Using Azolla

- Create your own robot class by deriving it from Cazolla class.
- Put consructor.
- Put `az_sim_fn()` function. This function is pure virtual and will be exceuted iteratively. So no need to give things like `while(true)` to keep the simulation running.

Example: Main File

```
#include "azolla/azolla.H"

class CMyRobot : public CAzolla
{
public:
    CMyRobot(CSimulationWindow *w)
    :CAzolla(w)
    {
        // Initialization
    }

    virtual void az_sim_fn()
    {
        // WRITE YOUR CODE HERE

        az_step();
    }
};

////////////////////////////////////////////////////////////////////////////
// MAIN
////////////////////////////////////////////////////////////////////////////

int main()
{
    CSimulationWindow win(600, 600,"azolla1");
    win.color(FL_GRAY);
    win.end();
    win.show();

    CMyRobot robot(&win);

    return(Fl::run());
}
```

Configuration File

- Check file `configure.H`
- All robot and simulation parameters are in configure.H file.

Example: configure.h

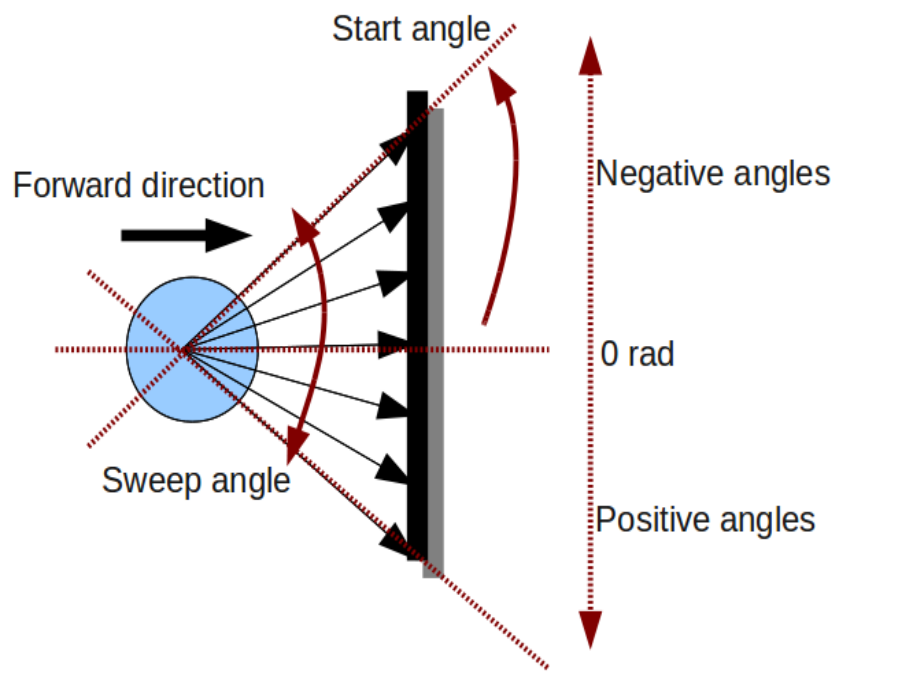
```
    /// Scale, 1 meter equals to how many pixels?
    const double    SCALE_FACTOR = 192.0;


    /// Simulation window refresh rate (in seconds)
    const double    SCREEN_TIMEOUT = 0.10;


    /// Diameter size of the robot (in meters)
    const double    ROBOT_SIZE   = 0.1 * SCALE_FACTOR;
    const double    ROBOT_SIZE_2 = ROBOT_SIZE/2;


    /// A grid area for probability of occupancy
    const int       GRID_MAP_H = 500;
    const int       GRID_MAP_W = 500;


    /// Laser sensor properties
    const double    LIDAR_VAR_2       = 0.1 * SCALE_FACTOR;   // Standard deviation (meters)
    const double    LIDAR_VAR         = sqrt(LIDAR_VAR_2);    // Variance
    const double    LIDAR_START_ANGLE = -M_PI_2;              // Starting angle of the first ray
    const double    LIDAR_SWEEP_ANGLE = M_PI;                 // Angle of area swept by the LIDAR
    const int       LIDAR_RAYS        = 100;                  // Number of rays
    const double    LIDAR_MAX         = 1.0 * SCALE_FACTOR;   // Maximum distance (meters)


    /// Position based odometry
    const int       CHOSEN_PARTICLE = 0;                    // Select one particle
    const int       ODOM_SAMPLES    = 1000;
    const double    KT              = 0.10 / SCALE_FACTOR;  // Translation error parameter
    const double    KD              = 0.10 / SCALE_FACTOR;  // Drift error parameter
    const double    KR              = 0.08 / SCALE_FACTOR;  // Rotation error parameter
    const double    MT              = 0.0;                  // Translation error mean
    const double    MD              = 0.0;                  // Drift error mean
    const double    MR              = 0.0;                  // Rotation error mean
```

Some Notices on configure.H

- Make sure that grid map (`GRID_MAP_H x GRID_MAP_W`) is bigger than your map size. Youe map file should be a PNG file. You should check its size first.

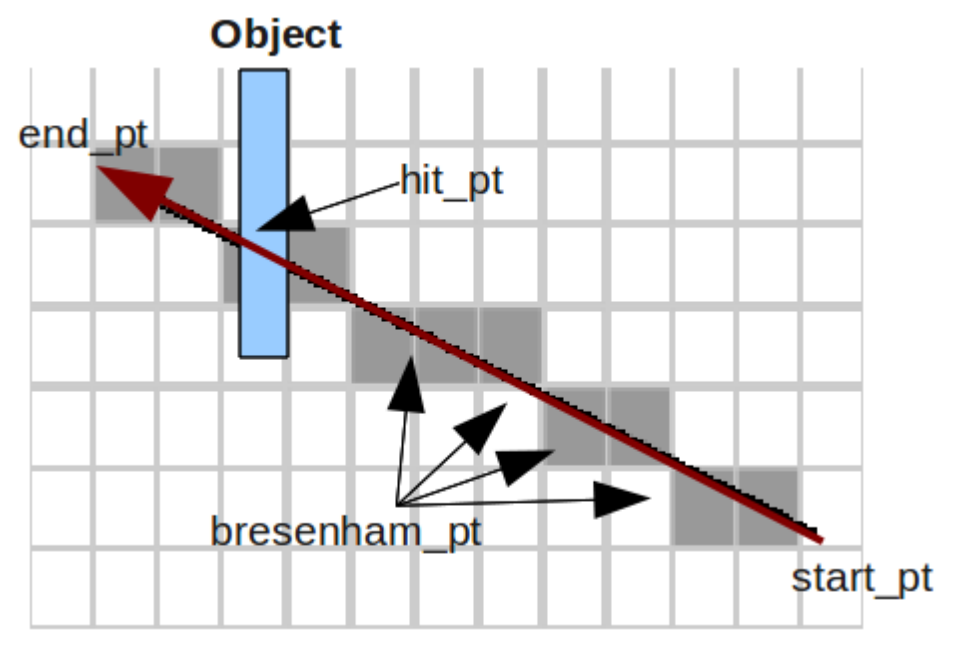
Laser Sensor

- Currently only laser sensor (LIDAR) is supported.
- To understand what `LIDAR_START_ANGLE` and `LIDAR_SWEEP_ANGLE` is, please refer the picture below.



Sensor Readings and Grid Mapping

- Please refer this picture below.
- This picture is describing what is happening in CSensor class.



Sensor Readings and Grid Mapping

- The occupancy probability of the grid map is updated per 1 pixel, which means your grid map resoluition is: your map real size divided by your PNG map size.
- Let say your real map size where the robot navigates is 10 m x 10 m, you scaled it down to a PNG file of 500 pixels x 500 pixels. It means your grid map resolution is 0.02 m x 0.02m or 2 cm x 2 cm.
- If you need more detail grid map, make the map file (the PNG file) larger using your favorite image editor software.

Further Information

- Kinematic model
  - See `az_step_fn() : void`
- Odometry error model
  - See `odometry_thead_worker() : void`
- Equations to compute occupancy probability of the grid map.
  - See `calc_occupancy(bool with_error) : void`

Dependencies

- FLTK library for GUI (mine is version 1.0.11)
- Boost libaries (mine is version 1.42)

- I will come with further updates later.
- Contact me at auralius@gmx.com if you are interested.