Emergency Evacuation Efficiency at Santa Fe High School

New Mexico
Supercomputing Challenge
Final Report
April 5, 2017

Team 02
Santa Fe High School

Team Members

Daniel Onstott

Micah Sulich

Luke Shankin

Marisa Tedori

Lileigh Thomas

Teacher

Brian Smith

Anita Nugent

Mentor

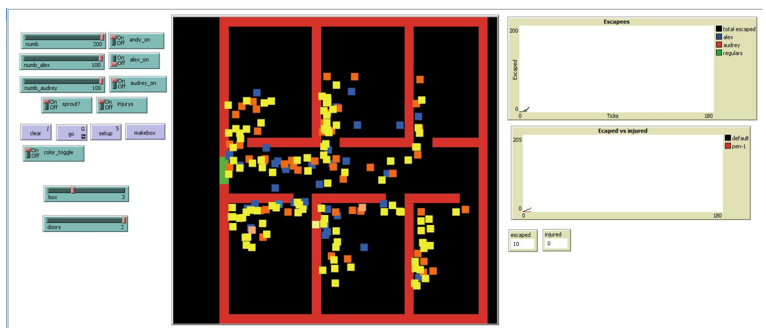
Brian Smith

Table of Contents

Executive Summary

This project stemmed from a personal experience our group has had at school with crowded hallways and tight choke points. During a genuine fire evacuation, people are much more likely to act irrationally then they are during a drill. Given that during drills, we had as many as two hundred people exiting a single door, we could only imagine how unsustainable that would be during a real emergency.

The project that was born from this concern was intended to better simulate many panicked individuals with a single goal. The final outcome, shown to the right, uses three different breeds of turtle in order to show more aggressive evacuees (blue), less aggressive evacuees (orange), and simplified neutral evacuees (yellow). Each breed of turtle has the same goal, the green exit to the left of the building, and different algorithms for how they will accomplish getting around obstacles such as walls and other turtles. Each breed also has varying chances of being pushed by other breeds and becoming "injured" which is increased based on the aggressiveness of turtles around them.



Currently the only modifiers to the building is the ability to change the size of the hallway that all turtles converge on. By changing this variable we have noticed some minor changes in evacuation speed as well as how our injury system reacts to larger or smaller groups of people in a condensed area.

Introduction

At the beginning of the 2016-2017 school year, Santa Fe High School held a fire drill in which it took roughly nine minutes to get from the classroom four doors down from the exit to the designated "safe zone." One may initially think this is because the safe zone is quite far away from the doors. On the contrary, this was due to the volume of students attempting to exit the building all at the same time. This prompted my team with a question: is there a more efficient way to move students through the exit doors and out to the safe zone? We began our research and found that an estimated 650 structure fires per year were reported in college classrooms and adult education centers (13%), while an average of 580 structure fires (11%) was reported annually in day care centers[10]. This accounts for 1% of all structure fires each year. Within this 1%, these fires caused an annual average of one civilian death, 79 civilian fire injuries and $88 million in direct property damage[10]. While educational facilities do not make up a large portion of structure fires, the facilities are extremely vulnerable to attacks of domestic terrorism and various other situations in which large amounts of people need to evacuate a building.

As the year persisted, we continued to time how long it took for each member of our team to exit the building during a drill from various places around campus. This presented us with the very real fact that our own school was incredibly ineffective in providing a safe exit, and was possibly making things even more difficult. The large volume of students evacuating from one set of double doors was (and still is) dangerously inefficient. That is why our project focuses particularly on the evacuations of Santa Fe High, and improving the efficiency of the process out of the school's academic building.

Background Information

Human Behavior Aspect

Human behavior is defined as the array of every physical action and observable emotion associated with individuals. The factors of one's human behavior include but are not limited to, attitude, perception, genetics, religious inclination, culture, social norms and ethnicities of society. Human behavior plays a major aspect in our project because the components of people are simulated through the code with aggressive and passive people. We show how the reaction of people in highly stressful situations affects an emergency exit.

The science/ psychology that happens when an emergency arises that is a threat to one's well-being stimulates the amygdala, a small structure located in a primitive part of the brain known as the limbic system. This act creates the survival instinct in a person. A human's survival behavior arises when they see other people reacting. For many don't react to emergencies unless others are in distress, a social and environmental aspect in emergencies.

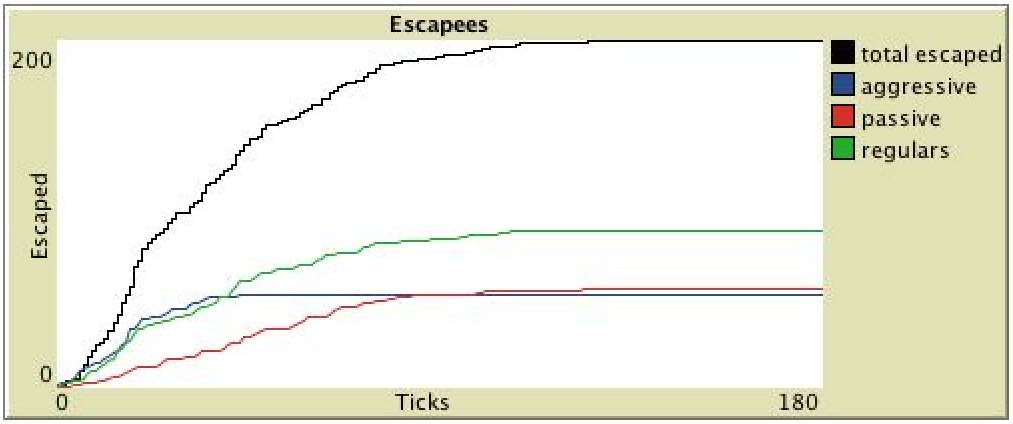
One's ability to handle the emergency comes down to the single component of human reaction. Which is a measurement of quickness an organism responds to some sort of stimulus, which inedible saves someone's life. This is measured through visual, auditory, and tactile. It takes 0.25s for a visual simulation, 0.17s for auditory stimulation, and 0.15s for tactile. This idea ties into reflexes of an individual.

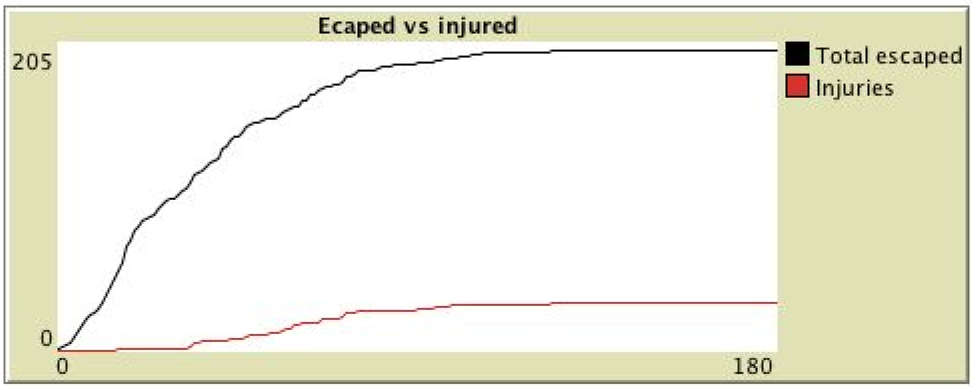
Another idea to introduce is American physiologist, Harvard medical school graduate; Walter Bradford Cannon's theory of Fight-or-Flight Response or Hyperarousal- the acute stress response. The psychological reaction that occurs in response to a perceived harmful event or threat to survival. His theory states that animals react to threats with a general discharge of the sympathetic nervous system when determining fighting or fleeing. This occurs with the adrenal medulla when it produces a hormonal cascade that results in the secretion of catecholamine (an organic compound that has a catechol) especially norepinephrine and epinephrine (organic chemicals in the catecholamine family that functions the brain and the body). This reaction is the same with humans.

Description

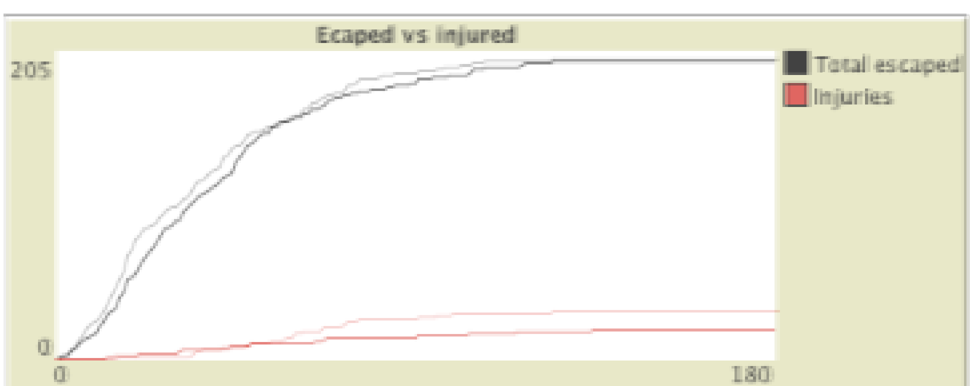
In our program, we attempted to make an accurate model of Santa Fe High School's exit route for evacuations. We then proceeded to model the export of students through the one and only exit during an emergency situation. As we secured the basics of our program, we then moved to implementing a reactionary element in which each turtle is assigned an aggressive or passive behavior during the situation. Our project is based in the language NetLogo because we felt it would be able to simulate a fire situation most effectively, as well as enable us to assign the different reactionary elements to individual turtles.

Results and Conclusions

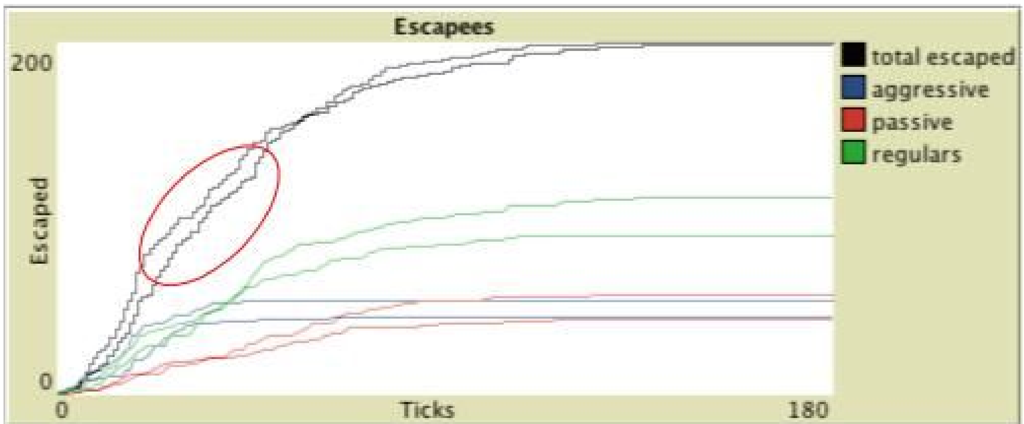




The graphs above shows an average cycle of our simulation with the widest hallways. Notice how the aggressive breed (graph 1) will push its way to the entrance first causing a spike in the blue line early on while the less aggressive breed will filter out much slower than the others. The graph below shows an average cycle where the hallways were as narrow as possible in our simulation.



The rate of injuries, or collisions between aggressive and passive breeds, is significantly lower dropping from a total 37 on average to only 20 average. This is most likely due to the fact that with less space in the hallways comes less people in one area at once which helps quell the amount of interactions between evacuees.



The final graph (above) shows only a minor increase in evacuation speed of total evacuees with wider hallways, demonstrating that it is more beneficial to keep students organized when there is an eventual choke point, rather than giving more space to move through.

Recommendations

While we believe our program demonstrates the most basic issues that arise during an evacuation drill, as well as adds a few more complex elements, there are always areas to improve in. The most prominent flaw in the program is the lack of realistic obstacles. Realistic obstacles may be defined as any object that presents itself as an obstruction or hurdle in the pathway of the exit. For instance, in a school fire evacuation, there will likely be many backpacks, chairs, desks, etc. that will interfere with a student's ability to exit the classroom in an efficient manner. We believe to make the simulation as effective as possible, it would be necessary to implement a way to randomize "obstructive objects" within the classrooms to better understand how individuals react to the hurdle. This would enhance the reality of the program, and therefore increase the effect of its results.

Aside from the above mentioned flaw, there are many other aspects of the program to consider improving. For example, many students at Santa Fe High School demonstrate a need for handicap accessibility when navigating the school. While the school has made strides in accommodating these students, the event of even a fire drill presents a setback for the whole process. In an extension of this project, it would be particularly interesting to introduce students who are limited terms of speed during evacuation. This would make our simulation both more specific to Santa Fe High School *and* more encompassing of issues that many public and private facilities face.

Finally, we would recommend enhancing the intensity of emotions the individual can act on. Our current program is fairly basic when it comes to emotional complexity, meaning our turtles can either be extremely aggressive, extremely passive or in the middle. This is based on the speed of the agent as well as different ways in which the agents navigates obstacles and it serves its purpose in the program as is, however human emotions tend to be on the extreme ends of the spectrum very infrequently. In a future rendition, it would be interesting to implement a spectrum of emotions that would be randomly assigned to each turtle to produce the effect of many individual minds interpreting the situation differently, potentially even trying to develope a way for turtle's emotions to be influenced by surrounding turtles. If one turtle is assigned a mild-frantic emotion, but is within close proximity to turtles assigned a more calm behavior, the original turtle may turn more calm due to its surroundings. This would increase the reality of the project, as well as hopefully provide insight into the effect of human emotion on crisis situations.

Acknowledgements

We would like to sincerely thank the following people for the time and help they dedicated to our Supercomputing Challenge 2017 project, and the knowledgeable input they gave to us.

Mr. Brian Smith, our programming mentor. We would like to thank him for enforcing our deadlines, assisting us in reaching out to the school administration, and offering expertise as a compute science teacher when we ran into problems we did not know how to fix.

Ms. Anita Nugent, our Santa Fe High advisor. We would like to thank her for being our Science Club representative, as well as helping us develop a clear idea for our project.

Bibliography

1. "Amygdala Activity, Fear, and Anxiety: Modulation by Stress." *Biological Psychiatry*. U.S. National Library of Medicine, 15 June 2010. Web. 03 Apr. 2017.
2. "Artists Resource for Fire." *ARF: Fire Safety: Minimum Safe Distances*. N.p., n.d. Web. 6 Nov. 2016.
3. "Behaviour Based Motion Simulation for Fire Evacuation Procedures - IEEE Xplore Document." *Behaviour Based Motion Simulation for Fire Evacuation Procedures - IEEE Xplore Document*. N.p., n.d. Web. 20 Nov. 2016.

4. "Experiment: How Fast Your Brain Reacts To Stimuli." *Experiment: How Fast Your Brain Reacts To Stimuli*. N.p., n.d. Web. 03 Apr. 2017.
5. Klote, John H. "Smoke Control and Fire Evacuation: What Every Smoke-control Designer Should Know about Fire Evacuation, the "panic Myth," and Human Behavior."*Heating/Piping/Air Conditioning Engineering.* (2008): 36. *Elportal*. Web. 1 Nov. 2016.
6. Koutamanis, Alexander. "Multilevel Analysis of Fire Escape Routes in a Virtual Environment." *CAAD Futures Digital Proceedings 1995* (1995): 331-42. Web. 5 Dec. 2016.
7. Layton, Julia. "How Fear Works." *HowStuffWorks Science*. HowStuffWorks, 13 Sept. 2005. Web. 03 Apr. 2017.
8. Li, Paul, and Jeannine Stamatakis. "What Happens in the Brain When We Experience a Panic Attack?" *Scientific American*. N.p., 17 June 2011. Web. 03 Apr. 2017. Ressler, Kerry J.
9. "Queuing Theory Accurately Models the Need for Critical Care Resources." *Home*. N.p., n.d. Web. 01 Dec. 2016.
10. "School Safety." *School Safety*. National Fire Protection Association, n.d. Web. 03 Apr. 2017.
11. "Survival – Mind and Brain." *Survival – Mind and Brain | The Psychologist*. N.p., n.d. Web. 03 Apr. 2017.
12. "Walter Bradford Cannon." *Walter Bradford Cannon*. N.p., n.d. Web. 03 Apr. 2017.

Appendix

Code

```
;Evacuation Efficiency (v.1.6.3 LAMBDA)
;Team SFHS-2 by Micah Sulich, Daniel Onstott, Luke Shankin, Lileigh Thomas, and Marisa Tedori
;Made in netlogo v6.0

globals [escaped alex_escaped audrey_escaped reg_escaped injured]
;the more aggressive agent set
breed [alex alexes]
;the more submissive agent set
breed [audrey audreys]
;the middle of the two
breed [andy andys]
to clear
  clear-all
end
to setup
  ifelse sprout? = true
  [
      ;reset graph
  set escaped 0
  set alex_escaped 0
  set audrey_escaped 0
  set reg_escaped 0
  set injured 0
  clear-turtles
  ;finds patches that are within the building to spawn from
  ask n-of numb (patches with ([pcolor != red and pcolor != green and pxcor > -13 and pxcor < 15]))
  [
    ;coinflip for regular andy
    ifelse (random(2)) = 0
    [
    sprout 1
    [set color yellow
     set breed andy
    set shape "square"]
    ][
     ;further coinflip o divide remaining andy
     ifelse (random(2)) = 0 [
     sprout 1
    [set color blue
     if color_toggle = false
      [
```

```
      set color yellow
     ]
    set breed alex
   set shape "square"]
   ][
    sprout 1
   [set color orange
    if color_toggle = false
    [
      set color yellow
     ]

    set breed audrey
   set shape "square"]
   ]
   ]
  ]

  ]
  ;---sprout off------------
  [
  set escaped 0
  set alex_escaped 0
  set audrey_escaped 0
  clear-turtles
  if andy_on = true
  [
   create-andy numb
   [

     setxy (((-12) + 2) + random (26)) (((-12) + (-2)) + random (29))
     set color yellow

  ]
  ]

  if alex_on = true
  [
   create-alex numb_alex
   [

   setxy (((-12) + 2) + random (26)) (((-12) + (-2)) + random (29))
   set color blue

   ]
  ]

 if audrey_on = true
```

```
 [
 create-audrey numb_audrey
 [

   setxy (((-12) + 2) + random (26)) (((-12) + (-2)) + random (29))
   set color orange

 ]
 ]
 ]
 reset-ticks
 setup-plots
end


to bounce_alex
   ;The bounce function is the dictates how the agents interact with each other.
; determines if the agent goes up or down
 if [pcolor] of patch-at dx 1 = red
 [
   if ycor > (-1 * box)[set heading (180)]
   if ycor < (-1 * box)[set heading (90)]
 ]
if not any? other audrey-on patch-ahead 1 = false
 [

   set heading (heading - 180)
   fd 1.2
   set heading(heading + 79)
   set heading(heading - 259)

 ]



end




to bounce_audrey
  ;The bounce function is the dictates how the agents interact with each other.
 ; In the case of this breed, if a red wall blocks its path, it will turn towards the door and try to move closer
to it until it can move around the obstacle.
 ;When coming into contact with another agent, It will turn around to find a clear path before trying to go
towards the door again.
 ;In this sense the agent â€œwaitsâ€ and â€œpushesâ€ its way around other agents.
```

```
 let flip (random 2)
; determines if the agent goes up or down
 if [pcolor] of patch-at dx 1 = red
 [
   if ycor > (-1 * box)[set heading (180)]
   if ycor < (-1 * box)[set heading (90)]
 ]

 if not any? other alex-on patch-ahead 1 = false
 [
   if flip = 1
   [

   set heading (heading - 180)
   fd .5
   set heading(heading + 85)
   set heading(heading - 259)

   ]

   if flip = 2
   [

   set heading (heading + 180)
   fd .5
   set heading(heading - 85)
   set heading(heading + 259)

   ]

 ]

 if not any? other andy-on patch-ahead 1 = false
 [
   if flip = 1
   [

   set heading (heading - 180)
   fd .75
   set heading(heading + 85)
   set heading(heading - 259)

   ]

   if flip = 2
   [

   set heading (heading + 180)
```

```
    fd .75
    set heading(heading - 85)
    set heading(heading + 259)

    ]

  ]

  if not any? other audrey-on patch-ahead 1 = false
  [
    if flip = 1
    [

    set heading (heading - 180)
    fd .9
    set heading(heading + 85)
    set heading(heading - 259)

    ]

    if flip = 2
    [

    set heading (heading + 180)
    fd .9
    set heading(heading - 85)
    set heading(heading + 259)

    ]

  ]

end

to bounce_andy
  ;The bounce function is the dictates how the agents interact with each other
  let flip (random 2)
; determines if the agent goes up or down
  if [pcolor] of patch-at dx 1 = red
  [
    if ycor > (-1 * box)[set heading (180)]
    if ycor < (-1 * box)[set heading (90)]
  ]

  if not any? other turtles-on patch-ahead 1 = false
  [
    if flip = 1
    [
```

```
  set heading (heading - 180)
  fd 1
  set heading(heading + 85)
  set heading(heading - 259)

  ]

  if flip = 2
  [

  set heading (heading + 180)
  fd 1
  set heading(heading - 85)
  set heading(heading + 259)

  ]

 ]

end
to makebox

 ;made by Micah
 ;SETTING UP THE SCHOOL and classroom environment
 clear-patches

 ;origin points
 let origin_x (-12)
 let origin_y (box)

 ;top room vars
 let top_room_1 (1 + box)
 let top_room_2 (1 + box)
 let top_room_3 (1 + box)

 ;hallway length vars
 let hall_leng_1 (-12)
 let hall_leng_2 (-12)
 let hall_leng_3 (-12)
 let hall_leng_4 (-12)

 ;bottom room vars
 let bt_room_1 (1 + box)
 let bt_room_2 (1 + box)
 let bt_room_3 (1 + box)

 ;back wall
```

```
let bk_wall_1 (16)
let bk_wall_2 (-16)
let bk_wall_3 (-16)


;making the front door/building edge
ask (patch (origin_x) (-1) )
  [
  set pcolor green
  ]
ask (patch (origin_x) (0) )
  [
  set pcolor green
  ]ask (patch (origin_x) (1) )
  [
  set pcolor green
  ]

  ask (patch (origin_x) (-2) )
  [
  set pcolor red
  ]
  ask (patch (origin_x) (-3) )
  [
  set pcolor red
  ]
  ask (patch (origin_x) (-4) )
  [
  set pcolor red
  ]

  ask (patch (origin_x) (2) )
  [
  set pcolor red
  ]
  ask (patch (origin_x) (3) )
  [
  set pcolor red
  ]
  ask (patch (origin_x) (4) )
  [
  set pcolor red
  ]

;TOP HALLWAY WALLs and doors
while [hall_leng_1 < (17)]
[
  ask (patch (hall_leng_1) (origin_y) )
```

```
  [
  set pcolor red
  ]

  set hall_leng_1 (hall_leng_1 + 1)
]
while [hall_leng_3 < (17)]
[
  ask (patch (hall_leng_3) (16) )
  [
  set pcolor red
  ]
  set hall_leng_3 (hall_leng_3 + 1)
]

   ;Making the upper hallway door gaps
ask (patch (0) (origin_y) )
  [
  set pcolor black
  ]
 ask (patch (-1) (origin_y) )
  [
  set pcolor black
  ]

 ask (patch (-11) (origin_y) )
  [
  set pcolor black
  ]
 ask (patch (-10) (origin_y) )
  [
  set pcolor black
  ]
 ask (patch (9) (origin_y) )
  [
  set pcolor black
  ]
 ask (patch (10) (origin_y) )
  [
  set pcolor black
  ]

;LOWER HALLWAY WALL
while [hall_leng_2 < (17)]
[
  ask (patch (hall_leng_2) (-1 * origin_y) )
  [
  set pcolor red
```

```
  ]
  set hall_leng_2 (hall_leng_2 + 1)
 ]
 while [hall_leng_4 < (17)]
 [
  ask (patch (hall_leng_4) (-16) )
  [
  set pcolor red
  ]
  set hall_leng_4 (hall_leng_4 + 1)
 ]
   ;Making the lower hallway door gaps
 ask (patch (-3) (-1 * origin_y) )
  [
  set pcolor black
  ]
 ask (patch (-4) (-1 * origin_y) )
  [
  set pcolor black
  ]
 ask (patch (6) (-1 * origin_y) )
  [
  set pcolor black
  ]
 ask (patch (7) (-1 * origin_y) )
  [
  set pcolor black
  ]
 ask (patch (15) (-1 * origin_y) )
  [
  set pcolor black
  ]
 ask (patch (14) (-1 * origin_y) )
  [
  set pcolor black
  ]

;THIS CODE MAKES the top rooms
 while [top_room_1 < (17)]
 [
  ask (patch  (origin_x) (top_room_1))
  [
  set pcolor red
  ]
  set top_room_1 (top_room_1 + 1)
 ]

 while [top_room_3 < (17)]
```

```
 [
   ask (patch  (origin_x + 10) (top_room_3))
   [
   set pcolor red
   ]
   set top_room_3 (top_room_3 + 1)
 ]

while [top_room_2 < (17)]
 [
   ask (patch  (origin_x + 20) (top_room_2))
   [
   set pcolor red
   ]
   set top_room_2 (top_room_2 + 1)
 ]

;this code makes the bottom rooms
while [bt_room_1 < (17)]
 [
   ask (patch  (origin_x) (-1 * bt_room_1))
   [
   set pcolor red
   ]
   set bt_room_1 (bt_room_1 + 1)
   ]
while [bt_room_3 < (17)]
 [
   ask (patch  (origin_x + 10) (-1 * bt_room_3))
   [
   set pcolor red
   ]
   set bt_room_3 (bt_room_3 + 1)
 ]
while [bt_room_2 < (17)]
 [
   ask (patch  (origin_x + 20) (-1 * bt_room_2))
   [
   set pcolor red
   ]
   set bt_room_2 (bt_room_2 + 1)
 ]

while [bk_wall_1 > (0)]
[
  ask patch (16) (bk_wall_1)
  [
    set pcolor red
```

```
  ]
  set bk_wall_1 (bk_wall_1 - 1)
 ]
 while [bk_wall_2 < (0)]
 [
  ask patch (16) (bk_wall_2)
  [
   set pcolor red
  ]
  set bk_wall_2 (bk_wall_2 + 1)
 ]
 ask (patch (16) (0) )
   [
   set pcolor red
   ]
  while [bt_room_1 < (17)]
  [
   ask (patch  (origin_x) (-1 * bt_room_1))
   [
   set pcolor red
   ]
   set bt_room_1 (bt_room_1 + 1)
   ]

;this code inpart prevents the backwall bug
let bugfix1 (-16)
while [bugfix1 < (17)]
[
ask (patch  (17) (bugfix1))
   [
   set pcolor red
   ]
set bugfix1 (bugfix1 + 1)
]

end
to fix_error_alex
  ;These following "fix error" procedures fix an error that resulted in the turtles going too far into the wall and the program crashing.
  ask alex
  [
  if xcor > 17
  [
   set heading (180)
   fd 1
  ]
  ]
```

```
end

to fix_error_audrey
 ask audrey
 [
 if xcor > 17
 [
  set heading (180)
  fd 1
 ]
 ]

end
to fix_error_andy
 ask andy
 [
 if xcor > 17
 [
  set heading (180)
  fd 1
 ]
 ]

end

to go
 reset-timer

 ;origin points
 let origin_x (-12)
 let origin_y (box)
 ;the likeley hood of an alex injuring the other agents
 let injury_chance_alex1 ( numb * .06)
 let injury_chance_alex2 ( numb * .03)
 let injury_chance_alex3 ( numb * .01)


ask alex
 [
  if injurys = true
  [
   ;audrey 6%
   if injury_chance_alex1 > random(numb)
   [
    ask audrey-here [set color 27]
   ]
   ;andy 3%
   if injury_chance_alex2 > random(numb)
```

```
    [
     ask andy-here [set color 47]
    ]
    ;alex 1%
    if injury_chance_alex3 > random(numb)
    [
     ask alex-here [set color 95]
    ]

   ]
    set heading towards patch (origin_x) ((0) + 1)
    bounce_alex
    fix_error_alex
    ifelse color != 27
    ;This set of commands dictates the movement of the healthy Alexes while the second one dictates how
they move when injured
    [
     fd 1.5
     ; this algorithm makes the Alex disaper as they go through the end door, simulating their escape.
     if xcor <= (origin_x) + .5
     [
      if xcor >= (origin_x) - .5
      [
       if ycor <= ((0) + 1.5)
       [
        if ycor >= ((0) + .5)
        [
         set escaped (escaped + 1)
         set alex_escaped (alex_escaped + 1)
         die
        ]
       ]
      ]
     ]
    ]
    ;if the Alexes are injured, they go 50% of their movement speed.
    [
     fd .7
    ; this algorithm makes the andy disappear as they go through the end door, simulating their escape.
     if xcor <= (origin_x) + .5
     [
      if xcor >= (origin_x) - .5
      [
       if ycor <= ((0) + 1.5)
       [
        if ycor >= ((0) + .5)
        [
         set escaped (escaped + 1)
```

```
              set alex_escaped (alex_escaped + 1)
              set injured (injured + 1)
              die
             ]
            ]
           ]
          ]
         ]
  ]

ask audrey
  [

     set heading towards patch (origin_x) ((0) + 1)
     bounce_audrey
     fix_error_audrey
     ifelse color != 27
     ;This set of commands dictates the movement of the healthy Audreyes while the second one dictates
how they move when injured
     [
      fd .9
      ; this algorithm makes the audrey disappear as they go through the end door, simulating their escape.
      if xcor <= (origin_x) + .5
      [
       if xcor >= (origin_x) - .5
       [
        if ycor <= ((0) + 1.5)
        [
         if ycor >= ((0) + .5)
         [
          set escaped (escaped + 1)
          set audrey_escaped (audrey_escaped + 1)
          die
         ]
        ]
       ]
      ]
     ]
     ;if the Audreys are injured, they go 50% of their movement speed.
     [
      fd .4
     ; this algorithm makes the andy disappear as they go through the end door, simulating their escape.
      if xcor <= (origin_x) + .5
      [
       if xcor >= (origin_x) - .5
       [
        if ycor <= ((0) + 1.5)
        [
```

```
        if ycor >= ((0) + .5)
        [
          set escaped (escaped + 1)
          set audrey_escaped (audrey_escaped + 1)
          set injured (injured + 1)
          die
        ]
      ]
    ]
  ]
 ]
]


ask andy
[

    set heading towards patch (origin_x) ((0) + 1)
    bounce_andy
    fix_error_andy
    ifelse color != 47
    ;This set of commands dictates the movement of the healthy andys, while the second one dictates how they move when injured
    [
     fd 1
     ; this algorithm makes the Andy disappear as they go through the end door, simulating their escape.
     if xcor <= (origin_x) + .5
     [
      if xcor >= (origin_x) - .5
      [
       if ycor <= ((0) + 1.5)
       [
        if ycor >= ((0) + .5)
        [
          set escaped (escaped + 1)
          set reg_escaped (reg_escaped + 1)
          die
        ]
       ]
      ]
     ]
    ]
    ;if the andys are injured, they go 50% of their movement speed.
    [
     fd .5
    ; this algorithm makes the andy disappear as they go through the end door, simulating their escape.
     if xcor <= (origin_x) + .5
     [
```

```
        if xcor >= (origin_x) - .5
        [
          if ycor <= ((0) + 1.5)
          [
            if ycor >= ((0) + .5)
            [
              set escaped (escaped + 1)
              set reg_escaped (reg_escaped + 1)
              set injured (injured + 1)
              die
            ]
          ]
        ]
      ]
     ]

]


  tick
end
```