گزارش شماره: ۷
تاریخ: **1401/01/26**

بسمه تعالی

گزارش هفتگی پیشرفت پایان‌نامه

نام دانشجو: نامی نذیری　　شماره دانشجویی: 9736223

1. کارهای انجام‌شده (شامل مقالات خوانده‌شده، برنامه‌های نوشته‌شده، آزمایش‌های انجام‌شده و نتایج مختصر آن‌ها):

- باگ موجود در وارد کردن کاراکتر ها رفع شد. البته هنوز می‌توان بر روی وارد کردن فایل ها وقت بیشتری گذاشت.
- کلاس های مربوط به State Machine اضافه شد و یک تست جزئی نیز صورت گرفت. باید تست های بیشتری را انجام داد.

توضیح کلی برای API فعلی ماشین حالت

ماشین حالت از سه کلاس کلی تشکیل شده. کلاس StateMachine ، State و Transition

```
Animator* animator = new Animator(animationClipManager.GetSkeleton(), [&]*anim, glfwGetTime());  // TODO new?????/

AnimationState* IDLE_STATE = new AnimationState("IDLE",animationClipManager.GetLoadedAnimationClips()[0]); // TODO new?????/
Transition* tranIdleToWalk = new Transition( to: "WALK",  evaluateFunction: []()->bool { return changeState; },  transitionTime: 0.1);
IDLE_STATE->AddNewTransition(tranIdleToWalk);

AnimationState* Walk_STATE = new AnimationState("WALK", animationClipManager.GetLoadedAnimationClips()[1]); // TODO new?????/
Transition* tranWalkToIdle = new Transition( to: "IDLE",  evaluateFunction: []()->bool {return !changeState; },  transitionTime: 0.14);
Walk_STATE->AddNewTransition(tranWalkToIdle);

AnimationStateMachine animState(animator, IDLE_STATE); // TODO new?????/
animState.AddNewState(Walk_STATE);
```

برای مثال در اینجا دو State حالت بیکار و راه رفتن را ساخته ام. یک transition نیز از حالت بیکار به حالت راه رفتن و یک transition دیگر نیز برای حالت راه رفتن به بیکار ساخته شده است.

سازنده هر State موارد زیر را به عنوان ورودی دریافت می کند

- **نام حالت**
- **اشاره گری به کلیپ موجود**

سازنده هر StateMachine موارد زیر را به عنوان ورودی دریافت می کند.

- **انیماتور(که وظیفه پخش کلیپ های انیمیشن را دارد)**
- **حالت اولیه**

هر transition مربوط به یک state است و ورودی هایش به صورت زیر است

- **استیتی که به وسیله‌ی این انتقال می خواهیم برویم**

- **تابعی که در صورت درست بودن باعث می شود این انتقال صورت گیرد (در اینجا از لامبدا استفاده شده. حتما باید تابع خروجی bool برگرداند.**
- **مدت زمانی که این انتقال طول میکشد. (در هنگام انتقال میان ژست پایانی انیمیشن فعلی در زمان حال با ژست ابتدایی انیمیشن بعدی ترکیب (blend) صورت می گیرد. این متغیر مدت زمانی که طول میکشد این ترکیب کامل شود را مشخص می کند.)**

در نهایت در لوپ نهایی تابع آپدیت کلاس StateMachine صدا زده می شود.

عملکرد تابع آپدیت به صورت زیر است:

ابتدا تمامی انتقال های حالت فعلی را بررسی می کند که آیا در فریم جاری انتقالی صورت می گیرد یا خیر.

در صورتی که انتقال صورت نگرفته باشد، آپدیت انیماتور را صدا میزد تا ژست کاراکتر آپدیت شود

در صورتی که انتقال صورت گیرد، ژست فعلی کاراکتر را بدست می آورد، ژست اولیه در استیت بعدی را نیز بدست می آورد و تابع TransitionUpdate را صدا می زند. این تابع خروجی bool دارد که نشان دهنده ی این است که آیا این انتقال تمام شده یا خیر. پس در فریم های بعدی اگر انتقال تمام نشده بود، آن را ادامه می دهد.

پس از اتمام انتقال، حالت فعلی عوض شده و کلیپ انیماتور را با استفاده از حالت فعلی تغییر می دهیم. و این عملیات دوباره تکرار می شود.

2. کارهای موردنیاز جهت ادامه در هفته های آتی:

- اضافه کردن CubeMaps به پروژه
- پیشنهادات:

i. بهتر کردن سرعت برنامه مخصوصا در هنگام اضافه کردن انیمیشن یا کاراکتر جدید (خواندن از فایل)

ii. دیباگ گرافیک و بهتر کردن آن (مشکلاتی در نورپردازی مشاهده شده. هنوز مطمئن نیستم چه چیزی دقیقا مشکل را درست می کند و آیا اصلا به عنوان مشکل حساب می شود یا خیر!)

iii. بهتر کردن ساختار کلاس ها و بازسازی کلاس ها (refactoring) به عنوان مثال با اینکه کلاس stateMachine تا جایی که تست شده است به خوبی کار می کند ولی فکر میکنم شاید راه های بهتری برای اضافه کردن استیت و تغییر حالت ها وجود داشته باشد. در حال حاضر حالت ها با استفاده از یک رشته (نام حالت) ذخیره شدن

(یک مپ بین نام و آبجکت) و انتقال نیز با استفاده از این نام صورت می گیرد. مطمئن نیستم که این روش بهینه باشد. همچنین با اضافه کردن کلاس هایی می توان تابع main را خلوت تر کرد.

iv. اضافه کردن یک روش دیگر برای skinning به جای linear blend skinning (مثلا روش dual quaternion skinning)

v. اضافه کردن یک کنترلر. حال که ماشین حالت را پیاده سازی کردم فکر میکنم که امکان این را داشته باشم که بتوانم یک کاراکتر را به وسیله ی کیبورد حرکت بدهم (با استفاده از حالت های مختلف مانند بیکار بودن، راه رفتن، دویدن، ایستادن و ...) البته اینکار نیاز به نوشتن کد لازم برای دوربین نیز می شود. درواقع این قسمت ساخت دمو با استفاده از کلاس های موجود است.

vi. شروع به نوشتن گزارش پروژه

3. مراجع

[1] https://learnopengl.com/Advanced-OpenGL/Cubemaps

[2] https://www.youtube.com/watch?v=4e_ToPH-I5o&ab_channel=LadislavKavan

[3] https://www.cs.utah.edu/~ladislav/dq/index.html

[4] http://rodolphe-vaillant.fr/entry/29/dual-quaternions-skinning-tutorial-and-c-codes