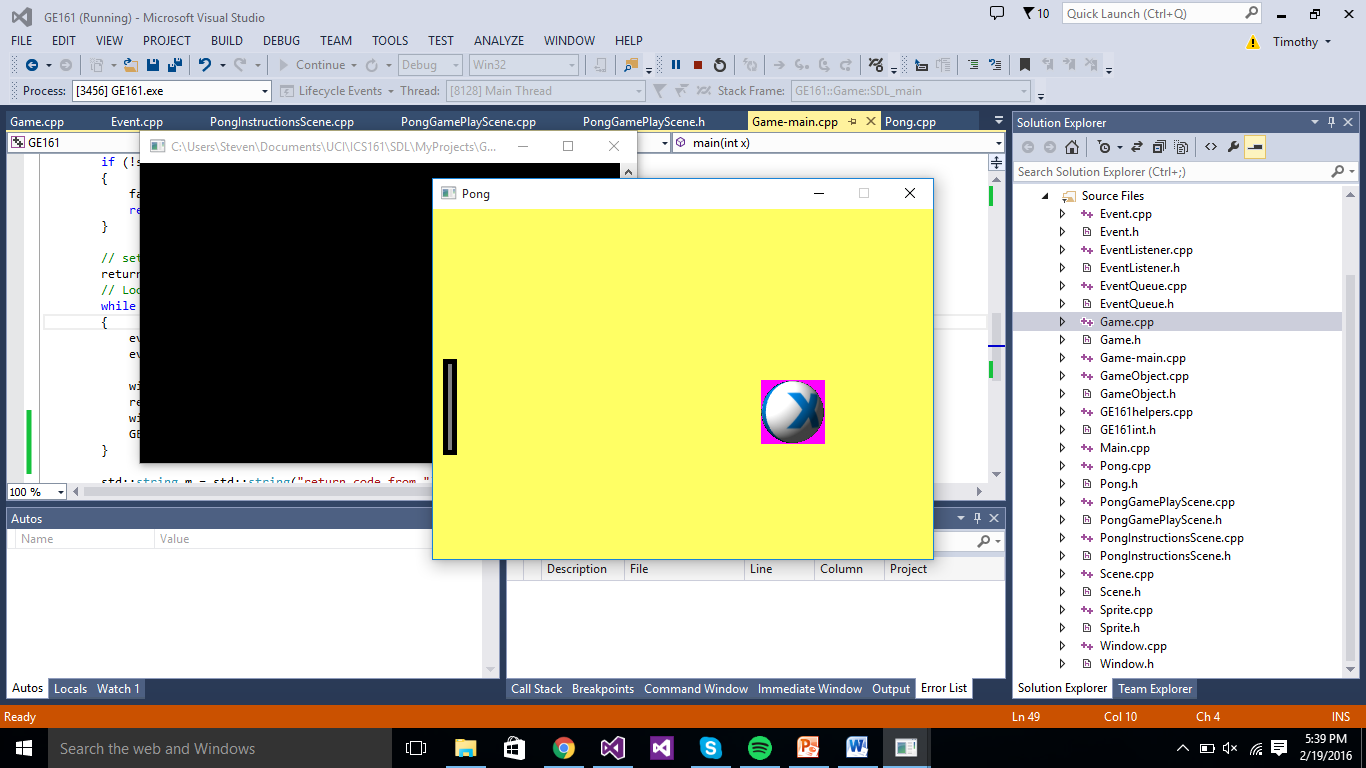
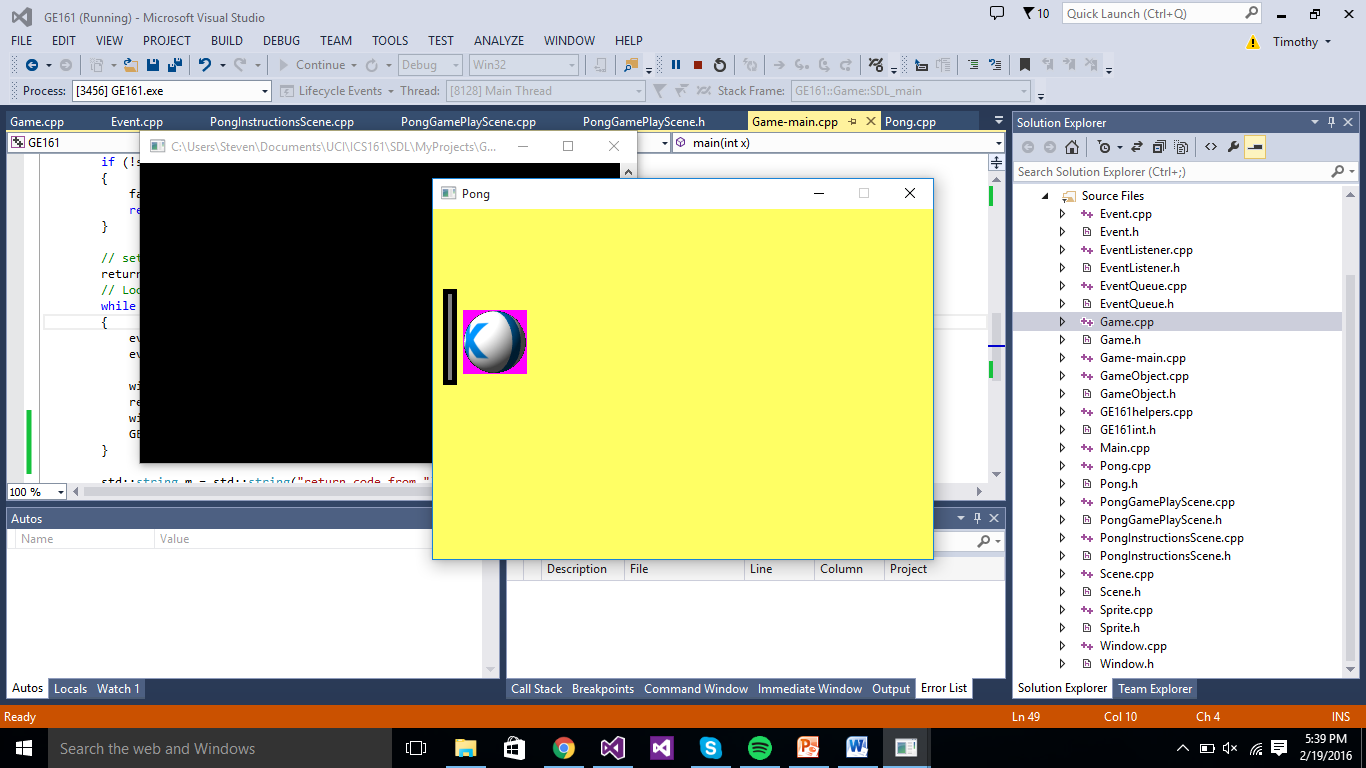
Timothy Pranoto  
38964311  
ICS161 HW6

Part2





In the PongPlayScene.cpp

Add #include "Event.h"

In the bool PongGamePlayScene::setup()

exitRequested = false;

paddleW = 14;

paddleH = 96;

GE161::Sprite\* paddleSpriteSheet = new GE161::Sprite(paddleW, paddleH);

int p=paddleSpriteSheet->makeFrame(GE161::Game::basePath() + "paddle.png", 0, 0);

paddleSpriteSheet->addFrameToSequence("paddle", p);

paddle->attachSprite(paddleSpriteSheet);

GE161::Game::registerAsListener(GE161::Event::KEY\_DOWN, this);

In the int PongGamePlayScene::draw()

paddle->draw("paddle");

In the if (ball->getX() < 0)

Add bounceOut\_++;

//this if case is for the collision between paddle and ball

if (ball->getX() <= paddle->getX() + paddleW){

if (ball->getY()<=paddle->getY() + paddleH&&

ball->getY()+frameHeight\_>=paddle->getY()){

x\_delta = rand() % 3 + 1;

}

}

//delete the bounce 10 times to this if case

if (exitRequested||bounceOut\_>=2)

{

return -1;

}

else

{

return GE161::Game::CONTINUE\_SCENE;

}

Add this function

void PongGamePlayScene::onEvent(GE161::Event & e){

if (e.type == GE161::Event::KEY\_DOWN)

{

if (e.key == "X")

exitRequested = true;

else if (e.key == "DOWN"){

if (paddle->getY() + paddleH >= theGame->window()->clientHeight())

paddle->setPos(paddle->getX(),

theGame->window()->clientHeight()- paddleH);

else

paddle->moveY(5);

}

else if (e.key == "UP"){

if (paddle->getY() <= 0)

paddle->setPos(paddle->getX(), 0);

else

paddle->moveY(-5);

}

}

}

In the PongGamePlayScene::~PongGamePlayScene() add

delete paddle;

Task 1

In the Game-main.cpp

Initialize const int FRAMES\_PER\_SECOND = 30;

Then add the delay in the while (returnCode == CONTINUE\_SCENE)

Put the delay in the end of the while loop

GE161::Game::delay((1000 / FRAMES\_PER\_SECOND) - GE161::Game::getTicks());

Before the while (returnCode == CONTINUE\_SCENE)

lastframetime = GE161::Game::getTicks();

inside while (returnCode == CONTINUE\_SCENE)

++fps;

int tick = GE161::Game::getTicks();

if (( tick - lastframetime) >=1000){

avef[i] = fps;

++i;

std::cout << "SDA";

fps = 0;

}

if (i == 5){

int sum = 0;

for (int a = 0; a < 5; ++a)

sum += avef[a];

i = 0;

std::string os = std::to\_string(sum / 5);

debugOut(os);

}

lastframetime = tick;

Task 2

Change the color default to white in void GE161::Window::clearBackground()

In the line

int rc = SDL\_SetRenderDrawColor(sdl\_renderer, 255, 255, 255, SDL\_ALPHA\_OPAQUE);

And add function to change color

void GE161::Window::changeColor(int r, int b, int g){

int rc = SDL\_SetRenderDrawColor(sdl\_renderer, r, b, g, SDL\_ALPHA\_OPAQUE);

if (rc != 0)

{

fatalSDLError("In Window::clearBackground(), SDL\_SetRenderDrawColor returned non-zero: ", SDL\_GetError());

return;

}

rc = SDL\_RenderClear(sdl\_renderer);

if (rc != 0)

{

fatalSDLError("In Window::clearBackground(), SDL\_RenderClear returned non-zero: ", SDL\_GetError());

return;

}

}

Task 3

In the Event.cpp

Add this to the switch statement

case SDLK\_BACKSPACE: key = "BACKSPACE"; break;

In the PongGamePlayScene.cpp

In void PongGamePlayScene::onEvent(GE161::Event & e) function add an extra if case for the backspace button

else if (e.key == "BACKSPACE"){

ball->setPos(theGame->window()->clientWidth() / 2-(frameWidth\_/2),

theGame->window()->clientHeight() / 2-(frameHeight\_/2));

}