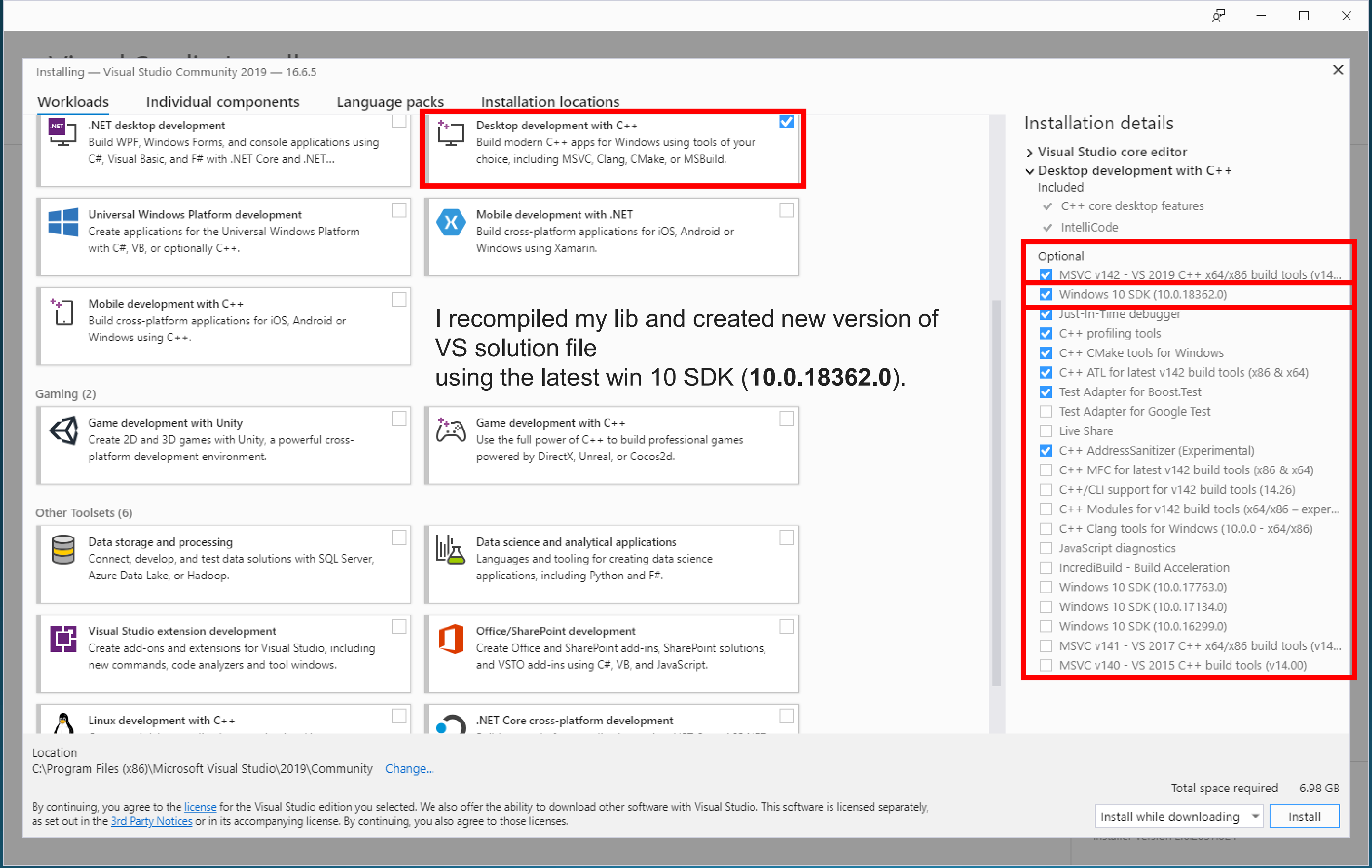


Installing — Visual Studio Community 2019 — 16.6.5

□ .NET desktop development
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET Core and .NET...

☑ Desktop development with C++
Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.

□ Universal Windows Platform development
Create applications for the Universal Windows Platform with C#, VB, or optionally C++.

□ Mobile development with .NET
Build cross-platform applications for iOS, Android or Windows using Xamarin.

□ Mobile development with C++
Build cross-platform applications for iOS, Android or Windows using C++.

I recompiled my lib and created new version of VS solution file
using the latest win 10 SDK (**10.0.18362.0**).

Gaming (2)

□ Game development with Unity
Create 2D and 3D games with Unity, a powerful cross-platform development environment.

□ Game development with C++
Use the full power of C++ to build professional games powered by DirectX, Unreal, or Cocos2d.

Other Toolsets (6)

□ Data storage and processing
Connect, develop, and test data solutions with SQL Server, Azure Data Lake, or Hadoop.

□ Data science and analytical applications
Languages and tooling for creating data science applications, including Python and F#.

□ Visual Studio extension development
Create add-ons and extensions for Visual Studio, including new commands, code analyzers and tool windows.

□ Office/SharePoint development
Create Office and SharePoint add-ins, SharePoint solutions, and VSTO add-ins using C#, VB, and JavaScript.

□ Linux development with C++

□ .NET Core cross-platform development

Installation details

> Visual Studio core editor

∨ Desktop development with C++

Included

✓ C++ core desktop features

✓ IntelliCode

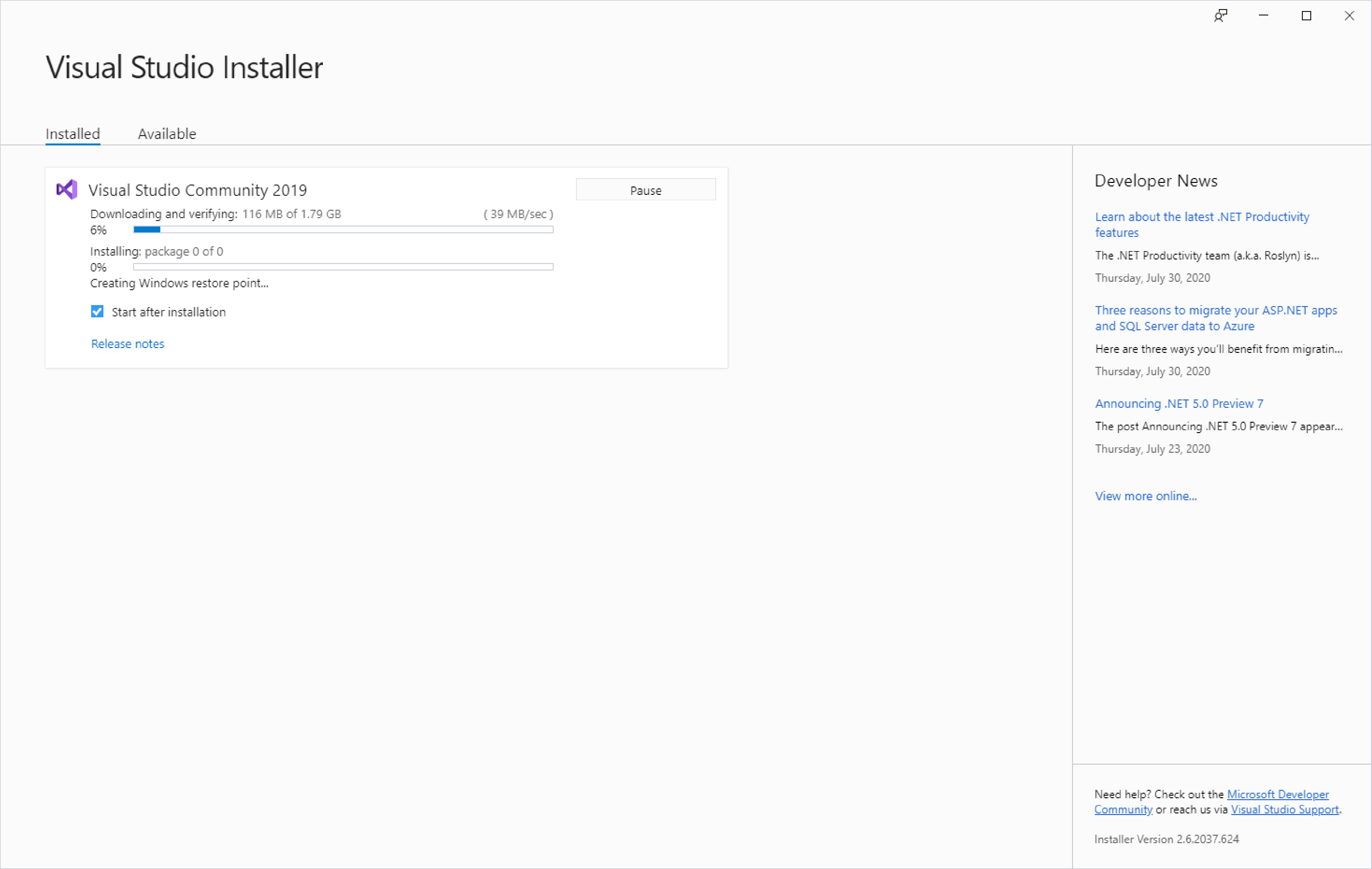
Optional

☑ MSVC v142 - VS 2019 C++ x64/x86 build tools (v14...
☑ Windows 10 SDK (10.0.18362.0)
☑ Just-In-Time debugger
☑ C++ profiling tools
☑ C++ CMake tools for Windows
☑ C++ ATL for latest v142 build tools (x86 & x64)
☑ Test Adapter for Boost.Test
□ Test Adapter for Google Test
□ Live Share
☑ C++ AddressSanitizer (Experimental)
□ C++ MFC for latest v142 build tools (x86 & x64)
□ C++/CLI support for v142 build tools (14.26)
□ C++ Modules for v142 build tools (x64/x86 – exper...
□ C++ Clang tools for Windows (10.0.0 - x64/x86)
□ JavaScript diagnostics
□ IncrediBuild - Build Acceleration
□ Windows 10 SDK (10.0.17763.0)
□ Windows 10 SDK (10.0.17134.0)
□ Windows 10 SDK (10.0.16299.0)
□ MSVC v141 - VS 2017 C++ x64/x86 build tools (v14...
□ MSVC v140 - VS 2015 C++ build tools (v14.00)

Location
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community Change...

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 6.98 GB



Visual Studio Installer

Visual Studio Community 2019

Downloading and verifying: 116 MB of 1.79 GB (39 MB/sec)

6%

Installing: package 0 of 0

0%

Creating Windows restore point...

☑ Start after installation

Release notes

Developer News

Learn about the latest .NET Productivity features

The .NET Productivity team (a.k.a. Roslyn) is...

Thursday, July 30, 2020

Three reasons to migrate your ASP.NET apps and SQL Server data to Azure

Here are three ways you'll benefit from migratin...

Thursday, July 30, 2020

Announcing .NET 5.0 Preview 7

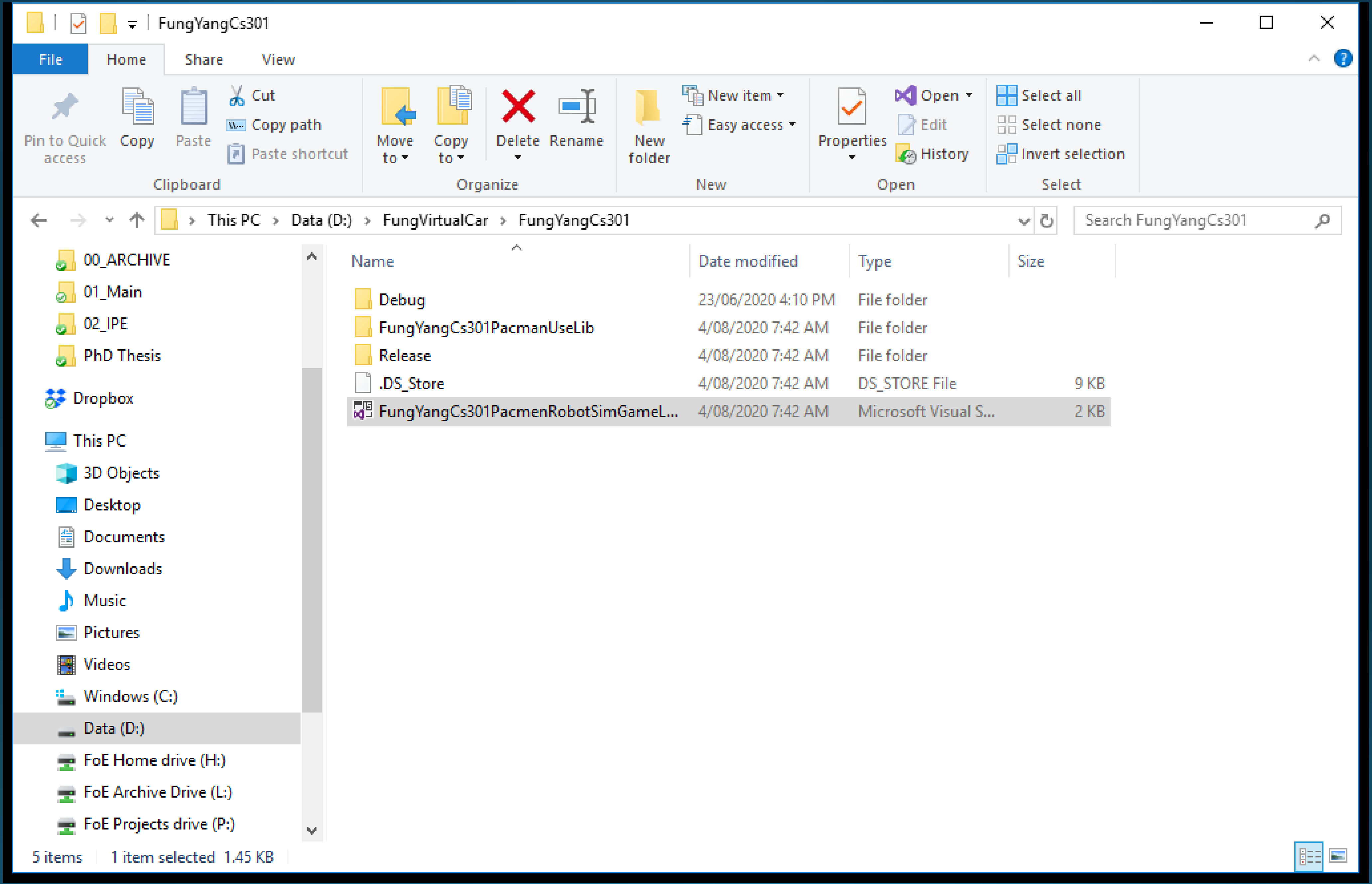
The post Announcing .NET 5.0 Preview 7 appear...

Thursday, July 23, 2020

View more online...

Need help? Check out the Microsoft Developer Community or reach us via Visual Studio Support.

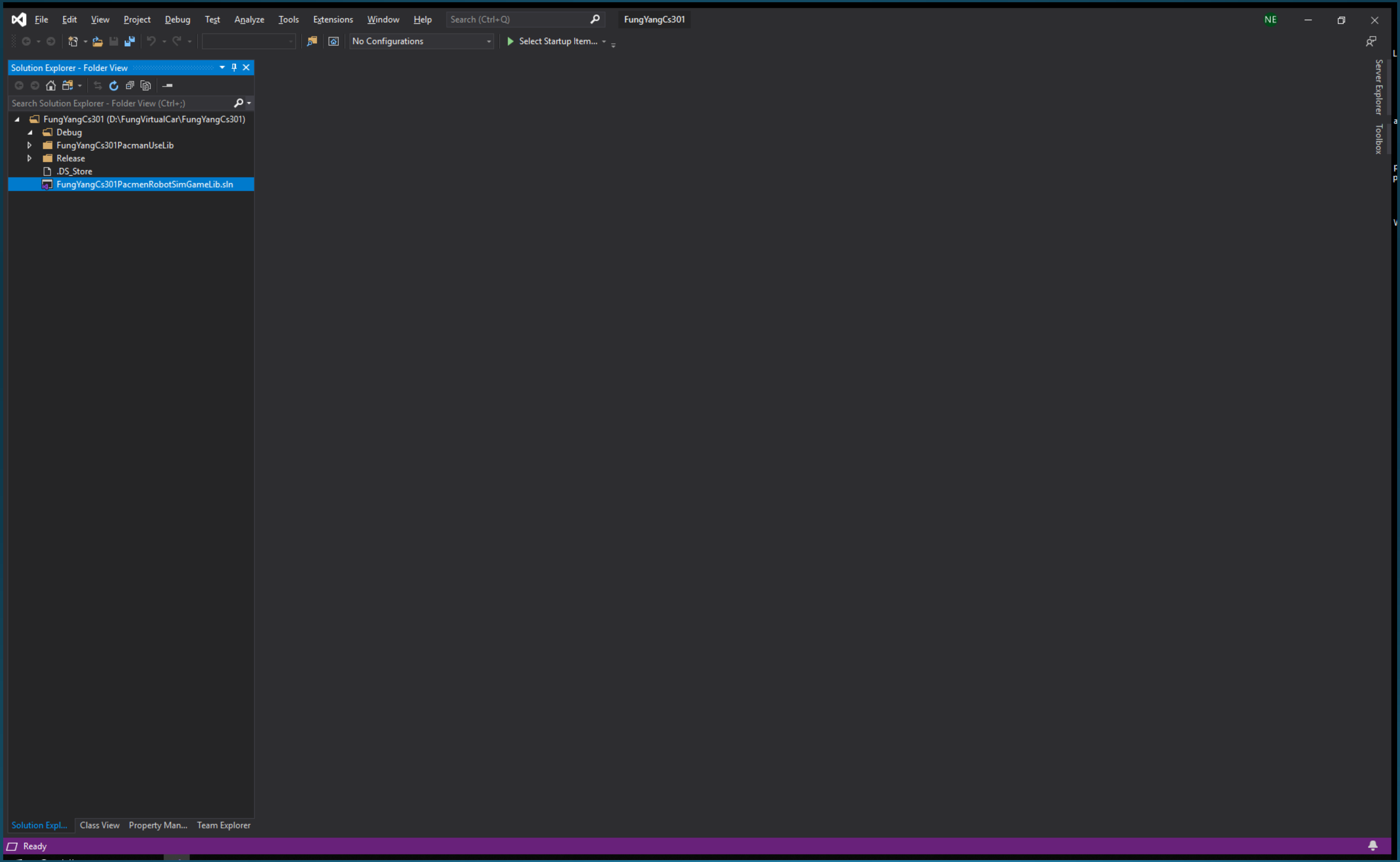
Installer Version 2.6.2037.624



This PC › Data (D:) › FungVirtualCar › FungYangCs301

Name	Date modified	Type	Size
Debug	23/06/2020 4:10 PM	File folder	
FungYangCs301PacmanUseLib	4/08/2020 7:42 AM	File folder	
Release	4/08/2020 7:42 AM	File folder	
.DS_Store	4/08/2020 7:42 AM	DS_STORE File	9 KB
FungYangCs301PacmenRobotSimGameL...	4/08/2020 7:42 AM	Microsoft Visual S...	2 KB

5 items 1 item selected 1.45 KB



FungYangCs301 (D:\FungVirtualCar\FungYangCs301)
Debug
FungYangCs301PacmanUseLib
Release
.DS_Store
FungYangCs301PacmenRobotSimGameLib.sln

Folder architecture

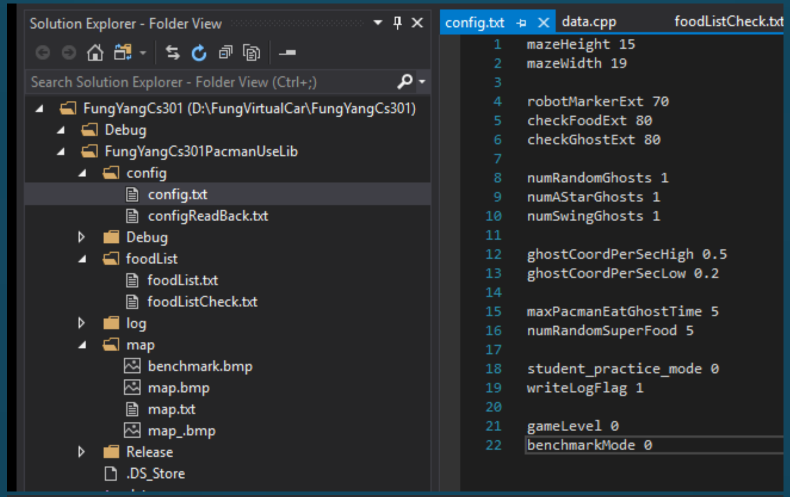
Main folder

- Debug ✖
- Release ✖
- FungYangCs301PacmanUseLib ⬅
- Solution (like a workspace, .sln)

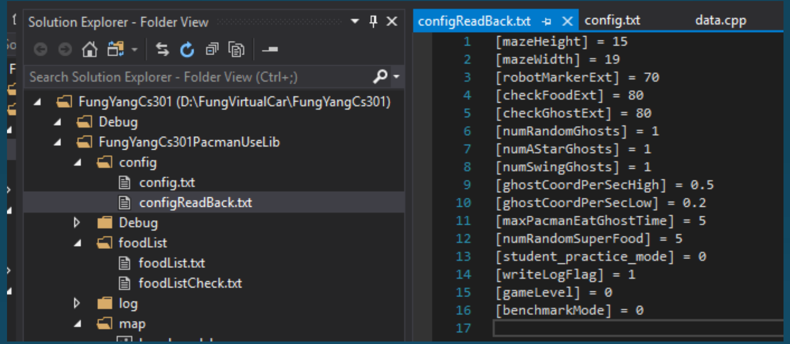
FungYangCs301PacmanUseLib

- config(2) ←
- Debug(0) ✕
- foodList(2) ←
- log(1) ←
    - exame ←
- map(4) ←
- Release ✕

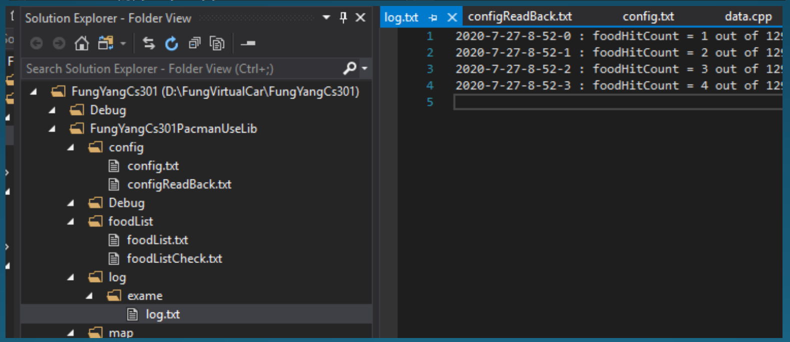
data.cpp
data.h
highPerformanceTimer.h
fungYangCs301SimMain.cpp
mainFungGLAppEngin.h



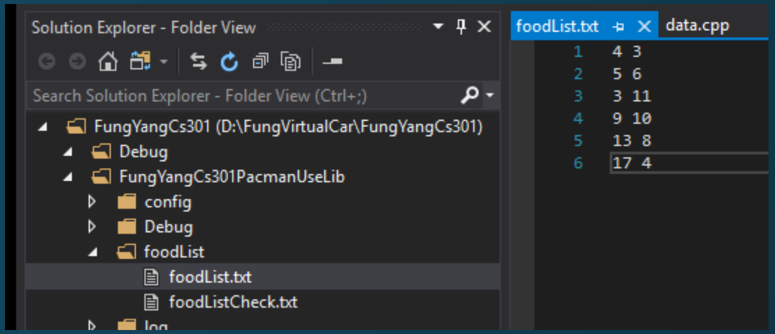
```
mazeHeight 15
mazeWidth 19

robotMarkerExt 70
checkFoodExt 80
checkGhostExt 80

numRandomGhosts 1
numAStarGhosts 1
numSwingGhosts 1

ghostCoordPerSecHigh 0.5
ghostCoordPerSecLow 0.2

maxPacmanEatGhostTime 5
numRandomSuperFood 5

student_practice_mode 0
writeLogFlag 1

gameLevel 0
benchmarkMode 0
```


Solution Explorer - Folder View
Search Solution Explorer - Folder View (Ctrl+;)
FungYangCs301 (D:\FungVirtualCar\FungYangCs301)
Debug
FungYangCs301PacmanUseLib
config
config.txt
configReadBack.txt
Debug
foodList
foodList.txt
foodListCheck.txt
log
map
configReadBack.txt
config.txt
data.cpp

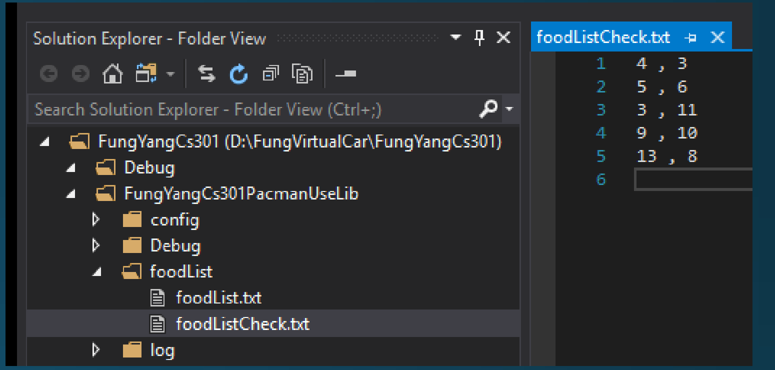
```
[mazeHeight] = 15
[mazeWidth] = 19
[robotMarkerExt] = 70
[checkFoodExt] = 80
[checkGhostExt] = 80
[numRandomGhosts] = 1
[numAStarGhosts] = 1
[numSwingGhosts] = 1
[ghostCoordPerSecHigh] = 0.5
[ghostCoordPerSecLow] = 0.2
[maxPacmanEatGhostTime] = 5
[numRandomSuperFood] = 5
[student_practice_mode] = 0
[writeLogFlag] = 1
[gameLevel] = 0
[benchmarkMode] = 0
```


Solution Explorer - Folder View
Search Solution Explorer - Folder View (Ctrl+;)
FungYangCs301 (D:\FungVirtualCar\FungYangCs301)
Debug
FungYangCs301PacmanUseLib
config
config.txt
configReadBack.txt
Debug
foodList
foodList.txt
foodListCheck.txt
log
exame
log.txt
map
log.txt
configReadBack.txt
config.txt
data.cpp

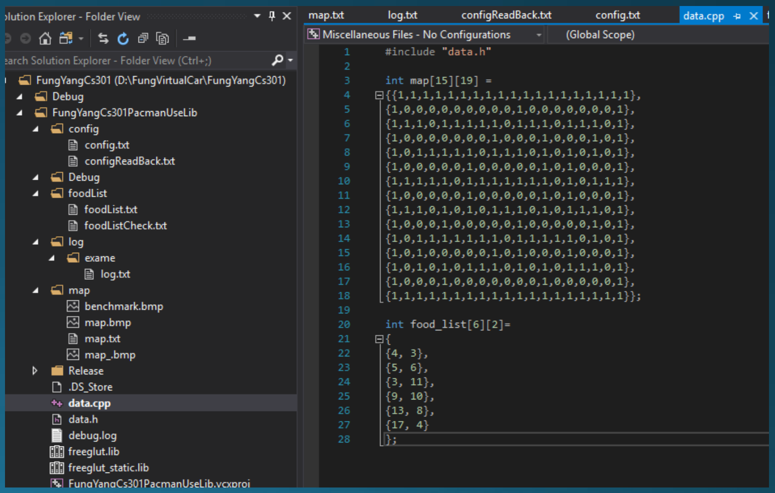
```
2020-7-27-8-52-0 : foodHitCount = 1 out of 12
2020-7-27-8-52-1 : foodHitCount = 2 out of 12
2020-7-27-8-52-2 : foodHitCount = 3 out of 12
2020-7-27-8-52-3 : foodHitCount = 4 out of 12
```


Solution Explorer - Folder View
Search Solution Explorer - Folder View (Ctrl+;)
FungYangCs301 (D:\FungVirtualCar\FungYangCs301)
Debug
FungYangCs301PacmanUseLib
config
Debug
foodList
foodList.txt
foodListCheck.txt
log
foodList.txt
data.cpp

```
4 3
5 6
3 11
9 10
13 8
17 4
```


Solution Explorer - Folder View
Search Solution Explorer - Folder View (Ctrl+;)
FungYangCs301 (D:\FungVirtualCar\FungYangCs301)
Debug
FungYangCs301PacmanUseLib
config
Debug
foodList
foodList.txt
foodListCheck.txt
log
foodListCheck.txt

```
4 , 3
5 , 6
3 , 11
9 , 10
13 , 8
```

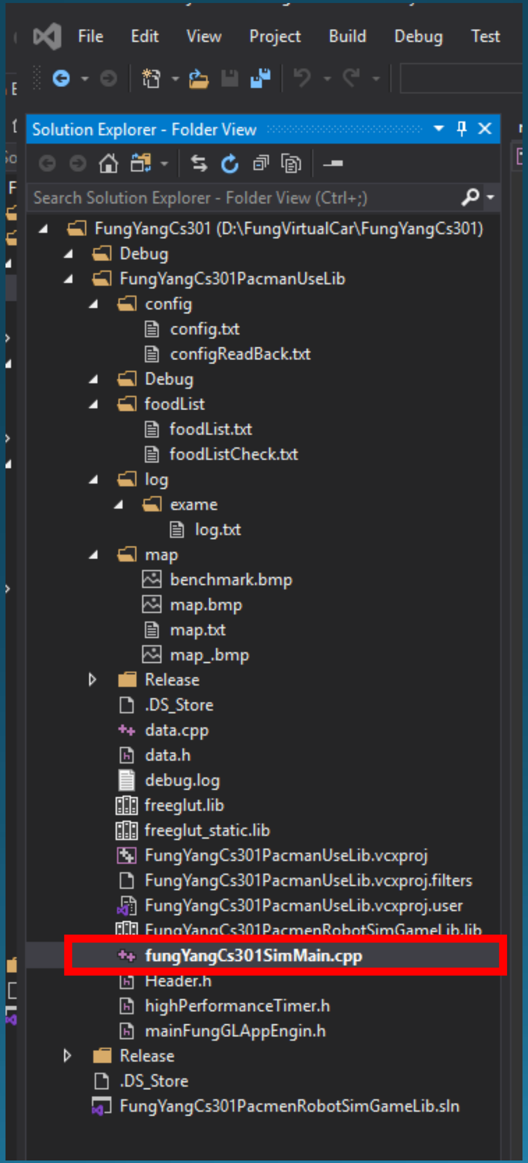


```
#include "data.h"

int map[15][19] =
{{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1},
{1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1},
{1,1,1,0,1,1,1,1,1,0,1,1,1,0,1,1,1,0,1},
{1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,1,0,1},
{1,0,1,1,1,1,1,0,1,1,1,0,1,0,1,0,1,0,1},
{1,0,0,0,0,0,1,0,0,0,0,0,1,0,1,0,0,0,1},
{1,1,1,1,1,0,1,1,1,1,1,1,1,0,1,0,1,1,1},
{1,0,0,0,0,0,1,0,0,0,0,0,1,0,1,0,0,0,1},
{1,1,1,0,1,0,1,0,1,1,1,0,1,0,1,1,1,0,1},
{1,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,1},
{1,0,1,1,1,1,1,1,1,0,1,1,1,1,1,0,1,0,1},
{1,0,1,0,0,0,0,0,1,0,1,0,0,0,1,0,0,0,1},
{1,0,1,0,1,0,1,1,1,0,1,0,1,0,1,1,1,0,1},
{1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,1},
{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1}};

int food_list[6][2]=
{
{4, 3},
{5, 6},
{3, 11},
{9, 10},
{13, 8},
{17, 4}
};
```

```
2020-7-27-8-52-0 : foodHitCount = 1 out of 129 ;  foodHitCellPos = (17 , 11) ; idxToErase = 106 ; superFoodHitCount = 0 out of 5 ; ghostHitCount = 0 ; ghostEatenCount = 0 ;
```

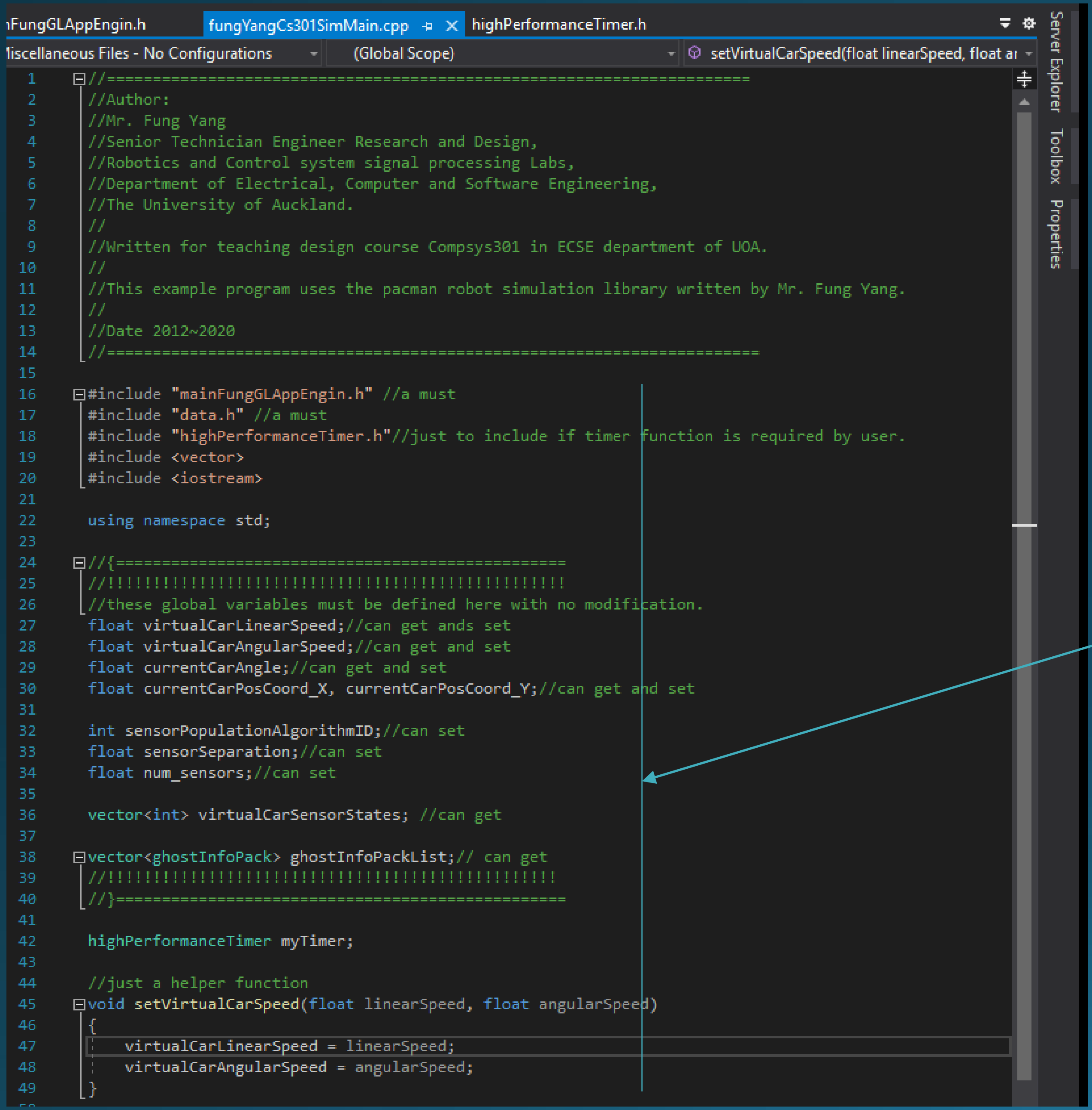
You will be working in 1 file (but you can create libraries if you wish)

fungYangCs301SimMain.cpp



File
Edit
View
Project
Build
Debug
Test
Solution Explorer - Folder View
Search Solution Explorer - Folder View (Ctrl+;)

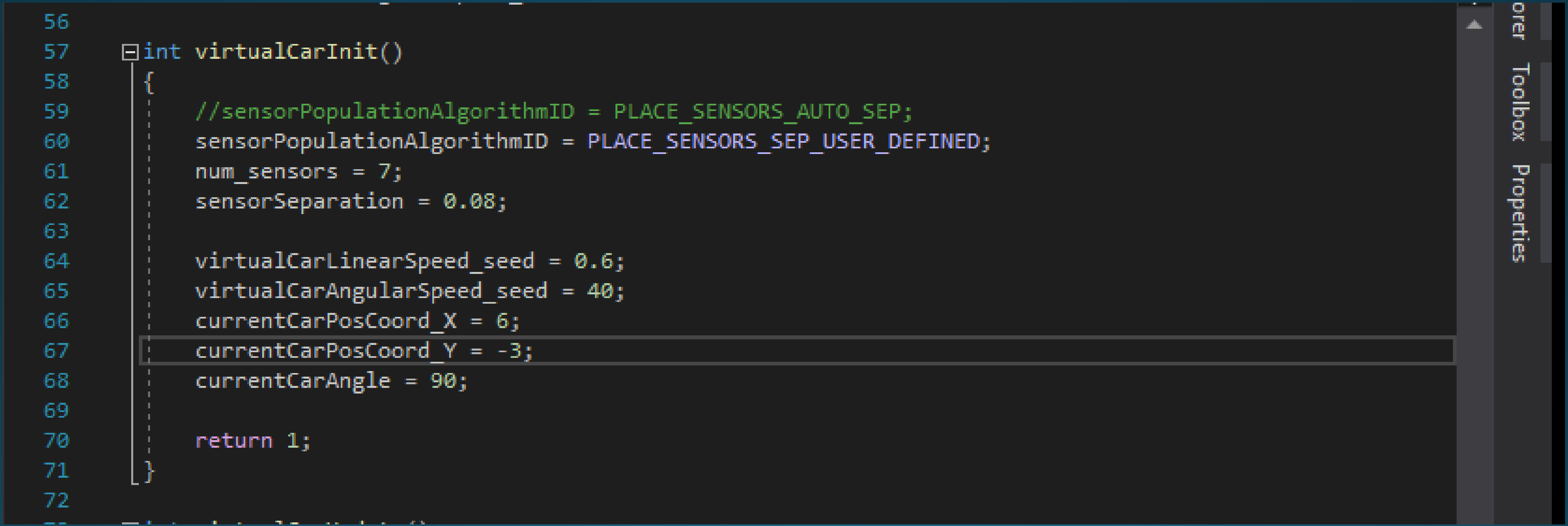
```
◢ FungYangCs301 (D:\FungVirtualCar\FungYangCs301)
  ◢ Debug
  ◢ FungYangCs301PacmanUseLib
    ◢ config
        config.txt
        configReadBack.txt
    ◢ Debug
    ◢ foodList
        foodList.txt
        foodListCheck.txt
    ◢ log
      ◢ exame
          log.txt
    ◢ map
        benchmark.bmp
        map.bmp
        map.txt
        map_.bmp
    ▷ Release
      .DS_Store
      data.cpp
      data.h
      debug.log
      freeglut.lib
      freeglut_static.lib
      FungYangCs301PacmanUseLib.vcxproj
      FungYangCs301PacmanUseLib.vcxproj.filters
      FungYangCs301PacmanUseLib.vcxproj.user
      FungYangCs301PacmenRobotSimGameLib.lib
      fungYangCs301SimMain.cpp
      Header.h
      highPerformanceTimer.h
      mainFungGLAppEngin.h
  ▷ Release
    .DS_Store
    FungYangCs301PacmenRobotSimGameLib.sln
```



```cpp
//======================================================================
//Author:
//Mr. Fung Yang
//Senior Technician Engineer Research and Design,
//Robotics and Control system signal processing Labs,
//Department of Electrical, Computer and Software Engineering,
//The University of Auckland.
//
//Written for teaching design course Compsys301 in ECSE department of UOA.
//
//This example program uses the pacman robot simulation library written by Mr. Fung Yang.
//
//Date 2012~2020
//=======================================================================

#include "mainFungGLAppEngin.h" //a must
#include "data.h" //a must
#include "highPerformanceTimer.h"//just to include if timer function is required by user.
#include <vector>
#include <iostream>

using namespace std;

//{=================================================
//!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
//these global variables must be defined here with no modification.
float virtualCarLinearSpeed;//can get ands set
float virtualCarAngularSpeed;//can get and set
float currentCarAngle;//can get and set
float currentCarPosCoord_X, currentCarPosCoord_Y;//can get and set

int sensorPopulationAlgorithmID;//can set
float sensorSeparation;//can set
float num_sensors;//can set

vector<int> virtualCarSensorStates; //can get

vector<ghostInfoPack> ghostInfoPackList;// can get
//!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
//}=================================================

highPerformanceTimer myTimer;

//just a helper function
void setVirtualCarSpeed(float linearSpeed, float angularSpeed)
{
    virtualCarLinearSpeed = linearSpeed;
    virtualCarAngularSpeed = angularSpeed;
}
```

Do not modify

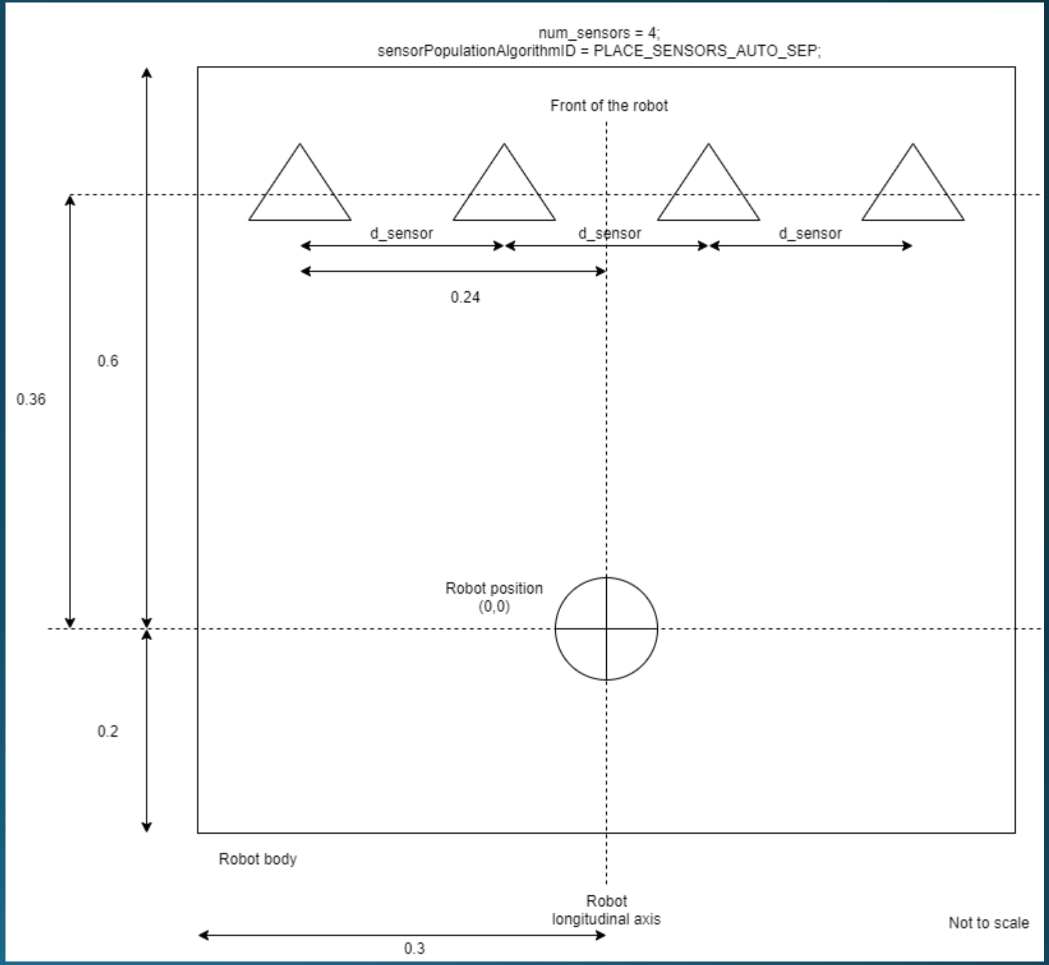
An “init” like Arduino, only run once, holds parameters about the robot sensors and the robot capabilities



```
int virtualCarInit()
{
    //sensorPopulationAlgorithmID = PLACE_SENSORS_AUTO_SEP;
    sensorPopulationAlgorithmID = PLACE_SENSORS_SEP_USER_DEFINED;
    num_sensors = 7;
    sensorSeparation = 0.08;

    virtualCarLinearSpeed_seed = 0.6;
    virtualCarAngularSpeed_seed = 40;
    currentCarPosCoord_X = 6;
    currentCarPosCoord_Y = -3;
    currentCarAngle = 90;

    return 1;
}
```

How to set up the light sensors: auto spacing or manual

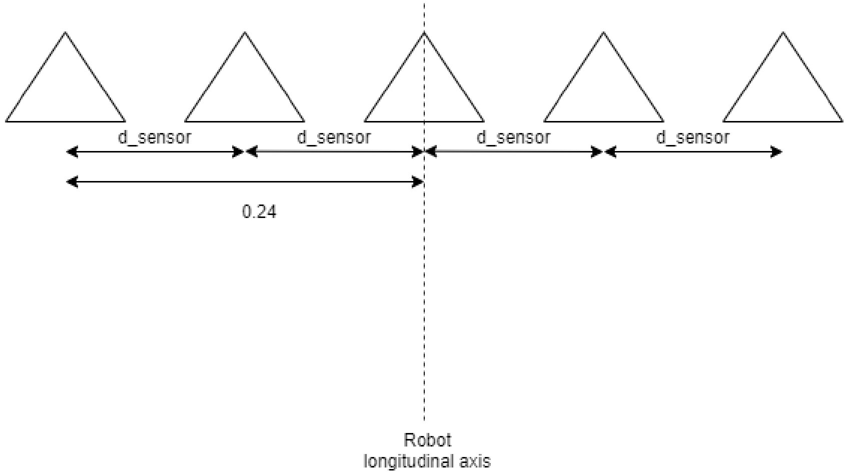
No SW limitation but we impose max 7 sensors
PLACE_SENSORS_SEP_USER_DEFINED/PLACE_SENSORS_AUTO_SEP



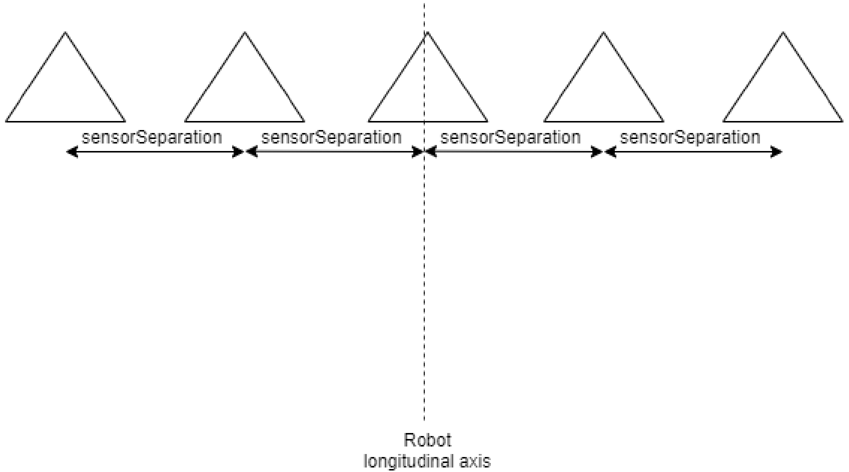
```
num_sensors = 4;
sensorPopulationAlgorithmID = PLACE_SENSORS_AUTO_SEP;
```

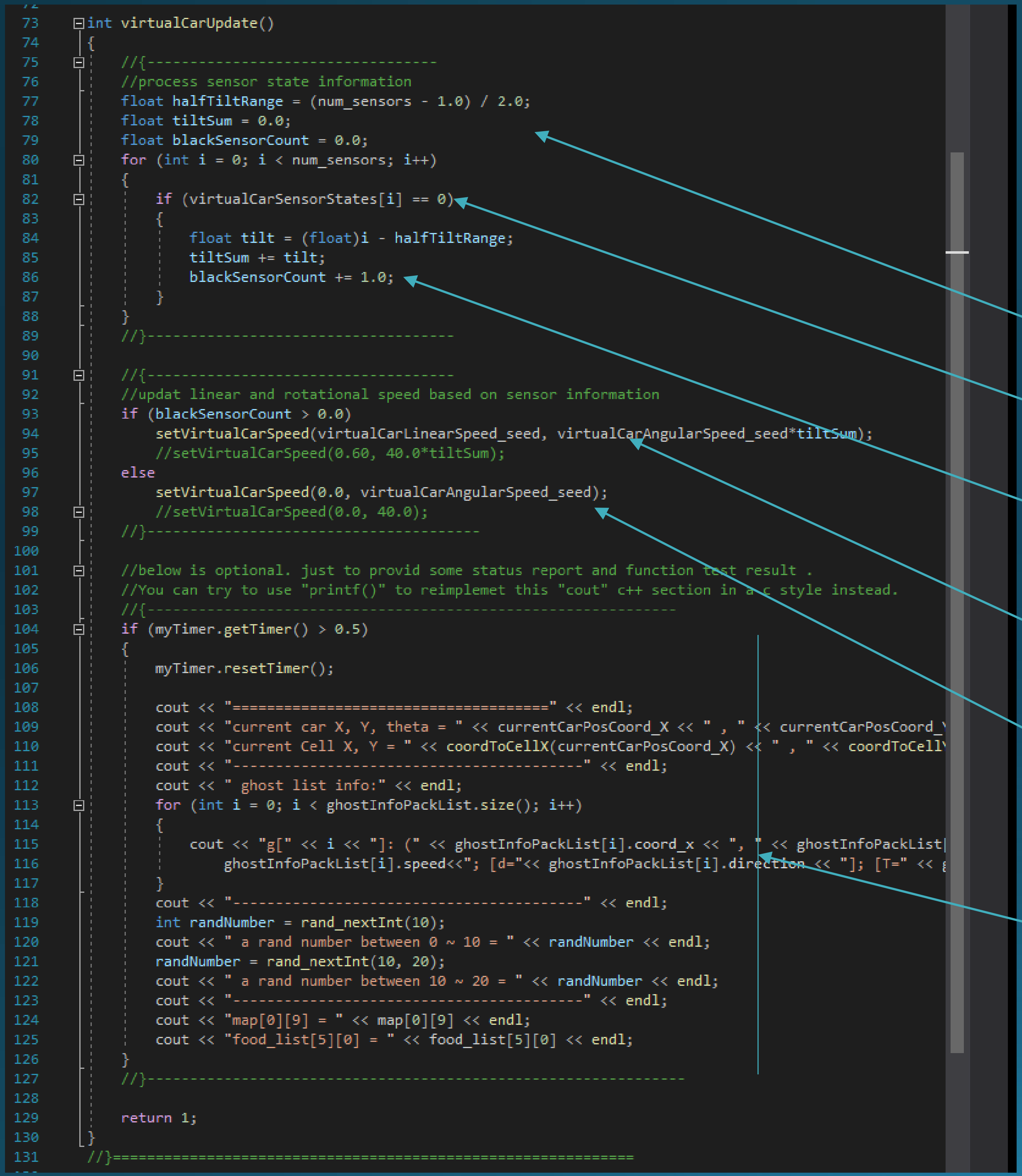
Questions:
How to read the state of the robot, ie light sensors
virtualCarSensorStates[i] // access the value of the light sensor (value of the pixel superposed to the center)

```
num_sensors = 5;
sensorPopulationAlgorithmID = PLACE_SENSORS_AUTO_SEP;
```



```
sensorPopulationAlgorithmID = PLACE_SENSORS_SEP_USER_DEFINED;
num_sensors = 5;
sensorSeparation = 0.08;
```





```
int virtualCarUpdate()
{
    //{----------------------------------
    //process sensor state information
    float halfTiltRange = (num_sensors - 1.0) / 2.0;
    float tiltSum = 0.0;
    float blackSensorCount = 0.0;
    for (int i = 0; i < num_sensors; i++)
    {
        if (virtualCarSensorStates[i] == 0)
        {
            float tilt = (float)i - halfTiltRange;
            tiltSum += tilt;
            blackSensorCount += 1.0;
        }
    }
    //}-----------------------------------

    //{-----------------------------------
    //updat linear and rotational speed based on sensor information
    if (blackSensorCount > 0.0)
        setVirtualCarSpeed(virtualCarLinearSpeed_seed, virtualCarAngularSpeed_seed*tiltSum);
        //setVirtualCarSpeed(0.60, 40.0*tiltSum);
    else
        setVirtualCarSpeed(0.0, virtualCarAngularSpeed_seed);
        //setVirtualCarSpeed(0.0, 40.0);
    //}--------------------------------------

    //below is optional. just to provid some status report and function test result .
    //You can try to use "printf()" to reimplemet this "cout" c++ section in a c style instead.
    //{------------------------------------------------------------
    if (myTimer.getTimer() > 0.5)
    {
        myTimer.resetTimer();

        cout << "=====================================" << endl;
        cout << "current car X, Y, theta = " << currentCarPosCoord_X << " , " << currentCarPosCoord_Y
        cout << "current Cell X, Y = " << coordToCellX(currentCarPosCoord_X) << " , " << coordToCellY
        cout << "----------------------------------------" << endl;
        cout << " ghost list info:" << endl;
        for (int i = 0; i < ghostInfoPackList.size(); i++)
        {
            cout << "g[" << i << "]: (" << ghostInfoPackList[i].coord_x << ", " << ghostInfoPackList[
                ghostInfoPackList[i].speed<<"; [d="<< ghostInfoPackList[i].direction << "]; [T=" << g
        }
        cout << "----------------------------------------" << endl;
        int randNumber = rand_nextInt(10);
        cout << " a rand number between 0 ~ 10 = " << randNumber << endl;
        randNumber = rand_nextInt(10, 20);
        cout << " a rand number between 10 ~ 20 = " << randNumber << endl;
        cout << "----------------------------------------" << endl;
        cout << "map[0][9] = " << map[0][9] << endl;
        cout << "food_list[5][0] = " << food_list[5][0] << endl;
    }
    //}------------------------------------------------------------

    return 1;
}
//}============================================================
```

An "update" like Arduino, runs continuously, holds code that need to be executed so the robot could move

Variables definition

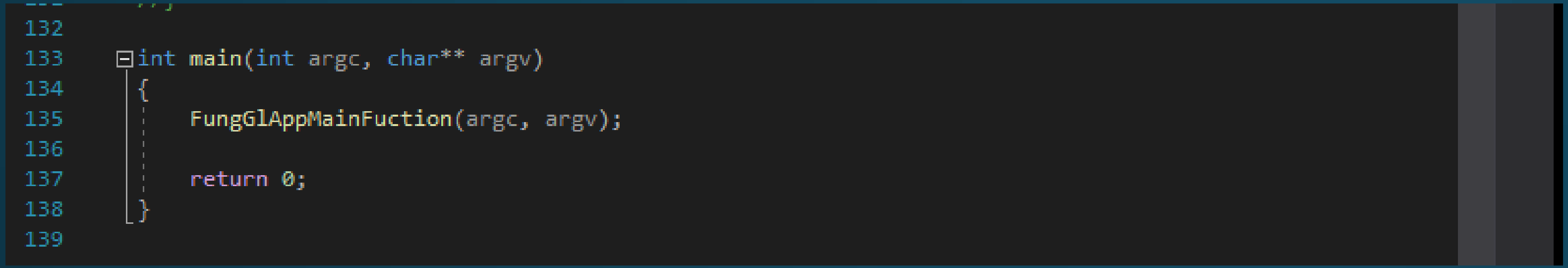
Checking the state of the light sensors

Counts how many sensors sees a line, ie black

If a line has been detected, do a composite movement of forward and turn

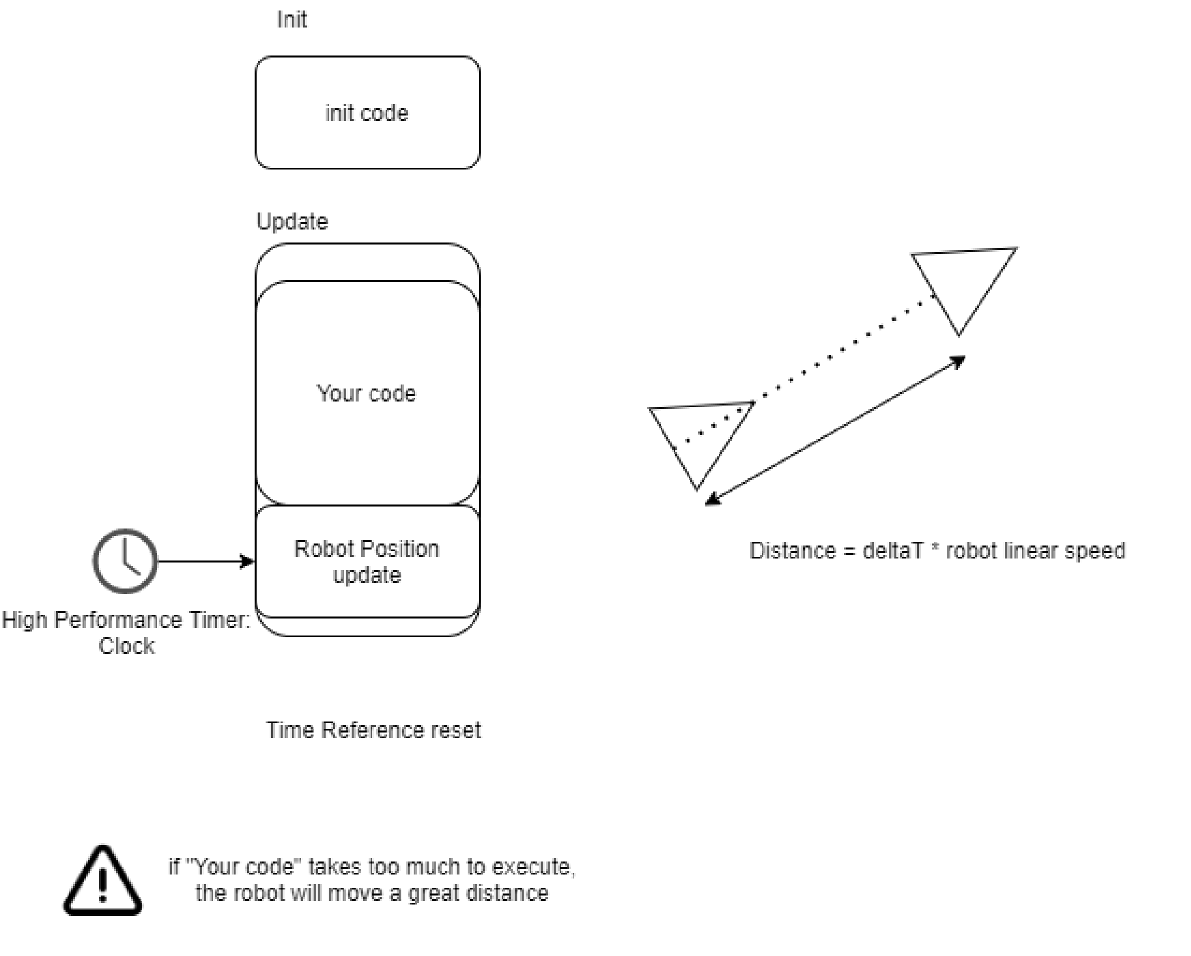
If no line has been detected, stop the robot (no linear speed, just angular speed)

Debug information: print to the console



```
int main(int argc, char** argv)
{
    FungGlAppMainFuction(argc, argv);

    return 0;
}
```

A main code with only 1 call to a function: FungGlAppMainFuction

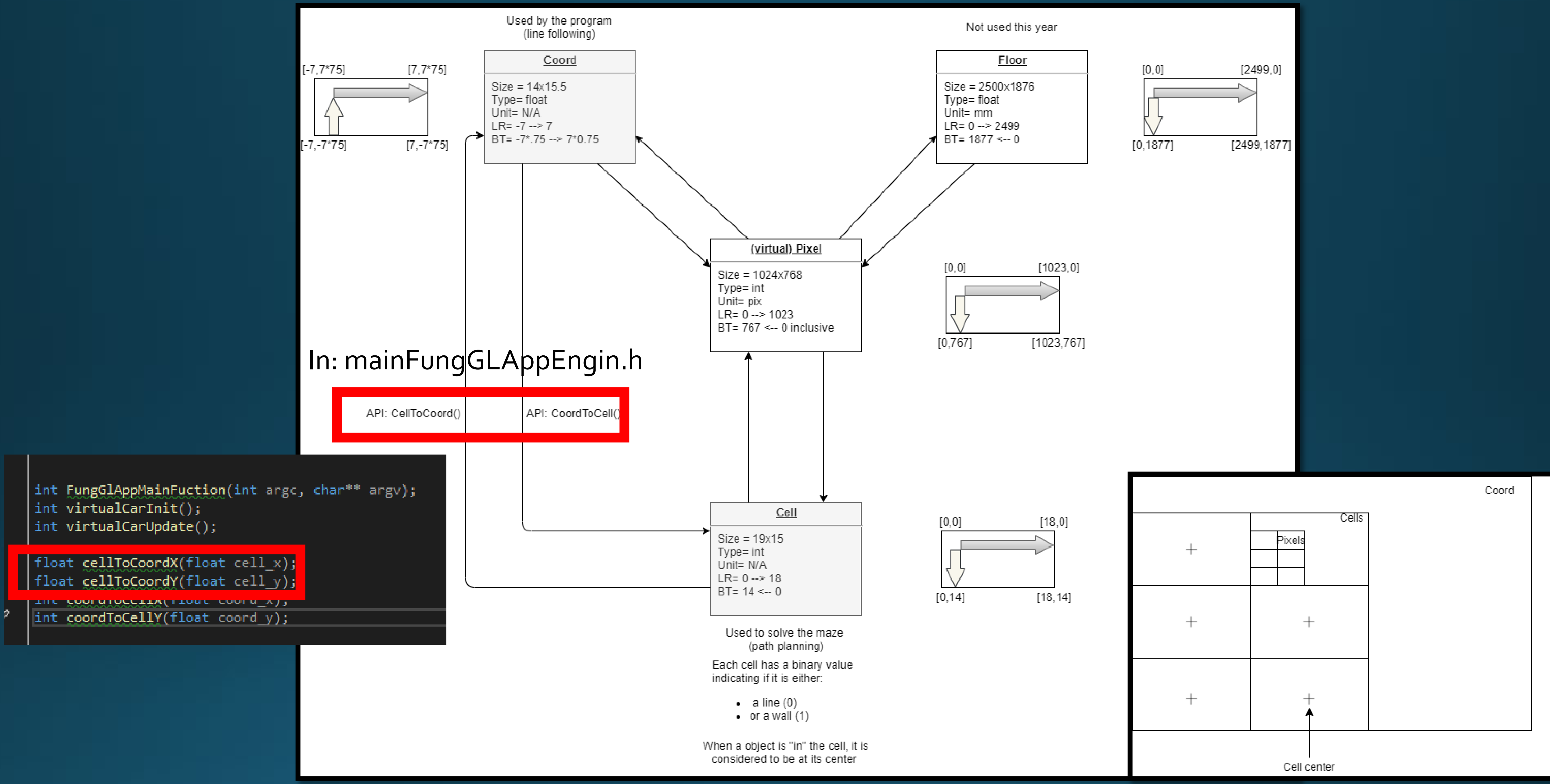


Distance = deltaT * robot linear speed

if "Your code" takes too much to execute, the robot will move a great distance

You will need to tweak some parameters depending on how calculation intensive your algorithm is and the speed of the computer running it

```
virtualCarLinearSpeed_seed = 0.6;
virtualCarAngularSpeed_seed = 40;
```



Used by the program
(line following)

Coord

Size = 14x15.5
Type= float
Unit= N/A
LR= -7 --> 7
BT= -7*.75 --> 7*0.75

Not used this year

Floor

Size = 2500x1876
Type= float
Unit= mm
LR= 0 --> 2499
BT= 1877 <-- 0

(virtual) Pixel

Size = 1024x768
Type= int
Unit= pix
LR= 0 --> 1023
BT= 767 <-- 0 inclusive

Cell

Size = 19x15
Type= int
Unit= N/A
LR= 0 --> 18
BT= 14 <-- 0

Used to solve the maze
(path planning)

Each cell has a binary value indicating if it is either:

- a line (0)
- or a wall (1)

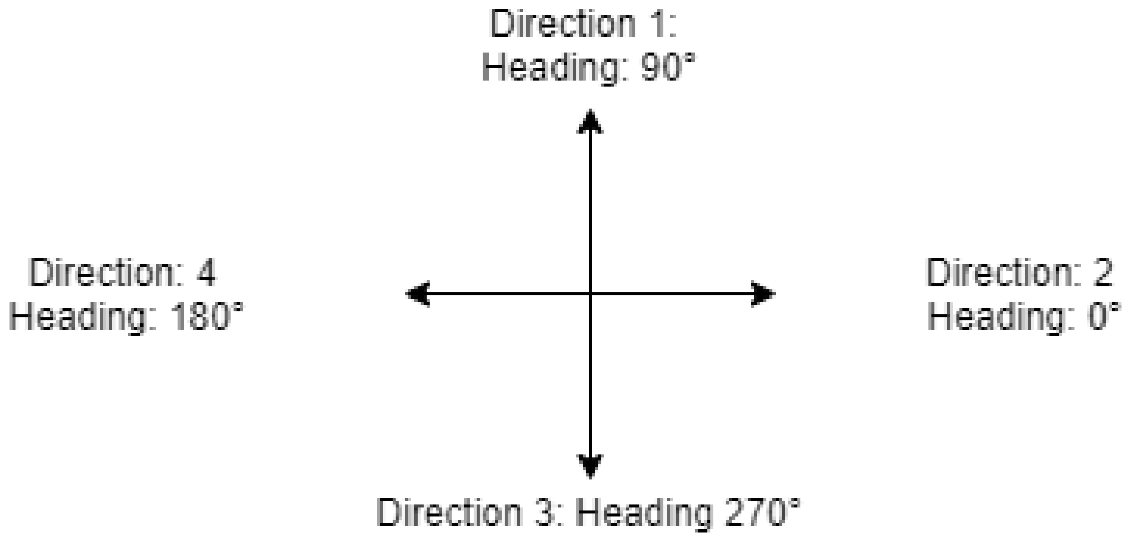
When a object is "in" the cell, it is considered to be at its center

In: mainFungGLAppEngin.h

API: CellToCoord()

API: CoordToCell()

```
int FungGlAppMainFuction(int argc, char** argv);
int virtualCarInit();
int virtualCarUpdate();

float cellToCoordX(float cell_x);
float cellToCoordY(float cell_y);
int coordToCellX(float coord_x);
int coordToCellY(float coord_y);
```



Direction 1:
Heading: 90°

Direction: 4
Heading: 180°

Direction: 2
Heading: 0°

Direction 3: Heading 270°

Direction: clock-wise
Heading: anti clock-wise

Robt use 0-90°
ghost use direction and not angle
because ghost can only go 4 directions

codes in:

int virtualCarInit(),

gives you the idea on how to set the car property through globle vars of the frame work.

codes in:

int virtualCarUpdate(),

gives you the idea on how to get the car and ghost property through globle vars of the frame work.
Through the cout<<... section.

And you will see that all the get/set properties of the virtual car/ghosts are all through
global variables of the framework, not even through a function! so should be very easy!

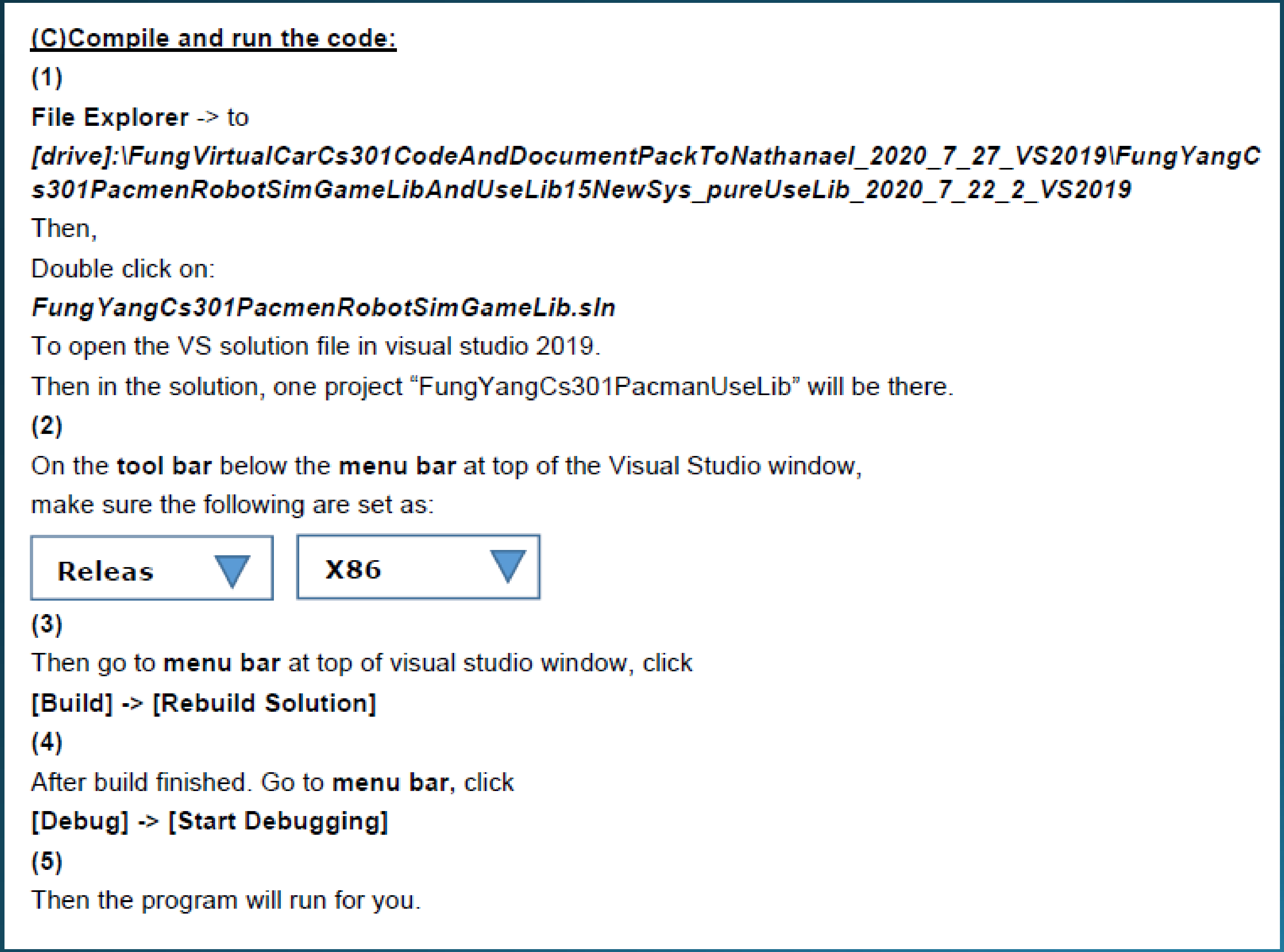
Your job is just to fill in your code in two functions of the framework:

virtualCarInit(), virtualCarUpdate().

One for initialization and one for loop.

Just like in coding Arduino,
there is only two functions you need to worry about:
setup(), loop().

Instructions to compile



(C)Compile and run the code:

(1)

File Explorer -> to

[drive]:\FungVirtualCarCs301CodeAndDocumentPackToNathanael_2020_7_27_VS2019\FungYangCs301PacmenRobotSimGameLibAndUseLib15NewSys_pureUseLib_2020_7_22_2_VS2019

Then,

Double click on:

FungYangCs301PacmenRobotSimGameLib.sln

To open the VS solution file in visual studio 2019.

Then in the solution, one project "FungYangCs301PacmanUseLib" will be there.

(2)

On the **tool bar** below the **menu bar** at top of the Visual Studio window, make sure the following are set as:

Releas ▼ **X86 ▼**

(3)

Then go to **menu bar** at top of visual studio window, click

[Build] -> [Rebuild Solution]

(4)

After build finished. Go to **menu bar**, click

[Debug] -> [Start Debugging]

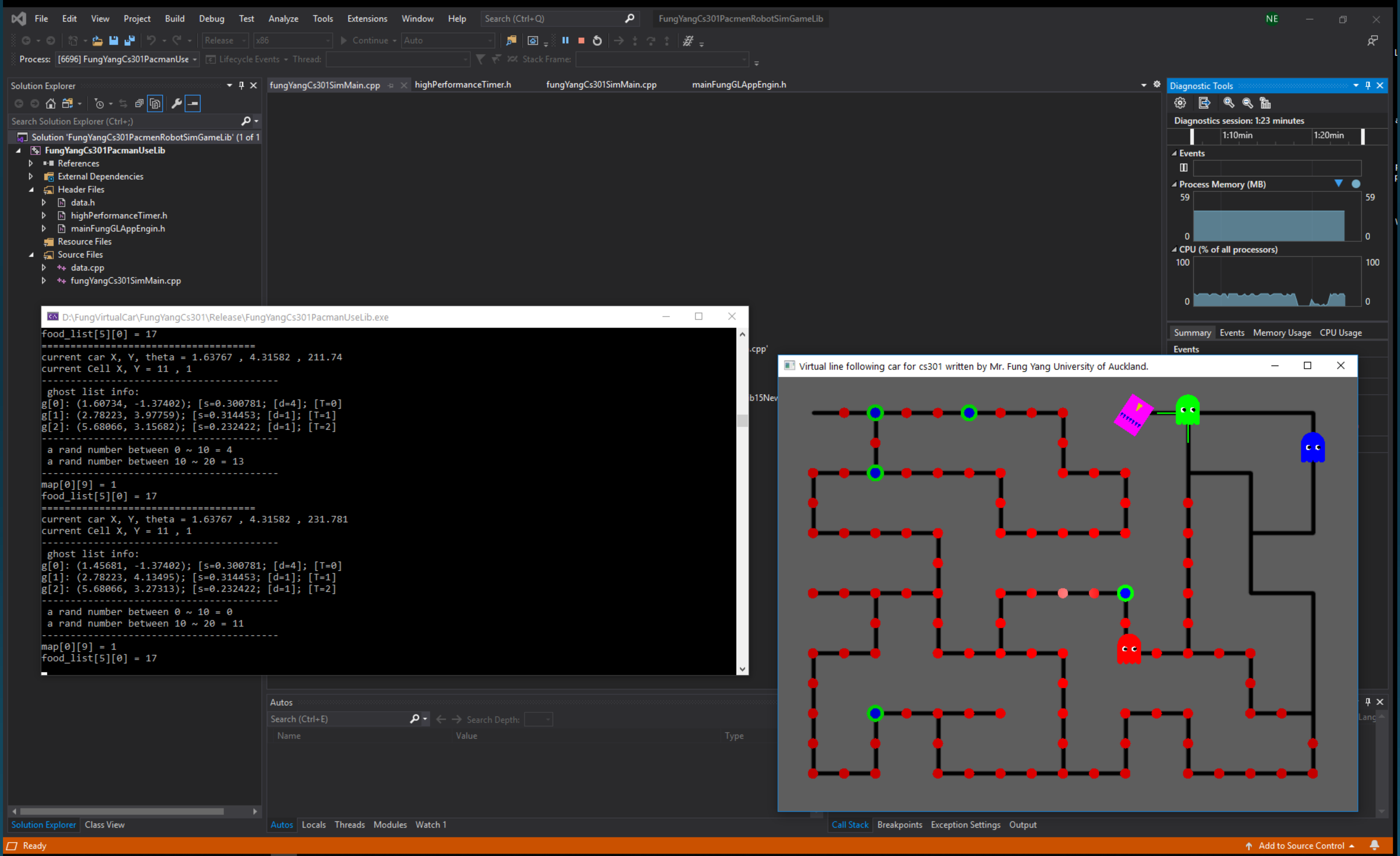
(5)

Then the program will run for you.

Make sure the project is selected in visual studio **before** you compile

```
1>------ Rebuild All started: Project: FungYangCs301PacmanUseLib, Configuration: Release Win32 ------
1>data.cpp
1>fungYangCs301SimMain.cpp
1>D:\FungVirtualCar\FungYangCs301\FungYangCs301PacmanUseLib\fungYangCs301SimMain.cpp(62,25): warnin
1>D:\FungVirtualCar\FungYangCs301\FungYangCs301PacmanUseLib\fungYangCs301SimMain.cpp(64,34): warnin
1>D:\FungVirtualCar\FungYangCs301\FungYangCs301PacmanUseLib\fungYangCs301SimMain.cpp(77,22): warning
1>D:\FungVirtualCar\FungYangCs301\FungYangCs301PacmanUseLib\fungYangCs301SimMain.cpp(113,21): warnin
1>Generating code
1>179 of 929 functions (19.3%) were compiled, the rest were copied from previous compilation.
1>  4 functions were new in current compilation
1>  77 functions had inline decision re-evaluated but remain unchanged
1>Finished generating code
1>FungYangCs301PacmenRobotSimGameLib.lib(mazeGen.obj) : warning LNK4099: PDB 'FungYangCs301Pacmen
1>FungYangCs301PacmenRobotSimGameLib.lib(gameManager.obj) : warning LNK4099: PDB 'FungYangCs301Pa
1>FungYangCs301PacmenRobotSimGameLib.lib(highPerformanceTimer.obj) : warning LNK4099: PDB 'FungYang
1>FungYangCs301PacmenRobotSimGameLib.lib(mainFungGLAppEngin.obj) : warning LNK4099: PDB 'FungYang
1>FungYangCs301PacmenRobotSimGameLib.lib(Geometry.obj) : warning LNK4099: PDB 'FungYangCs301Pacme
1>FungYangCs301PacmenRobotSimGameLib.lib(my_GL_Draw_Funcs.obj) : warning LNK4099: PDB 'FungYangCs
1>FungYangCs301PacmenRobotSimGameLib.lib(AStarManager.obj) : warning LNK4099: PDB 'FungYangCs301Pa
1>FungYangCs301PacmenRobotSimGameLib.lib(texture.obj) : warning LNK4099: PDB 'FungYangCs301PacmenR
1>FungYangCs301PacmenRobotSimGameLib.lib(Lighting.obj) : warning LNK4099: PDB 'FungYangCs301Pacmen
1>FungYangCs301PacmenRobotSimGameLib.lib(Utilities.obj) : warning LNK4099: PDB 'FungYangCs301PacmenR
1>FungYangCs301PacmenRobotSimGameLib.lib(PathFinder.obj) : warning LNK4099: PDB 'FungYangCs301Pacme
1>FungYangCs301PacmanUseLib.vcxproj -> D:\FungVirtualCar\FungYangCs301\Release\FungYangCs301PacmanU
1>Done building project "FungYangCs301PacmanUseLib.vcxproj".
========== Rebuild All: 1 succeeded, 0 failed, 0 skipped ==========
```

To stop a simulation, close one of the windows



```
food_list[5][0] = 17
==================================
current car X, Y, theta = 1.63767 , 4.31582 , 211.74
current Cell X, Y = 11 , 1
----------------------------------------
 ghost list info:
g[0]: (1.60734, -1.37402); [s=0.300781; [d=4]; [T=0]
g[1]: (2.78223, 3.97759); [s=0.314453; [d=1]; [T=1]
g[2]: (5.68066, 3.15682); [s=0.232422; [d=1]; [T=2]
----------------------------------------
 a rand number between 0 ~ 10 = 4
 a rand number between 10 ~ 20 = 13
----------------------------------------
map[0][9] = 1
food_list[5][0] = 17
==================================
current car X, Y, theta = 1.63767 , 4.31582 , 231.781
current Cell X, Y = 11 , 1
----------------------------------------
 ghost list info:
g[0]: (1.45681, -1.37402); [s=0.300781; [d=4]; [T=0]
g[1]: (2.78223, 4.13495); [s=0.314453; [d=1]; [T=1]
g[2]: (5.68066, 3.27313); [s=0.232422; [d=1]; [T=2]
----------------------------------------
 a rand number between 0 ~ 10 = 0
 a rand number between 10 ~ 20 = 11
----------------------------------------
map[0][9] = 1
food_list[5][0] = 17
```

Accessing data

current car X position	currentCarPosCoord_X
current car Y position	currentCarPosCoord_Y
current car angle	currentCarAngle
current Cell X	coordToCellX(currentCarPosCoord_X)
current Cell Y	coordToCellX(currentCarPosCoord_Y)
number of ghosts	ghostInfoPackList.size()
Ghost i position X	ghostInfoPackList[i].coord_x
Ghost i position Y	ghostInfoPackList[i].coord_y
Ghost i speed	ghostInfoPackList[i].speed
Ghost i direction	ghostInfoPackList[i].direction
Ghost i type	ghostInfoPackList[i].ghostType
Path on the map (is this a line (1) or not (0)?)	map[ligne#(0 to 15)][column#(0 to 19)]
Coordinate of the food pellets	food_list[x coordinate (0 to 15)][y coordinate(0 to 19)]

The web link to the YouTube video is listed below:
https://www.youtube.com/watch?v=5TStc7MPi9U&t=1s