Darwin UML Diagram

Darwin

_height : const int
_width : const int
_size : const int
_turns : int
_turns_gone : int
_grid : std::vector<int>
_creatureInfo : std::unordered_map<int, creatureInfo>
_creaturesOnGrid : std::vector<int>

Darwin(int x, int y, int t);
void addCreature(Creature s, int r, int c);
void printGrid(std::ostream& w);
void simulate();
std::pair<front_t, int> front(int pos, dir_t dir);
std::pair<int, int> front_coordinate(int pos, dir_t dir);
bool in_bounds(std::pair<int, int> coord);
int coordToPosition(std::pair<int, int> coord);
void orderCreatureTurn();

Creature

_s : Species
_dir : dir_t
_pc : int

Creature(s : Species, dir : std::string);
dir_t getDirection();
bool execute(front_t, Creature&);
bool compareSpecies(const Creature &);
const std::string getSpeciesName() const;

1..1 — 0..*

0..1

1..1

Instruction

_n : int
_i : inst_t

0..* — 0..1

Species

_instructions : Vector<isntruction>
_name : std::string

void addInstruction(std::string)
inst_t executeControls(int, front_t)
Species(std::string)
bool operator==(const Species)
const std::string getName() const