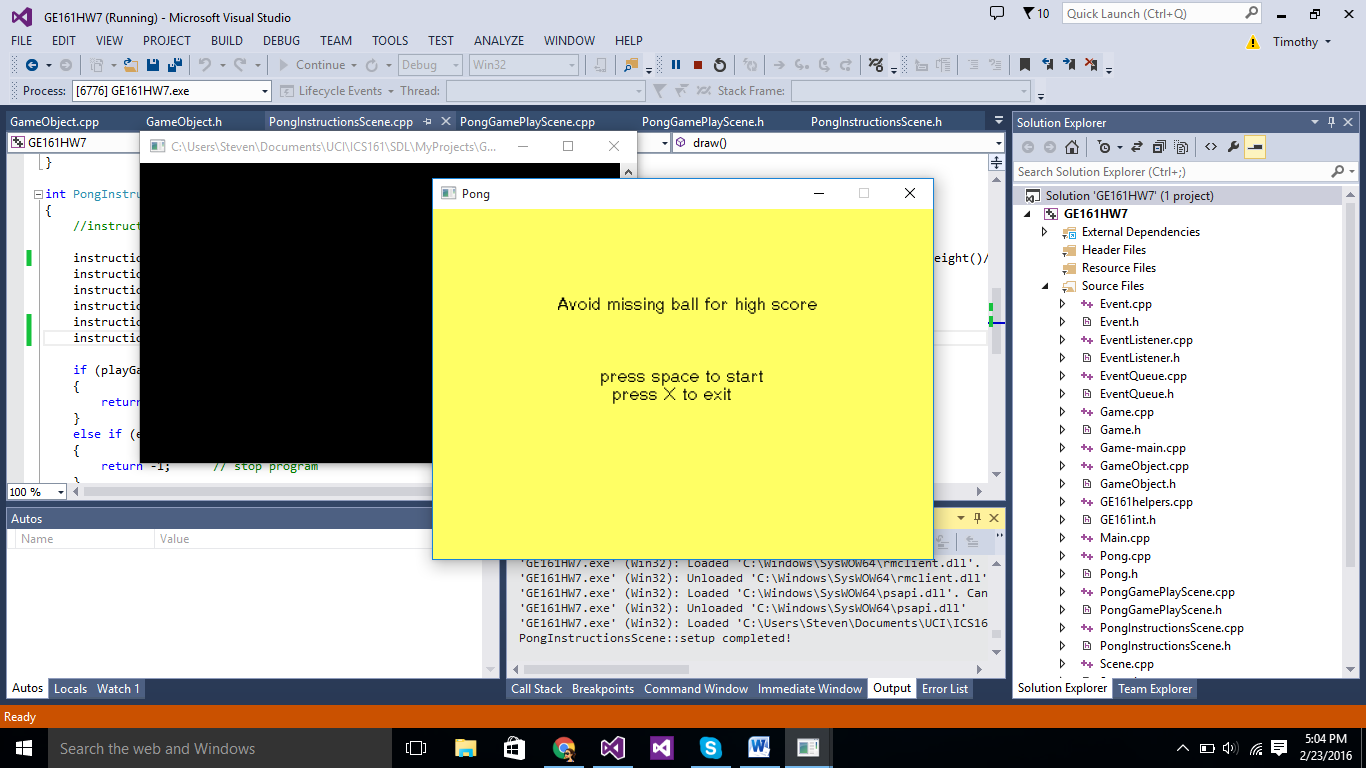
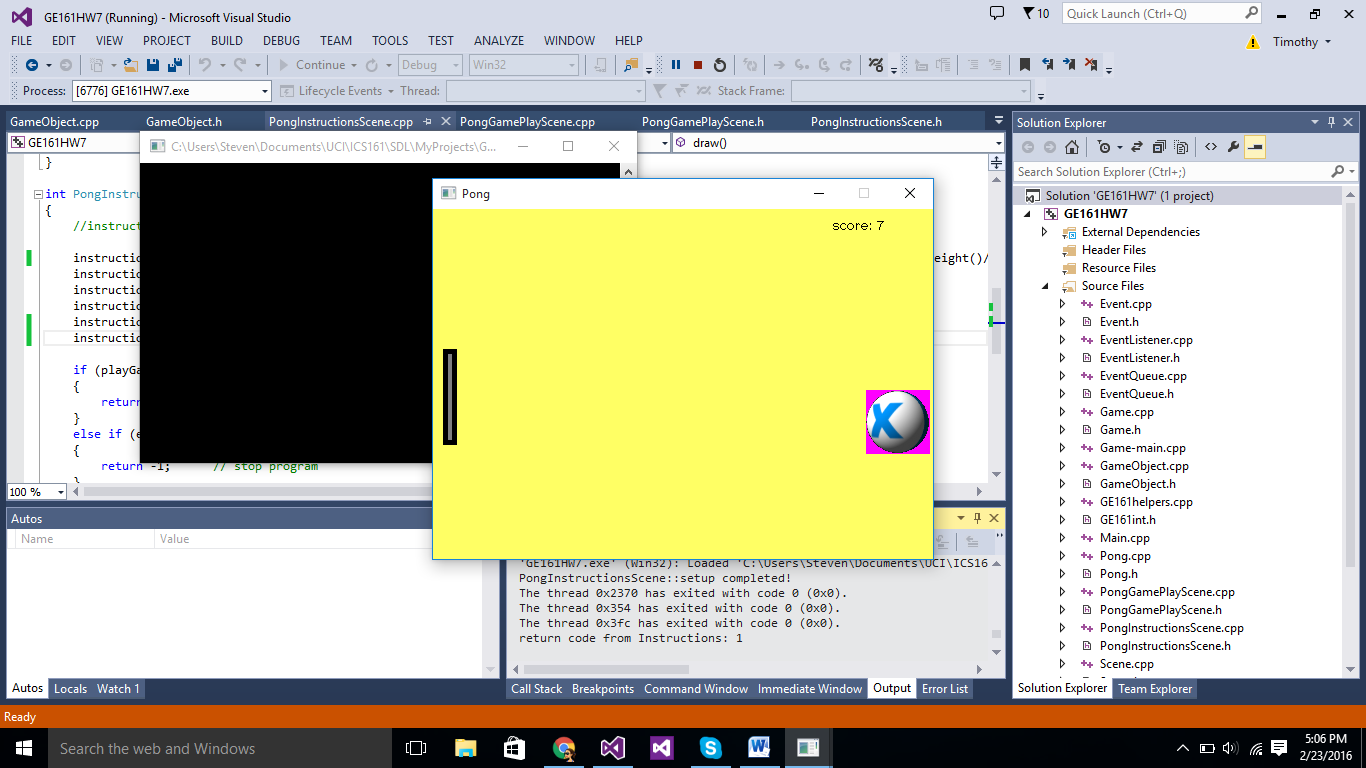
Timothy Pranoto  
38964311  
ICS161 HW7





Design issue: from my design of the score feature, sometimes when the ball hit the right side of the window, it counts the hit twice. This is probably cause by when the ball goes beyond the windows so the code tried to bounce the ball back but it still beyond the window and count it twice because it collide with another right side window

I added two extra files, Texty.h and Texty.cpp

Texty.h:

#pragma once

#include <iostream>

#include <sstream>

#include <SDL.h>

#include "SDL\_ttf.h"

#include "GE161int.h"

class Texty

{

friend class GameObject;

friend class Sprite;

public:

Texty( std::string fontName, int fontSize, bool visible);

~Texty();

enum writeOption\_t { CONTINUE, NEXT\_LINE };

void write(std::string text, writeOption\_t option);

void write(std::string text, int x, int y);

void yikes(std::string message);

private:

SDL\_Renderer\* renderer;

std::string fontName;

bool visible;

bool error;

int fontSize;

SDL\_Color color;

TTF\_Font\* font;

int nextX, nextY;

int lineStartX;

SDL\_Texture \* texture;

};

Texty.cpp:

#include "Texty.h"

#include "Game.h"

Texty::Texty( std::string fontName, int fontSize, bool visible) :

renderer(GE161::Game::theGame->getRenderer()),

fontName(fontName),

visible(visible),

error(false)

{

if (!TTF\_WasInit())

{

int s = TTF\_Init();

if (s)

{

yikes("Failed on TTF\_Init()");

return;

}

}

// Open the font

font = TTF\_OpenFont(fontName.c\_str(), fontSize);

if (font == nullptr)

{

yikes("TTF\_OpenFont can't open " + fontName);

return;

}

color = { 0, 0, 0, 255 };

nextX = nextY = 10;

lineStartX = nextX;

}

Texty::~Texty()

{

if (error)

return;

TTF\_CloseFont(font);

}

void Texty::write(std::string text, writeOption\_t option)

{

if (error)

return;

if (text.length() == 0)

{

if (option == NEXT\_LINE)

{

nextX = lineStartX;

nextY += TTF\_FontHeight(font) + 1;

}

// nop if CONTINUE

return;

}

//We need to first render to a surface as that's what TTF\_RenderText

//returns, then load that surface into a texture

SDL\_Surface \*surface = TTF\_RenderText\_Blended(font, text.c\_str(), color);

if (surface == nullptr)

{

TTF\_CloseFont(font);

yikes("TTF\_RenderText error");

return;

}

texture = SDL\_CreateTextureFromSurface(renderer, surface);

if (texture == nullptr){

yikes(std::string("SDL\_CreateTextureFromSurface errror ") + SDL\_GetError());

return;

}

int w, h;

SDL\_QueryTexture(texture, NULL, NULL, &w, &h);

SDL\_Rect dest;

if (option == CONTINUE)

{

dest.x = nextX;

dest.y = nextY;

}

else // option == NEXT\_LINE

{

dest.x = lineStartX;

dest.y = nextY + h;

}

dest.w = w;

dest.h = h;

SDL\_RenderCopy(renderer, texture, NULL, &dest);

nextX += w;

nextY = dest.y;

SDL\_FreeSurface(surface);

SDL\_DestroyTexture(texture);

}

void Texty::write(std::string text, int x, int y)

{

nextX = x;

nextY = y;

lineStartX = x;

write(text, CONTINUE);

}

void Texty::yikes(std::string message)

{

std::cout << "\*\*Texty error " << message << " SDL\_GetError: " << SDL\_GetError() << std::endl;

std::ostringstream errMsg;

errMsg << "\*\*Texty error " << message << " SDL\_GetError: " << SDL\_GetError() << std::endl;

OutputDebugString(errMsg.str().c\_str());

error = true;

}

I also added #include "Texty.h" in the GameObject.h preprocessor

Added friend class Texty; in the Game.h just below the friend class Sprite;

For part1, I need to change the png picture in the instructions to all text from texty. To do that, I added Texty\* instructionText; in the PongInstructionsScene.h as a private variable. Then, in bool PongInstructionsScene::setup() definition in the PongInstructionsScene.cpp, I changed this line of code:

GE161::Sprite\* instructionsImage = new GE161::Sprite(450, 300);

frameIndex = instructionsImage->makeFrame(GE161::Game::basePath() + "Instructions.png", 0, 0);

instructions->attachSprite(instructionsImage);

to:

instructionText = new Texty(GE161::Game::basePath() + "sample.ttf", 20, true);

in the int PongInstructionsScene::draw() definition, I changed:

instructions->draw(frameIndex);

to:

instructionText->write("Avoid missing ball for high score", theGame->window()->clientWidth()/ 4, theGame->window()->clientHeight()/ 4);

instructionText->write(" ", Texty::NEXT\_LINE);

instructionText->write(" ", Texty::NEXT\_LINE);

instructionText->write(" ", Texty::NEXT\_LINE);

instructionText->write(" press space to start", Texty::NEXT\_LINE);

instructionText->write(" press X to exit", Texty::NEXT\_LINE);

to draw all the text.

For the next part1, I need to be able to track the score when the game is being played. To do that, I added:

int score=0;

Texty\* instructionText;

As private variables in PongGamePlayScene.h

Then, in PongGamePlayScene.cpp, in the bool PongGamePlayScene::setup() definition, I added:

instructionText = new Texty(GE161::Game::basePath() + "sample.ttf", 16, true);

Next, in the int PongGamePlayScene::draw() definition, I added:

instructionText->write("score: ", 400, 10);

as the first line

and then I added a line of code in if (ball->getX() + frameWidth\_ >= theGame->window()->clientWidth())

++score;

to count the wherever the ball hit the right side of the window.

I also added:

instructionText->write(std::to\_string(score), Texty::CONTINUE);

to write and keep updating the score.