Missile Command

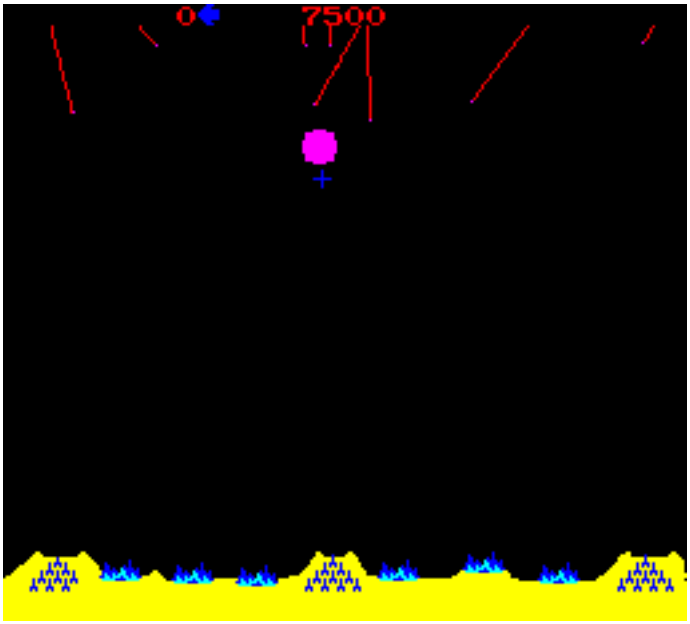
Specification

Implement a Java version of the game Missile Command

- Only hostile missiles (ICBM) and anti-missiles (ABM) has to be implemented
- Missiles do not have to be able to split into sub missile clusters
- Cascading explosions has to be implemented, i.e. expanding missile explosions shall explode enveloped missiles and create new explosions
- Game Over, start screen and points calculation does not have to be included
- The game does not have to become progressively harder

Analysis

In classic Missile Command, the player defended six cities using three anti-missile bases. Each anti-missile base had a finite amount of missiles. Missiles were fired from the base nearest to the intended target.



The player's game ended when all six cities were destroyed.

Full details of the game are found here: http://en.wikipedia.org/wiki/Missile_Command.

Alternative versions of the game have only one anti-missile base defending the six cities.

Components

- ICBM
  - Start from a random point along the top of the screen.
  - Moves toward the bottom of the screen where they explode on impact.
  - Leave a trail behind them.
  - Trail disappears when the ICBM explodes.
- ABM
  - Originate at one of the anti-missile bases
  - Moves toward the intended target then explodes.
  - Leave a trail behind them.
  - Trail disappears when the ABM explodes.
- ABM explosion
  - Expands to a maximum diameter of X before shrinking back to nothing.
  - Any ICBM caught in the blast radius will also explode.
- ICBM explosion
  - Expands to a maximum diameter of Y before shrinking back to nothing.
  - Any ICBM caught in the blast radius will also explode.
- City
  - Explode when impacted by an ICBM
  - Cities are not recreated between levels.
  - At least one city must survive a level for the player to proceed to the next.
  - When all cities are destroyed the game is over.
- Missile Base
  - Explodes when impacted by an ICBM
  - Contain a finite number of missiles.
  - Missiles fired in response to the appropriate key press.
  - When all of a base's missiles have been fired, further requests for missile launches will be ignored.
  - Missile bases destroyed in one level of the game are restored in the next.
- Targeting Crosshair
  - Controlled using the mouse.
  - Determines the target for ABMs.

Implementation

Initial Tasklist

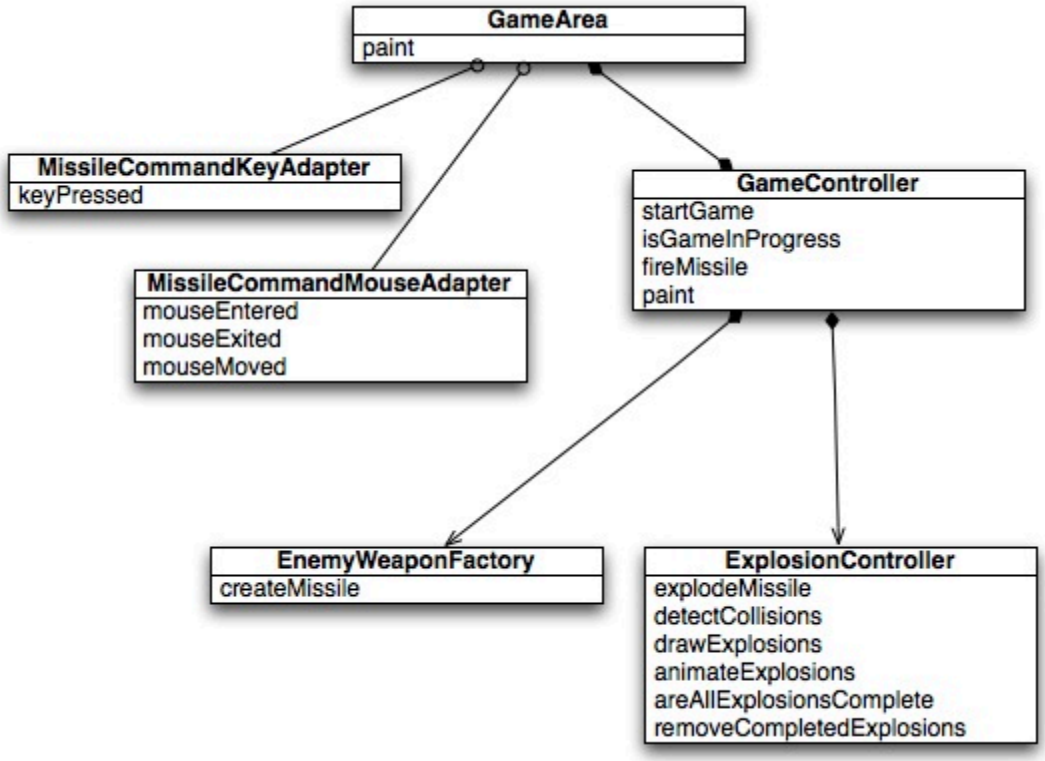
Generate Project Structure	Use maven.
Game Area	A class that manages the game area and the various background elements.
ICBM Movement	Generate an ICBM with an initial starting point somewhere along the top edge of the game area. Animate the ICBM so that it moves from the top to the bottom of the screen. The ICBM should leave a trail behind it.
ICBM Target	ICBM should select one of the targets in the defensive area (cities and missile bases), using the coordinates of the target as the point along the bottom of the screen to which it should aim.
Defensive Area	Add the six cities and three missile bases along the bottom of the game area.
Create Missiles	Missiles should be added to the game at fixed intervals up to a maximum.
ICBM collision with Defensive Area	Detect the collision of an ICBM with the defensive area. Handle the destruction of the cities and anti-missile sites.
ICBM explosion	When an ICBM is caught in an explosion or collides with the defensive area it will explode, expanding to a predefined radius before shrinking to nothing.
Targeting Crosshair	When the mouse pointer is within the game area, the pointer should be represented by a cross hair.
Fire ABM	Fire an ABM from the chosen anti-missile site toward the current position of the targeting crosshair. ABM's are fired using the keyboard – left ('a') , middle ('s') or right ('d').
ABM Explosion/ICBM collision	Any ICBM caught in an explosion's burst radius is destroyed. Remove the ICBM and its trail from the game area.
ABM Explosion	When an ABM reaches its intended target it should explode, expanding to a predefined size before shrinking away to nothing.

Design

NB. All class diagrams are presented in pseudo-UML.

Overview of Key Classes

The following diagram outlines the key classes, including the most important methods, within the project.

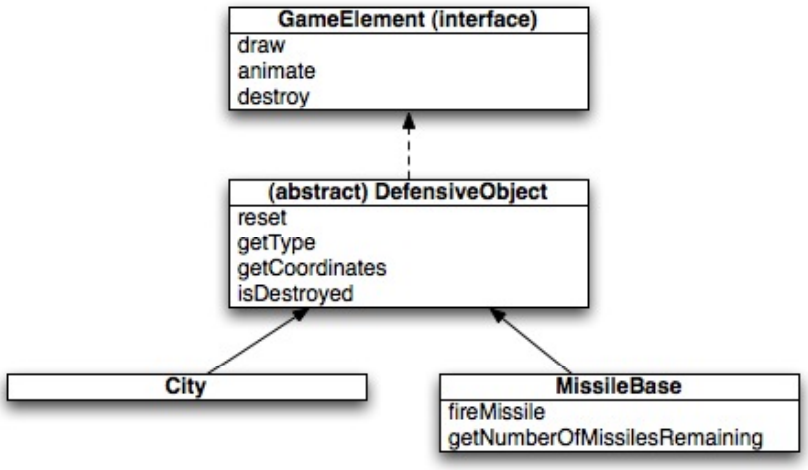


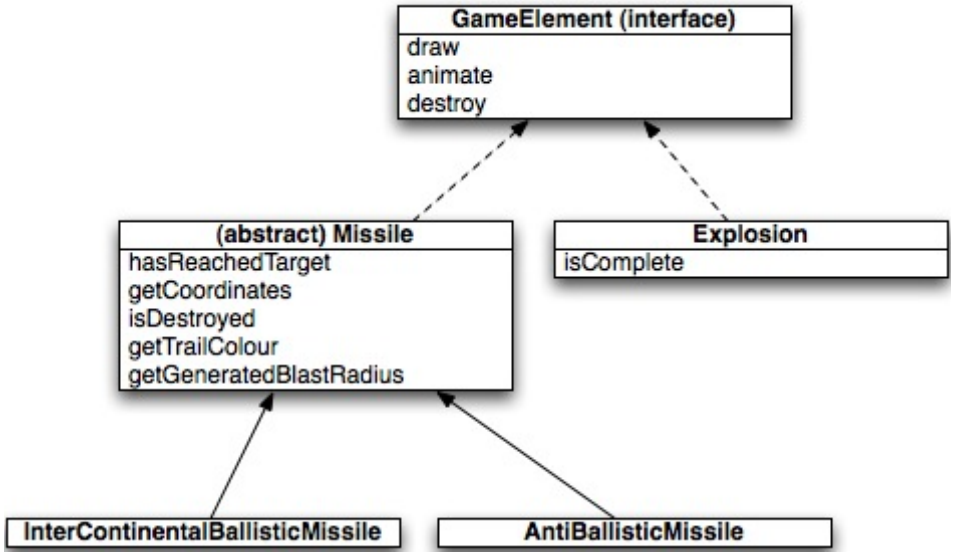
GameArea	Responsible for the creation of the area in which the game is played, the various static elements of the game, e.g. cities, and the interaction between player and game via keyboard and mouse.
MissileCommandKeyAdapter	Captures keypresses and, where relevant, passes them to the GameController
MissileCommandMouseAdapter	Manages the state of the mouse cursor and captures the location of the mouse when it is within the game area.

GameController	Controls the flow of the game. Manages the various non-static elements of the game, e.g. missiles and explosions.
EnemyWeaponFactory	Responsible for the creation of enemy weapons.
ExplosionController	Manages the creation, animation and drawing of Explosions. Also manages the detection of collisions between explosions and other game elements such as missiles, cities and missile bases.

Game Elements

There are two types of game element: static and non-static, i.e. those elements that move across the screen and those that do not. All game elements implement the GameElement interface.




GameElement (interface)
draw
animate
destroy
(abstract) Missile
hasReachedTarget
getCoordinates
isDestroyed
getTrailColour
getGeneratedBlastRadius
Explosion
isComplete
InterContinentalBallisticMissile
AntiBallisticMissile

Future Improvements

- User Instructions
  - Display ‘press spacebar to start’ message.
  - ‘Game Over’ message.
  - ‘How to play’
- Demo mode
  - Demo of game to play until user starts the game.
- Scoring
  - Award points for destruction of incoming missiles.
  - Bonus points for cities and missile bases still standing at the end of each level.
  - Display current score at the top of the screen.
- High Score
  - Record and store the high score. This should persist across executions of the game.
- Game Levels
  - Should be more structured, with a more gentle increase in difficulty.
- Look and Feel
  - Give the player some indication of missiles remaining in a missile base.
  - Replace the various elements at the bottom of the screen (land, cities, missile bases) with custom graphics.
  - Etc …too many to detail!