Earth Defence Force Elite Asteroid Division

Table of contents

Introduction

Earth Defence Force: Elite Asteroid Division is based on the popular arcade game from the 1980s called Missile Command by Atari Inc. The concept of the arcade game EDF:EAD, is for the player to stop the asteroids that rain from the sky from destroying the cities below. The player will be able to stop the asteroids from hitting the cities by firing a missile directly at the asteroid to be destroyed before hitting the cities. The game is played by moving the mouse and clicking on the anticipated location of the asteroids and firing a missile directly at the asteroids. The game has a scoring system: 100 points are gained for each asteroid shot down, 100 points are lost for each city hit, and 300 points are lost for each city destroyed. This scoring system is enhanced with a high scores system as well. After each game, a locally-saved file is read and checked to see if the most recent game has set a new high score. It updates the file if necessary, and displays the high score screen. The game also features difficulty modes: easy, medium and hard, which change how many shots the player can fire at once as well as the frequency that the meteors rain down. Pressing the escape key during gameplay pauses the game and brings up the menu, and pressing it at the menu closes the game.

What's new

This is the initial release of Earth Defence Force: Elite Asteroid Division. Therefore, everything is new. Enjoy!

System requirements

CPU: Intel Pentium D or AMD Athlon 64 (K8) 2.6 GHz
RAM: 2GB
GPU (Integrated): Intel HD Graphics or AMD (formerly ATI) Radeon HD Graphics with DirectX 9
GPU (Discrete): Nvidia GeForce 9600 GT or AMD Radeon HD 2400 with DirectX 9
Disk space: 200MB

Getting help

If you find yourself needing assistance, feel free to read through this help document. But you were probably doing that already anyways.

How to play

The gameplay of Earth Defence Force: Elite Asteroid Division is fairly straightforward.

1. Click the start button
2. Select your difficulty
3. Aim with the mouse
4. Click to fire a missile at your mouse location
5. Shoot down the meteors before all the cities are destroyed

Difficulty levels

Earth Defence Force: Elite Asteroid Division has three difficulty levels: Easy, Medium, and Hard.

Easy:

You can fire four missiles at a time. As they go offscreen you can fire more.
Meteors spawn at half speed.

Medium:

You can fire two missiles at a time. As they go offscreen you can fire more.
Meteors spawn at normal speed.

Hard:

You can fire only one missile at a time. If you miss you need to wait for it to go offscreen before firing another.
Meteors spawn faster than normal.

Scoring

For each meteor you shoot down, you get 100 points.
For each city hit, you lose 100 points.
For each city destroyed, you lose 300 points.

Your score can go into the negatives if you let too many meteors through.

A high score screen is accessible from the main menu, showing high scores.
If you set a new high score, your score will bump the others down.

Technical Information

This game was made with XNA 4.0 for Windows and Microsoft Visual Studio 2013.
The game classes are outlined in more detail in subtopics, if you are so inclined to read them.

Classes

Classes and Their Purposes

Button.cs

A class for buttons on menu screens. It draws a rectangle of a given colour along with the text supplied of a given font. Contains click listener in Update() method and isClicked boolean property for triggering code when the button is clicked.

City.cs

A class for each city. Contains frames for each destruction state, which are changed when the city is hit by a meteor. Contans methods for checking if the city was hit, or destroyed, as well as a getRect() method for collision detection.

CollisionManager.cs

A class for keeping track of all collisions. Deletes shots when they go offscreen, spawns explosions and

deletes objects when they blow up. Also calls ScoreManager methods to keep track of scores on object collision.

Explosion.cs

A class for generating animated explosions. Takes the position of the explosion, and depending on the location, either draws an airburst explosion if it's above a given height, or a mushroom cloud explosion if it is below. This class also plays an explosion sound on every explosion.

Game1.cs

Main game class that ties everything together. Contains all the textures, objects, initializations, and so on and so forth. Its update() method calls all the necessary object's update methods, Draw() calls draws, and so on.

Meteor.cs

A class for each animated meteor. Has a start point (random spot on the top of the screen), a random end point (at the bottom), and a random , normalised, speed. The meteors are spawned off-screen and just move along their speed vector. They are removed in the CollisionManager class.

Program.cs

The basic XNA game wrapper.

ScoreManager.cs

A class to keep track of current game scores which also contains the methods for reading and writing high scores to file.

ScrollingBackground.cs

A class that allows the backgrounds of the game to Parallax Scroll. This is a technique where the background image moves slower than foreground image, which will create a illusion of depth and 2 dimensional scene adding to the realization of the game. This will allow to instantiate multiple parallax scrolling backgrounds throughout the game. This game uses three parallax scrolling backgrounds to create the illusion of moving planetary stars.

Shot.cs

A class for each missile shot by the turret. They are angled and shoot towards where the mouse click occurred, and make a sound when created. The speed is randomly generated within a set range, and is normalised. The shots, like meteors, merely move along their speed vector until the CollisionManager disposes of them.

TextScreen.cs

A class for any menu screen that needs to display text. It takes a string array as input, uses the first item as the title, and iterates through the rest to draw the text to the screen. This is used for the How To Play, About, Help, and High Scores screens.

Turret.cs

A class for the turret object. Update() method positions the turret to face the mouse cursor. Clicking generates a new Shot object.

Class relationship diagram

Class Relationship Diagram of Earth Defence Force: Elite Asteroid Division

cd UMLClassDiagram1
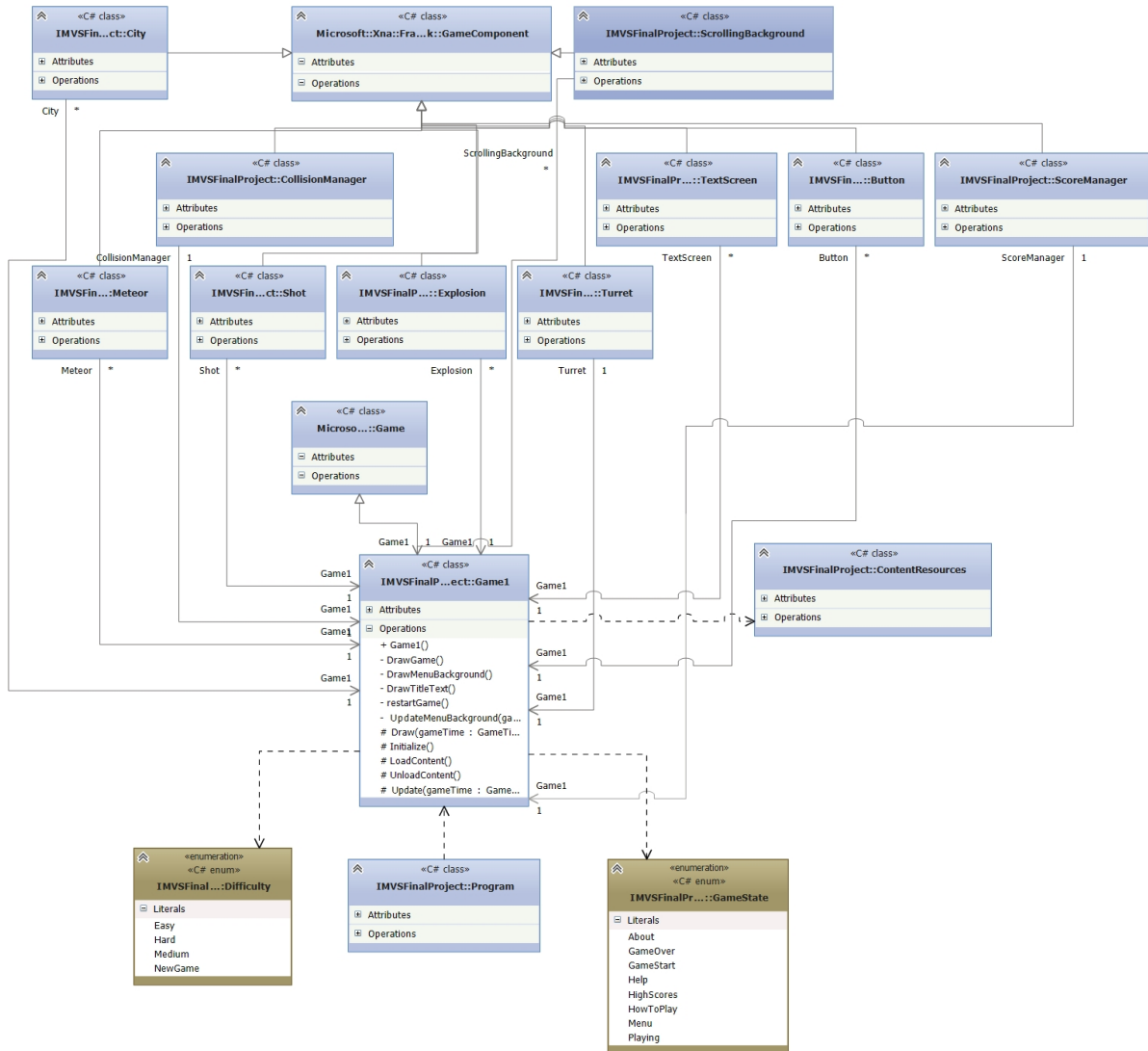
Class attributes diagram

Class Attributes and Methods Diagram of Earth Defence Force: Elite Asteroid Division

The attributes didn't fit inside the relationships diagram, so they are included here without relationships.

cd AttributeAndMethodDiagram

«C# class» IMVSFin...ct::City

Attributes
- + Hits : Integer
- + Position : Vector2
- + Texture : Texture2D
- - dimension : Vector2
- - frameIndex : Integer
- - FRAMENUMBER
- - frames : List<Recta...
- - hits : Integer
- - MAX_HITS
- - position : Vector2
- - screenSize : Vector2
- - spriteBatch : SpriteB...
- - texture : Texture2D
- - xPosition : Integer

Operations
- + City(game : Game,...
- + Draw()
- + GetRect() : Rectangle
- + GotHit()
- + Initialize()
- + IsDestroyed() : Bool...
- + Update(gameTime :...
- - createFrames()

«C# class» IMVSFinalProject::CollisionManager

Attributes
- + NewCityHit : Integer
- + NewMeteorsShotDown : Integer
- - cityList : List<City>
- - explosionList : List<Explosion>
- - explosionSoundEffect : SoundEffect
- - game : Game1
- - meteorList : List<Meteor>
- - newCityHit : Integer
- - newMeteorsShotDown : Integer
- - screenSize : Vector2
- - shotList : List<Shot>

Operations
- + CollisionManager(game : Game, shotLis...
- + Initialize()
- + Update(gameTime : GameTime)

«C# class» IMVSFinalP...::Explosion

Attributes
- + AnimationOver : Boolean
- - AIR_COLUMNS
- - AIR_ROWS
- - animationOver : Boolean
- - asteroidExplosionInstance : ...
- - delay : Integer
- - delayCounter : Integer
- - dimension : Vector2
- - explosionSoundEffect : Sou...
- - frameIndex : Integer
- - frames : List<Rectangle>
- - GROUND_COLUMNS
- - GROUND_ROWS
- - inAir : Boolean
- - origin : Vector2
- - position : Vector2
- - soundPlayed : Boolean
- - spriteBatch : SpriteBatch
- - texture : Texture2D

Operations
- + Draw()
- + Explosion(game : Game, sp...
- + Initialize()
- + Update(gameTime : Game...
- - createFrames()

«C# class» IMVSFin...:Meteor

Attributes
- + Dimension : Vector2
- + Position : Vector2
- + Texture : Texture2D
- - delay : Integer
- - delayCounter : Integer
- - dimension : Vector2
- - frameIndex : Integer
- - FRAMENUMBER
- - frames : List<Recta...
- - hitPoint : Integer
- - line : List<Rectangle>
- - origin : Vector2
- - position : Vector2
- - rand : Random
- - rotation : Single
- - speed : Vector2
- - spriteBatch : SpriteB...
- - texture : Texture2D

Operations
- + Draw()
- + GenerateSpeed(scre...
- + GetRect() : Rectangle
- + Initialize()
- + Meteor(game : Gam...
- + RandomX(screenSiz...
- + Update(gameTime : ...
- - createFrames()
- - createLine(screenSiz...

«C# class» IMVSFin...ct::Shot

Attributes
- + Position : Vector2
- + Texture : Texture2D
- - delay : Integer
- - delayCounter : Integer
- - dimension : Vector2
- - frameIndex : Integer
- - FRAMENUMBER
- - frames : List<Recta...
- - origin : Vector2
- - position : Vector2
- - rotation : Single
- - screenSize : Vector2
- - shotSoundEffect : So...
- - soundPlayed : Boolean
- - speed : Vector2
- - spriteBatch : SpriteB...
- - target : Vector2
- - texture : Texture2D
- - turret : Turret

Operations
- + Draw()
- + GetRect() : Rectangle
- + Initialize()
- + Shot(game : Game,...
- + Update(gameTime : ...
- - createFrames()

«C# class» IMVSFinalProject::ScrollingBackground

Attributes
- - backGroundRectangle : Rectangle
- - backGroundTexture : Texture2D
- - position : Vector2
- - positionOne : Vector2
- - positionTwo : Vector2
- - speed : Vector2
- - spriteBatch : SpriteBatch

Operations
- + Draw()
- + Initialize()
- + ScrollingBackground(game : Game, spriteBatch : ...
- + Update(gameTime : GameTime)

«C# class» IMVSFinalPr...::TextScreen

Attributes
- - bodyFont : SpriteFont
- - headingFont : SpriteFont
- - messages : String[*]
- - spriteBatch : SpriteBatch
- - titlePos : Vector2

Operations
- + Draw()
- + Initialize()
- + TextScreen(game : Game, spr...
- + Update(gameTime : GameTi...

«C# class» IMVSFin...::Button

Attributes
- + Colour : Color
- + IsClicked : Boolean
- - buttonRect : Rectan...
- - colour : Color
- - isClicked : Boolean
- - message : String
- - padding : Integer
- - position : Vector2
- - spriteBatch : SpriteB...
- - spriteFont : SpriteFont
- - texture : Texture2D

Operations
- + Button(game : Gam...
- + Draw()
- + Initialize()
- + Update(gameTime : ...

«C# class» IMVSFinalProject::ScoreManager

Attributes
- + BoolHighScoresRun : Boolean
- + CurrentScore : Integer
- + ScoreMessages : String[*]
- - boolHighScoresRun : Boolean
- - cm : CollisionManager
- - currentScore : Integer
- - maxHighScores : Integer
- - scoreMessages : String[*]
- - spriteBatch : SpriteBatch
- - spriteFont : SpriteFont
- - textHighScoresNew : String[*]
- - textHighScoresOld : String[*]
- - theFileRead : FileStream
- - theFileWrite : FileStream
- - theScoreRead : StreamReader
- - theScoreWrite : StreamWriter

Operations
- + AddPoints(pointsToAdd : Integer)
- + Draw()
- + Initialize()
- + ScoreManager(game : Game, spriteB...
- + Update(gameTime : GameTime)
- ~ GetHighScores()
- ~ ReadHighScores()

«C# class» IMVSFinalProject::ContentResources

Attributes
- + air_explosion : Byte[*]
- + AstroidExplosion : Byte[*]
- + backgroundSky_double_flipped : Byte[*]
- + BodyFont : Byte[*]
- + buildings_bigger_2048 : Byte[*]
- + city_small_sprite : Byte[*]
- + Culture : CultureInfo
- + Dramatic_Music : Byte[*]
- + ground_explosion : Byte[*]
- + HeadingFont : Byte[*]
- + MenuScreenSoundV1 : Byte[*]
- + MissileLaunch : Byte[*]
- + NuclearAlarm : Byte[*]
- + nuclear_explosion_sound_effect : Byte[*]
- + nuclear_holocaust : Byte[*]
- + ResourceManager : ResourceManager
- + rocket_spritesheet_vertical : Byte[*]
- + stars : Byte[*]
- + trail_4 : Byte[*]
- + turret_mount_new : Byte[*]
- + turret_sheet : Byte[*]
- + whitePixel : Byte[*]
- - resourceCulture : CultureInfo
- - resourceMan : ResourceManager

Operations
- ~ ContentResources()

«C# class» IMVSFinal...::Program

Attributes

Operations
- - Main(args : String[*])

«enumeration» «C# enum» IMVSFinalP...t::Difficulty

Literals
- Easy
- Hard
- Medium
- NewGame

«C# class» IMVSFinalProject::Game1
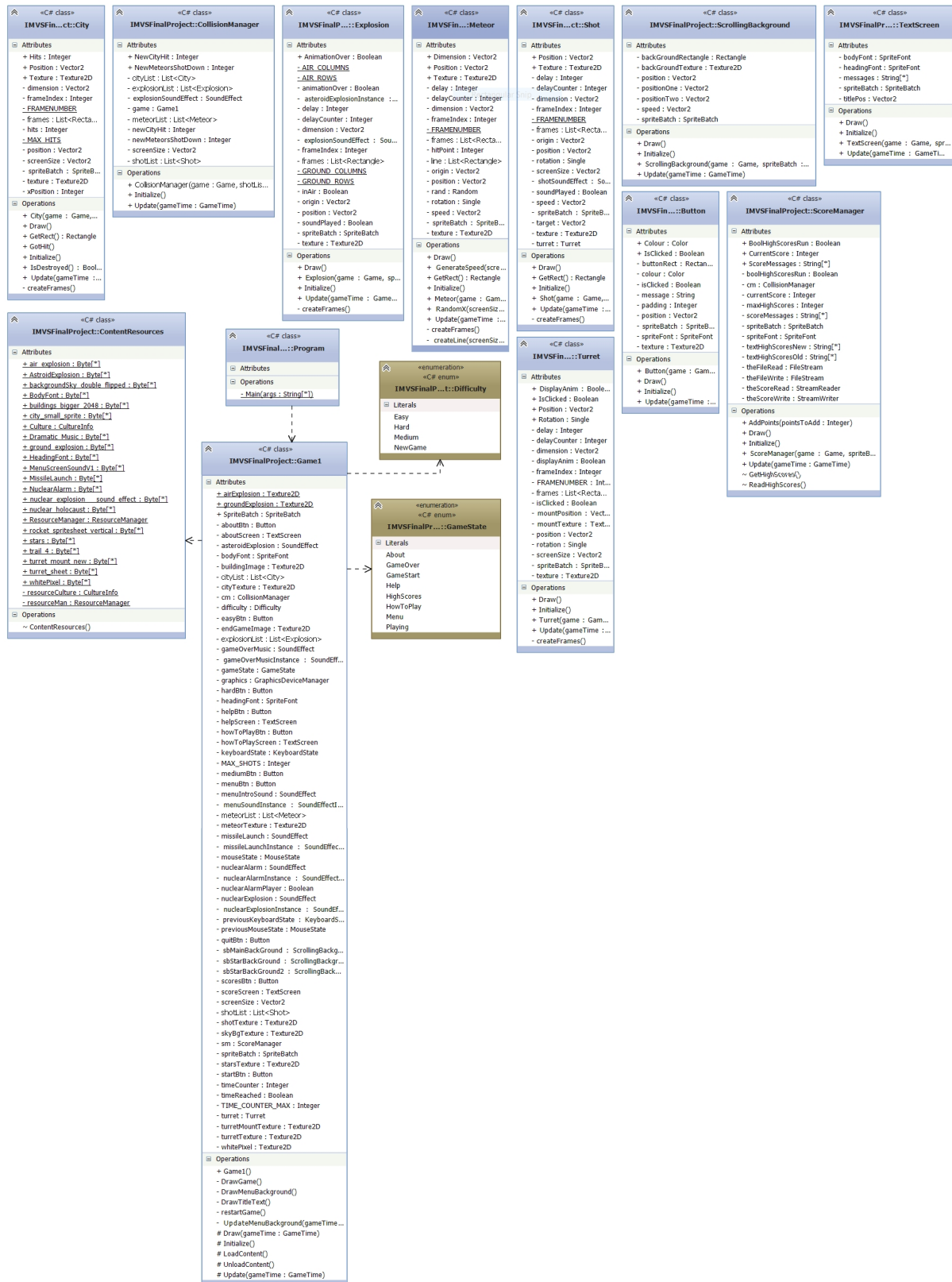
Attributes
- + airExplosion : Texture2D
- + groundExplosion : Texture2D
- + SpriteBatch : SpriteBatch
- - aboutBtn : Button
- - aboutScreen : TextScreen
- - asteroidExplosion : SoundEffect
- - bodyFont : SpriteFont
- - buildingImage : Texture2D
- - cityList : List<City>
- - cityTexture : Texture2D
- - cm : CollisionManager
- - difficulty : Difficulty
- - easyBtn : Button
- - endGameImage : Texture2D
- - explosionList : List<Explosion>
- - gameOverMusic : SoundEffect
- - gameOverMusicInstance : SoundEff...
- - gameState : GameState
- - graphics : GraphicsDeviceManager
- - hardBtn : Button
- - headingFont : SpriteFont
- - helpBtn : Button
- - helpScreen : TextScreen
- - howToPlayBtn : Button
- - howToPlayScreen : TextScreen
- - keyboardState : KeyboardState
- - MAX_SHOTS : Integer
- - mediumBtn : Button
- - menuBtn : Button
- - menuIntroSound : SoundEffect
- - menuSoundInstance : SoundEffectI...
- - meteorList : List<Meteor>
- - meteorTexture : Texture2D
- - missileLaunch : SoundEffect
- - missileLaunchInstance : SoundEffec...
- - mouseState : MouseState
- - nuclearAlarm : SoundEffect
- - nuclearAlarmInstance : SoundEffect...
- - nuclearAlarmPlayer : Boolean
- - nuclearExplosion : SoundEffect
- - nuclearExplosionInstance : SoundEf...
- - previousKeyboardState : KeyboardS...
- - previousMouseState : MouseState
- - quitBtn : Button
- - sbMainBackGround : ScrollingBackg...
- - sbStarBackGround : ScrollingBackgr...
- - sbStarBackGround2 : ScrollingBack...
- - scoresBtn : Button
- - scoreScreen : TextScreen
- - screenSize : Vector2
- - shotList : List<Shot>
- - shotTexture : Texture2D
- - skyBgTexture : Texture2D
- - sm : ScoreManager
- - spriteBatch : SpriteBatch
- - starsTexture : Texture2D
- - startBtn : Button
- - timeCounter : Integer
- - timeReached : Boolean
- - TIME_COUNTER_MAX : Integer
- - turret : Turret
- - turretMountTexture : Texture2D
- - turretTexture : Texture2D
- - whitePixel : Texture2D

Operations
- + Game1()
- - DrawGame()
- - DrawMenuBackground()
- - DrawTitleText()
- - restartGame()
- - UpdateMenuBackground(gameTime...
- # Draw(gameTime : GameTime)
- # Initialize()
- # LoadContent()
- # UnloadContent()
- # Update(gameTime : GameTime)

«enumeration» «C# enum» IMVSFinalPr...::GameState

Literals
- About
- GameOver
- GameStart
- Help
- HighScores
- HowToPlay
- Menu
- Playing

«C# class» IMVSFin...::Turret

Attributes
- + DisplayAnim : Boole...
- + IsClicked : Boolean
- + Position : Vector2
- + Rotation : Single
- - delay : Integer
- - delayCounter : Integer
- - dimension : Vector2
- - displayAnim : Boolean
- - frameIndex : Integer
- - FRAMENUMBER : Int...
- - frames : List<Recta...
- - isClicked : Boolean
- - mountPosition : Vect...
- - mountTexture : Text...
- - position : Vector2
- - rotation : Single
- - screenSize : Vector2
- - spriteBatch : SpriteB...
- - texture : Texture2D

Operations
- + Draw()
- + Initialize()
- + Turret(game : Gam...
- + Update(gameTime : ...
- - createFrames()

Attribution

The assets for this game were borrowed from a number of sources:

Images:

Turret Stand:

https://pixabay.com/en/tower-gun-weapon-turret-cannon-35984/

Air Explosion:

http://makingof.astrobabe.net/2013_10_01_archive.html

Turret:

http://www.moddb.com/games/nebulawars/news/animated-turrets

Missiles:

http://spritedatabase.net/file/5321

Meteors:

http://meteorsspriterstorage.smackjeeves.com/comics/740222/my-sprites/

Buildings:

http://www.freelogovectors.net/skyscrapers-vectors/

Flames on buildings:

http://imgkid.com/fire-sprite-animation.shtml

Ground Explosion:

http://www.neogaf.com/forum/showthread.php?t=1110593

Sky Background:

http://montco.happeningmag.com/rediscover-night-sky-mcccs-observatory/night-sky-hd-wallpaper-2/

Skyline Silhouette:

http://www.clipartbest.com/chicago-skyline-silhouette-free

Stars cut from cover image of:

https://www.youtube.com/watch?v=oWA38bbmfGI

End Game Image:

http://armageddononline.org/tag/nuclear-holocaust/

Sounds:

Missile Launch Sound:
https://www.youtube.com/watch?v=bGgvuDyIm6g

Nuclear alarm sound:
https://www.youtube.com/watch?v=c9uwhuWRpY

Explosion sound:
https://www.youtube.com/watch?v=1spZBEFTexI

Game Over Explosion Sound:
https://www.youtube.com/watch?v=byyhFHgbrG0

Background music:
https://www.youtube.com/watch?v=BeEDQMLtyGY

Dramatic Game Ending sound:
https://www.youtube.com/watch?v=c7uNShVD4Hk

Code:

Code concepts for XNA game menu and button class:
http://www.spikie.be/blog/page/Building-a-main-menu-and-loading-screens-in-XNA-Page-3.aspx

High scores code adapted from XNA 4 Course Ch 17 by VideoGameDev:
https://www.youtube.com/watch?v=WfRjGnEEM3M