

Documentación Proyecto Final Programación IV – Missile Command

Las especificaciones del juego son las siguientes:

MissileCommand

Missile Command es un juego desarrollado por Atari en 1980, en el cual se simula la invasión hacia las ciudades terrestre, y una serie de torretas de misiles deben detener a los invasores.

ESPECIFICACIONES MISSILE COMMAND

En base al juego Missile Command de 1980 realizar una versión 2.0 en la cual se tendrán las siguientes condiciones:

- El juego tendrá misiles que aleatoriamente caen del cielo con el objetivo de derrumbar las ciudades.
- El juego tendrá ciudades las cuales podrán ser destruidas por los misiles
- El juego tendrá cañones los cuáles serán las defensas de las ciudades, y van a atacar los misiles. Ellos atacarán según en las coordenadas las cuales indiquen el cursor. Dos de los cañones estarán ubicados en las esquinas de las ciudades y otro estará en la mitad de ellas.
- Si cae un misil en una ciudad la destruirá por completo, al igual que si cae un misil en los cañones.
- El jugador perderá si todas las ciudades están destruidas.
- Cuando se presente un choque el juego deberá representar gráficamente la colisión
- El objetivo final del jugador es defender las ciudades atacando todos los misiles que caerán del cielo, tratando de durar la mayor cantidad de tiempo.

Para mejor interpretación mirar el siguiente

link: https://www.youtube.com/watch?v=8eC_6QzvLrE

En el desarrollo del proyecto se manejaron las siguientes ideas:

Se implementó la librería JavaFX la cual permitía apoyarse de clases que implementaban métodos abstractos para una mejor organización del código. La idea principal fue apoyarnos en un framework llamado GameFx, pero al fin se terminó trabajando con los métodos de la clase Application.

Como tal se creó la interfaz que permitía interactuar con un ambiente de una ciudad en las horas de la noche la cual era atacada por misiles que eran lanzados

desde el cielo. Los misiles como tal son lanzados aleatoriamente desde el cielo, obteniendo una posición X aleatoria cada vez que se crea. A medida que van bajando (incrementando su variable Y) van dejando rastro por toda la pantalla hasta su destino final, que sería la parte del rascacielos de los edificios (pixel en Y 390) ya que todos los edificios tienen la misma altura.

Cuando un misil impacta con un edificio, este edificio se destruirá; para el manejo de esta idea se implementó dos arreglos, uno llamado `buildingsI` que se encargaba de almacenar las imágenes de los edificios, y otro arreglo `buildingsN` que se encarga de almacenar 1 y 0, 1 indicando que hay un edificio y 0 indicando que no hay edificio. Esto con el fin de que cada vez que se pintara la pantalla, en especial los edificios, él se pintara de acuerdo a la información de unos y ceros que había en el vector `buildingsN`, y crear la simulación de destrucción.

Cada vez que se manda a pintar la pantalla con el método `draw()`; él verifica primero que en el arreglo de números (`buildingsN`) por lo menos haya todavía un 1 (un edificio) si lo hay retorna `true`, indicando que todavía es válido el lanzamiento de misiles, y si ya no queda ningún uno él retorna `false` indicando que el lanzamiento de misiles debe de parar ya que no queda ningún edificio por destruir.

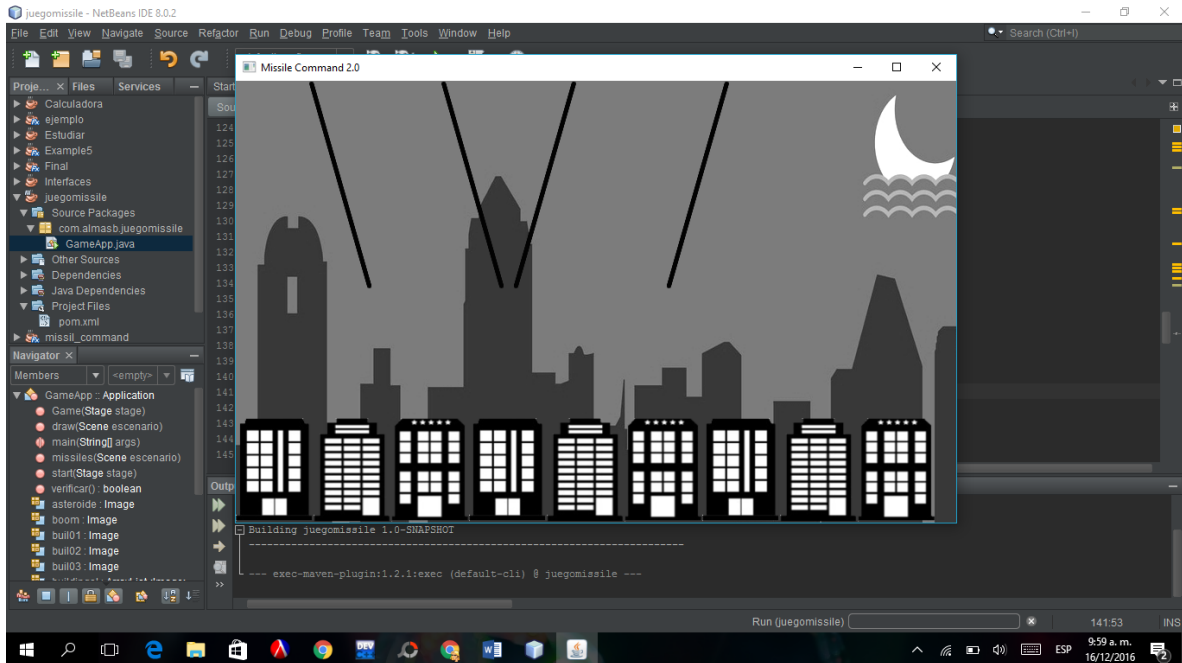
En la parte de la animación de misiles `AnimationTimer()`, dentro de ella se especificó el evento del mouse, tratando de capturar si había un clic por parte del jugador en la pantalla; lo primero que verifica es que el clic haya sido en las márgenes asignadas a la pantalla, seguido de esto se pregunta si el clic fue por parte del botón PRIMARY (normalmente el botón izquierdo del mouse), si fue así, él captura la coordenada X y la coordenada Y, después de esto se declaró una línea de tiempo `Timeline`, para que creará una animación de fotogramas desde la parte terrenal de la ciudad hasta la parte (x,y) del clic por parte del mouse, creando así el efecto de un misil por parte de la defensa (de la ciudad).

Retomando la parte de los misiles, cuando un misil impacta sobre un edificio, él programa verifica dentro de cuales pixeles fue la detonación y asigna un 0 al vector de números indicando que ese edificio ya fue destruido.

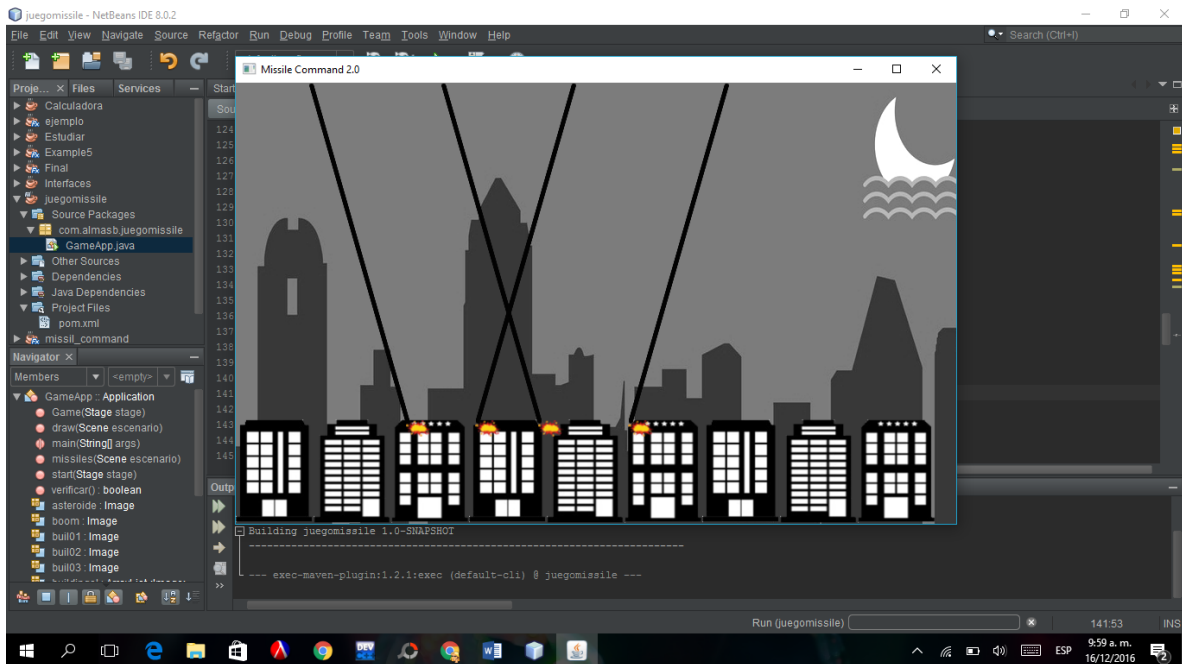
Lentamente, a medida que van cayendo los misiles, el propio juego aumenta de nivel, haciendo que cada vez bajen más misiles para la destrucción completa de la ciudad.

Imágenes de las partes más esenciales del juego.

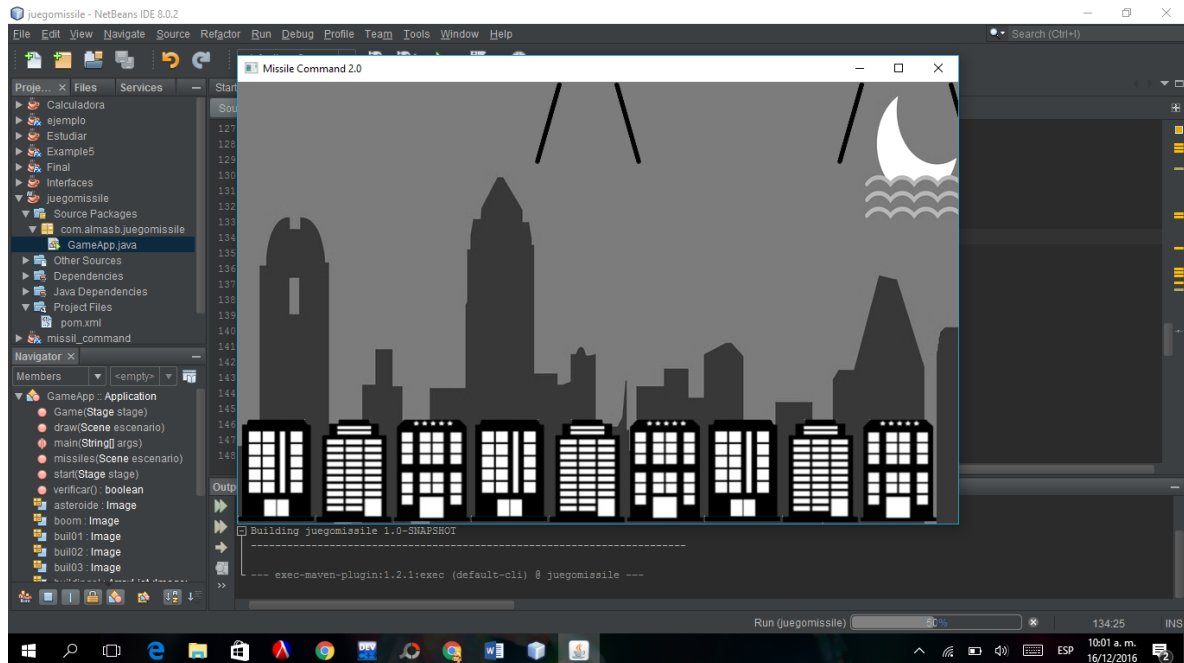
Lanzamiento de misiles



Colisión de misiles



Lanzamiento de misiles



Destrucción de Edificios

