**UNIVERSIDAD NACIONAL DE INGENIERÍA**

# **FACULTAD DE CIENCIAS**

****

# **Tópicos de Ciencia de la Computación CC0F6 A**

# **Modelamiento de Cúpula de Hierro usando Agentes Inteligentes**

***Ademar Fatama Ruiz – 20192148F***

***Joel Jimenez Chacon – 20160559k***

***Israel Danilo Blas Salas – 20150522G***

***Cleber Aguado Gutierrez - 20210386G***

**Profesor:**  
 **GIPSY MIGUEL ANGEL ARRUNATEGUI ANGULO**

1. Introducción

En un contexto global donde la seguridad y la defensa son prioridades fundamentales, la necesidad de desarrollar sistemas eficaces para contrarrestar amenazas aéreas se ha vuelto imperativa. En particular, el sistema de defensa anti-misiles conocido como Iron Dome, o Cúpula de Hierro, desempeña un papel crucial en la protección del territorio de Israel frente a posibles ataques con misiles. Este sistema, compuesto por una red de estaciones de defensa distribuidas estratégicamente, tiene como objetivo detectar, interceptar y neutralizar misiles enemigos, contribuyendo así a la seguridad nacional.

En este informe, nos proponemos abordar la evaluación y optimización de la efectividad del Iron Dome mediante la implementación de un modelo de simulación en NetLogo. Este modelo permitirá analizar diversos escenarios y parámetros que influyen en la capacidad defensiva del sistema. Al igual que en situaciones del mundo real, se considerarán elementos clave, como la presencia de múltiples misiles, diversos sitios de lanzamiento de ataques, la visibilidad y sensibilidad de los detectores, el tiempo de recarga, y otros factores relevantes que puedan afectar el rendimiento del Iron Dome.

1. Descripción del problema

El propósito es simular la efectividad de la defensa antimisiles de Israel en NetLogo, explorando diversos escenarios. Se analizarán factores clave como múltiples misiles, sitios de lanzamiento, visibilidad del detector, tiempo de recarga y la capacidad de guiado de misiles defensores. También se considerará la ubicación de los misiles en la cúpula y su habilidad para explotar guiadamente. Esta simulación brindará una comprensión detallada de la capacidad de respuesta del sistema en situaciones variadas, facilitando la toma de decisiones estratégicas y mejoras continuas.

1. Descripción del Modelo de Agentes Inteligentes (ABM)

En este proyecto, se utiliza un Modelo de Agentes Inteligentes (ABM) para simular la operación de un sistema de defensa de misiles similar al Iron Dome de Israel. Un ABM es un enfoque de modelado que se basa en la interacción y comportamiento autónomo de agentes individuales para comprender los patrones emergentes de un sistema complejo.

En este caso, el modelo de ABM consta de los siguientes componentes:

Agentes: Los misiles entrantes y los misiles de defensa se representan como agentes autónomos en el modelo. Cada misil tiene su propio conjunto de características, como velocidad, dirección y estado (vivo o no).

Sistema de detección: El sistema de detección es una parte crucial del modelo. Detecta los misiles entrantes dentro de un cierto radio y activa la respuesta de defensa.

Interacción: Los misiles de defensa se desplazan hacia los misiles entrantes detectados. Al llegar a un misil entrante, lo neutralizan y ambos misiles se eliminan del modelo.

Restricciones y variables: El modelo considera diversas restricciones, como el tiempo de recarga entre los disparos de los misiles de defensa y la velocidad de los misiles. Estas variables afectan la eficacia del sistema de defensa.

Visualización: El modelo proporciona representaciones visuales animadas de las trayectorias de los misiles, el área de detección y otras variables relevantes. Esto permite observar el comportamiento del sistema de defensa y analizar los resultados de la simulación.

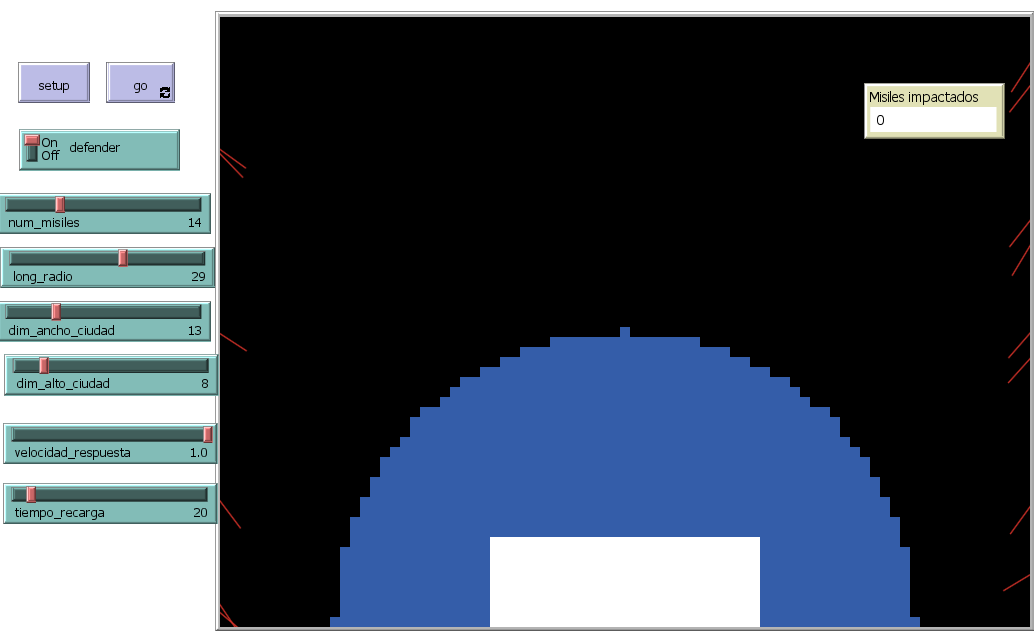
El modelo de Agentes Inteligentes permite simular de manera realista la operación de un sistema de defensa de misiles similar al Iron Dome. Al ajustar los parámetros y probar diferentes escenarios, podemos obtener información valiosa sobre la eficiencia y seguridad del sistema, lo que facilita la toma de decisiones informadas en situaciones donde la defensa contra misiles es crucial.

1. Parámetros y variables del modelo

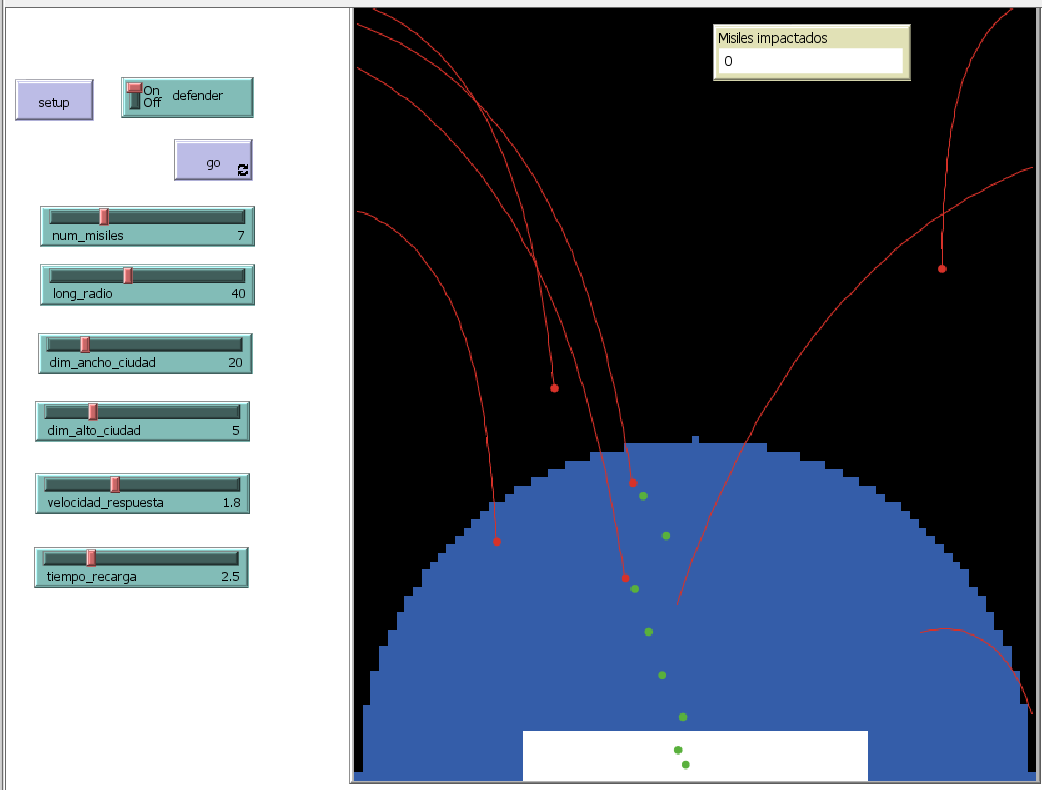
Código:

**globals** [  
 cant\_amenazas  
 gravity  
 ticks\_desde\_ultimo\_disparo ; nuevo contador global para rastrear el tiempo desde el último disparo  
 misiles\_impactados\_ciudad  
 fortaleza  
]  
**turtles-own** [  
 bandera  
 direccion  
 velocity  
 target-turtle  
 vivo  
 op\_heading  
]  
**to** setup  
 clear-all  
 sensor  
 ciudad  
 set fortaleza **true**  
 set gravity -0.04  
 create-turtles num\_misiles  
 [ set shape "circle"  
 set bandera 0  
 set size 1  
 set color red  
 set vivo **true** ; inicialmente, todos los misiles están vivos  
 randomize  
 pen-down  
 ]  
 set ticks\_desde\_ultimo\_disparo tiempo\_recarga ; inicialmente, se puede disparar un misil de inmediato  
 reset-ticks  
**end**  
  
  
**to** sensor  
 ask patches with [sqrt(pxcor ^ 2 + pycor ^ 2) <= long\_radio ]  
 [ set pcolor blue ]  
**end**  
  
; fronteras de la ciudad  
**to** ciudad  
 ask patches with [abs pxcor <= dim\_ancho\_ciudad and pycor <= dim\_alto\_ciudad]  
 [ set pcolor white ]  
**end**  
  
**to** randomize  
 ifelse random-float 1 < 0.5  
 [ set xcor max-pxcor ]  
 [ set xcor min-pxcor ]  
 set ycor random-ycor  
 ifelse ycor > max-pycor / 3  
 [ set velocity 0.5 + random-float 1.5  
 ifelse xcor > 0  
 [ set heading (210 - random 15) ]  
 [ set heading (-30 + random 15) ]]  
 [ set velocity 0.5 + random-float 0.5  
 ifelse xcor > 0  
 [ set heading (115 - random 15) ]  
 [ set heading (75 - random 15) ] ]  
  
 if pcolor = red or ycor < min-pycor or ycor > max-pycor  
 [ randomize ]  
**end**  
  
**to** go  
 set cant\_amenazas count turtles with [color = red and bandera = 1 and vivo = **true**] ; cuenta las amenazas al inicio de cada tick  
 ask turtles  
 [ if color = red and vivo = **true** ; solo los misiles rojos vivos se mueven y pueden ser una amenaza  
 [ let angle heading ; Ángulo de lanzamiento  
 ; Fórmulas de movimiento parabólico  
 let vx velocity \* cos angle  
 let vy velocity \* sin angle  
 ; Actualizar posición  
 setxy (xcor + vx) (ycor + vy)  
 if xcor < min-pxcor or xcor > max-pxcor or ycor < min-pycor + 1.5 ; or ycor > max-pycor  
 [ die ]  
 ; Actualizar velocidad y  
 set vy vy + gravity  
 set heading atan vy vx  
 if pcolor = blue and bandera = 0  
 [ set bandera 1 ]  
 if pcolor = blue  
 [ set direccion heading]  
 if abs pxcor <= dim\_ancho\_ciudad and pycor <= min-pycor + dim\_alto\_ciudad ; si el misil ha llegado a la ciudad  
 [ set vivo **false** ; el misil ya no está vivo  
 set fortaleza **false**  
 ask patches with [pcolor = white]  
 [ set pcolor brown ]  
 set misiles\_impactados\_ciudad (misiles\_impactados\_ciudad + 1)  
 die ; elimina el misil  
 ] ; incrementa el contador de misiles que impactan en la ciudad  
 ]  
 if color = green and defender = **true**  
 [ fd velocidad\_respuesta ; aumenta la velocidad de los misiles de respuesta  
 if is-turtle? target-turtle and [color] of target-turtle = red and [bandera] of target-turtle = 1 and [vivo] of target-turtle = **true** ; solo ajusta la dirección si el objetivo es un misil entrante vivo y aún existe  
 [ set heading towards target-turtle  
 if distance target-turtle < 0.5 [ ; aumenta la distancia a la que consideramos que los misiles han colisionado  
 hatch 1 [ ; crea una nueva tortuga  
 ask myself [ die ] ; elimina el misil de defensa  
 ask target-turtle [ set vivo **false** ; el misil objetivo ya no está vivo  
 die ] ; elimina el misil objetivo  
 die  
 ]  
 ]  
 ]  
 if xcor < min-pxcor + 1.5 or xcor > max-pxcor - 1.5 or ycor > max-pycor - 1.5 ; or ycor > max-pycor  
 [ die ]  
 ]  
 ]  
 tick  
 if ticks\_desde\_ultimo\_disparo < tiempo\_recarga ; si no ha pasado suficiente tiempo desde el último disparo, incrementa el contador  
 [ set ticks\_desde\_ultimo\_disparo (ticks\_desde\_ultimo\_disparo + 1) ]  
 if cant\_amenazas > 0 and defender = **true** and ticks\_desde\_ultimo\_disparo >= tiempo\_recarga and fortaleza; solo se puede disparar un misil si ha pasado suficiente tiempo desde el último disparo  
 [ activar  
 set ticks\_desde\_ultimo\_disparo 0 ] ; restablece el contador después de disparar un misil  
**end**  
  
**to** activar  
 let target one-of turtles with [color = red and bandera = 1 and vivo = **true**] ; selecciona un misil entrante vivo como objetivo  
 if target != nobody [ ; si hay un objetivo  
 create-turtles 1  
 [ set shape "circle"  
 set size 1  
 set color green  
 set xcor 0  
 set ycor min-pycor  
 set heading towards target  
 set target-turtle target ] ; establece el objetivo del misil  
 ]  
**end**

Modelo:



1. Resultados



Como primer modelo se tuvo una idea de realizar una esfera vista desde arriba llegando misiles desde todas direcciones, pero luego se optó por cambiar a una vista de lado para generar movimientos parabólicos en los misiles.  
  
Como primer modelo teníamos misiles representados como líneas que chocaban, pero se mejoró por la representación final que nos permite visualizar la trayectoria de los misiles atacantes, una mejor representación del área de los misiles, retardo en el lanzamiento de los misiles de defesa además detección de los misiles que llegan al sensor.

1. Conclusiones

En este proyecto, se implementó con éxito un modelo de Agentes Inteligentes (ABM) para simular la defensa de un domo con un área como sensor de misiles.

1. Implementación exitosa del modelo: Se logró desarrollar e implementar el modelo de ABM utilizando la plataforma NetLogo, lo que nos permitió simular la interacción entre el área del domo antimisiles y los misiles atacantes.

2. Restricciones en la interacción de agentes: Se adapto del comportamiento de los misiles de ataque y también los de defensa para que interactúen como si estuvieran en condiciones reales (Como la gravedad y movimiento parabólico) con los parámetros pasados en los widgets de netLogo.

3. Importancia de futuras investigaciones: En nuestro modelo a pesar de haber implementado la mayor cantidad de características posibles aun podríamos mejorar si la ciudad tuviera más puntos de lanzamiento de misiles de defensa o si el campo de visión fuera más grande para poder visualizar que siempre lleguen desde una zona alta.

4. Optimización de parámetros y variables: Se podrían utilizar técnicas de ML para poder obtener parámetros óptimos generando multiples simulaciones tomando como variable predictiva e ideal la cantidad de misiles impactados siendo 0 está la óptima.

1. Referencias

* NetLogoWeb. (s.f.). Differences Between NetLogoWeb and NetLogo. Recuperado de <https://www.netlogoweb.org/docs/differences>