Проект

По предметот Компјутерска Графика

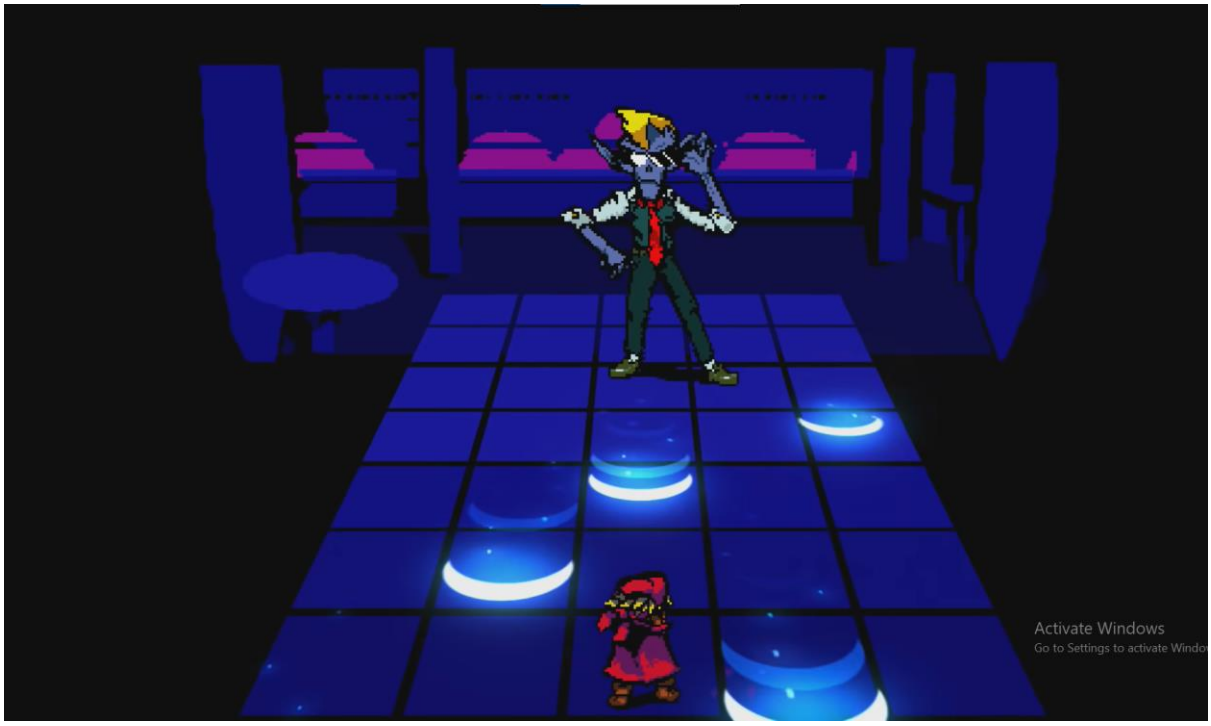
Тема на проект: Компјутерска Игра

Документ за кратко објаснување на чекорите и фазите при работењето на проектот како, резултатот и начинот на изработување

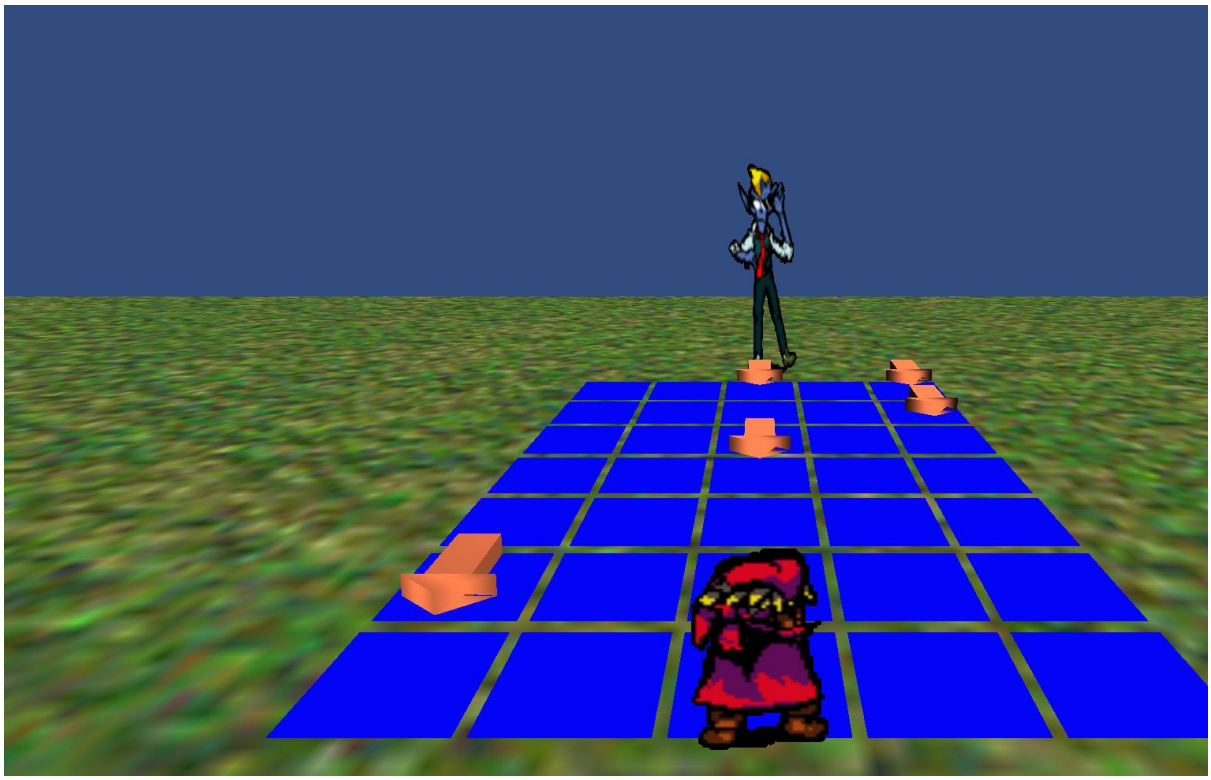
Од Филип Костов 191115, СИИС

RHYTHM GAME

Мојот проект по предметот Компјутерска Графика е 3D компјутерска игра налик на играта Everhood (https://everhoodgame.com/). Дизајнирањето на игрите како целина од интерактивна околина, интересна приазна и сцени, чисти анимации и детални текстури и графика, е комплексна постапка за која е потребно време и соработка и обично не е индивидуална, туку тимска работа. Денешните AAA “Tripple A” игри се обично финансирани од големи компании и се надградување на веќе постоечките. За анимациите и текстурите често се користат веќе готови енџини кои вклучуваат големо количество код и библиотеки за готови рендерирања. Преку овој проект јас си зададов еден вид на предизвик да направам интерактивна компјутерска игра само врз основа на OpenGL програмирање и без помош на “надворешни” апликации како Blender, Unity…

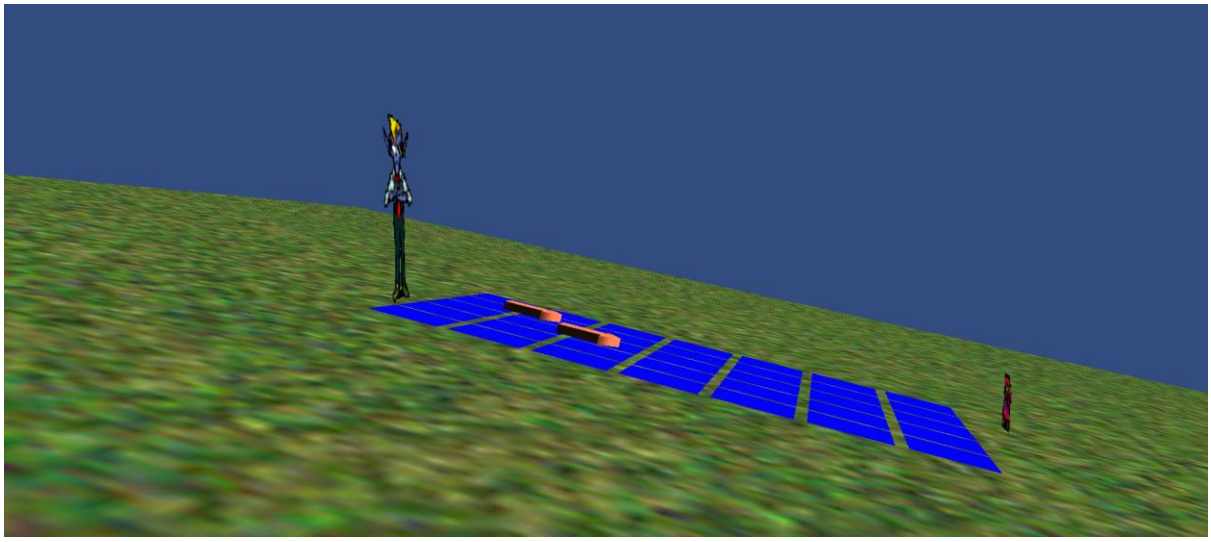


Слика од играта Everhood



Слика од финалниот продукт од мојот проект

Моделите на секој објект од мојата игра се прилично едноставнии и спој од најосновните објекти (триаголници, правоаголници). Моделот на подвижниот карактер, “Red” е едноставен правоаголник врз кој налепив транспарентни .png слики. Анимацијата за движење во лева и десна насока, како и стоечката позиција (IDLE) на “Red” и “Zigg” ги направив од исечените слики од главниот Spreadsheet за анимации кој го имам прикачено во датотеката. 3Д стрелките се комбинација од коцка (6 правоаголници) за телото на стрелката, како и 2 паралелни триаголници, 2 нормални правоаголници и еден правоаголник, хипотенузата на триаголниците. Играта ја направив на начин да изгледа природно 3Д преку некои чекори за илузија. Еве слика ја од друг агол



Текстурата на “Red” и “Zigg”скоро и не се гледат бидејќи се од обичен правоаголен модел. Единственото нешто во проектот што би можеле да речеме дека е 3Д се стрелките како и нивниот пат на движење и оддалеченост од карактерите.

РАБОТА ВО OPENGL

Како што спомнав, сакав проектот да го направам “од почеток”, па користев само OpenGL. Прво креирав 1920x1080 прозорец со сина позадина. Рамнината на која лежат моделите е едноставен правоаголник со “grass” текстура од трева и го ротирав моделот. Полето модели од квадрати со вектори за сина боја, и ги ротирав и поставив едно до друго. За моделите на карактерите Ред и Зиг наидов на проблем при прикачувањето на текстурите. Текстурите се .png транспарентни слики, но не ја покажуваа карактеристиката на транспарентност и се рендерираа со црна позадина. Решението на проблемот беше поставување на glEnable(GL_BLEND); наредбата и додадов APLHA (RGB-A) канал при декларирање на текстурите.Исто така fragment shader-от го изменував со

```
vec4 texColor = texture(texture1, TexCoord);
    if(texColor.a < 0.1)
    discard;
    FragColor = texColor;
```

за текстури без канал за транспарентност да не ги користат вредностите.

Исто така поставив

glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP_TO_EDGE);

glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP_TO_EDGE);

затоа што се појави проблем при промената на транспарентните текстури краевите од текстурата се рендерираа со црн border наместо целосно транспарентни.

Во овој момент ми се појавија проблеми со “прелапување” на текстурите, иако моделите се гледаше дека беа одделени, па поставив камера, движење околу 3Д просторот” со отклучување со копчето “F” за подобро да го следам изработувањето на проектот. Камерата ја поставив со KeyCallback функција во која беше вгнездена flag променлива преку која следев дали е притиснато или не копчето. После оваа постапка почнав да ги додавам текстурите за анимациите на карактерот кои ги изрежав на еднакви димензии преку Adobe PhotoShop, додадов и транслација на моделот на Ред и го ограничив движењето на 5 различни позиции преку променлива со која ја следев тековната позиција. Оваа беше сложена постапка бидејќи наидов на голем број проблеми со преклопување на мирувачките (IDLE) анимации со анимациите за движење па користев клуч (семафор) променливи flag_signalA, flag_signalD.

```
491  if ((flag_moveA==1) && (flag_signalD==1)) {
492      flag_signalA=0;
493      if(glfwGetTime()>0.05f && pomosna<5){
494
495
496          transl_pom-=0.12f;
497          pomosna++;
498          glfwSetTime( time: 0.0f);
499      }
500      if(pomosna==5)
501      {
502          flag_signalA=1;
503          pomosna=0;
504          br_lenta--;
505          flag_moveA=0;
506      }
507
508
509  }
```

Анимациите ги направив чисти и одделени за единица време со користење glfwGetTime(), го споредував со одредена единица време (пример анимациите на

движење лево кон десно (промената на текстурите) ги поставив на 5ms, додека на миравачката анимација на 20мс.

Со низа променливи ги следев тековните текстури и која требаше да се постави за следна.

За моделите на стрелките ги направив прилично едноставно, но ги рендерирав со ray tracing и додадов извор на светлина. Долната страна на објектите ми појавување проблем поради тоа што имав минимална грешка при пишењето на векторите. За оваа постапка ги додадов новите fragment и vertex shader-и бидејќи претходните ги користев само за текстури а не и за боја и извор на светлина. При рендерирањето на стрелките исто така настана проблем т.ш едниот од паралелните триаголници не се рендерирање коректно за стрелките на 2, 3 ,4 ,5 линија од полето. Пробав да ги сменам поставките за блендирање, исто го извадив и

glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP_TO_EDGE);,

пробав двата shader-и да ги комбинирам во еден бидејќи мислев дека причината е во тоа затоа што ги користев истовремено, но неможев да најдам решение. Начинот на рендерирање на стрелките го направив преку променлива која зима рандом број од 1 до 5 за бројот на редот (полето) во која би се придвижувала. Исто така ставив и експоненцијален инкрементор со кој можев да го контролирам бројот на стрелки (бројот на стрелки и времето на нивното движење, односно растојанието на транслација според y-оската)

```
1221        {
1222            if(pom4==0) {
1223                pom3=1;
1224                for (int i = 0; i < incrementor; i++) {
1225                    if (distance[i] >= 5.8f) {
1226                        picker[i] = random_pole[i] * 0.65f;
1227                        distance[i] = 0.0f;
1228                    }
1229                }
1230                pom3=0;
1231            }
1232            if(incrementor<=12) {
1233                if(incrementor>=4){
1234                    if(pom2>=6) {
1235                        incrementor++;
1236                        pom2 = 0;
1237                    }
1238                }
1239                else{
1240                    incrementor++;
```

Во овој дел од програмата на лош начин го правев менувањето на трансформациите, лошо ги именував променливите и на лоши места ги изцртував buffer-ите, па при рендерирањето на стрелките се рендерираше само една, и на таа единствена стрелка

се менуваше полето преку рандом генераторот, како и нејзината транслација, но не се рендерираа и други, па оваа променлива инкрементор главно ја користев баш за тоа, за на почетокот да се рендерираат повеќе стрелки, а не и за да се контролира нивното движење, но потоа добив идеја да го имплементирам и тоа.

Исто така ставив и health променлива како интерактивен додаток

И доколку стрелка која би стигнала до последниот вектор за транслација истовремено се пречека и со тековната положба, линија на моделот на Ред, да се намали неговиот живот и доколку стигне 0, прекинува да се извршува програмата. И во овој чекор се појави проблем па морав да користам пак семафори затоа што променливата се намалуваше од два дела од програмата, доколку се помрдне карактерот врз стрелка, и доколку е во мирување и стрелката пристигне до него, па морав да ја менувам променливата само еднаш во дадено време.