Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное

Учреждение высшего образования

«ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Физико-технический институт

Дисциплина «Технология программирования»

Отчет по командному проекту:

**«Написание 2D-игры на языке С++ с использованием библиотеки SFML»**

Выполнил:

Студент 21317 группы

Савин Иван Алексеевич

Проверил:

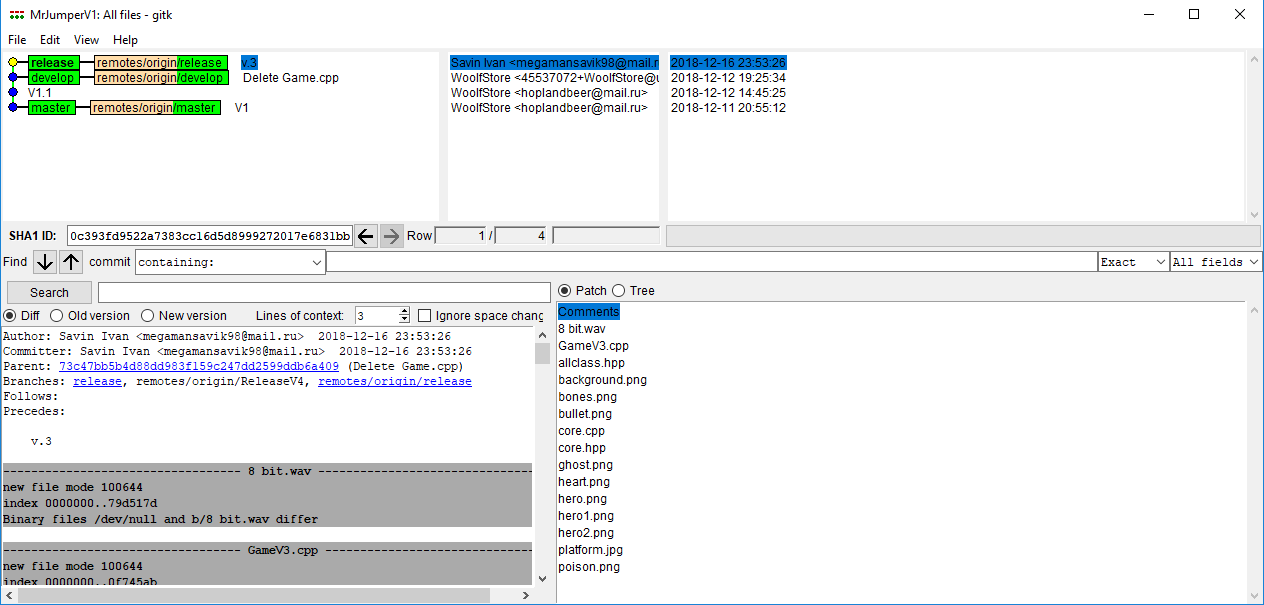
к. ф.-м. н.

Бульба Артем Владимирович

Петрозаводск 2018г.

Пояснение сделанных коммитов:

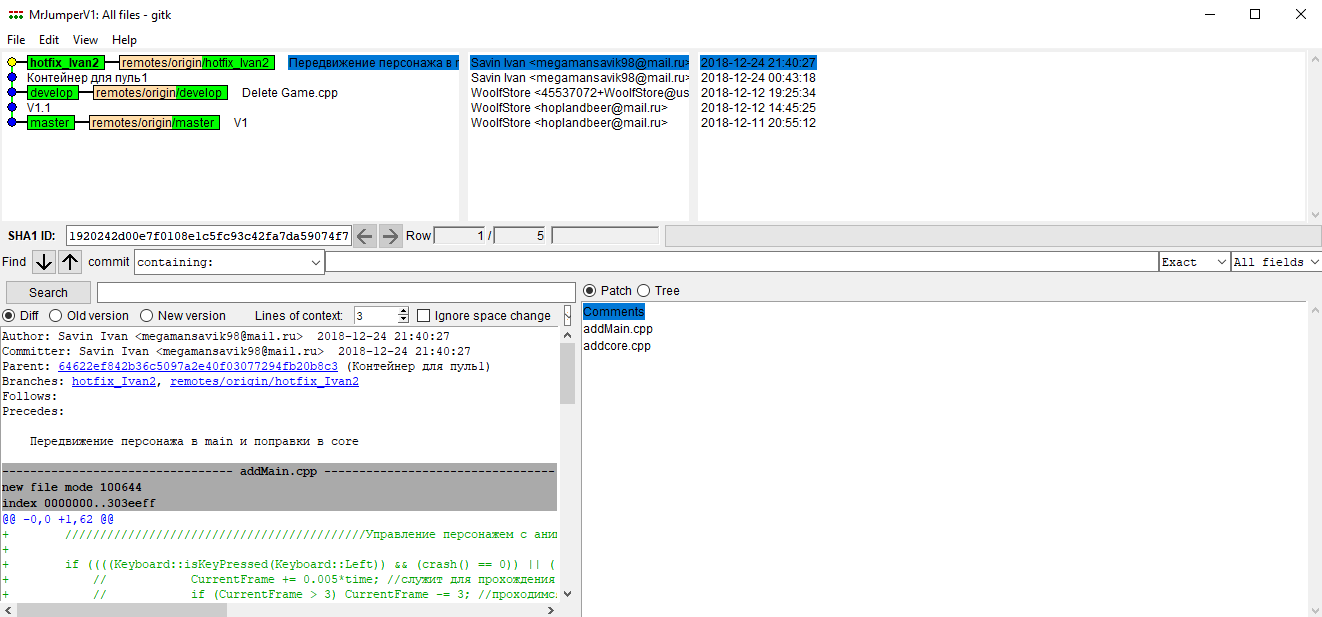
1)



Commit: 0c393fd9522a7383cc16d5d8999272017e6831bb

Добавили классы и музыку. Таким образом получили 3 версию кода.

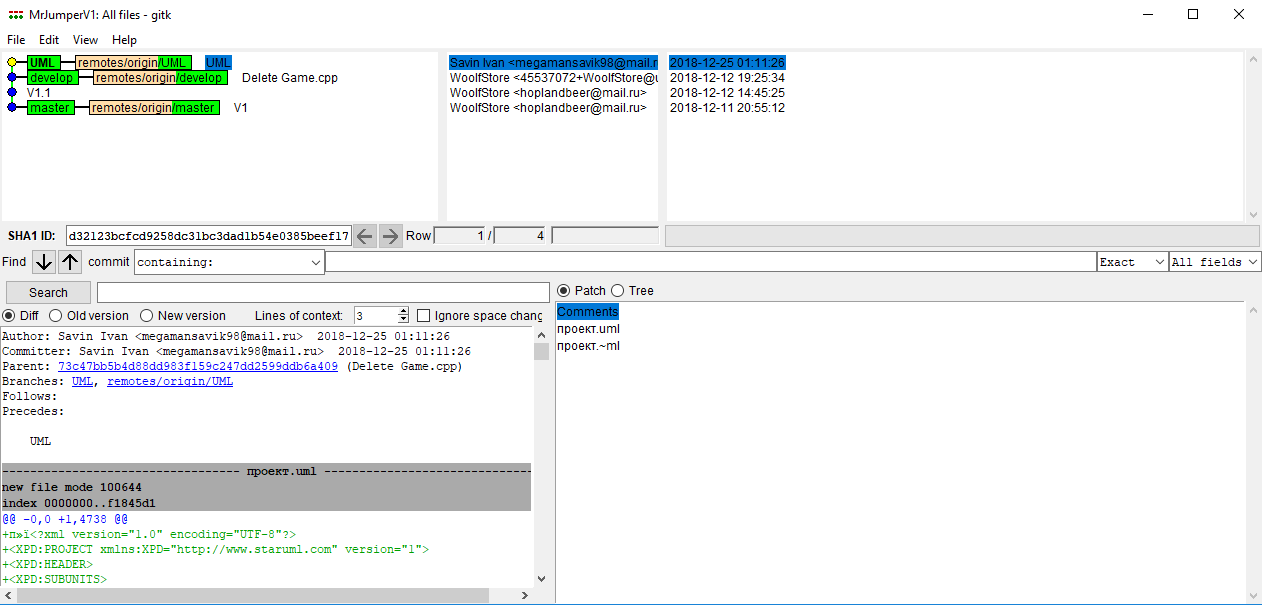
2)



Commit: 1920242d00e7f0108e1c5fc93c42fa7da59074f7

Попытка создания контейнера для пули. И поправки по передвижении персонажа.

3)



Commit: d32123bcfcd9258dc31bc3dad1b54e0385beef17

Была сделана диаграмма вариантов использования и диаграммы классов.

**GameBegins.cpp:**

**1)**  
///////////////////////////////////////////Управление персонажем с анимацией////////////////////////////////////////////////////////////////////////

if ((((Keyboard::isKeyPressed(Keyboard::Left)) && (crash() == 0)) || ((Keyboard::isKeyPressed(Keyboard::Left)) && (enemyLive))) && (stop() != 2)) { //если нажата клавиша стрелка влево

CurrentFrame += 0.005\*time; //служит для прохождения по "кадрам". переменная доходит до трех суммируя произведение времени и скорости. изменив 0.005 можно изменить скорость анимации

if (CurrentFrame > 6) CurrentFrame -= 6; //проходимся по кадрам с первого по третий включительно. если пришли к третьему кадру - откидываемся назад.

herosprite.setTextureRect(IntRect(64 \* int(CurrentFrame), 256, 64, 128)); //проходимся по координатам Х. получается 96,96\*2,96\*3 и опять 96

herosprite.move(-0.2\*time, 0);

changeHeroPosition(herosprite.getPosition().x, herosprite.getPosition().y);

dirhero = false;

}

if ((((Keyboard::isKeyPressed(Keyboard::Right)) && (crash() == 0)) || ((Keyboard::isKeyPressed(Keyboard::Right)) && (enemyLive))) && (stop() !=2)) {

CurrentFrame += 0.005\*time; //служит для прохождения по "кадрам". переменная доходит до трех суммируя произведение времени и скорости. изменив 0.005 можно изменить скорость анимации

if (CurrentFrame > 6) CurrentFrame -= 6; //проходимся по кадрам с первого по третий включительно. если пришли к третьему кадру - откидываемся назад.

herosprite.setTextureRect(IntRect(64 \* int(CurrentFrame), 384, 64, 128)); //

herosprite.move(0.2\*time, 0);//происходит само движение персонажа вправо

changeHeroPosition(herosprite.getPosition().x, herosprite.getPosition().y);

dirhero = true;

}

if ((Keyboard::isKeyPressed(Keyboard::Space)) && (stop() != 3) && !endOfJump() && HeroCanJump == 1) {

herosprite.move(0, -0.8\*time);

changeHeroPosition(herosprite.getPosition().x, herosprite.getPosition().y);

herosprite.setTextureRect(IntRect(0, 0, 64, 128));

}

if (endOfJump() && on\_ground == 0)

HeroCanJump = 0;

if (on\_ground !=0)

HeroCanJump = 1;

// ПУЛЯ

if ((Keyboard::isKeyPressed(Keyboard::A) && youCanShoot)){

youCanShoot = false;

if (dirhero)

pow.dirBullet = 1;

else

pow.dirBullet = -1;

generatebullet = true;

}

if (generatebullet){

pow.bulletLive = true;

if (pow.dirBullet == 1)

pow.bulletsprite.setScale(1, 1);

else

pow.bulletsprite.setScale(-1, 1);

pow.bulletsprite.move(0.8 \* pow.dirBullet \* time, 0);

changeHeroBulletPostion(pow.bulletsprite.getPosition().x, pow.bulletsprite.getPosition().y);

if (pow.bulletsprite.getPosition().x > 1280 || pow.bulletsprite.getPosition().x < 0){

pow.bulletLive = false;

}

}

if (!pow.bulletLive){

pow.bulletsprite.setPosition(herosprite.getPosition().x + 64, herosprite.getPosition().y + 64);

changeHeroBulletPostion(pow.bulletsprite.getPosition().x, pow.bulletsprite.getPosition().y);

generatebullet = false;

youCanShoot = true;

}

**2)**

// ДВИЖЕНИЕ ПЕРСОНАЖА С ПЛАТФОРМОЙ

if (on\_ground == 2){

if (!stopPlatform2){

if (platformMove){

herosprite.move(0.2\*time, 0);

changeHeroPosition(herosprite.getPosition().x, herosprite.getPosition().y);

} else {

herosprite.move(-0.2\*time, 0);

changeHeroPosition(herosprite.getPosition().x, herosprite.getPosition().y);

}

}

}

**3)**

// ПЛАТФОРМА ТОЛКАЕТ ПЕРСА

if (crash()==1 && on\_ground==1 && !platformCantMove){

herosprite.move(0.4 \* time, 0);

changeHeroPosition(herosprite.getPosition().x, herosprite.getPosition().y);

}

else{

if (crash()==2 && on\_ground==1 && !platformCantMove)

herosprite.move(-0.4 \* time, 0);

changeHeroPosition(herosprite.getPosition().x, herosprite.getPosition().y);

}

window.clear();

window.draw(backgroundsprite);

window.draw(poisonsprite);

window.draw(herosprite);

window.draw(p1.platformsprite);

window.draw(p2.platformsprite);

**Core.cpp:**

**1)**

void changeHeroPosition(int x, int y){

mainHero.x = x;

mainHero.y = convertYCoordinate(y);

**2)**

int onGround(){// возвращает 1 если стоит на колонне, 2 если стоит на двигующейся платформе, 0 если не стоит

for (int i = 0; i < quantityOfPlatforms; i++){

if ((mainHero.x >= arrayOfPlatforms[i].x && mainHero.x <= arrayOfPlatforms[i].x + 256) && (mainHero.y - 128 <= arrayOfPlatforms[i].y))

return 1;

}

if ((mainHero.x >= movingPlatform.x && mainHero.x <= movingPlatform.x + 256) && (mainHero.y - 128 <= movingPlatform.y)){

return 2;

}

return 0;

}

**3)**

int crash() { // столкновение с двигающийся платформой

if ((mainHero.y - 128 >= platform.y) || (mainHero.y <= platform.y - 64)){

if (mainHero.x + 64 >= movingPlatform.x + 256+64 && mainHero.x + 64 <= movingPlatform.x + 256+64+5)

return 1;

else

if (mainHero.x >= movingPlatform.x - 69 && mainHero.x <= movingPlatform.x - 64 )

return 2;

}

return 0;

}

**4)**

int stop(){ // 1 - right 2 - left, 3 - top 0 - ok

if (mainHero.x <= 0) return 2;

if (mainHero.x + 64 >= 1280) return 1;

if (mainHero.y >= 720) return 3;

for (int i = 0; i < quantityOfPlatforms; i++){

if ((arrayOfPlatforms[i].y <= mainHero.y) && (arrayOfPlatforms[i].y - 64 >= mainHero.y - 128)){

if (arrayOfPlatforms[i].x <= mainHero.x + 64){

return 1;

}

if (arrayOfPlatforms[i].x + 256 <= mainHero.x){

return 2;

}

}

}

return 0;

}