

Cover Page Info

Group Project: Doodlebugs Vs. Ants  
CS 162

May 2018

Kimberly Broz  
John Casey III  
Kevin Ohrlund  
Kelly Usenko  
Scott Wickersham

Division of labor:

# Critter

- Protected Data Members
  - RowPosition
  - ColumnPosition
- Logic
  - Constructor that takes in Row and Column Positions as Parameters:  
Critter(row, column)
  - ~Destructor
  - Virtual Void move()  
move(Grid &grid)
    - random number between 0-3 is used to move up, down, left, or right.
      - if cell is null or off the grid, don't move
      - else, move update row and column.
  - Virtual Void breed() (abstract function)
  - Void Increment Age()
    - stepsSurvived++

## Ant

- Logic
  - Overriding breed(Grid &grid)
    - If the steps survived are divisible by 3
      - If an adjacent cell is empty
        - antSpawned = false
        - until antSpawned
        - random number between 0-3 is used to assign cell
        - if the cell is empty, spawn a new ant in the cell
        - set antSpawned to true

## Doodlebug

- Data Members
  - stepsWithoutEating
- Logic
  - Overriding breed(Grid &grid)
    - if the steps survived are divisible by 8
      - if an adjacent cell is empty
        - doodlebugSpawned = false
        - until antSpawned
        - random number between 0-3 is used to assign cell
        - if the cell is empty, spawn a new doodlebug in cell
        - set doodlebugSpawned = true
  - Overriding move( Grid &grid)
    - If an ant is in an adjacent cell, delete ant in the cell. Move Doodlebug to the cell and update row and column

# Grid

- Data Members
  - Critter\*\*\*
  - int rows
  - int columns
- Logic
  - Constructor to initialize a grid Grid(row, columns)
  - ~Destructor -
  - Grid getGrid
  - int getRows
  - int getColumns
  - void printGrid

# Game

- Data Members
  - int currentTimeStep
- Logic
  - Constructor Game()
  - 
  - void run()
    - Create a 20X20 Grid
      - Embedded for loop that takes in number of rows, columns \*from user.
    - Randomly Place 5 Doodlebugs on Grid \*or take in # from user
      - for # of Doodlebugs
        - randomly generate a row and column
      - for # of Ants
        - randomly generate a row and column
    - Randomly Place 100 Ants on Grid \*or take in # from user
    - do
      - user input for number of timeSteps
      - for the number of timeSteps
        - Doodlebugs Move (eat)
        - Ants Move
        - Breed Ants
        - Breed Doodlebugs
        - For all Doodlebugs
          - if timeSteps without 'eating' is 3, delete the Doodlebug
        - Display the resulting grid at end of Turn
        - incrementAge of all critters
      - Display Menu asking if User would like to Continue or Quit

## Program Flow:

## Challenges:

## Reflection:

## Test Plan:

[illegible]