2D Predator Prey Simulation by Aayush Shrestha

Homework3

Generated by Doxygen 1.8.15

Chapter 1

Bug List

File cell.cpp

No known bugs.

File cell.hpp

No known bugs.

File controller.cpp

No known bugs.

File controller.hpp

No known bugs.

File coyote.cpp

No known bugs.

File grid.cpp

No known bugs.

File grid.hpp

No known bugs.

File organism.cpp

No known bugs.

File organism.hpp

No known bugs.

File organismcreator.cpp

No known bugs.

File organismcreator.hpp

No known bugs.

File organismimpl.cpp

No known bugs.

File organismimpl.hpp

No known bugs.

File roadrunner.cpp

No known bugs.

File roadrunner.hpp

No known bugs.

File simulation.cpp

No known bugs.

File simulation.hpp

No known bugs.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 5

Class Documentation

5.1 Cell Class Reference

Public Member Functions

- Cell ()
  default constructor
- Cell (int xVal, int yVal)
- std::string str () const

Public Attributes

- int **x**
- int **y**

Friends

- bool operator== (const Cell &cell1, const Cell &cell2)

5.1.1 Constructor & Destructor Documentation

5.1.1.1 Cell() [1/2]

```
Cell::Cell ( )
```

default constructor

Brief: Default Constructor Description: empty constructor to prevent default compiler generated constructur

5.1.1.2 Cell() [2/2]

```
Cell::Cell (
            int xVal,
            int yVal )
```

Brief: Constructor with x and y co-ordinates provided Description : Initializes a cell with x and y co-ordinates

Parameters

	int xVal - x co-ordinate
	int yVal - y co-ordinate

5.1.2 Member Function Documentation

5.1.2.1 str()

```
std::string Cell::str ( ) const
```

Brief: Utility function to return string notation of cell Description : Returns a string representation of the cell with its x and y co-ordinates

Returns

: string representation of cell

5.1.3 Friends And Related Function Documentation

5.1.3.1 operator==

```
bool operator== (
            const Cell & cell1,
            const Cell & cell2 )  [friend]
```

Brief: Overladed equality operator Description : Utility overload to compare to cells

Parameters

	const Cell& a - first cell to be compared
	const Cell& b - second cell to be compared

Returns

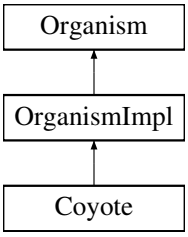
: true if equal

The documentation for this class was generated from the following files:

- cell.hpp
- cell.cpp

5.2 Coyote Class Reference

Inheritance diagram for Coyote:



Public Member Functions

- Coyote (const Cell &cell)
- void incrementTimeWithoutEating ()
- bool **isStarving** () const
- void eat ()

Additional Inherited Members

5.2.1 Constructor & Destructor Documentation

5.2.1.1 Coyote()

```
Coyote::Coyote (
            const Cell & cell )
```

Brief : Coyote Constructor Description : Creates a Coyote object using OrganismImplementor and sets its properties as per the spec

5.2.2 Member Function Documentation

5.2.2.1 eat()

```
void Coyote::eat ( )
```

Brief : update time without eating if just ate something

5.2.2.2 incrementTimeWithoutEating()

```
void Coyote::incrementTimeWithoutEating ( )
```

Brief : Increment time without eating for each step coyote goes without eating a roadrunner

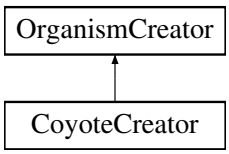
The documentation for this class was generated from the following files:

- coyotee.hpp
- coyote.cpp

5.3 CoyoteCreator Class Reference

```
#include <coyotee.hpp>
```

Inheritance diagram for CoyoteCreator:



Public Member Functions

- Coyote * create (const Cell &cell)

Additional Inherited Members

5.3.1 Detailed Description

Brief : Factory to create Coyotes

5.3.2 Member Function Documentation

5.3.2.1 create()

```
Coyote * CoyoteCreator::create (
            const Cell & cell )  [virtual]
```

Brief : Coyote factory

Parameters

	const Cell& cell - location of coyote

Returns

: pointer to coyote.

Implements OrganismCreator.

The documentation for this class was generated from the following files:

- coyotee.hpp
- coyote.cpp

5.4 Grid Class Reference

Public Member Functions

- Grid (int x, int y)
- void display () const
- Cell getRandomEmptyCellInGrid () const
- void clearCell (const Cell &cell)
- void setCellValue (const Cell &cell, Organism *organism)
- Organism * getCellValue (const Cell &cell) const
- Organism * getCellValue (int x, int y) const
- CellVector getAdjacentCells (const Cell &cell) const
- CellVector getRandomizedAdjacentCells (const Cell &cell) const
- Cell getEmptyCell (const CellVector &adjacentCells, bool &found) const
- Cell getRandomMatchingCell (const CellVector &cellSet, OrganismComparator comparator, bool &found) const

5.4.1 Constructor & Destructor Documentation

5.4.1.1 Grid()

```
Grid::Grid (
            int x,
            int y )
```

Description : Constructor with x and y ranges provided initalizes the vector with xMax and yMax which are the x and y dimensions of the grid the user wants to run the simulation for sets the set as the vector of all possible cells in the vector intializes 2D array of organisms.

Returns

: nothing

5.4.2 Member Function Documentation

5.4.2.1 clearCell()

```
void Grid::clearCell (
            const Cell & cell )
```

Description : Clears contents of provided cell

Parameters

	const Cell& cell - cell to delete from the grid

Returns

: nothing

5.4.2.2 display()

```
void Grid::display ( ) const
```

Description : Utility function to display the grid

5.4.2.3 getAdjacentCells()

```
CellVector Grid::getAdjacentCells (
            const Cell & cell ) const
```

Description : Gets adjascent cells that the agent could move to

Returns

: vector of possible adjascent cells

5.4.2.4 getCellValue() [1/2]

```
Organism * Grid::getCellValue (
            const Cell & cell ) const
```

Description : utility function to get x and y location of organism in the array using the cell vector

Returns

: reference to organism contained

5.4.2.5 getCellValue() [2/2]

```
Organism * Grid::getCellValue (
            int x,
            int y ) const
```

Description : Get organism contained at any given cell @params: int x, int y - location of the organism in the array

Returns

: reference to the organism

5.4.2.6 getEmptyCell()

```
Cell Grid::getEmptyCell (
            const CellVector & adjacentCells,
            bool & found ) const
```

Description : Finds empty adjascent cell

Parameters

	const CellVector& adjacentCells - vector of adjascent cells
	bool& found - true if found empty adjascent cell that can be moved to

5.4.2.7 getRandomEmptyCellInGrid()

```
Cell Grid::getRandomEmptyCellInGrid ( ) const
```

Description : Utility function used to load organisms in the grid

Returns

: empty cell in the grid for an organism to occupy

5.4.2.8 getRandomizedAdjacentCells()

```
CellVector Grid::getRandomizedAdjacentCells (
            const Cell & cell ) const
```

Description : Utility function to select random adjascent cell to move to @params : const Cell& cell - cell to check adjascents for

Returns

: adjascent cell to move to

5.4.2.9 getRandomMatchingCell()

```
Cell Grid::getRandomMatchingCell (
            const CellVector & cellSet,
            OrganismComparator comparator,
            bool & found ) const
```

Description : Utility function to find matching cell @params : const CellVector& cellSet - vector of cells to scan @params : OrganismComparator comparator - organism to find @params : bool& found - true if found else false

Returns

: random matching cell

5.4.2.10 setCellValue()

```
void Grid::setCellValue (
            const Cell & cell,
            Organism * organism )
```

Description : Sets cell value with organism provided in the grid

Parameters

:const	Cell& cell - reference to cell
	Organism∗ organism - organism to be loaded

Returns

: nothing

The documentation for this class was generated from the following files:

- grid.hpp
- grid.cpp

5.5 GridController Class Reference

Public Member Functions

- GridController ()
- GridController (Grid ∗gridPtr)
- void createRoadrunners (int count)
- void createCoyotes (int count)
- void **display** () const
- void writeDataToCsv (std::ostream &csvFileStream, int iteration) const
- void step ()

5.5.1 Constructor & Destructor Documentation

5.5.1.1 GridController() [1/2]

```
GridController::GridController ( )
```

Brief : Default Constructor

5.5.1.2 GridController() [2/2]

```
GridController::GridController (
            Grid * gridPtr )
```

Brief : Constructor with pointer to a grid object Description : initializes the controller with a grid object pointer also initializes coyote and roadrunner factories.

5.5.2 Member Function Documentation

5.5.2.1 createCoyotes()

```
void GridController::createCoyotes (
            int count )
```

Brief : creates Coyotes

Parameters

	int count - number of Coyotes to create

5.5.2.2 createRoadrunners()

```
void GridController::createRoadrunners (
            int count )
```

Brief : creates road runners

Parameters

	int count - number of road runners to create

5.5.2.3 step()

```
void GridController::step ( )
```

Brief : Moves road runners and coyotes.

5.5.2.4 writeDataToCsv()

```
void GridController::writeDataToCsv (
            std::ostream & csvFileStream,
            int iteration ) const
```

Brief : CSV writer for data visualization

Parameters

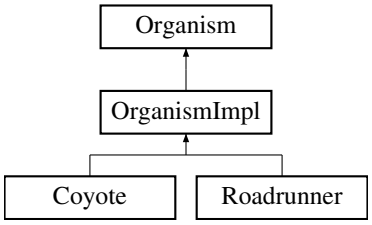
	ostream& csvFileStream - stream to write to
int	iterations - number of iterations of simulation

The documentation for this class was generated from the following files:

- controller.hpp
- controller.cpp

5.6 Organism Class Reference

Inheritance diagram for Organism:



Public Member Functions

- Organism (int requiredSurvivalTime, char organismSymbol, const Cell &cell)
- virtual ~Organism ()=0

  destructor
- virtual void **move** (const Cell &newCell)=0
- virtual Cell **getCell** () const =0
- virtual void **setCell** (const Cell &cell)=0
- virtual int **getTimeSurvived** () const =0
- virtual void **incrementTimeSurvived** ()=0
- virtual bool **readyToBreed** () const =0

Public Attributes

- const char **symbol**

Protected Attributes

- const int **required_survival_time_for_breeding**
- int **timeSurvived**
- Cell **myCell**

Friends

- char getSymbolOrEmptyChar (Organism ∗organism)

5.6.1 Constructor & Destructor Documentation

5.6.1.1 Organism()

```
Organism::Organism (
            int requiredSurvivalTime,
            char organismSymbol,
            const Cell & cell )
```

Description : Constructor to initialize an organism

Parameters

	int requiredSurvivalTime - time before breed
	char organismSymbol - type of organism
	const Cell& cell - cell location

5.6.1.2 ∼Organism()

```
Organism::∼Organism ( )  [pure virtual]
```

destructor

Default Destructor

5.6.2 Friends And Related Function Documentation

5.6.2.1 getSymbolOrEmptyChar

```
char getSymbolOrEmptyChar (
            Organism * organism )  [friend]
```

Description : Get what organism is residing in cell

Parameters

	Organism* organism - organism pointer to cell

Returns

: Symbol of organism if present else empty space symbol

The documentation for this class was generated from the following files:

- organism.hpp
- organism.cpp

5.7 OrganismComparator Class Reference

```
#include <organism.hpp>
```

Public Member Functions

- OrganismComparator (char value)
- OrganismComparator (Organism *organism)
- bool compare (Organism *organism)

5.7.1 Detailed Description

Utility class to compare organisms

5.7.2 Constructor & Destructor Documentation

5.7.2.1 OrganismComparator() [1/2]

```
OrganismComparator::OrganismComparator (
            char value )
```

Brief : compare cells to see what organism resides there

Parameters

	char value - value to be compared

5.7.2.2 OrganismComparator() [2/2]

```
OrganismComparator::OrganismComparator (
            Organism * organism )
```

Brief : compare cells to see what organism resides there Description : Uses the organism object to compare value

Parameters

	Organism* organism - pointer to organism in cell

5.7.3 Member Function Documentation

5.7.3.1 compare()

```
bool OrganismComparator::compare (
            Organism * organism )
```

Description : Overloaded operator to compare organism values

Returns

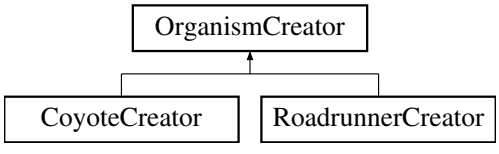
: true if the match else false

The documentation for this class was generated from the following files:

- organism.hpp
- organism.cpp

5.8 OrganismCreator Class Reference

Inheritance diagram for OrganismCreator:



Public Member Functions

- OrganismCreator ()
- virtual Organism * **create** (const Cell &cell)=0
- OrganismVector **getOrganisms** ()
- void remove (const Cell &cell)
- int **count** () const

Protected Member Functions

- void addOrganism (Organism *organismPtr)
- int findIndex (const Cell &cell)

5.8.1 Constructor & Destructor Documentation

5.8.1.1 OrganismCreator()

```
OrganismCreator::OrganismCreator ( )
```

Brief : Default Constructor

5.8.2 Member Function Documentation

5.8.2.1 addOrganism()

```
void OrganismCreator::addOrganism (
            Organism * organismPtr )  [protected]
```

Brief : Add organism to our organisms vector

Parameters

	Organism* organismPtr organism to be added

5.8.2.2 findIndex()

```
int OrganismCreator::findIndex (
            const Cell & cell )  [protected]
```

Brief : gets index of organism we're trying to find

Parameters

	const Cell& cell -> cell where the org resides

Returns

: index of cell

5.8.2.3 remove()

```
void OrganismCreator::remove (
            const Cell & cell )
```

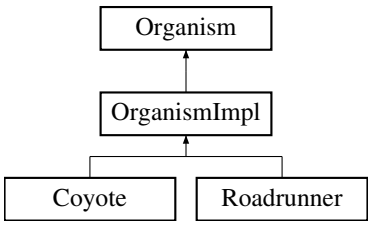
Brief : Removes the organism from the vector

The documentation for this class was generated from the following files:

- organismcreator.hpp
- organismcreator.cpp

5.9 OrganismImpl Class Reference

Inheritance diagram for OrganismImpl:



Public Member Functions

- OrganismImpl (int requiredSurvivalTime, char organismSymbol, const Cell &cell)
- virtual void move (const Cell &newCell)
- Cell **getCell** () const
- void **setCell** (const Cell &cell)
- int **getTimeSurvived** () const
- void **incrementTimeSurvived** ()
- bool readyToBreed () const
- void **die** ()

Additional Inherited Members

5.9.1 Constructor & Destructor Documentation

5.9.1.1 OrganismImpl()

```
OrganismImpl::OrganismImpl (
            int requiredSurvivalTime,
            char organismSymbol,
            const Cell & cell )
```

Brief : Constructor that creates an organism with time for survival, type and location as parameters

Parameters

	int requiredSurvivalTime - survival time for organsim
	char organismSymbol - type of organism
	const Cell& cell - location of cell

5.9.2 Member Function Documentation

5.9.2.1 move()

```
void OrganismImpl::move (
            const Cell & newCell )  [virtual]
```

Brief : utility function to make agent move cells

Parameters

	const Cell& newCell - cell to move to

Implements Organism.

5.9.2.2 readyToBreed()

```
bool OrganismImpl::readyToBreed ( ) const  [virtual]
```

Brief: Determines if it is time for the agent to breed yet

Returns

: true if conditions are met

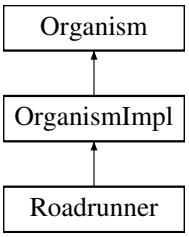
Implements Organism.

The documentation for this class was generated from the following files:

- organismimpl.hpp
- organismimpl.cpp

5.10 Roadrunner Class Reference

Inheritance diagram for Roadrunner:



Public Member Functions

- Roadrunner (const Cell &cell)

Additional Inherited Members

5.10.1 Constructor & Destructor Documentation

5.10.1.1 Roadrunner()

```
Roadrunner::Roadrunner (
            const Cell & cell )
```

Brief : Initializes a road runner with its cell location

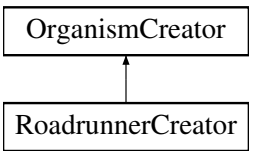
The documentation for this class was generated from the following files:

- roadrunner.hpp
- roadrunner.cpp

5.11 RoadrunnerCreator Class Reference

`#include <roadrunner.hpp>`

Inheritance diagram for RoadrunnerCreator:



Public Member Functions

- Roadrunner * create (const Cell &cell)

Additional Inherited Members

5.11.1 Detailed Description

Brief : Roadrunner factory

5.11.2 Member Function Documentation

5.11.2.1 create()

```
Roadrunner * RoadrunnerCreator::create (
            const Cell & cell )  [virtual]
```

Brief : Creates a new road runner at cell specified.

Returns

: a reference to a road runer returned by the factory

Implements OrganismCreator.

The documentation for this class was generated from the following files:

- roadrunner.hpp
- roadrunner.cpp

5.12 Simulation Class Reference

Public Member Functions

- void getBoardSize ()
- void getRoadrunners ()
- void getCoyotes ()
- void getSteps ()
- void runSimulation ()

Static Public Member Functions

- static Simulation * **getInstance** ()

Protected Attributes

- std::ofstream **out_stream**
- std::string **temp**

5.12.1 Member Function Documentation

5.12.1.1 getBoardSize()

```
void Simulation::getBoardSize ( )
```

Brief : Get board size and validate if valid range.

5.12.1.2 getCoyotes()

```
void Simulation::getCoyotes ( )
```

Description : Get number of Coyotes and validate if in range

5.12.1.3 getRoadrunners()

```
void Simulation::getRoadrunners ( )
```

Description : Get number of roadrunners and validate if in range

5.12.1.4 getSteps()

```
void Simulation::getSteps ( )
```

Brief : Get number of steps user wants to run the simulation for

5.12.1.5 runSimulation()

```
void Simulation::runSimulation ( )
```

Brief : Collect data and start the simulation Description : Collects all necessary information instantiates the board and the agents and begins running simulations for steps specified. if all agents die -> simulation is stopped if board is filled with roadrunners -> simulation is stopped. @output - writes information about number of agents alive at each step to a csv.

- this csv is then used by graph.py to plot the information
- in order to make the graph use the make graph command

The documentation for this class was generated from the following files:

- simulation.hpp
- simulation.cpp

Chapter 6

File Documentation

6.1 cell.cpp File Reference

Implementation of the Cell Class.

```
#include "cell.hpp"
```

Functions

- bool operator== (const Cell &a, const Cell &b)

6.1.1 Detailed Description

Implementation of the Cell Class.

This file contains the implementation for all the functions described in Cell.hpp

Author

Aayush B Shrestha

Bug No known bugs.

6.1.2 Function Documentation

6.1.2.1 operator==()

```
bool operator== (
            const Cell & a,
            const Cell & b )
```

Brief: Overladed equality operator Description : Utility overload to compare to cells

Parameters

	const Cell& a - first cell to be compared
	const Cell& b - second cell to be compared

Returns

: true if equal

6.2 cell.hpp File Reference

Header of the cell class.

```
#include <string>
#include <vector>
```

Classes

- class Cell

Typedefs

- typedef std::vector< Cell > CellVector
  vector to hold cells in the simulation

6.2.1 Detailed Description

Header of the cell class.

Author

Aayush B Shrestha

Bug No known bugs.

6.3 controller.cpp File Reference

Implementation for the GridController Class.

```
#include "controller.hpp"
```

6.3.1 Detailed Description

Implementation for the GridController Class.

Description : Contains controls for all elements of the simulation

Author

Aayush B Shrestha

Bug No known bugs.

6.4 controller.hpp File Reference

Header for the GridController Class.

```
#include "roadrunner.hpp"
#include "coyotee.hpp"
#include "organismcreator.hpp"
#include "cell.hpp"
#include "grid.hpp"
#include "organism.hpp"
#include <fstream>
#include <iostream>
```

Classes

- class GridController

6.4.1 Detailed Description

Header for the GridController Class.

Author

Aayush B Shrestha

Bug No known bugs.

6.5 coyote.cpp File Reference

Implementation of the Coyote class which inherits from Organismimpl.

```
#include "coyotee.hpp"
```

6.5.1 Detailed Description

Implementation of the Coyote class which inherits from Organismimpl.

Author

Aayush B Shrestha

Bug No known bugs.

6.6 grid.cpp File Reference

Implementation for the Grid Class.

```
#include "grid.hpp"
```

Functions

- CellVector getCellSet (int xMax, int yMax)
- Organism *** get2dOrganismArray (int xMax, int yMax)

Variables

- const int **POSSIBLE_ADJACENT_CELL_COUNT** = 4

6.6.1 Detailed Description

Implementation for the Grid Class.

Author

Aayush B Shrestha

Bug No known bugs.

6.6.2 Function Documentation

6.6.2.1 get2dOrganismArray()

```
Organism*** get2dOrganismArray (
             int xMax,
             int yMax )
```

Description : Get 2D array of agents after creating it using the dimensions provided by user @params : int xMax - x value of grid provided by user @params : int yMax - y value provided by user

Returns

: 2D array of agents

6.6.2.2 getCellSet()

```
CellVector getCellSet (
             int xMax,
             int yMax )
```

Description : Get all x-y cords of matrix in cell vector @params : int xMax - x value of grid provided by user @params : int yMax - y value provided by user

Returns

: vector of all cells in the grid

6.7 grid.hpp File Reference

Header for the Grid Class.

```
#include "cell.hpp"
#include "organism.hpp"
#include <algorithm>
#include <cstdlib>
#include <iostream>
```

Classes

- class Grid

6.7.1 Detailed Description

Header for the Grid Class.

Author

Aayush B Shrestha

Bug No known bugs.

6.8 organism.cpp File Reference

Implementation for the Organism Class.

```
#include "organism.hpp"
```

Functions

- char getSymbolOrEmptyChar (Organism *organism)

6.8.1 Detailed Description

Implementation for the Organism Class.

Author

Aayush B Shrestha

Bug No known bugs.

6.8.2 Function Documentation

6.8.2.1 getSymbolOrEmptyChar()

```
char getSymbolOrEmptyChar (
            Organism * organism )
```

Description : Get what organism is residing in cell

Parameters

	Organism* organism - organism pointer to cell

Returns

: Symbol of organism if present else empty space symbol

6.9 organism.hpp File Reference

Header for the Organism Class.

```
#include "cell.hpp"
#include <vector>
```

Classes

- class Organism
- class OrganismComparator

Typedefs

- typedef std::vector< Organism * > OrganismVector
  vector of organism pointers

Variables

- const char **ROADRUNNER_CHAR** = 'o'
- const char **COYOTE_CHAR** = 'X'
- const char **EMPTY_SPACE_CHAR** = '-'

6.9.1 Detailed Description

Header for the Organism Class.

Author

Aayush B Shrestha

Bug No known bugs.

6.10 organismcreator.cpp File Reference

Implementation for the Organism Creator Class.

```
#include "organismcreator.hpp"
```

6.10.1 Detailed Description

Implementation for the Organism Creator Class.

Author

Aayush B Shrestha

Bug No known bugs.

6.11 organismcreator.hpp File Reference

Header for the Organism Creator Class.

```
#include "cell.hpp"
#include "grid.hpp"
#include "organism.hpp"
#include <iostream>
```

Classes

- class OrganismCreator

6.11.1 Detailed Description

Header for the Organism Creator Class.

Author

Aayush B Shrestha

Bug No known bugs.

6.12 organismimpl.cpp File Reference

Implementation for the Organism Implementor Class.

```
#include "organismimpl.hpp"
```

6.12.1 Detailed Description

Implementation for the Organism Implementor Class.

Author

Aayush B Shrestha

Bug No known bugs.

6.13 organismimpl.hpp File Reference

Header for the Organism Implementor Class.

```
#include "cell.hpp"
#include "organism.hpp"
```

Classes

- class OrganismImpl

6.13.1 Detailed Description

Header for the Organism Implementor Class.

Author

Aayush B Shrestha

Bug No known bugs.

6.14 roadrunner.cpp File Reference

Implementation for the Roadrunner Class.

```
#include "roadrunner.hpp"
```

6.14.1 Detailed Description

Implementation for the Roadrunner Class.

Author

Aayush B Shrestha

Bug No known bugs.

6.15 roadrunner.hpp File Reference

Header for the Roadrunner Class.

```
#include "cell.hpp"
#include "organismcreator.hpp"
#include "organismimpl.hpp"
```

Classes

- class Roadrunner
- class RoadrunnerCreator

6.15.1 Detailed Description

Header for the Roadrunner Class.

Author

Aayush B Shrestha

Bug No known bugs.

6.16 simulation.cpp File Reference

Implementation for the Simulation Class.

```
#include "simulation.hpp"
```

6.16.1 Detailed Description

Implementation for the Simulation Class.

Author

Aayush B Shrestha

Bug No known bugs.

6.17 simulation.hpp File Reference

Header for the Simulation Class.

```
#include <iostream>
#include "controller.hpp"
#include "grid.hpp"
#include <cstdlib>
#include <ctime>
#include <fstream>
#include <string>
```

Classes

- class Simulation

6.17.1 Detailed Description

Header for the Simulation Class.

Author

Aayush B Shrestha

Bug No known bugs.

Index