

# 1 AI-Course-Project

We have implemented 3 scenarios of agent based search environment where each agent has to collect 5 of its targets spawned randomly in the environment with a fixed location.

- Scenario 1: **Competition:** Here only public broadcast is allowed. Each agent may lie or broadcast the correct information about other agent's targets into a public channel that every agent can see. An iteration is over as soon as one agent collect all of its targets.
- Scenario 2: **Collaboration:** Here public and private broadcast is allowed. An agent can broadcast information publicly or specifically to an agents channel. Here we may or may not lie in the public channel and always tell the truth in private communication. Iteration is over when all the agents collects all the targets.
- Scenario 3: **Compassionate:** This is the same as Collaboration except no one will ever lie and the game is over when one agent collect all of its targets.

The Executables can be found in the `bin` directory

## 2 Setup

### 2.1 SDL2 is the main rendering library used

**SDL2image** add-on is also used to load add-ons

#### 2.1.1 On Windows

##### 1. Setup SDL2

- Download **SDL2** libs from [here](#)
- Specifically you need the Developmental libs.

- I am using Visual Studio right now and I downloaded `SDL2-devel-2.0.8-VC.zip`
- Extract it somewhere, I extracted it to `C:\Dev\SDL2-2.0.8`
- Create a new file named `sdl2-config.cmake` in there with the following content

```
set(SDL2_INCLUDE_DIRS "${CMAKE_CURRENT_LIST_DIR}/include")
# Support both 32 and 64 bit builds
if (${CMAKE_SIZEOF_VOID_P} MATCHES 8)
    set(SDL2_LIBRARIES "${CMAKE_CURRENT_LIST_DIR}/lib/x64/SDL2.lib;${CMAKE_CURRENT_LIST_DIR}/lib/x64/SDL2main.lib")
else ()
    set(SDL2_LIBRARIES "${CMAKE_CURRENT_LIST_DIR}/lib/x86/SDL2.lib;${CMAKE_CURRENT_LIST_DIR}/lib/x86/SDL2main.lib")
endif ()
string(STRIP "${SDL2_LIBRARIES}" SDL2_LIBRARIES)
```

- Now open control panel to `Edit environment variables` , you can just search for this term too
- Add a new **System variable** named `SDL2_DIR` and set the value to the directory where you extracted the lib for me it will be `C:\Dev\SDL2-2.0.8` .
- I will be using **x64** libs so I will also edit the `Path` variables for the user and add `C:\Dev\SDL2-2.0.8\lib\x64` folder to path.

## 2. Setup SDL2image

- Download **SDL2image** libs from [here](#)
- Specifically you need the Developmental libs.
- I am using Visual Studio right now and I downloaded `SDL2_image-devel-2.0.3-VC.zip`
- Extract it somewhere, I extracted it to `C:\Dev\SDL2_image-2.0.3`
- Create a new file named `sdl2_image-config.cmake` in there with the following content.

```
#+BEGIN_SRC cmake
set(SDL2_IMAGE_INCLUDE_DIRS "${CMAKE_CURRENT_LIST_DIR}/include")
# Support both 32 and 64 bit builds
if (${CMAKE_SIZEOF_VOID_P} MATCHES 8)
```

```

    set(SDL2_IMAGE_LIBRARIES "${CMAKE_CURRENT_LIST_DIR}/lib/x64/SDL2_image.
        lib")
else ()
    set(SDL2_IMAGE_LIBRARIES "${CMAKE_CURRENT_LIST_DIR}/lib/x86/SDL2_image.
        lib")
endif ()
string(STRIP "${SDL2_IMAGE_LIBRARIES}" SDL2_IMAGE_LIBRARIES)

```

- Now open control panel to **Edit environment variables**
- Add a new **System variable** named **SDL2\_image\_DIR** and set the value to the directory where you extracted the lib for me it will be **C:\Dev\SDL2\_image-2.0.3**
- I will be using **x64** libs so I will also edit the **Path** variables for the user and add **C:\Dev\SDL2\_image-2.0.3\lib\x64** folder to path.
- **Restart the system so the environment variables are set and the path re-read.**
- Now you can build the code with CMake

### 2.1.2 On Linux(Ubuntu)

Download both **SDL2** and **SDL2\_image** like this

```

sudo apt install libsdl2-dev
sudo apt install libsdl2-image-dev

```

The **CMake** file I have can automatically find and build these libs

## 2.2 Qt is used to manage front-end and back-end threads and their communication

### 2.2.1 On Windows

- Download qt libs from here <https://www.qt.io/download>
- Add the library directory to path as **QT\_DIR**, for me it is **C:\Dev\Qt\5.9.1**
- Also add the library binaries to **Path** for me it is **C:\Dev\Qt\5.9.1\msvc2015\_64\bin**

### 2.2.2 On Linux(Ubuntu)

- Install necessary Qt modules from an updated **PPA** like this one: <https://launchpad.net/~beineri/+archive/ubuntu/opt-qt592-trusty>
- Install necessary modules with these commands after adding the **PPA**

```
sudo apt install qt59base  
sudo apt install qt59declarative
```

### 2.3 Now you should be able to use CMake to build the source code