BATS - FINAL REPORT

School Name: Melrose Junior High **Team Number: Melrose High 99**

Project Title: Bats take flight **Area of Science: Zoology/ Animal behavior**

Computer Language: NetLogo

Team Members: Name – Grade – and email address:

Marilyn Lopez 8th Marilyn.lopez@melroseschools.org

Anjalina Sanchez 8th Anjalina.sanchez@melroseschools.org

Madison Garrett 8th Madison.garrett@melroseschools.org

Evelyn Woods 8th Evelyn.woods@melroseschools.org

Executive Summary:

Our team has recently visited Carlsbad Caverns and watched the bats fly in and out of the cave. This made us interested in how they are able to fly from inside the cave and all the way out despite the many obstacles. Our teams noticed that they use a spiral pattern while flying out and that they then fly on their separate ways in order to hunt.

We wanted to show how the bats flock out of the cave to get food for themselves and for their young one, so that is what we made them do in the NetLogo program they are in the back of the cave bundled together and then when the GO button is pressed they skitter their way out of the cave to get the energy that they need, in the form of food and water. We wanted to show how the bats fly because they don't fly like a bird, they skitter around when they are flying. When they leave their cave they usually fly out when the sunset and come back before the sun rises again.

Problem Statement:

We have researched bat populations and behavior with how they group together and split into different bands. We started learning about the NETLOGO 3D program at the beginning of the year, we realized that we needed help on how to make it work properly, so Mr. Daugherty helped us out a lot and now we have a pretty good idea on how to use it better and properly without messing something up. We looked for help with the 3D model from individuals we met at the Kickoff.

Methods Used:

We talked to park officials at Carlsbad to get more information. Also by reading books and going online to find information on the topic. Also some relatives and friends know more about bats, especially since quite a few fly around Melrose.

We are making our model using the NETLOGO program. We have studied how the pre-made "flocking" procedure worked and we used the 3D version to make our model more interesting for this assignment and attractive.

Some variables that our model includes are: the number of bats, the amount of obstacles, how tightly they can turn and maneuver, and the amount of time they have to go back and forth from their feeding areas. Our model will have the size of the cavern opening stay as a fixed variable.

Verification and Validation:

We made a model of bats flying out of a cave, and yes the bats flew out of the cave. We made them spiral like they do in the program, but it's not as uniform as we hoped. However, they DO exit the cave, and they fly with a skittering action like we wanted.

Results and Conclusions:

Our team was expecting to find out how the bats fly out of the cave to find water and food for themselves and their young ones and use this to make a model of their behavior and so we did with the help of the scientists in Carlsbad. We have used the 3D modeling program to show how they fly as if they separate to forage with groups of other bats.

Another main result we hope to achieve is learning more about computer modelling and how to use the NETLOGO programs. We also tried NetLogo 3D this year, and found out how much of a change that is!

Products of Our Work:

On our NetLogo program we made bats fly out of a cave simulating Carlsbad Caverns, one famous area where many bats hibernate, find food and water, and reproduce.

Significant Achievements:

Our major achievement is that we learned more programming skills.

Using premade procedures from others.

Working in the 3D environment.

Citations:

1. Field Trip: Carlsbad Caverns, Carlsbad New Mexico 88220 Fall 2016.

2. Bats- Wikipedia.com. Website showing how bats fly.

3. Wdfw.wa.gov/living/bats.html. Website showing bat ecology.

4. Batworlds.com. Website showing how bats live.

5. Book- "The Secret Lives of Bats: My Adventures with the World's Most Misunderstood Mammals". Author: Merlin D. Tuttle Published: 2015 by Houghton Mifflin Harcourt, Boston 2015

Acknowledgements:

We would like to thank the park officials at Carlsbad, Parents, teachers like Mrs. Montague and Mr. Daugherty.

Also, we would like to recognize the author of the Flocking code that we used on the NetLogo Models, both at the 2D and 3D levels.

BATS SOURCE CODE:

```
turtles-own [              ;; This is to enable the 'Flocking' procedure to work.
        flockmates         ;; agentset of nearby turtles
        nearest-neighbor] ;; closest one of our flockmates
breed [bats bat]           ;; Our main agents are 'bats'


to setup            ;;  THIS PROCEDURE SETS UP OUR PROGRAM
               clear-all
                ask patches [set pcolor 5]  ;; color of the background
                cave
                clouds
                crt population          ;; the number of bats
                 [set breed bats
                  set color 0 + random 3
                  set shape "bat"
                  set size 3
                 setxy  162 37 ]
                  reset-ticks           ;; allows a clock
end


to go                       ;;  THIS PROCEDURE CAUSES THE MOVEMENTS OF OUR BATS.
              ask turtles [ flock ]                     ;;  Tells the bats to fly together.  This is where the premade 'flock' commands are used by us.
              repeat 5 [ ask turtles [ sonar                ;;  The 'sonar' procedure asks the bats to look for and avoid cave walls.
                        skitterflight           ;;  The 'skitterflight' procedure makes the bats fly less like birds, and more eratically like bats.
                         fd 0.2  rt 2 - random 2      ;;  This gives some random motion to the bat flights, but keeps them circling to the right
                        sonar fd 0.2
                        skitterflight sonar         ;;  More bat-like motion commands.
                        scatter ] ]             ;;  The 'scatter' procedure is called whenever the bats exit the cave.  This represents them going off to feed.
              ask turtles [ if (heading > 260) and (heading < 280) [ fd 1.5 ] ]    ;;
              tick    ;;   Advances the clock.
```

```
end


to sonar            ;; THIS PROCEDURE IS USED TO KEEP THE BATS FROM FLYING INTO THE CAVE WALLS.
              if pcolor = 33 [set heading towardsxy -5 26]   ;; If a bat starts a move 'underground', it heads to the surface.
             if [pcolor] of patch-ahead 3 = 33 [ rt 35]     ;; This makes the bats look ahead, and turn away if they detect a cave wall.
end


to skitterflight      ;; THIS PROCEDURE MAKES THE BATS FLY LIKE BATS INSTEAD OF LIKE BIRDS (MAKES ERATIC MOVEMENTS).
            ask bats [ set xcor xcor + 4 fd 4 set ycor ycor + 4 set xcor xcor - 4 bk 4 set ycor ycor - 4 ]            ;; Makeing them skitter across like a real bat
             ask bats [ if heading  < 90 [set heading (270 + random 20 - 10) if heading > 90 and heading < 180 [set heading ((270 + random 10 - 5))]]]
                        ;; Heading 270 aims the bats toward the opening.
end


to scatter          ;; THIS PROCEDURE MAKES THE BATS DISAPPEAR, TO SIMULATE THEM SCATTERING OUT TO GO FEED.
              if pcolor = white [die]          ;; White patches represent clouds to let the bats know they are out of the cave.  Then they 'die' so show them disappearing.
end


to cave             ;; THIS PROCEDURE IS USED TO DRAW OUR CAVE.
         ask patches [ if pxcor >  178 [set pcolor 33]]            ;; This set the end of the cave as solid rock color.
        ask patches [ if  pycor > 25 * sin pxcor + 50 [set pcolor 33]]   ;; Uses math do draw the cave ceiling to follow a 'sine curve'. (Quicker than hand drawing a convoluted cave.
        ask patches [ if  pycor > 80  [set pcolor 17]]            ;; This makes the area above the cave be shown as an evening sky color.
        ask patches [ if  pycor < 25 * sin pxcor [set pcolor 33]]      ;; Drawing the cave floor to follow the sine curve.
       ask patches [ if  pycor > 2 * pxcor + 350 [set pcolor 17]]      ;; Draws 'open air' to represent the cave mouth.
end


to clouds           ;; THIS PROCEDURE DRAWS IN CLOUDS TO ALLOW BATS TO FINISH THEIR FLIGHT.
             ask patches [if pxcor < -195 and pycor > -35 [ set pcolor white]]   ;; Make clouds in the program along the edges, and make them white
             ask patches [if pxcor < -100 and pycor > 95 [ set pcolor white]]
end
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;   BORROWED CODE!!!  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


;;  ALL OF THE FOLLOWING IS FROM THE 'FLOCKING' PROGRAM

to flock  ;; turtle procedure
  find-flockmates
  if any? flockmates
    [ find-nearest-neighbor
    ifelse distance nearest-neighbor < minimum-separation
      [ separate ]
      [ align
      cohere ] ]
end

to find-flockmates  ;; turtle procedure
 set flockmates other turtles in-radius vision   ;; tells the bats to find and follow other bats
end

to find-nearest-neighbor ;; turtle procedure
 set nearest-neighbor min-one-of flockmates [distance myself]  ;; Tells them to find the closest bats near them
end

to separate  ;; turtle procedure
 turn-away ([heading] of nearest-neighbor) max-separate-turn
end

to align  ;; turtle procedure
 turn-towards average-flockmate-heading max-align-turn
end

to-report average-flockmate-heading
```

```
  let x-component sum [dx] of flockmates
  let y-component sum [dy] of flockmates
  ifelse x-component = 0 and y-component = 0
    [ report heading ]
    [ report atan x-component y-component ]
end

to cohere  ;; turtle procedure
  turn-towards average-heading-towards-flockmates max-cohere-turn
end

to-report average-heading-towards-flockmates  ;
  let x-component mean [sin (towards myself + 180)] of flockmates
  let y-component mean [cos (towards myself + 180)] of flockmates
  ifelse x-component = 0 and y-component = 0
    [ report heading ]
    [ report atan x-component y-component ]
end

to turn-towards [new-heading max-turn]  ;; turtle procedure
  turn-at-most (subtract-headings new-heading heading) max-turn
end

to turn-away [new-heading max-turn]  ;; turtle procedure
  turn-at-most (subtract-headings heading new-heading) max-turn
end

to turn-at-most [turn max-turn]  ;; turtle procedure
  ifelse abs turn > max-turn
    [ ifelse turn > 0
      [ rt max-turn ]
      [ lt max-turn ] ]
    [ rt turn ]
end
```