

# *Defender Computer Game Specification Document*

---

Barry Belcher and Aaron Curley  
DirectX 2D Computer Game Project  
CIS 587 Computer Game Design and Implementation I  
Professor: Bruce Maxim  
10/29/2007  
Revision 1

**Revision Sheet**

---

Revision	Date	Brief Summary of Changes
Draft Version A	10/27/2007	Changes reviewed and Accepted
Version 1.0	10/29/2007	Final submittal

<b>1. ASSIGNMENT OVERVIEW .....</b>	<b>1</b>
<b>2. EXECUTIVE SUMMARY .....</b>	<b>1</b>
2.1 BACKGROUND STORY .....	1
2.1.1 Overview of Game.....	1
2.1.2 Background Story.....	1
<b>3. GAME PLAY LOOK AND FEEL .....</b>	<b>2</b>
3.1 APPEARANCE .....	2
3.1.1 Spacecraft of Defender .....	3
3.2 PLAYERS ROLES AND ACTIONS .....	6
3.3 STRATEGIES AND MOTIVATIONS .....	7
3.4 LEVEL SUMMARY/STORY PROGRESSION.....	9
<b>4. DEVELOPMENT SPECIFICATIONS .....</b>	<b>10</b>
4.1 SYSTEM REQUIREMENTS .....	10
4.2 DEVELOPMENT SPECIFICATIONS.....	10
4.3 ALGORITHM STYLE .....	10
4.3.1 Graphics and Sounds .....	10
<b>5. DISCLAIMER.....</b>	<b>15</b>
<b>6. REFERENCES.....</b>	<b>16</b>
6.1 GRAPHICS REFERENCES .....	16
6.2 SOUND REFERENCES .....	16
6.3 MISCELLANEOUS.....	16



## **1. Assignment Overview**

In this assignment, a complete 2D DirectX computer game is designed and implemented. Included is reasonable storyline for its genre, in this case, an arcade-style shooter. This document describes the various game components from background story and game look and feel to the software/hardware requirements.

## **2. Executive Summary**

### **2.1 Background Story**

#### **2.1.1 Overview of Game**

The name of the game is Defender, a 2-D computer arcade shooter game about you, the defender, given a mission of delivering vital supplies to your Insectoid colony world, to counteract the effects of genocidal weapons used by the dreaded R.A.I.D. Foundation.

#### **2.1.2 Background Story**

The story takes place in the Milky Way-Andromeda galaxy millions of years from now as the two galaxies began to merge. The tidal forces of the galaxial merge created many tunnels in space that were discovered to be navigable by your ancestral home Insectoid world, Xandor, allowing for space travel at untold distances and to previously unreachable worlds. Theirs' was a period of prosperity, and the Insectoid race flourished, colonizing many new worlds, including Xanthu VII, your place of birth, on the outskirts of the Xandor Consortium Federation of Worlds.

Unfortunately for you, Earth also discovered the space-tunnels and too was in a period of expansion. The SETI project had long since been abandoned, never having received a signal of other life, so it was thought that humans were the only intelligent life. Being an expansionist race, humanity soon took advantage of "space-tunnel" travel, expanding into the outer reaches of space.

Twenty years before the timeline of "Defender", humans and insectoids made first contact. Although the Insectoids had never met humans before, they were excited by the encounter of a new species and naively sent a fleet of ships to greet the humans in peace and friendship. However, humans had been raised to dislike insects and mistook the approaching fleet as an act of aggression and immediately opened fire. This started the Xando-Earth war, a war that has lasted 19 years. Now, however, both sides have reached an uneasy peace, although tensions are at astronomical levels.

Not everyone agreed with ending the war, especially many humans that did not believe in negotiating with the "bugs". In their view, bugs are to be exterminated, not negotiated with in any form, thus were thought of as things to be destroyed. The radical group of "bug haters" formed the Race Against Insectoid Dominion or R.A.I.D. for short. The foundation has many

followers, and through many secret projects and covert operations, they assembled a fleet and developed an Insectoid bio-weapon. The foundation then decided to secretly test this weapon on planet XR-271-B, which is Xanthu VII, your home world.

The fleet has deployed the weapon with devastating effects. To save its people, the Xanthu VII government has employed your services to deliver vital supplies to neutralize the effects of the bio-weapon, while destroying as many spacecraft as possible to clear the way for other supply vessels (not shown in the game). The game begins as you are on your way back to Xanthu VII and must blast through the R.A.I.D. fleet to save your people.

### **3. Game Play Look and Feel**

#### **3.1 Appearance**



**Figure 1. Defender Concept Screen**

The sample screen above illustrates the look of Defender. You will play the role of an Insectoid pilot in a blockade runner delivering vital supplies to your embattled home world of Xanthu VII. The game is a vertical-scrolling arcade style shooter. The enemy ships will approach from the

top of the screen firing various weapons at our hero. There are three types on enemies; fighters, fighter-bombers, and bombers.

### **3.1.1 *Spacecraft of Defender***

#### **3.1.1.1 Insectoid Blockade Runner**



**Figure 2. Insectoid Blockade-Runner**

##### **3.1.1.1.1 Blockade Runner Details**

Hull Rating: 100

Weapons

- Primary: Dual Lasers - Do 60 points of damage
- Secondary: Guided Rocket – Does 290 points of damage
  - Capacity: Limited

The player (You) pilot an Insectoid Blockade Runner, especially designed to blast through enemy ships. The blockade runner is stronger than any of the enemy ships and is equipped with a dual laser system and guided rockets. The dual-forward firing laser system has a high-rate of fire, more powerful than energy bullets and photon bombs. The most powerful weapon in the player's arsenal is the rocket that only blockade runners carry. A hit from one of these will destroy virtually every craft in the game, but the ship can only carry a limited number. Once fired, it will target the nearest ship in front of the blockade runner. There is no way to acquire more rockets so the player must take this into consideration.

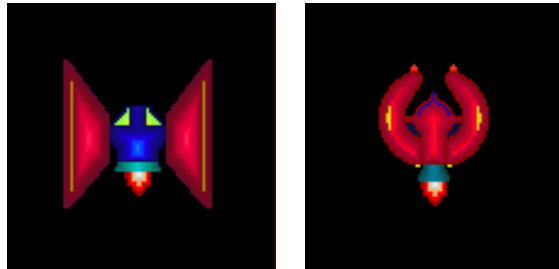
##### **3.1.1.1.2 Information on Enemy Ship Properties**

Each enemy ship has certain “properties” that are hard coded into the game. These properties allow the “difficulty” of the enemy ship to be adjusted. These properties, and their meanings, are listed below:

- **Health/Armor Rating** - The initial amount of health enemy ships of this type have. Each time the ship is hit with a weapon, the weapon's damage amount is subtracted from the current health of the ship. Once the health reaches zero, the ship explodes.
- **Kill Score** - The number of points the player is awarded for killing this ship.
- **Fire Probability** - The probability that this weapon will actually be fired every reload cycle. All ships classes have a fire probability for both Photons and Bullets. Note, that since some ships classes do not fire a certain weapon, their fire probabilities for that weapon are 0%.
- **Reload Time** - The amount of time that must elapse between checks if the ship should fire. Every time this amount of time here has elapsed, the ship either fires the weapon or does not fire, depending on the result of the probability-based decision. Even if the ship

does not fire, the reload timer is reset, so the ship must wait until the next cycle to again do the fire probability check.

### 3.1.1.3 R.A.I.D. Attack Fighters



**Figure 3. Attack Fighters (A.K.A. F1 and F2)**

Fighters are the fastest, yet weakest of the enemy craft in the game. Enemy fighters can only fire energy bullets. Energy bullets do the least amount of damage per shot in the game, but should not be taken too lightly by the player since clusters of energy bullets can easily prove to be lethal.

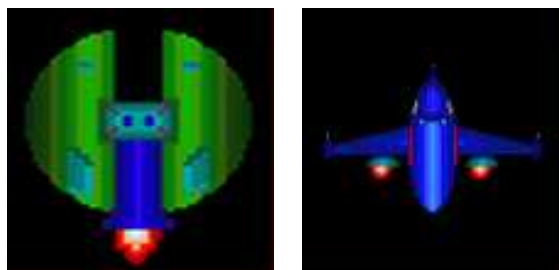
#### 3.1.1.3.1 R.A.I.D. Fighter (F1) Details

Armor Rating	Kill Score	Weapon Damage	Fire Probability	Reload Time
100	100	10 points/energy bullet	60%	1 Second

#### 3.1.1.3.2 R.A.I.D. Fighter (F2) Details

Armor Rating	Kill Score	Weapon Damage	Fire Probability	Reload Time
175	200	10 points/energy bullet	60%	500ms

### 3.1.1.4 R.A.I.D. Superiority Fighter-Bombers



**Figure 4. Superiority Fighter-Bombers (A.K.A. FB1 and FB2)**



Fighter-Bombers are the general purpose spacecraft. In addition, fighter-bombers can take more damage than fighters. Fighter-bombers can also fire energy bullets and photon bombs.

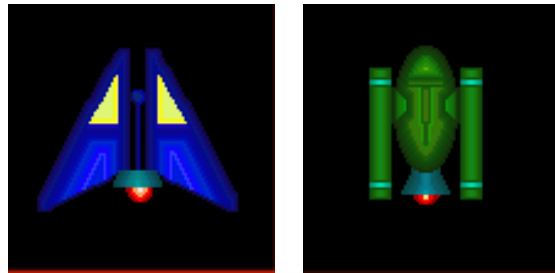
**3.1.1.4.1 R.A.I.D. Fighter-Bomber (FB1) Details**

Armor Rating	Kill Score	Weapon Damage	Fire Probability	Reload time
200	200	10 points/energy bullet 30 points/photon bomb	40% - Energy Bullets 20% - Photons	1 second

**3.1.1.4.2 R.A.I.D. Fighter-Bomber (FB2) Details**

Armor Rating	Kill Score	Weapon Damage	Fire Probability	Reload time
250	300	10 points/energy bullet 30 points/photon bomb	40% - Energy Bullets 20% - Photons	500ms

**3.1.1.5 R.A.I.D. Heavy Bombers**



**Figure 5. Heavy Bombers (A.K.A. B1 and B2)**

Bombers are the heavy-hitters purpose craft. Bombers only fire photon bombs, but at a faster rate than fighter-bombers, making them lethal when attacking in groups.

**3.1.1.5.1 R.A.I.D. Bomber (B1) Details**

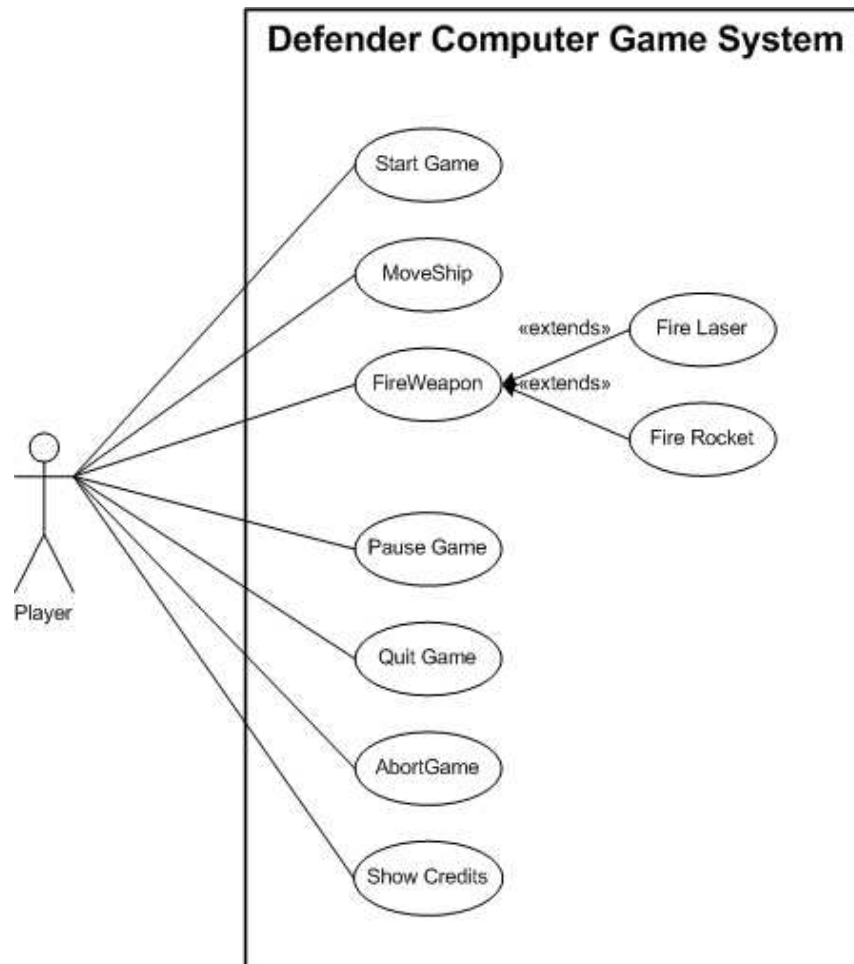
Armor Rating	Kill Score	Weapon Damage	Fire Probability	Reload time
200	300	30 points/photon bomb	60%	1 second

**3.1.1.5.2 R.A.I.D. Bomber (B2) Details**

Armor Rating	Kill Score	Weapon Damage	Fire Probability	Reload time
340	400	30 points/photon bomb	60%	500 ms

### 3.2 Players Roles and Actions

As stated earlier, you are an Insectoid that pilots a blockade runner delivering vital supplies to Xanthu VII, to aid the résistance against the R.A.I.D. attack fleet. The diagram below illustrates the actions available to the player.



**Figure 6. Player Use Case Diagram**

- **Start Game** – The user presses ENTER or space bar on the intro screen to dismiss the introduction and start playing the game.
- **Move Ship** – The player moves the ships via the arrow keys allowing for 8-directional movement of the ship on screen. However, the ship always faces forward (towards the top of the screen) since it is imperative the supplies must reach Xanthu VII. To move the ship the following keys are used.
  - Up: Up arrow or ‘W’
  - Down: Down arrow or ‘S’
  - Left: Left arrow or ‘D’
  - Right: Right arrow or ‘A’

- **Fire Weapon** – The player has two types of weapons at his/her disposal; dual forward-firing lasers and rockets. To fire the lasers, press the space bar. There is unlimited laser ammo, so it is expected to be the main player weapon. Rockets however are more powerful than lasers, but limited in number. To launch a rocket the player presses and releases the ENTER key.
- **Pause Game** - To temporarily halt the game, the player presses the “P” key. The game will display “Game Paused” at the center of the screen. Switching away from the game will place the game in a paused state upon return.
- **Quit Game** – To stop playing Defender, the player can quit by pressing the ESC key.
- **Abort Game** – If however, the game hangs up (never good and must be avoided), the player can abort the game by selecting CTRL-ALT-DEL keys simultaneously. In addition, in the event of certain errors, the player can end the game by following the on-screen instructions. Again, game errors must be avoided.
- **Show Credits** – The screen can either be displayed on demand by the player by pressing the ‘C’ key or will display automatically when the player beats the game or the game ends.

### **3.3 Strategies and Motivations**

The currently thinking of the game design is to have three levels of game play, with a clear player objective of survival. However, due to the limitation of time, no “boss” could be added for our game. Instead, the player will have to survive fourth level of heavy enemy bombardment before reaching Xanthu VII.

To add to the realism of the game, real space images were used. Bright, colorful, high resolution graphics were selected with a goal for sharper game appearance. The 1024 x 768 resolution of the game is a very common resolution supported by virtually all modern graphic cards.

The player will be motivated to continue playing the game by starting off fairly easy allowing the player to quickly learn how to play the game. As the player continues game play, the enemy becomes more difficult to destroy, but should not overwhelm the player. The sound effects add to the enjoyment of the game, while not being a distraction. The controls are arranged to allow for ease of use, using commonly used keys for most functions. Lastly, there is a clear player goal of survival in the game. It is the also a game design goal to motivate the player to replay the game and each level.

The strategy will vary for each player. The player may choose for example to attack fighter first due to their speed or may go after bombers to avoid an onslaught of photon bombs. Since rockets are limited the player must decide when best to use them and on what target.

On the game design side, the enemy will appear off screen and attack in a series of pattern and varying velocities. The first level will be fairly easy to master, but the later levels will be progressively harder. The primary goal of each level is survival, but on later levels, the player will soon discover that unless certain enemy ships are destroyed quickly, the screen is soon populated with too many bullets and photons to realistically avoid. To add to the game play, the enemy ships will also be worth different points. Because of this, the player may choose to win the game as quickly as possible or try to beat the game while killing as many enemy ships as possible, achieving maximum points.

The player will be given four lives to complete the game. When a player is destroyed, another ship will materialize after a number of seconds. During the period of time that the player is “dead” and waiting for rematerialization, new enemy ships will stop entering the screen, to give the player some “breathing room” and a chance to get oriented before the level continues.

### **3.4 Level summary/Story progression**

There are three levels to the game plus a single final “assault” level. Completion of each level requires the player to survive attacks by a number of enemy spacecraft. Each following level becomes progressively harder. After completion of the final level or when the number of lives reaches zero, the game is over.

The story is straight-forward. The player must get through three levels of combat to reach planet Xanthu VII, their home world. The design of the game was done in such a way that if expanded on the entire project could be a single mission in a much larger game. For example, the player has reached the planet, but now must battle on the surface. The story was written in such a way that it could be easily expanded on.

The preliminary plan for level progression is shown below.

- Number of Lives: 4, but configurable in the level file
- Number of Rockets: 10, but configurable in the level file
- Number of Levels: 3 plus boss assault level
- Number of Enemy Craft: Various
- Level Information
  - Level 1
    - Enemy Craft: F1, FB1
    - Entry Points: From Top of Screen
  - Level 2
    - Enemy Craft: F1, F2, FB1, B1
    - Entry Points: From Top and Sides of Screen
  - Level 3 (Final Basic)
    - Enemy Craft: F1, F2, FB1, FB2, B1, B2
    - Entry Points: Anywhere
  - Boss Assault (Final Level)
    - Enemy Craft: F1, F2, FB1, FB2, B1, B2
    - Entry Points: Anywhere

## **4. Development Specifications**

### **4.1 System Requirements**

- Pentium 4 1.5 GHz
  - May run on much slower machines
- 512 MB Recommended (To ensure enough space for Windows)
- 32 MB Graphics Card
- DirectX 8.1 or higher.

### **4.2 Development Specifications**

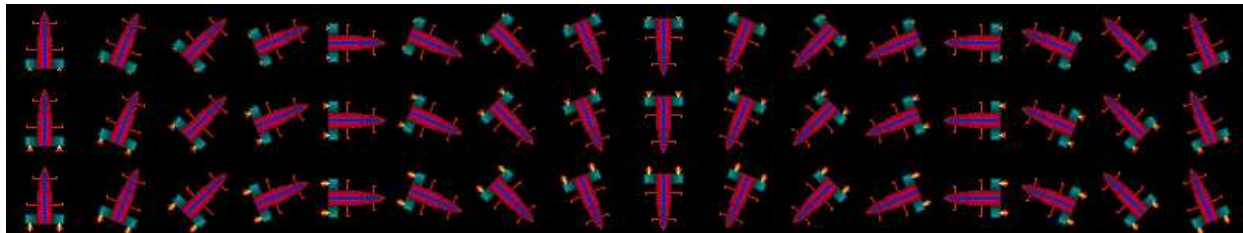
- Built in Visual Studio 2005
  - Can also be built with C++ Builder 6 Professional by uncomment a single #define statement.
- C++ Language
- Built with DirectX 9.0 and above SDK.
- UML diagrams generated with Microsoft® Visio® 2003 Professional
- Graphic bitmaps were modified/created using following tools
  - Microsoft® Paint
  - Paint Shop Pro® 7
  - Adobe PhotoShop® Elements 2.0
  - SpriteWorks™ (purchased originals)
- Sounds were using Switch Plus Audio File Conversion Software

### **4.3 Algorithm Style**

#### **4.3.1 Graphics and Sounds**

##### **4.3.1.1 Graphics**

The scenery (see Figure 1.) was created using multiple bitmaps and clipped images via DirectX to give the illusion of space travel. Several sources were use to provide the images and will be listed in the References section at the end of the document. Sprite sheets were used to provide the animation effects.



**Figure 7. Rocket Sprite Sheet Example**

#### 4.3.1.2 Sound effects

The inspiration for the sounds effects of Defender were provided from various sources.

- Liam Bradbury – Background Music
- Star Wars™ X-Wing vs. Tie Fighter – Sound effects
- André LaMothe - Outpost project – Sound effects.

All sounds in the game (except the background music) has the following properties

- Audio Sample Rate: 11Khz
  - Background Music: 44KHz
- Bit Rate: 88kps
  - Background Music: 352kbps
- Audio Channels: 1 (Mono)
- Audio format: PCM

#### 4.3.1.3 Development Algorithm

Listed below are the key features of Defender

- Fixed frame rate (30 FPS).
- Fixed point numbers (to allow fractional pixel movement).
- Some use of inheritance and data encapsulation (See the UML diagram below)
- Ability to restore lost surfaces, in case game is minimized and restored.
- Level Data File – Level information is loaded from a text file.

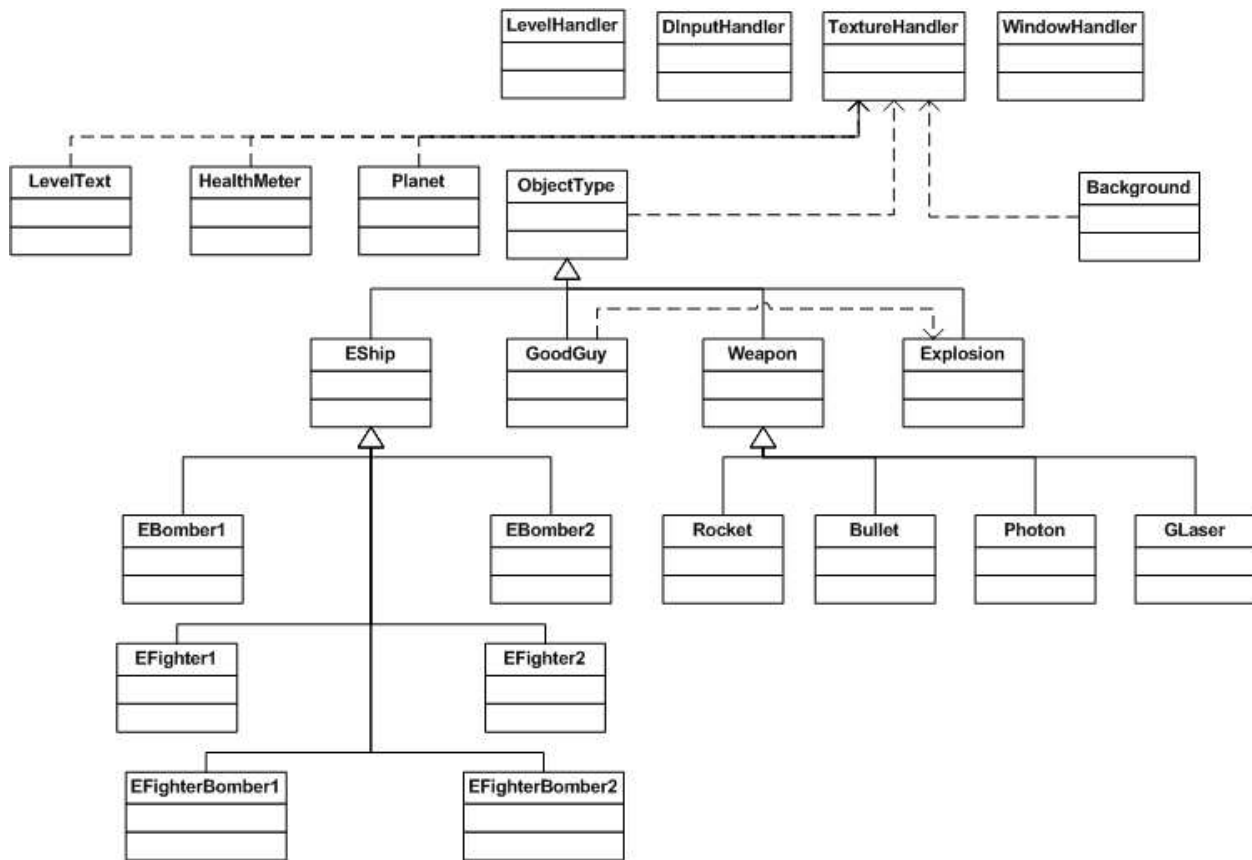


Figure 8. Defender Object Structure

#### 4.3.1.3.1 Game Area Logical View

Only of the more interesting components of the game is the level design. To understand the level design structure a brief explanation of the screen structure must be given.

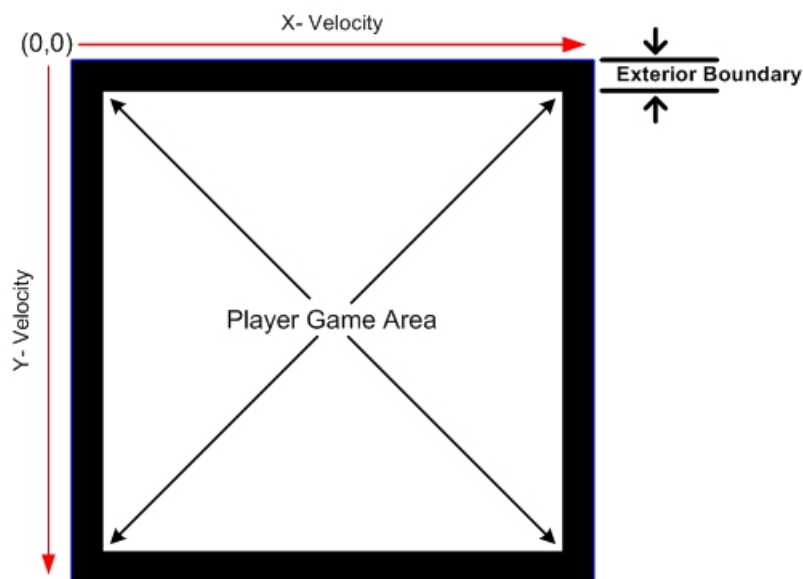


Figure 9. Game Area Logical View



The game area consists of the player viewable area and the exterior boundary. Given that the origin of a computer screen is at the upper-left corner, the positive horizontal and vertical directions are front left-to-right and top-to-bottom respectively. Assuming no acceleration at the moment, to travel from left to right requires a positive X-Velocity and the Y-Velocity is zero. Pure positive vertical travel involves a positive Y-Velocity and the X-Velocity is zero. To travel in the reverse direction involves the appropriate negative velocity. To travel at an angle, say 45 degrees from left to right, the X and Y velocities are equal. To summarize, a constant travel in a straight line simply involves supplying the appropriate X and Y values, plus a point of origin to start. If however a travel “curve” is desired, a horizontal and vertical acceleration factor is added.

#### **4.3.1.3.2 General Format of the Level File**

The level file consists of the following types of information.

- Level Header Information
- Game Version Number – must match the game
- Number of Players
- Number of Rockets
- Enemy Level Records

NOTE: The LevelData.txt file contains a full description the information contained within.

The generic level entry form is given here. The general description of each level is given in the general format

```
NEWLVL
DIST,TYPE,INITX,INITY,IvX,IvY,AccX,AccY,
DIST,TYPE,INITX,INITY,IvX,IvY,AccX,AccY,
DIST,TYPE,INITX,INITY,IvX,IvY,AccX,AccY,
NEWLVL
DIST,TYPE,INITX,INITY,IvX,IvY,AccX,AccY,
END
```

Where

- NEWLVL – New Game Level
- DIST – Distance “traveled” in the game when the enemy appears. Since this is a vertical scrolling game, the blockade runner is always moving forward, so the DIST values in the level file must be in increasing order.
- TYPE – The type of enemy
- INITX, INITY – The initial ‘X’ and “Y” coordinates of the enemy ship. This can be either an absolute number or a percentage of the screen resolution width. For example:
  - (0,0) - the upper-left corner of the screen
  - (-5,0) - 5 pixels off the left edge of the viewable area.
  - (%-5, %-5) - The percentage out of the game width and height of the screen. Basically, to the left and above the player game area.

- (105%, %50) - Five percent to the right of the right-most player game area.
- IvX,IvY – IvX and IvY denote the initial velocity vector of the ship. Again, this can be either a literal or a percentage of the player game area's width and height.
- AccX,AccY - These last two fields specify the enemy ship's acceleration in the X and Y directions. Again, these numbers can be specified as either literals or percentages which represent the change to the velocity vector per second.

Look at the example level description below to see how this could be applied.

4.3.1.3.2.1 Example Level Description

```
0.80          ;Defender version associated with this file (must match game)
;
4             ; start out with 4 lives
10            ; start out with 10 rockets
;
;
;
NEWLVL
; Middle FB1 zooms straight down and back up
100,FB1,%50,-100,%0,%50,%0.0,%-0.45,
; Group of 5 fighters in a pattern moving to the right
500,F1,%00,-100,%20,%20,%0,%0,
550,F1,%10,-100,%20,%20,%0,%0,
600,F1,%20,-100,%20,%20,%0,%0,
650,F1,%30,-100,%20,%20,%0,%0,
700,F1,%40,-100,%20,%20,%0,%0,
; Group of 5 fighters in a pattern moving to the left
1200,F1,%100,-100,%-20,%20,%0,%0,
1250,F1,%90,-100,%-20,%20,%0,%0,
1300,F1,%80,-100,%-20,%20,%0,%0,
1350,F1,%70,-100,%-20,%20,%0,%0,
1400,F1,%60,-100,%-20,%20,%0,%0,
;
NEWLVL
;Around the Clock
16900,F1,%50,%-5,%0,%30,%0,%0,      ; F1 - Straight Down
17000,F2,%105,%-5,%-30,%30,%0,%0,    ; F2 - Left Diagonal Down
17100,FB1,%105,%50,%-30,%0,%-0,%0,    ; FB1 - To the Left
17200,FB2,%105,%105,%-30,%-30,%0,%0,  ; FB2 - Left Diagonal Up
17300,B1,%50,%105,%0,%-30,%0,%0,      ; B1 - Straight Up
17400,B2,%-5,%105,%30,%-30,%0,%0,     ; B2 - Right Diagonal up
17500,F1,%-5,%50,%30,%0,%0,%0,        ; F1 - To the Right
17600,F2,%-5,%-5,%30,%30,%0,%0,       ; F2 - Left Sweep Down
END
```

## **5. Disclaimer**

It is not our intention to ever sell, misrepresent, distribute, or intentionally steal any sound effect and image. This purpose of this purely educational project is to demonstrate the power of DirectX on a 2D game environment at the University of Michigan – Dearborn and must not be modified or distributed in any form. This is due to several factors:

- Spriteworks™ Developments has specific provisions about that the images must be compiled, thus not directly readable, except by the project, in this case the game, unless I get written authorization stating otherwise.
- The compilers used have an academic license, thus no commercial code can be sold under said license.

In summary, we can only use the software and most objects for personal, educational, non-commercial use only, so I will respect their wishes in this matter. The read me file will contain this information with a copy of the author permissions.

## 6. References

Below are the main references used, without whose past expertise, this project would have been much harder, so we give out personal thanks. Although this was just a school project, their hard work must not go unnoticed.

### 6.1 Graphics References

Introduction Image	NASA, Jet Propulsion Laboratories, California Institute of Technology, <b>PIA08042: Extreme Planets</b> , <a href="http://photojournal.jpl.nasa.gov/jpeg/PIA08042.jpg">http://photojournal.jpl.nasa.gov/jpeg/PIA08042.jpg</a> , Modified for use by Barry Belcher
Background Image	Hubble Space Image, location unknown (possibly property of NASA)
Spacecraft and Weapon Images	Spriteworks™ Developments, <a href="http://www.spriteworks.com/sworks.html">http://www.spriteworks.com/sworks.html</a> , copyright © 2003, All rights reserved
Explosion and Bullet Images	Lomibao, Steven , <b>Asteroids</b> , <a href="http://www.engin.umd.umich.edu/CIS/course.des/cis587/games/fo3/Lomibao.zip">http://www.engin.umd.umich.edu/CIS/course.des/cis587/games/fo3/Lomibao.zip</a> , Final Project Fall 2003, Modified for game use by Barry Belcher and Aaron Curley
Planet Image	Activision, Star Trek: Armada II, copyright © 2001, all rights reserved, modified for game use by Aaron Curley

### 6.2 Sound References

Background Music	Bradbury, Liam, <b>techno_01_339-royalty_free_music_for_your_games-liam_bradbury-18.mp3</b> , <a href="http://www.blastmymusic.com/liambradbury">www.blastmymusic.com/liambradbury</a> , copyright © 2007, All rights reserved, Converted for game use by Barry Belcher
Sound Effects	Holland, Lawrence, Totally Games, <b>Star Wars: X-Wing vs. TIE Fighter</b> , Published by LucasArts, copyright © 1997, All rights reserved, Renamed for game use by Barry Belcher  LaMothe, André, Sams Publishing, <b>Tricks of the Windows Game Programming Gurus, 2<sup>nd</sup> Edition</b> , copyright © 2002, All rights reserved

### 6.3 Miscellaneous

Library Functions and other effects	LaMothe, André, Sams Publishing, <b>Tricks of the Windows Game Programming Gurus, 2<sup>nd</sup> Edition</b> , copyright © 2002, All rights reserved
-------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------