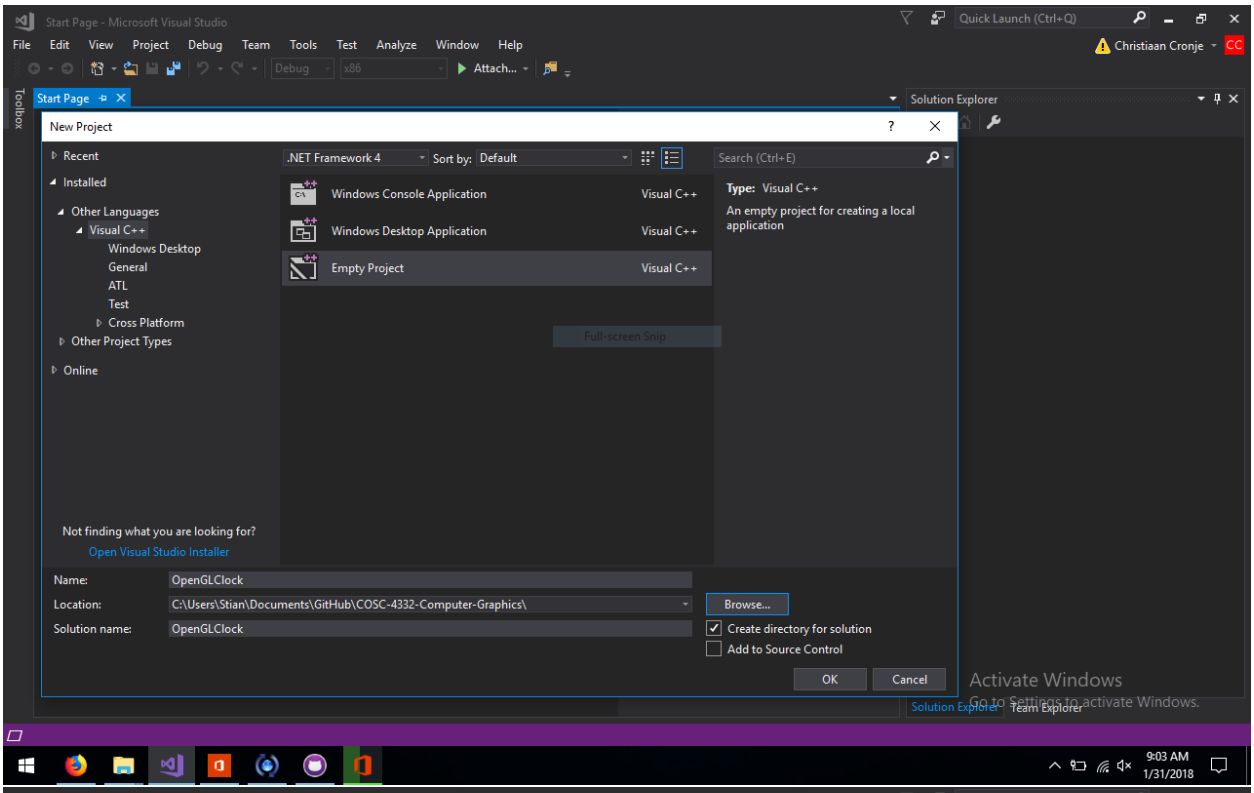
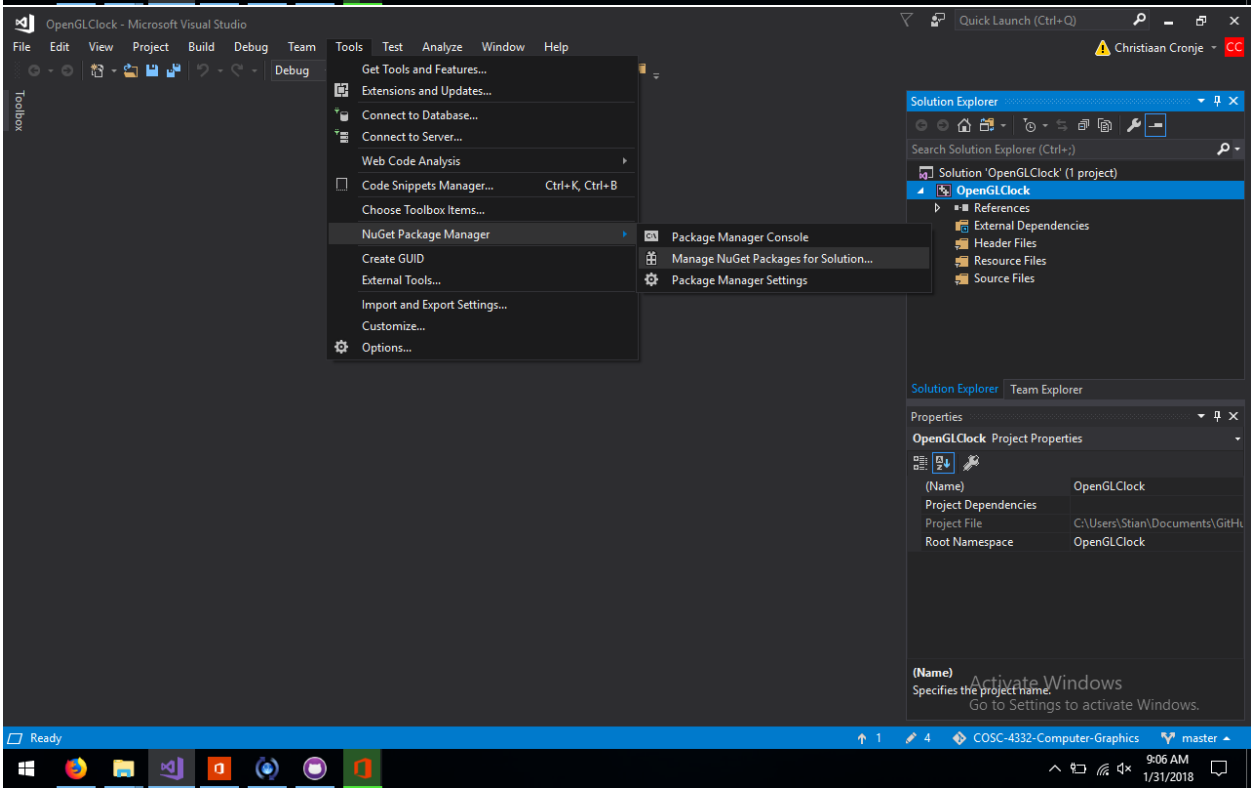
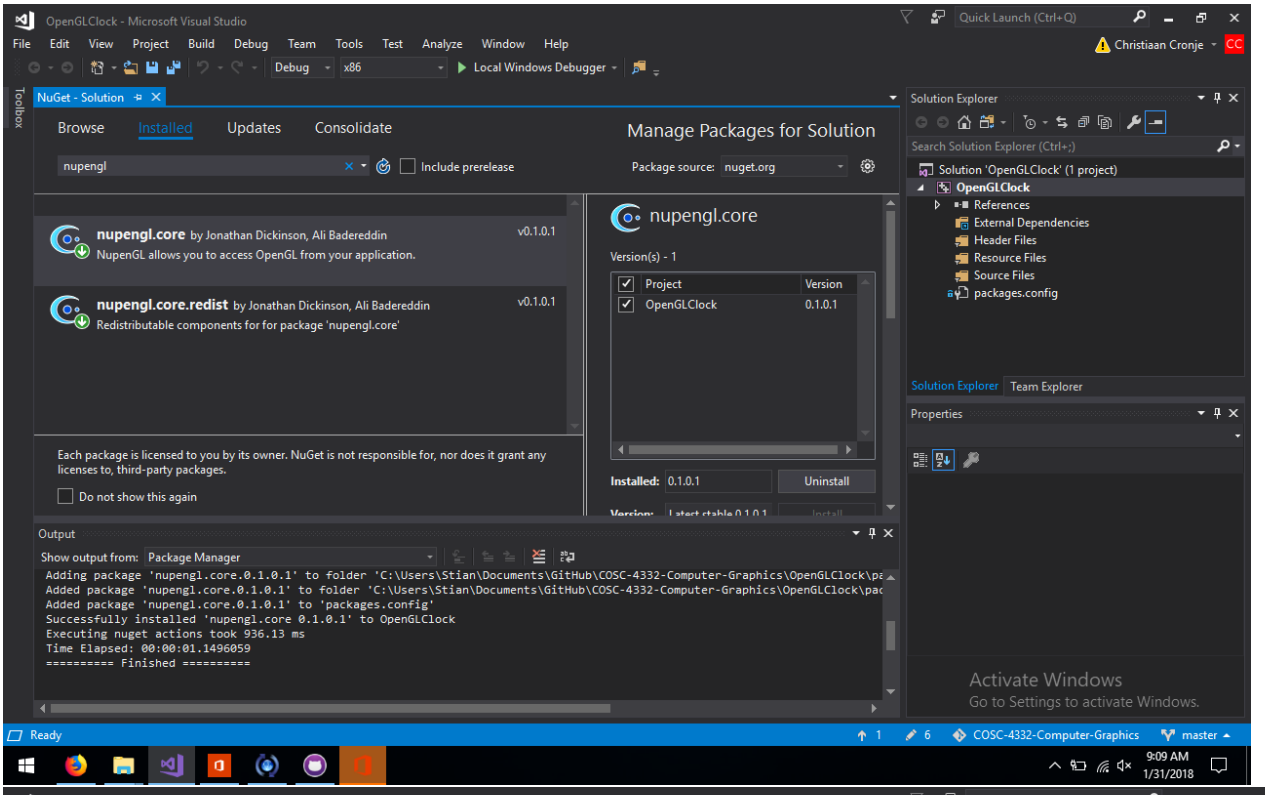
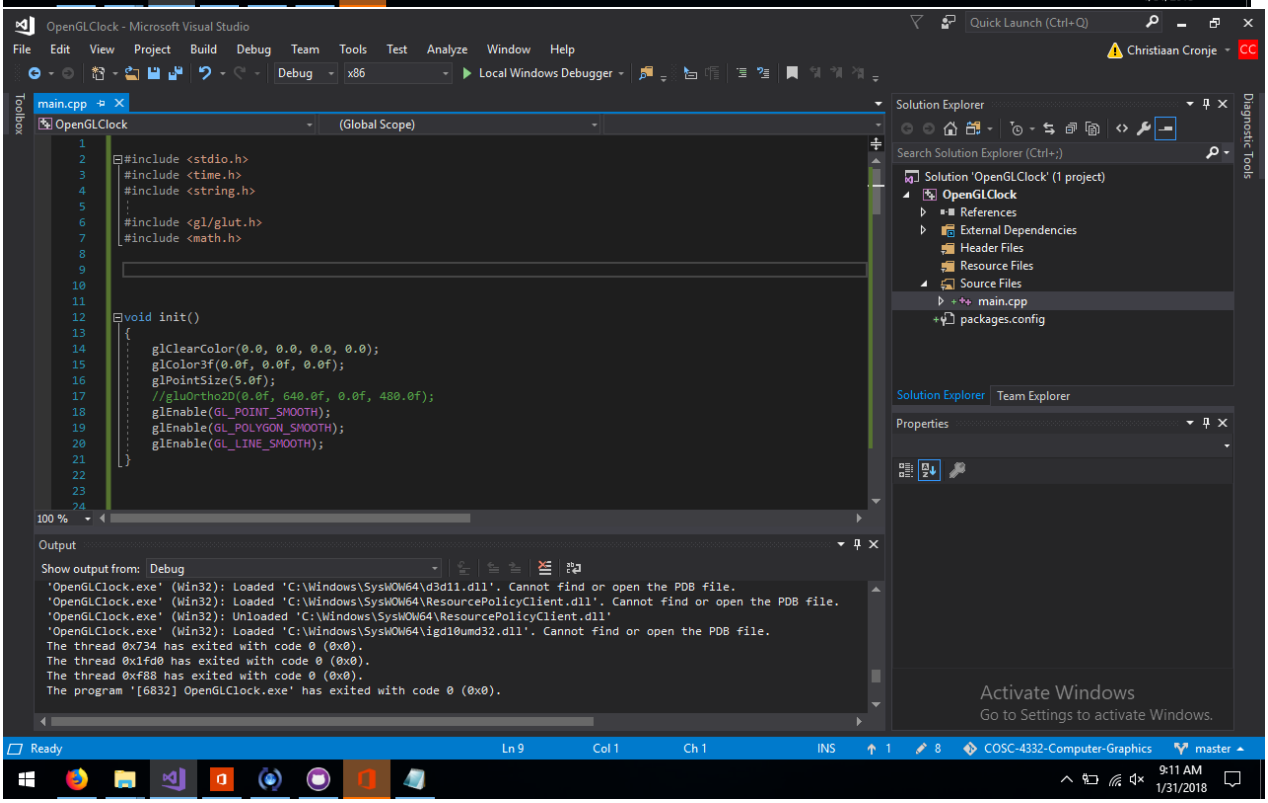
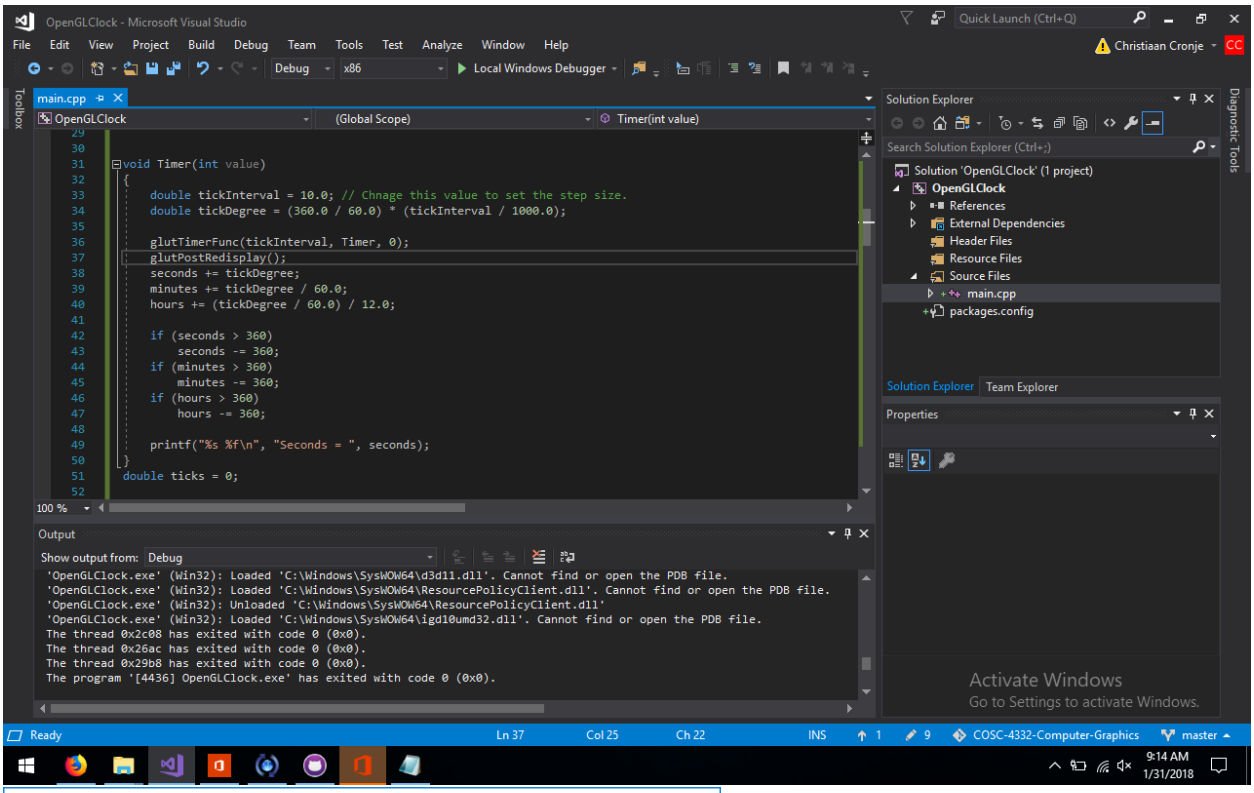
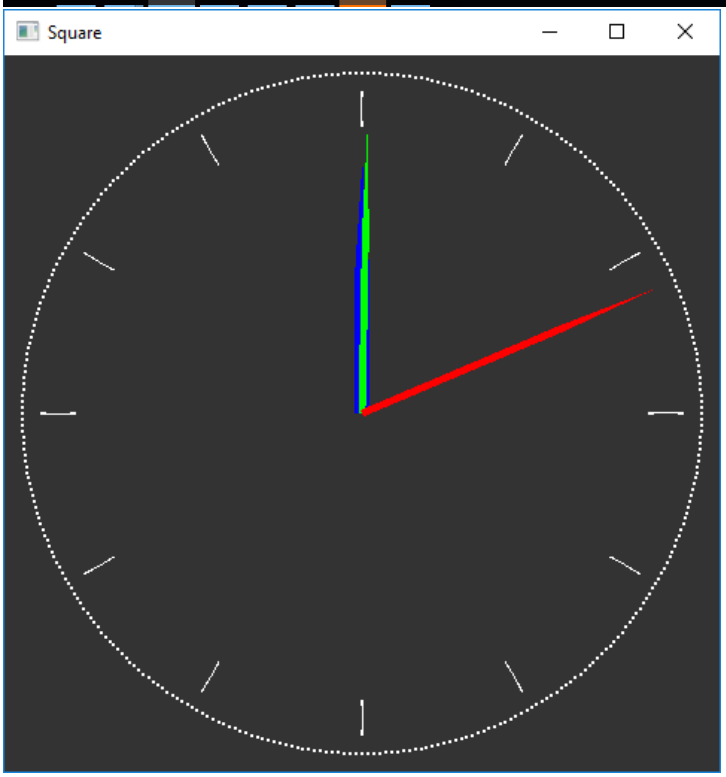

Start Page - Microsoft Visual Studio
File Edit View Project Debug Team Tools Test Analyze Window Help
Quick Launch (Ctrl+Q)
Christiaan Cronje
Start Page
Solution Explorer
New Project
Recent
Installed
Other Languages
Visual C++
Windows Desktop
General
ATL
Test
Cross Platform
Other Project Types
Online
.NET Framework 4
Sort by: Default
Windows Console Application Visual C++
Windows Desktop Application Visual C++
Empty Project Visual C++
Type: Visual C++
An empty project for creating a local application
Not finding what you are looking for?
Open Visual Studio Installer
Name: OpenGLClock
Location: C:\Users\Stian\Documents\GitHub\COSC-4332-Computer-Graphics\
Solution name: OpenGLClock
Browse...
Create directory for solution
Add to Source Control
OK
Cancel
Activate Windows
Go to Settings to activate Windows.
Solution Explorer Team Explorer
9:03 AM
1/31/2018


OpenGLClock - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
Get Tools and Features...
Extensions and Updates...
Connect to Database...
Connect to Server...
Web Code Analysis
Code Snippets Manager... Ctrl+K, Ctrl+B
Choose Toolbox Items...
NuGet Package Manager
Create GUID
External Tools...
Import and Export Settings...
Customize...
Options...
Package Manager Console
Manage NuGet Packages for Solution...
Package Manager Settings
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'OpenGLClock' (1 project)
OpenGLClock
References
External Dependencies
Header Files
Resource Files
Source Files
Properties
OpenGLClock Project Properties
(Name) OpenGLClock
Project Dependencies
Project File
Root Namespace OpenGLClock
(Name)
Specifies the project name.
Activate Windows
Go to Settings to activate Windows.
Ready
COSC-4332-Computer-Graphics
master
9:06 AM
1/31/2018


OpenGLClock - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Christiaan Cronje
Debug x86 Local Windows Debugger
NuGet - Solution
Browse Installed Updates Consolidate
Manage Packages for Solution
nupengl
Include prerelease
Package source: nuget.org
nupengl.core by Jonathan Dickinson, Ali Badereddin v0.1.0.1
NupenGL allows you to access OpenGL from your application.
nupengl.core.redist by Jonathan Dickinson, Ali Badereddin v0.1.0.1
Redistributable components for for package 'nupengl.core'
Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.
Do not show this again
nupengl.core
Version(s) - 1
Project Version
OpenGLClock 0.1.0.1
Installed: 0.1.0.1
Uninstall
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'OpenGLClock' (1 project)
OpenGLClock
References
External Dependencies
Header Files
Resource Files
Source Files
packages.config
Solution Explorer Team Explorer
Properties
Output
Show output from: Package Manager
Adding package 'nupengl.core.0.1.0.1' to folder 'C:\Users\Stian\Documents\GitHub\COSC-4332-Computer-Graphics\OpenGLClock\pa
Added package 'nupengl.core.0.1.0.1' to folder 'C:\Users\Stian\Documents\GitHub\COSC-4332-Computer-Graphics\OpenGLClock\pac
Added package 'nupengl.core.0.1.0.1' to 'packages.config'
Successfully installed 'nupengl.core 0.1.0.1' to OpenGLClock
Executing nuget actions took 936.13 ms
Time Elapsed: 00:00:01.1496059
========== Finished ==========
Activate Windows
Go to Settings to activate Windows.
Ready
COSC-4332-Computer-Graphics master
9:09 AM
1/31/2018


OpenGLClock - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Christiaan Cronje
Debug x86 Local Windows Debugger
main.cpp
OpenGLClock
(Global Scope)
#include <stdio.h>
#include <time.h>
#include <string.h>
#include <gl/glut.h>
#include <math.h>
void init()
{
glClearColor(0.0, 0.0, 0.0, 0.0);
glColor3f(0.0f, 0.0f, 0.0f);
glPointSize(5.0f);
//gluOrtho2D(0.0f, 640.0f, 0.0f, 480.0f);
glEnable(GL_POINT_SMOOTH);
glEnable(GL_POLYGON_SMOOTH);
glEnable(GL_LINE_SMOOTH);
}
100 %
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'OpenGLClock' (1 project)
OpenGLClock
References
External Dependencies
Header Files
Resource Files
Source Files
main.cpp
packages.config
Solution Explorer Team Explorer
Properties
Output
Show output from: Debug
'OpenGLClock.exe' (Win32): Loaded 'C:\Windows\SysWOW64\d3d11.dll'. Cannot find or open the PDB file.
'OpenGLClock.exe' (Win32): Loaded 'C:\Windows\SysWOW64\ResourcePolicyClient.dll'. Cannot find or open the PDB file.
'OpenGLClock.exe' (Win32): Unloaded 'C:\Windows\SysWOW64\ResourcePolicyClient.dll'
'OpenGLClock.exe' (Win32): Loaded 'C:\Windows\SysWOW64\igd10umd32.dll'. Cannot find or open the PDB file.
The thread 0x734 has exited with code 0 (0x0).
The thread 0x1fd0 has exited with code 0 (0x0).
The thread 0xf88 has exited with code 0 (0x0).
The program '[6832] OpenGLClock.exe' has exited with code 0 (0x0).
Activate Windows
Go to Settings to activate Windows.
Ready
Ln 9 Col 1 Ch 1 INS
COSC-4332-Computer-Graphics master
9:11 AM
1/31/2018


OpenGLClock - Microsoft Visual Studio
main.cpp
void Timer(int value)
{
double tickInterval = 10.0; // Chnage this value to set the step size.
double tickDegree = (360.0 / 60.0) * (tickInterval / 1000.0);
glutTimerFunc(tickInterval, Timer, 0);
glutPostRedisplay();
seconds += tickDegree;
minutes += tickDegree / 60.0;
hours += (tickDegree / 60.0) / 12.0;
if (seconds > 360)
seconds -= 360;
if (minutes > 360)
minutes -= 360;
if (hours > 360)
hours -= 360;
printf("%s %f\n", "Seconds = ", seconds);
}
double ticks = 0;
Output
'OpenGLClock.exe' (Win32): Loaded 'C:\Windows\SysWOW64\d3d11.dll'. Cannot find or open the PDB file.
'OpenGLClock.exe' (Win32): Loaded 'C:\Windows\SysWOW64\ResourcePolicyClient.dll'. Cannot find or open the PDB file.
'OpenGLClock.exe' (Win32): Unloaded 'C:\Windows\SysWOW64\ResourcePolicyClient.dll'
'OpenGLClock.exe' (Win32): Loaded 'C:\Windows\SysWOW64\igd10umd32.dll'. Cannot find or open the PDB file.
The thread 0x2c08 has exited with code 0 (0x0).
The thread 0x26ac has exited with code 0 (0x0).
The thread 0x29b8 has exited with code 0 (0x0).
The program '[4436] OpenGLClock.exe' has exited with code 0 (0x0).
Solution Explorer
Solution 'OpenGLClock' (1 project)
OpenGLClock
References
External Dependencies
Header Files
Resource Files
Source Files
main.cpp
packages.config
Properties
Activate Windows
Go to Settings to activate Windows.
Ready
9:14 AM
1/31/2018


Square

Modified Code:

```
#include <stdio.h>
#include <time.h>
#include <string.h>

#include <gl/glut.h>
#include <math.h>



void init()
{
	glClearColor(0.2, 0.2, 0.2, 1.0);
	glColor3f(0.0f, 0.0f, 0.0f);
	glPointSize(2.0f);
	//gluOrtho2D(0.0f, 640.0f, 0.0f, 480.0f);
	glEnable(GL_POINT_SMOOTH);
	glEnable(GL_POLYGON_SMOOTH);
	glEnable(GL_LINE_SMOOTH);
}



double seconds = 0;
double minutes = 0;
double hours = 0;


void Timer(int value)
{
	double tickInterval = 10; // Chnage this value to set the step size.
	double tickDegree = (360 / 60) * (tickInterval / 1000);

	glutTimerFunc(tickInterval, Timer, 0);
	glutPostRedisplay();
	seconds += tickDegree;
	minutes += tickDegree / 60;
	hours += (tickDegree / 60) / 12;

	if (seconds > 360)
		seconds -= 360;
	if (minutes > 360)
		minutes -= 360;
	if (hours > 360)
		hours -= 360;

	printf("%s %f\n", "Seconds = ", seconds);
}
double ticks = 0;

// Your first Point The hello world application
void drawpoint()
```

```
{
	glPushMatrix();
	glRotated(ticks += 1, 0, 0, 1);
	glTranslated(0.95, 0, 0);
	glColor3f(1, 1, 1);
	glBegin(GL_POINTS);
	glVertex2d(0, 0);
	glEnd();
	glPopMatrix();
}

double mark = 0;
double space = 360 / 12;
void drawTimeMarker()
{
	glPushMatrix();
	glRotated(mark += space, 0, 0, 1);
	glColor3f(1, 1, 1);
	glBegin(GL_POLYGON);
	glVertex2d(-0.003, 0.8);
	glVertex2d(0.003, 0.8);
	glVertex2d(-0.003, 0.9);
	glVertex2d(0.003, 0.9);
	glEnd();
	glPopMatrix();
}

void drawSecondsHand()
{
	glPushMatrix();  //Tells OpenGL to store the current state that we are in.
	glRotated(-seconds, 0, 0, 1);
	glColor3f(1.0, 0.0, 0.0);
	glBegin(GL_POLYGON);
	glVertex2d(-0.01, 0);
	glVertex2d(0.01, 0);
	glVertex2d(0.01, 0.6);
	glVertex2d(0, 0.9);
	glVertex2d(-0.01, 0.6);
	glEnd();
	glPopMatrix();   //Then when we want to go back to our previous state, we call
glPopMatrix().
}
void drawMinuteHand()
{
	glPushMatrix();
	glRotated(-minutes, 0, 0, 1);
	glColor3f(0.0, 1.0, 0.0);
	glBegin(GL_POLYGON);
	glVertex2d(-0.01, 0);
	glVertex2d(0.01, 0);
	glVertex2d(0.01, 0.5);
	glVertex2d(0, 0.8);
	glVertex2d(-0.01, 0.5);
	glEnd();
	glPopMatrix();
}
void drawHoursHand()
{
```

```
	glPushMatrix();
	glRotated(-hours, 0, 0, 1);
	glColor3f(0.0, 0.0, 1.0);
	glBegin(GL_POLYGON);

	glVertex2d(-0.02, 0);
	glVertex2d(0.02, 0);
	glVertex2d(0.02, 0.4);
	glVertex2d(0, 0.7);
	glVertex2d(-0.02, 0.4);
	glEnd();
	glPopMatrix();
}


void drawcircle()
{

	//	glLoadIdentity(); //Reset the drawing perspective
	for (int i = 0; i < 360; i++)
	{
		drawpoint();
	}

	for (int i = 0; i < 12; i++)
	{
		drawTimeMarker();
	}
	/*
	OpenGL keeps a stack of matrices to quickly apply and remove transformations.
glPushMatrix copies the top matrix and pushes it onto the stack, while glPopMatrix pops
the top matrix off the
	stack. All transformation functions (glScaled, etc.) function on the top matrix,
and the top matrix is what all rendering commands use to transform their data.
	By pushing and popping matrices, you can control what transformations apply to
which objects, as well as apply transformations to groups of objects, and easily reverse
the transformations
	so that they don't affect other objects.
	*/

	// hours hand
	drawHoursHand();
	// Minutes hand
	drawMinuteHand();
	// Seconds hand
	drawSecondsHand();
}
void display(void) {

	glClear(GL_COLOR_BUFFER_BIT);
	glLineWidth(2.5);
	glColor3f(1.0, 0.0, 0.0);
	drawcircle();
	glFlush();
}
int main(int argc, char **argv) {

	glutInit(&argc, argv);
```

```
	glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
	glutInitWindowPosition(200, 200);
	glutInitWindowSize(480, 480);
	glutCreateWindow("Square");
	glutDisplayFunc(display);
	init();
	Timer(0);
	glutMainLoop();
	return 0;
}
```