۴}.
- امتیازی: نوشتن گزارش کامل از پروژه.

نکات مهم در زمان تحويل

- این پروژه برای انجام به صورت انفرادی طراحی شده و هیچ گونه "همفکری" و "با هم کد زدن" در آن معنا ندارد. کدها برای تست تشابه توسط سامانه "Moss" بررسی خواهند شد. با این حال استفاده از منابع اینترنتی برای قسمت‌های گرافیکی آزاد و برای قسمت‌های منطق با ذکر منبع (از جمله سایت Learn OpenGL یا مخزن گیت معرفی شده در صورت استفاده) ایرادی ندارد.
- در صورت تشخیص تقلب، به هر دو نفر نمره منفی تعلق خواهد گرفت و با موارد تقلب به سختی برخورد خواهد شد.
- این پروژه به صورت غیر حضوری تحويل گرفته خواهد شد و نمره هر فرد زمان تحويل مشخص می‌شود.
- شما باید تسلط لازم را به همه‌ی قسمت‌های پروژه خود داشته باشید. این تسلط می‌تواند با پرسش در مورد قسمت‌های پیاده‌سازی شده یا با تغییر در قسمت‌هایی از برنامه توسط شما در زمان تحويل مشخص شود.
- در زمان تحويل تسلط شما به پروژه مورد بررسی قرار گرفته و به آن نمره‌ای بین ۰ تا ۱ تعلق می‌گیرد. این نمره در نمره بقیه بخش‌های پروژه ضرب می‌شود و حاصل آن نمره نهایی پروژه شمامست. برای مثال اگر فردی بدليل تسلط پایین نمره ۰.۵ را از قسمت تسلط دریافت کند، حداکثر نمره‌ای که از پروژه می‌تواند بگیرد نصف نمره آن است.
- فایل ارسالی شما باید یک فایل `zip != rar` باشد و از فرمت نام زیر پیروی کند:

`<studentId>_proj_<IDENName>.zip`

نام IDE مورد استفاده خود را به صورتی اختصاری ننویسید که قابل تشخیص نباشد و از حروف کوچک به هم چسبیده استفاده کنید. در صورتی هم که از CMake استفاده می‌کنید، به جای نام IDE عبارت `cmake` را بنویسید. مثلا:

9531095_GHW2_intellij.zip
9531095_GHW2_visualstudio.zip
9531095_GHW2_cmake.zip

- در صورت داشتن هر گونه سوال یا ابهام، با ایمیل samimd.77@gmail.com یا اکانت تلگرامی @SamiMD ارتباط برقرار کنید.

موفق باشید