#==============================#

| Yuan Wang, Yizhen Wang |

| ywang19, ywang22 |

| PA6 README |

#==============================#

#===================#

| Summary |

#===================#

**This is the Polymorphism program which is about PA6.**

1. The Standard Command Line input is : ./PA6 [#gridSize] [#doodlebugs] [#ants] [#time\_steps] [#seed] [#pause]
2. Note: if you do not enter any argument in the command line, just press enter button, and it will show the default value: gridSize -> 20, doodlebugs-> 5, ants-> 100, time\_steps-> 1000, seed-> 1,

**The whole program includes:**

1. grid.h (header class for grid.cpp):

Defined the class for grid which has organisms, size, organism\_at, put\_organism, get\_size, get\_occupied\_surrounding, get\_unoccupied\_surrounding.

1. grid.cpp

Defined the above functions in details:

Organism\* grid::organism\_at(int x, int y)

void grid::put\_organism(int x, int y, Organism\* o)

int grid::get\_size()

list<Organism\*>\* grid::get\_occupied\_surrounding(int x, int y)

list<Organism\*>\* grid::get\_unoccupied\_surrounding(int x, int y)

grid::grid(int size)

grid::grid()

1. organism.h (header class for organism.cpp)

Defined :

Classs of the Organism;

DoodleBug -> Organism;

Ant -> Organism;

GhostBug -> Organism;

And their individual functions.

Here is the Class of organism, and ant, doodlebug, ghostbug choose their individual methods from this class;

        virtual void move()           =  0;

virtual int  get\_x()          =  0;

virtual int  get\_y()          =  0;

virtual int  set\_x(int a)     =  0;

virtual int  set\_y(int a)     =  0;

virtual int  get\_breed()      =  0;

virtual int  set\_breed(int a) =  0;

virtual int  get\_satiation()  =  0;

virtual void set\_sat(int s)   =  0;

virtual char get\_rep()        =  0;

virtual int  get\_ID()         =  0;

1. organism.cpp: nothing special, just the method in details
2. simulation.cpp: main function:

int get\_random(int min, int max) : random numbers

list<DoodleBug\*>\* iterate\_doodles(list<DoodleBug\*>\* bugs, grid\* g)//iterate the doodlebug

list<Ant\*>\* iterate\_ants(list<Ant\*>\* ants, grid\* g) //iterate the ant

void doodle(grid\* g, int count, list<DoodleBug\*>\* db) // doodlebug

void ant(grid\* g, int count, list<Ant\*>\* ants) // ant

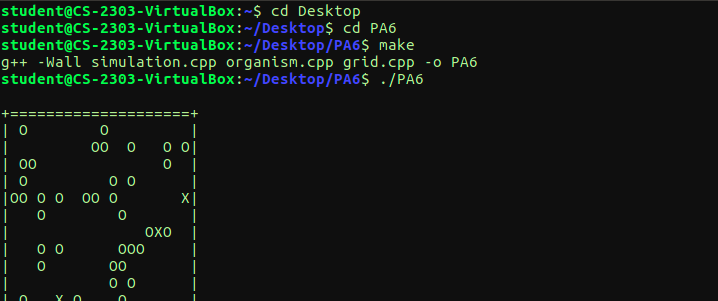
ostream& operator << (ostream& os, Organism& o) // << organism

ostream& operator << (ostream& os, grid& g) // << grid

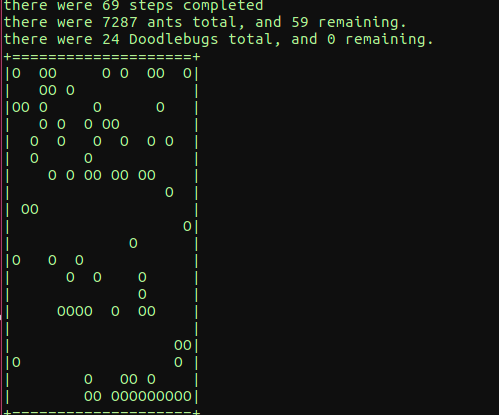
int main(int argc, char\*argv[]) // main function

Outputs:

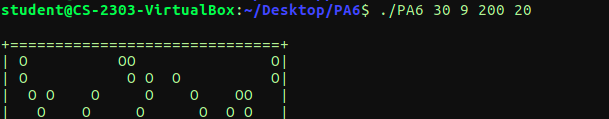
Here is how to run the default setting:



Here is the final output about the default setting:



Here is the self\_testing outpt:



Here is the final output:

