

Design Document for:
Defender 2: R.A.I.D. On Xanthu VII

Written by:
Barry Belcher
Aaron Curley
Michael Hardy
Evan Musu

Game Design and Implementation I
Dr. Bruce R. Maxim

Version: 5

DECEMBER 14, 2007

TABLE OF CONTENTS

1	GAME OVERVIEW	1
1.1	APPEARANCE	1
1.2	STORY ABSTRACT	1
1.3	FREQUENTLY ASKED QUESTIONS	1
1.4	DEVELOPMENT PLATFORM	2
2	GAME MECHANICS	2
2.1	OVERVIEW	2
2.2	CAMERA POSITION	2
2.3	USER INTERFACE DESIGN	2
2.3.1	<i>World</i>	3
2.3.2	<i>Movement</i>	3
2.3.3	<i>Weapons</i>	3
2.3.4	<i>Objects</i>	3
2.3.5	<i>Overall Description</i>	3
2.3.6	<i>Key Screen Images</i>	4
2.3.7	<i>State Transition Diagram</i>	6
2.3.8	<i>Design Rules</i>	6
2.4	USES CASES	8
2.4.1	<i>System Overview</i>	8
2.4.2	<i>Play Game</i>	9
2.4.3	<i>Change Options</i>	9
2.4.4	<i>Exit Game</i>	9
2.4.5	<i>Load Game Menu</i>	9
2.4.6	<i>View Objectives</i>	10
2.4.7	<i>Return to Game</i>	10
2.4.8	<i>Restart Area</i>	10
2.4.9	<i>Quit to Main Menu</i>	10
2.4.10	<i>Move Xophad</i>	11
2.4.11	<i>Perform Mind Push</i>	11
2.4.12	<i>Use Neutralizer</i>	11
2.4.13	<i>Pick Up Neutralizer/Agent</i>	12
2.4.14	<i>Toggle Console</i>	12
2.4.15	<i>Toggle First/Third Person Views</i>	12
2.5	REPLAYING AND SAVING	12
2.6	IN-GAME CONTROL SUMMARY	13
2.7	GAME PLAY DETAILS	13
3	ARTIFICIAL INTELLIGENCE	14
3.1	OPPONENT AI	14
3.1.1	<i>Player Detection</i>	14
3.1.2	<i>Motion</i>	14
3.1.3	<i>Path Finding</i>	14
3.1.4	<i>Special Actions</i>	15
3.1.5	<i>Combat</i>	15
3.2	NON-PLAYABLE CHARACTERS	15
3.3	REACTIVE ITEMS	15
4	GAME ELEMENTS	18
4.1	DATA DICTIONARY	18
4.2	CHARACTER BIBLE	19
5	STORY OVERVIEW	20

5.1	PLOT SUMMARY	20
5.2	STORY BOARD	22
6	GAME PROGRESSION	23
6.1	FLOWCHARTS	23
7	CONCEPTUAL LEVEL DESIGN	26
7.1	LOCATION	26
7.2	PLAYER INFORMATION	26
7.3	MISSION OBJECTIVES	26
7.3.1	<i>Mission Objectives</i>	26
7.3.2	<i>Mission Area Breakdown</i>	26
7.4	CHALLENGE HIGHLIGHTS	27
7.4.1	<i>Cut Scenes</i>	27
7.4.2	<i>Storytelling</i>	27
7.5	WOW FACTOR	27
7.6	MAP DESCRIPTION	27
7.7	ASSETS AND REQUIREMENTS	28
8	LESSONS LEARNED	28
8.1	POST MORTEM EVALUATIONS	29
8.1.1	<i>Barry Belcher Post Mortem</i>	29
8.1.2	<i>Aaron Curley Post Mortem</i>	30
8.1.3	<i>Mike Hardy Post Mortem</i>	31
8.1.4	<i>Evan Musu Post Mortem</i>	32
9	BIBLIOGRAPHY	34

Revision History

Version Draft A (11/13/07)	Draft A contains the first three sections of the design document.
FTR Draft (11/17/07)	Contains all sections ready for the FTR scheduled on 11/19/07.
Version 1	Initial Final version of Design Document. Removing the stinger as a weapon due to timing constraints and added use case descriptions. Added more recent images
Version 2	Add Level Detail information
Version 3	New AI and Reviews performed
Version 4	Add Post Mortem Reviews and Reference Information
Version 5	Some minor corrections.

1 Game Overview

1.1 Appearance

Defender 2: R.A.I.D. on Xanthu VII is a first-person-shooter (FPS) in an alien environment. Lighting, objects, and characters will appear colorful in nature, using purple, green, and deep red colors. As the player encounters the “human-built” factory, the environment will have an industrial-metallic look.

1.2 Story Abstract

Defender 2: R.A.I.D. on Xanthu VII is the sequel to the DirectX 2D Project: *Defender*. Volume 2 is a 3D computer game about you, the defender, given a mission of counteracting the effects of genocidal weapons used by the dreaded R.A.I.D. Foundation.

1.3 Frequently Asked Questions

What is the game?

The game is called *Defender 2: R.A.I.D. on Xanthu VII*, the sequel to the 2D arcade shooter, *Defender*. Due to the limited project time, this game will consist of only one mission.

Who is the intended target Audience?

Defender 2: R.A.I.D. on Xanthu VII is targeted for teens due to animated violence and death.

Why create this game?

This game is a logical extension to the *Defender* arcade game. This project allows us to explore the power of the Torque Game Engine in a 1st/3rd person game environment. In addition, there are not many games where the hero is an Insectoid, normally thought of as evil in many games.

Where does the game take place?

This game takes place on the Insectoid colony world of Xanthu VII, just after the Insectoid blockade-runner reached the planet service and learns of the recent events of his planet's demise.

What does the player control?

The player controls the actions of an Insectoid character throughout the game. The player can walk, run, jump and defend himself, although stealth is a major part of the game.

What is the main focus?

The player's main focus is stealth, since this is a “thief” style game. The mission consists of several goals that must be achieved. The player must get past the various obstacles to achieve the game's ultimate goal to destroy the biogenic weapon and kill the humans in the mission area. To this end there are several sub goals:

- Sneak past the guards and avoid being seen.

- Locate and acquire the special chemical agent planted in the mission area by the human secret agent.
- Locate the bio-processing area.
- Combine the special chemical agent with the neutralizer the player has at the beginning of the game.
- Get out of the area before the resulting bio-cloud kills the player. Although the player is more resistant to the resulting cloud than the human guards, the player is not immune.
- Make it back to the ship.

What's different about Defender 2?

This game is different in that the hero is an Insectoid, not the usual human. There are not many games where stealth is the main focus either.

1.4 Development Platform

Defender 2 is intended to be played on Windows 2000/XP systems meeting the following minimum system requirements:

- Pentium 4, 2.0 GHz
- 512 MB RAM
- 64 MB OpenGL or DirectX Compatible 3D Graphics Accelerator Card
- DirectX compatible Soundcard
- 40MB Hard Drive Space
- Minimum screen resolution of 800x600 (1024x768 Recommended)

2 Game Mechanics

This section discusses the game mechanics of *Defender 2: RAID on Xanthu VII*.

2.1 Overview

Defender 2 is a FPS-style game. However, the player will play the game from either the first-person or third person perspective. Defender 2 will be developed using the Torque Game Engine (v1.5) and Torque Script by GarageGames®. For more information on the Torque game Engine, please go to <http://www.garagegames.com>.

2.2 Camera Position

The Torque Game Engine allows a game to be displayed from the first-person or third-person perspective. Defender 2 uses the standard engine setting of allowing the user to toggle between the views via the <TAB> key. The third-person view is from behind and just above the player.

2.3 User Interface Design

The user embodies an Insectoid soldier in this First Person Shooter (FPS). The conventions used will be similar to other games in the FPS genre which a player will be familiar with.

2.3.1 World

The player's view will be of an alien world denoted by red and purple hues.

2.3.2 Movement

The user can move about the world using standard FPS conventions (more details in section 2.6, In-Game Control Summary). The player's movement is constrained by buildings and objects in the world. Architecture will be used as puzzles, creating unique way to move from area to area that is not linear running from room to room. The player will have the ability to jump further distances compared to human players in similar games (and the opponent human soldiers in this game). Because the player is an Insectoid alien, falling from great distances will not kill him the way a human soldier would die from the same fall.

2.3.3 Weapons

The user can employ two weapons, which are selectable one at a time by the user.

- A. **Mind Push** – The player starts with the mind push weapon. When a player shoots this weapon at an enemy, the enemy is thrown in the opposite direction equivalent to a jump. The mind push does not affect other objects or NPCs.
- B. **Neutralizer** – Early in the first level the player finds the neutralizer. When the player shoots the neutralizer at insecticide goo, it eliminates the goo. Spraying the neutralizer at an enemy soldier causes the soldier to be temporarily frozen in place. The weapon has no effect on any other object or NPC.

2.3.4 Objects

All objects in the level cannot be moved. The player can grab the chemical agent to be used later at the climax of the level. The agent is not directly used, but is a qualifying condition of poisoning the master vat and hence winning the game.

2.3.5 Overall Description

The user interface will be logically divided into two sections: the “main menu” and the “in-game” areas. The user interface will be fairly simple, with a focus on making its use a quick and easy process.

The main menu will present a title screen to the user and allow the user to choose between starting the game and viewing the generic Torque options panel, which allows changing of the screen resolution and audio settings. Once the user clicks the “start game” button on the main menu, the game will transition to the “in-game” user interface.

Upon transition to the in-game user interface, the player object and the environment will be displayed. The user will control the player throughout the game by using the keyboard and mouse much like the “standard” FPS (first-person-shooter) controls in many games. See section 2.6 for a detailed description of player controls.

If the user presses the ESC (escape) key when playing the game, an in-game menu is displayed. This menu will allow the player to resume the game, quit to the main menu, or restart the area. Choosing the “restart area” option will be equivalent to dying; that is, the player will start over at the last checkpoint reached.

For the in-game interface, a “health bar” will be displayed near the bottom-left corner of the screen and will represent the current health of the player. Above this bar, small icons will represent the possible tools (weapons or items) that the user can choose from. The active tool will be highlighted. See Figure 3 in section 2.3.6 for an illustration and section 2.6 for more information on the selection of the current tool.

2.3.6 Key Screen Images

This section contains proposed images for various parts of the in-game user interface.

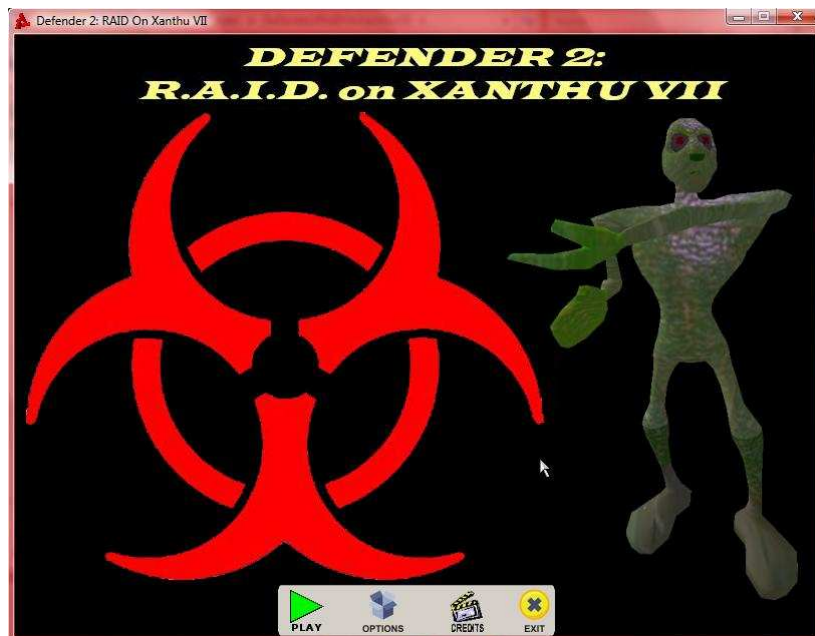


Figure 1 – Main Title Screen

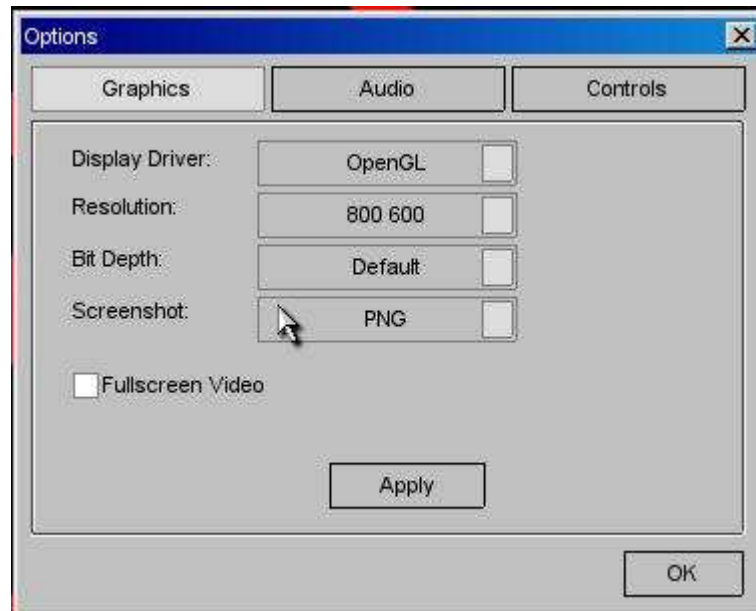


Figure 2 – Torque Options Panel



Figure 3 – In-Game User Interface

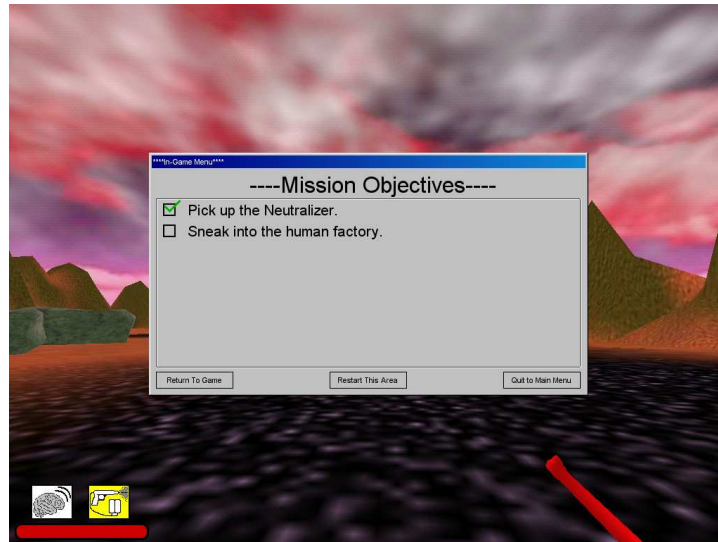


Figure 4 – The In-Game Menu

2.3.7 State Transition Diagram

The state transition diagram given here illustrates the progression of the user interface as the user performs certain actions. In general, the user will start at the main menu and begin the game. After the user decides to quit, the user will press the escape key and will choose the “Quit to Main Menu” option. Then, the user will exit the program from the main menu by choosing the “exit” button.

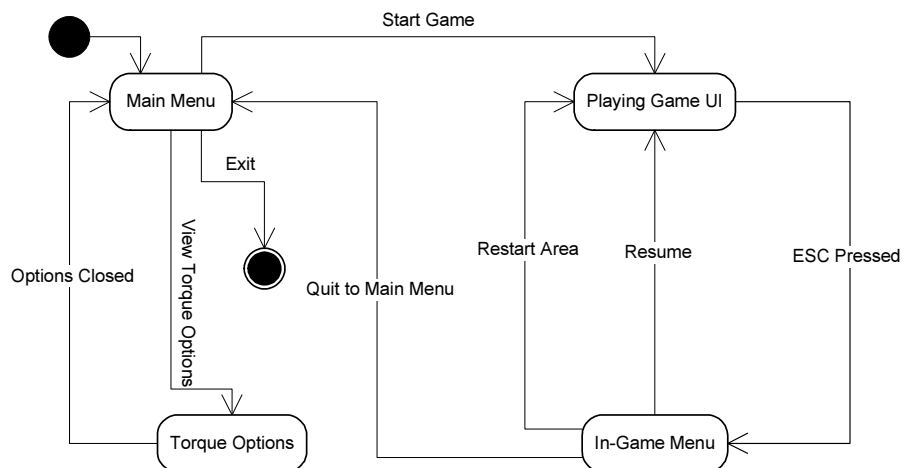


Figure 5 – State Transition Diagram

2.3.8 Design Rules

One of the goals of this project is to ensure a well-designed user interface that is easy to learn and use. Although the current design is fairly simple, it is still necessary to correct any usability issues to avoid user frustration. Therefore, the design of the user interface will be prototyped, tested, and redesigned as necessary.

Time constraints do not allow for formal usability testing outside the team, which will require that user interface evaluation be primarily carried out by team members. To strive for a more accurate evaluation of the user interface, all team members will test each revision to the user interface and give feedback to the designer, who will then make changes as appropriate.

The “main menu” portion of the interface will be an extremely simplistic menu-style user interface that will consist primarily of choosing a command button that matches the desired action. The in-game player controls are designed to match the “standard” first-person-shooter game controls as much as possible. Designing the user interface in this way will minimize learning time of users that have experience with other games.

Specific User Interface Goals:

- Intuitive (easy to learn)
- Consistent
- Intuitive Icon Images (if no text labeling is included)
- Intuitive Labeling for UI Components
- Simple (focus on helping the user play; avoid complexity that is “just for show”)

2.4 Uses Cases

2.4.1 System Overview

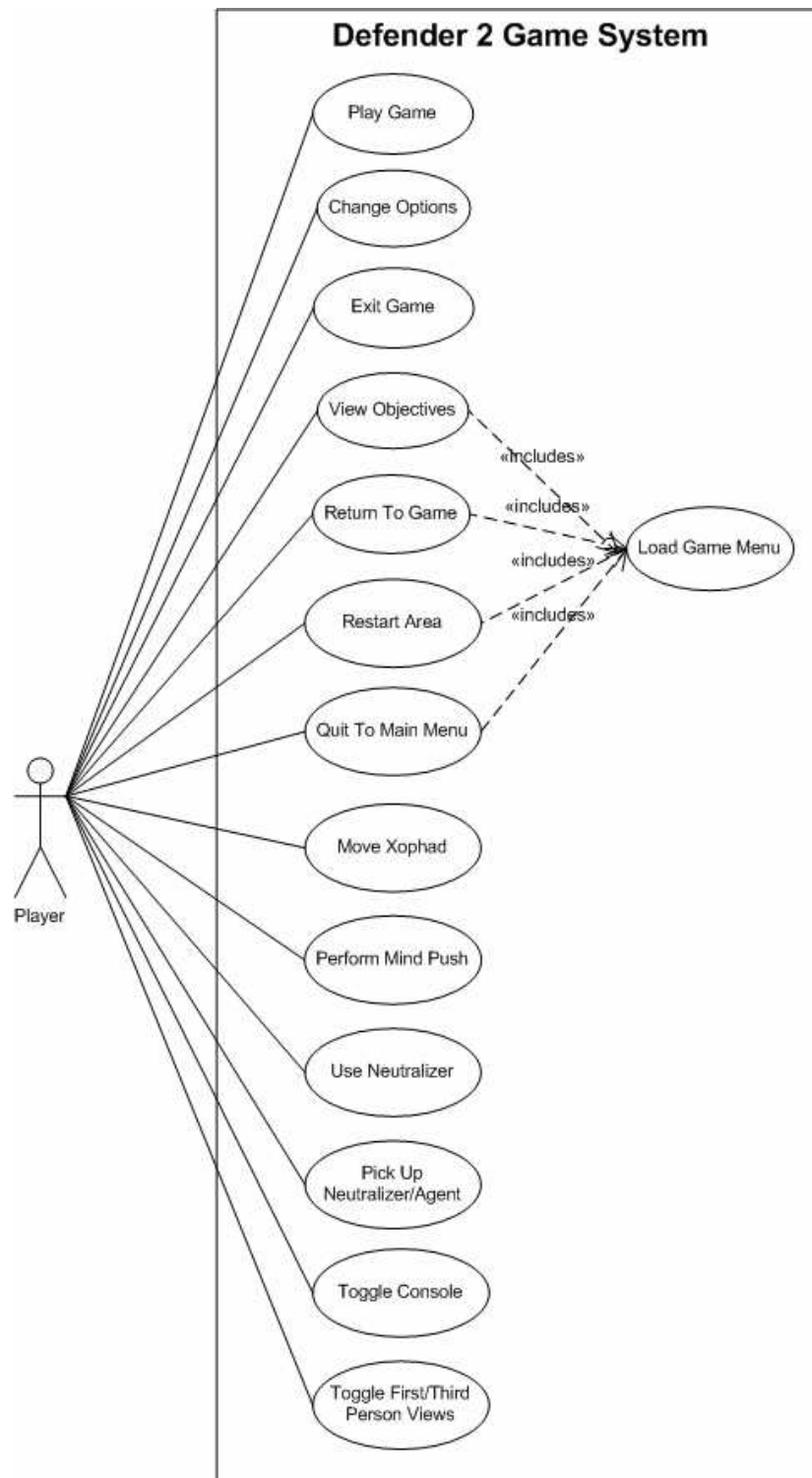


Figure 6. Defender 2 Use Case Overview

2.4.2 Play Game

Brief Description

The *Play Game* use case allows the player to play the Defender 2 game

Step-by-Step Description

The player must currently be at the Defender 2 main menu screen.

1. The player selects the “Play” icon at the Defender 2 main menu screen.
2. The system responds by Loading the mission details (datablocks, objects, and mission lighting.)
 - 2.1 If the player selects “Cancel” the mission loading ends and the player is returned to the Defender 2 main menu screen

2.4.3 Change Options

Brief Description

The *Change Options* use case allows the player to modify the Visual/Audio options of Defender 2.

Step-by-Step Description

The player must currently be at the Defender 2 main menu screen.

1. The player selects the “Options” icon at the Defender 2 main menu screen.
2. The system responds by displaying the options dialog box.
3. The player sets the options accordingly.
 - 3.1 If the player selects “Cancel” no changes are made and the options box is removed.
 - 3.2 If the player applies the changes, the new settings take effect.
4. The player selects the OK button on the options box screen
5. The system responds by removing the options box and returning the player back to the Defender 2 main menu screen.

2.4.4 Exit Game

Brief Description

The *Exit Game* use case allows the player to quit Defender 2 game

Step-by-Step Description

The player must currently be at the Defender 2 main menu screen.

1. The player selects the “Exit” icon at the Defender 2 main menu screen.
2. The system responds by unloading the Defender 2 game.

2.4.5 Load Game Menu

Brief Description

The *Load Game Menu* use case allows the player to load and display the in-game menu of Defender 2

Step-by-Step Description

The player must currently be in the Defender 2 mission area.

1. The player selects the “ESC” key at the Defender 2 live display screen.
2. The system responds by loading the Defender 2 In-Game menu.
 - 2.1 The mission objectives are also updated.

2.4.6 View Objectives

Brief Description

The *View Objectives* use case allows the player to load and display the in-game menu of Defender 2

Step-by-Step Description

The player must currently be in the Defender 2 mission area.

1. The player loads the in-Game Menu via the *Load Game Menu* use case
2. The player may now review the objectives.

2.4.7 Return to Game

Brief Description

The *Return To Game* use case allows the player to return to the mission area from the In-Game menu popup.

Step-by-Step Description

The player must currently be in the In-Game menu

1. The player selects the *Return To Game* button
2. The system responds by removing the In-Game menu and returning the player to the mission area

2.4.8 Restart Area

Brief Description

The *Restart Area* use case allows the player to restart Defender 2 from the last checkpoint reached in the game.

Step-by-Step Description

The player must currently be in the In-Game menu popup

1. The player selects the *Restart This Area* button
2. The system responds by displaying a confirmation popup
3. One of the following happens:
 - 3.1 If the player selects “No”, the system responds by clearing the popup, returning the player to the in-game menu.
 - 3.2 If the player selects “Yes”, the system responds by clearing the popup, checking the items that were modified since the last checkpoint was reached and restoring them back to the game and restarting the player from the last checkpoint reached.

2.4.9 Quit to Main Menu

Brief Description

The *Quit to Main Menu* use case allows the player to restart Defender 2 from the last checkpoint reached in the game.

Step-by-Step Description

The player must currently be in the In-Game menu popup

1. The player selects the *Quit to Main Menu* button
2. The system responds by displaying a confirmation popup
3. One of the following happens:
 - 3.1 If the player selects “No”, the system responds by clearing the popup, returning the player to the in-game menu.
 - 3.2 If the player selects “Yes”, the system responds by clearing the popup, ending the mission and returning the player to the Defender 2 main menu screen.

2.4.10 Move Xophad

Brief Description

The *Move Xophad* use case allows the player to perform movements of the Insectoid

Step-by-Step Description

The player must currently be in the mission

1. The player selects the appropriate key as described in section 2.6 (In-Game Control Summary)
2. The system responds by moving Xophad in the appropriate direction.

2.4.11 Perform Mind Push

Brief Description

The *Perform Mind Push* use case allows the player to perform a Mind Push attack on an enemy player

Step-by-Step Description

The player must currently be in the Defender 2 mission area.

1. The player presses the Number “1” key on the keyboard to perform the attack
2. The system responds by equipping the mind push weapon and displaying the Mind push icon.
3. The player presses the left mouse button to activate the weapon
4. The system responds by performing the following:
 - Play the mind push sound file
 - Apply any mind push visual effects
 - Push the enemy back if within range
- 4.1 If the enemy takes damage, the system responds by applying deducting the health of the affected
- 4.2 If the enemy is killed (0 health) the system responds by playing the enemy death sequence, playing any death sound file, and records the information for later retrieval in the event the games resumes at a previous checkpoint.
5. The player continues game play.

2.4.12 Use Neutralizer

Brief Description

The *Perform Mind Push* use case allows the player to perform a Mind Push attack on an enemy player

Step-by-Step Description

The player must currently be in the Defender 2 mission area.

1. The player presses the Number “2” key on the keyboard to perform the attack
2. The system responds by equipping the neutralizer and displaying the Mind push icon.
3. The player presses the left mouse button to activate the neutralizer
4. The system responds by performing the following:
 - Play the neutralizer sound file
 - Apply any neutralizer visual effects
- 4.1 If an enemy is hit the system responds by playing the enemy death sequence, playing any death sound file, and records the information for later retrieval in the event the games resumes at a previous checkpoint.
- 4.2 If and pool or vat of insecticide is struck, the system responds by render the goo as harmless (causing no damage to the player is the player enters the pool or vat).
- 4.3 If the neutralizer has a low charge, the system respond by notifying the player The charge is too low.
5. The player releases the left mouse button.
6. The system responds by ending the neutralizer special effects.

2.4.13 Pick Up Neutralizer/Agent

Brief Description

The *Pick Up Neutralizer/Agent* use case allows the player to retrieve the neutralizer and special chemical agent in Defender 2

Step-by-Step Description

The player must currently be in the Defender 2 mission area.

1. The player moves Xophad over the neutralizer or special chemical agent
2. The system responds by removing the neutralizer/agent object from the play Screen and adds the object information to the players inventory
3. The player continues game play

2.4.14 Toggle Console

Brief Description

The *Toggle Console* use case allows the player to Toggle the game console

Step-by-Step Description

The player must currently be in the Defender 2 mission area.

1. The player presses the “~” (TILDA) key
2. The system responds by displaying the in-game console.
3. The player enters a command in the command line.
4. The system responds by executing the command or displaying the appropriate message.
5. The player presses the “~” key again.
6. The system responds by removing the console from view
7. The player continues game play

2.4.15 Toggle First/Third Person Views

Brief Description

The *Toggle First/Third Person Views* use case allows the player to Toggle the first and third person views in Defender 2

Step-by-Step Description

The player must currently be in the Defender 2 mission area.

1. The player presses the <TAB> key.
2. The system responds by displaying the third person view if currently in first –person view or vice-versa
4. The player continues game play.
3. The player presses the <TAB> key again.
4. The system responds by displaying the third person view if currently in first –person view or vice-versa.
7. The player continues game play

2.5 Replaying and Saving

Defender 2 will provide no save functionality due to the limited development time. However, there are checkpoints in the game to aid the player. If the player dies or chooses to restart the area, the game will resume from the last checkpoint reached. If the player exits back to the main menu, all progress in the area is lost and he/she will have to start the mission from the beginning,

2.6 In-Game Control Summary

When playing the game, the user will perform various actions as described below:

- Player movement through the W, A, S, and D keys
 - W – Forward movement.
 - A – “Sideways” movement to the left.
 - S – Backward movement.
 - D – “Sideways” movement to the right
- Player “jump” using spacebar.
- Player rotation and “looking” by moving the mouse.
- Tool (weapon or item) selection using the number keys to select a specific tool or the mouse wheel to move to the next or previous tool.
- Firing/activation of the current “tool” by clicking the left mouse button.
- Activation of the in-game menu by pressing ESC.
- TAB key to switch from first person to third person view or vice-versa.

These control choices are intended to match the “standard” FPS control set so that the learning time of new users can be minimized.

2.7 Game Play Details

The user will first appear at some point outside of the factory complex where he/she will be informed of the mission objective. To aid the player, the in-game menu will list the current objectives, checking off each one as the goals are reached. Game play will be fairly linear, although the user may deviate somewhat from the intended game path. For example, the game is intended to be executed in a “stealth” fashion, but the user is free to slug it out with all the guards.

The story board in the later section lists the critical tasks that must be performed during game play, but the user is free to determine to a degree how to accomplish each goal and reach each checkpoint.

The game will communicate with the player via sounds and messages at various points in the game as to what checkpoints are reached, what items are collected, etc. There will also be messages and/or audible cues to let the player know if a goal must be reached in order to continue or function. For example, if the player attempts to use the neutralizer, but has used it a number of times in quick succession, the player may have to wait for the canister to recharge.

The player may take damage during the course of game play. If this occurs the Insectoid will scream to indicate pain and the health bar will decrease. However, he has regenerative powers, so the health will gradually increase back to full health as long no other damage is taken. If the player health bars is completely drained the player dies, and will resume from the last checkpoint reached.

3 Artificial Intelligence

This section describes how the artificial intelligence works and how the player interacts with the AI and various items in the world.

3.1 Opponent AI

The opponent AI consists of enemy soldiers armed with guns (ranged weapon):

3.1.1 Player Detection

Players are detected using two methods:

3.1.1.1 Hearing

The AI checks for the player within a radius. If he is at alert, this radius is larger.

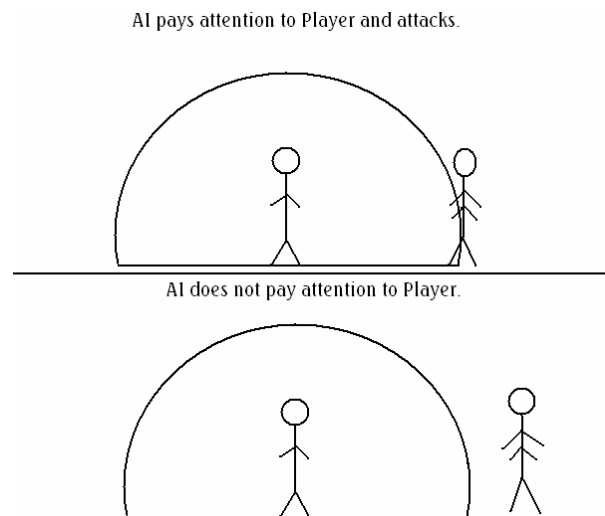


Figure 7 – Auditory Aspect of AI

3.1.1.2 Visual

The AI scans for the player where all three checks have to succeed:

- The player is within visual range (distance)
- The player is within the AI's Field of View
- No terrain, buildings, or objects are directly between the player and the AI

When the AI is at an alert state, he can see further.

3.1.2 Motion

Each Soldier AI can follow a unique path where they turn around at the end and pace the path the other way. This mode is called "Patrol."

3.1.3 Path Finding

The levels have stairs, ramps, and other 3D level design. The Soldier AI stays on its path. If attacked by the neutralizer, they will eventually return to their path. If knocked off course by the mind push, they stay in place.

3.1.4 Special Actions

Soldier AIs that see the player for the first time yell “Over there” so that nearby AIs (within a hearing range) can turn toward the player and try to see them.

3.1.5 Combat

Soldier AIs stop to fire at the player. If the player goes out of sight, they wait looking in the same direction for a while before returning to patrol.

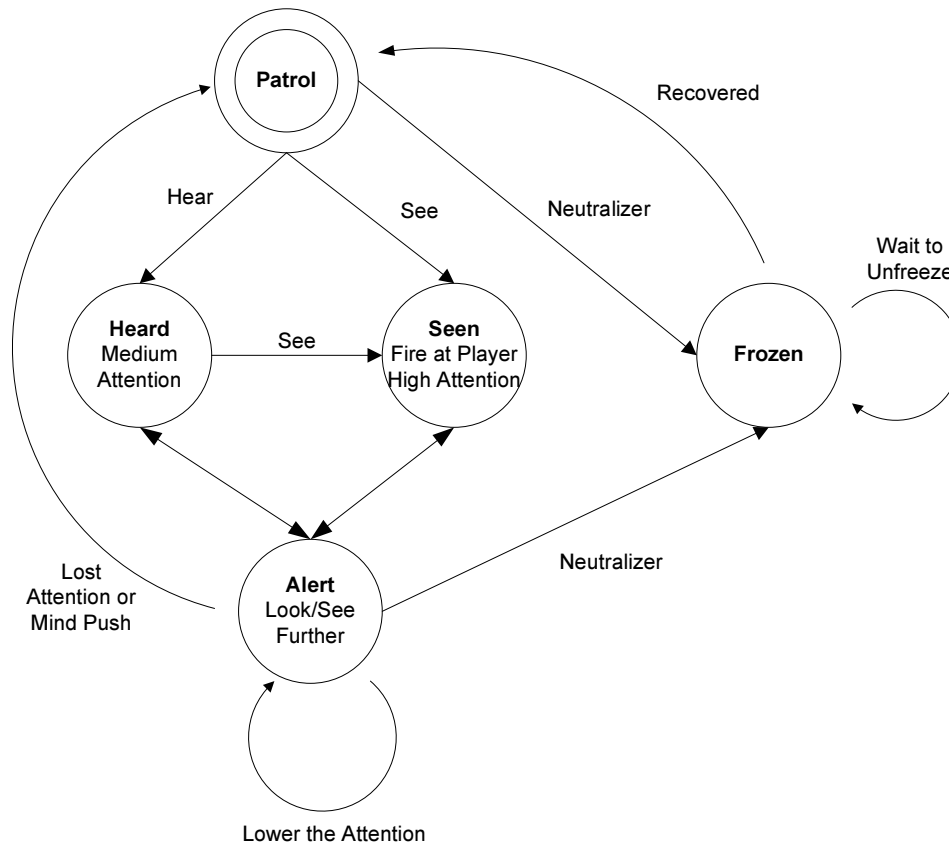


Figure 9 – Soldier AI State Diagram

3.2 Non-Playable Characters

The player will encounter non-playable characters (NPCs) before entering the factory. These non-playable characters will tell the story to the player and tell the player how to play the game. The player does not need to necessarily interact with all of these non-playable characters to progress through the game, but they could supply useful information to the player.

3.3 Reactive Items

There are a number of reactive items that interact with or affect the player:

1. Neutralizer

The neutralizer is a weapon that clears paths for the player. These blocked paths are puddles of insecticide. (A visual example of this weapon is shown below.) The neutralizer freezes the Soldiers momentarily if hit with enough of it.

2. Mind Wave/Force Push

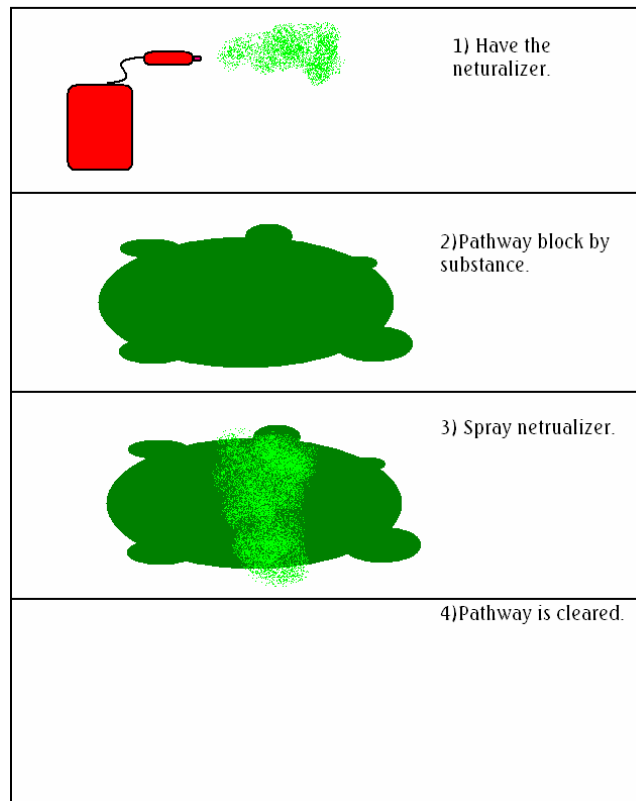
The mind wave is a weapon for the player to push the enemy AI without actually having to touch the AI enemy. (A visual example of this weapon is shown below.) The fall from the Mind Push or falling in a vat can damage the enemy AI.

3. Insecticide Pool

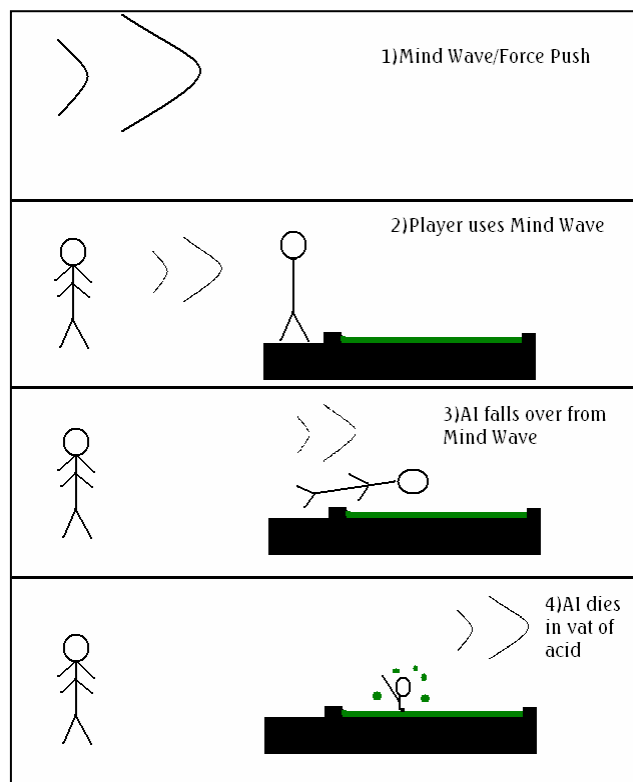
If touched by the player, an insecticide pool will damage the player's health or kill the player depend on how long the player remains in contact with it.

4. Poisonous Cloud

The poisonous cloud will appear after the player achieves the main goal of introducing the neutralizer and agent into the "master vat." The poisonous cloud will spread across the area and will do severe damage to the player's health and can kill the player. The poisonous cloud is designed to harm the enemy soldiers much worse, killing them in a relative short time.



3.3.1.1.1 Figure 9 – Neutralizer's Function Example



3.3.1.1.2 Figure 10 – Mind Wave Function Example

4 Game Elements

4.1 Data Dictionary

Listed below are proposed items to be included in Defender 2.

Item	Weapon – Mind Push
Goal in Context	Moves enemy soldiers, so if done at correct time/place, they die from the fall or landing in a toxic goo vat.
Scope	Entire level
Primary Actor	Player
Priority	High, needed for the story mechanics
Frequency	Mind push is unlimited use, but has a cooling off period after each use.

Item	Weapon – Neutralizer
Goal in Context	Destroys toxic goo which prevents the player from reaching the goal
Scope	Only usable until after the player collides with the Neutralizer weapon object
Primary Actor	Player
Priority	High, needed for the story mechanics
Frequency	Only a certain amount can be used at a time, there is no ammo limit, but it has a recharge rate.

Item	Agent
Goal in Context	Make goo into human destructive agent
Scope	Only available until after the player collides with the Agent object
Primary Actor	Player
Priority	High, needed for level objective
Frequency	Only once in game

Item	Goo
Goal in Context	Goo damages Insectoids, preventing player from traversing the level
Scope	Within vats or pools on the ground.
Primary Actor	Goo
Priority	High, crucial to story
Frequency	Often, goo is placed on the ground and in vats throughout the level

Item	Enemy soldier's Weapon
Goal in Context	Stops player from succeeding
Scope	When bullets fired from weapon collide with player, they do damage.
Primary Actor	Enemy Soldier
Priority	High, needed as an outside enemy
Frequency	Interspersed throughout the level

Item	10 Master Goo Source
Goal in Context	When hit with neutralizer and agent, creates poisonous cloud
Scope	In master goo vat
Primary Actor	Goo
Priority	High, level objective
Frequency	Once in level

Item	Poisonous cloud
Goal in Context	Kills all humans and Insectoid aliens within range
Scope	Spreads over area of factory
Primary Actor	Poisonous cloud
Priority	Medium, needed for story but could be changed to another story element
Frequency	Once in level

4.2 Character Bible

Role	Player
Name	Xophad Beetlebrax
Race	Insectoid
Age	27 Years old
Place of Birth	Xanthu VII on the outskirts of the Xando Consortium Federation of Worlds
History	After completing hive school, Xophad joined the Xando Space Fleet as a ship's pilot. 14 years of exceptional service in the Xando-Earth war and numerous promotions helped Xophad reach the rank of Captain. His triumph at the Vogon bypass inspired the Xanthu VII government to hire him to clear the way through R.A.I.D. forces and deliver the crucial neutralizing agent.
Current Events	Meet the Insectoid Underground on Xanthu VII to destroy the bio-weapon and stop the human threat on his home planet.
Abilities	Intelligent, able to figure out spatial puzzles. Mind push that throws humans. Great vision: able to see miles in the distance. Jump higher and for more distance than humans. Dangerous Insectoid stinger. Can withstand falls from any height due to high air resistance. Uses devices like the neutralizer. Can carry inventory such as the agent.

Role	NPC
Name	Xent Rthur
Race	Insectoid
Age	54 Years old
Place of Birth	Xord
History	Xent had an ordinary life before the war as the Police Prefect of a small town on Xord. But the Xando-Earth war woke Xent out of his bucolic slumber of a life when human soldiers landed on Xord. He led his small town, the county, and then the nation against the human invaders. Injured in the war, he has spent the last year watching humanity for any signs of betrayal. His vigilance paid off as the secret R.A.I.D. factory on Xanthu VII was discovered.
Current Events	Too old and injured to fight the R.A.I.D. forces himself, he has tracked down the landing spot of Xophad Beetlebrax and informs him about the bio-weapon known as Goo and the existence of a chemical agent, which can be used to destroy the factory.
Abilities	Espionage.

Role	AI Enemy
Name	Various (George, John, Tom, Jim, etc)
Race	Human
Age	19-34 Years old
Place of Birth	Earth and Earth Colonies
History	R.A.I.D. forces come from throughout the galaxy. Xenophobes who would rather attack first and ask questions later.
Current Events	R.A.I.D. forces guard the bio-weapon facility
Abilities	Low intelligence relative to Insectoids. Proficient in handheld weapons. Only sees short distances. Can hear enemies close by. Impatient, bores of searching for enemies for long.

5 Story Overview

5.1 Plot Summary

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 galaxy 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. There 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 your race, 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.

However, a radical group of “bug haters” formed the Race Against Insectoid Dominion or R.A.I.D. for short. The foundation assembled a secret fleet and developed an Insectoid biogenic weapon, deploying the weapon with devastating effect.

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

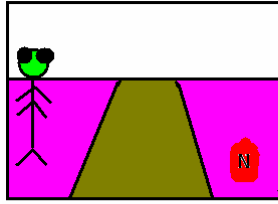
spacecraft as possible to clear the way for other supply vessels. Although you reached your world, you were the only surviving ship able to reach the surface and much of the planet's population had evacuated to other worlds.

You reached your rendezvous point and have met with an Insectoid Underground contact, who has informed you of your world's evacuation and the human takeover of a factory that has been transformed to produce more of the bio-agent for use against other Insectoid worlds.

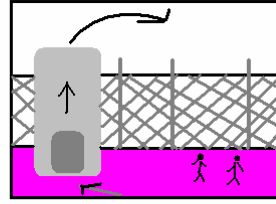
You also learned that the Underground obtained information stating there was a Human saboteur in the factory that secretly placed a special agent within, working with the Xandor and Earth governments to prevent another war. When the agent is combined with your neutralizer, not only will the bio-weapon be neutralized it also produces a highly poisonous gas that will destroy every human in the factory.

Although the Earth government is totally against the actions of the R.A.I.D Foundation, they cannot act openly, not knowing the extent that the R.A.I.D. Foundation has grown. Naturally, this means you are on your own. The game begins you outside the factory perimeter trying to gain access to the facility.

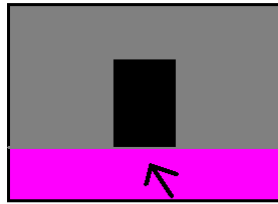
5.2 Story Board



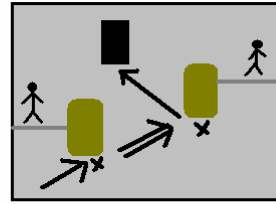
You meet a covert-ops insectoid who tells you your mission and gives you the naturalizer.



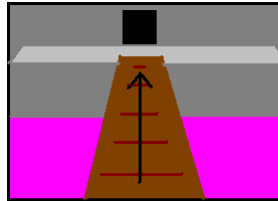
You see the perimeter of the factory and go up a spiral staircase without guards seeing you. You jump over and make it into the factory area.



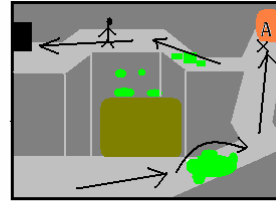
You enter the first factory area.



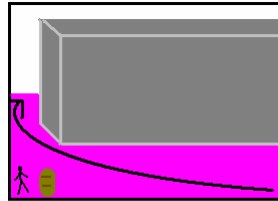
Avoid being caught by the guards and exit first factory.



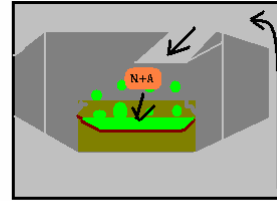
When you exit the first factory you walk across a plank which leads to the second factory.



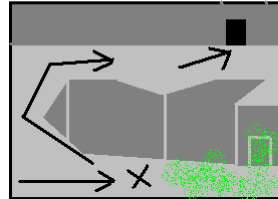
You are inside the second factory. Get pass the acid, pick up the "Agent", eliminate the guard, and get out of the second factory.



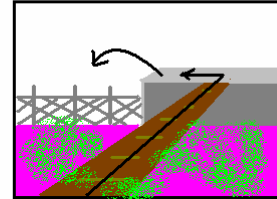
You make a run for the third factory while avoiding more guards. The entrance is in the back of the building.



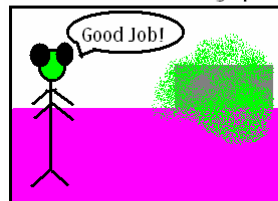
You place the "stuff" into the master vat and it sets off a chain reaction setting off a poisonous gas.



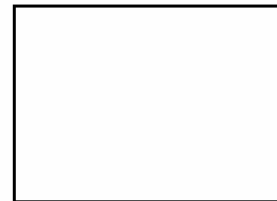
You notice the entrance you came in is blocked so you go through the exit on the top of the building. Poisonous gas is building up.



You take a plank which connects with the first factory and jump out of the factory's perimeter. Poisonous gas is building up.



You meet up with you covert-ops insectoid and he thank you for what you did. Mission Ends.



6 Game Progression

6.1 Flowcharts

The following flow charts are set to illustrate the key features of the game and how it fits into the game flow. For the game flow portion of the game we are only looking at a very high overview of a single game tick of play.

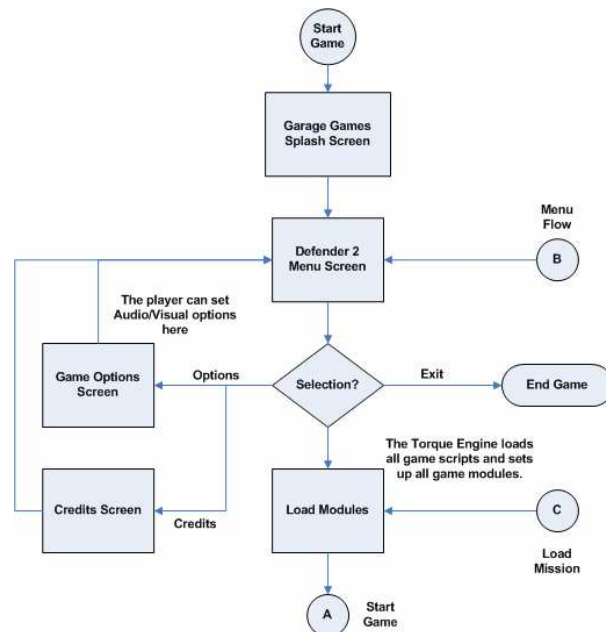


Figure 11 – Defender 2 Menu Flow

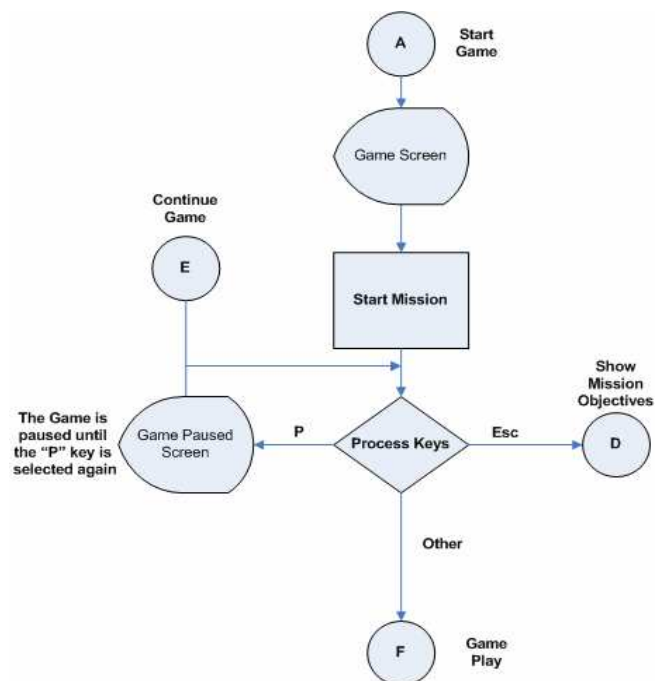


Figure 12 – Defender 2 Game Flow

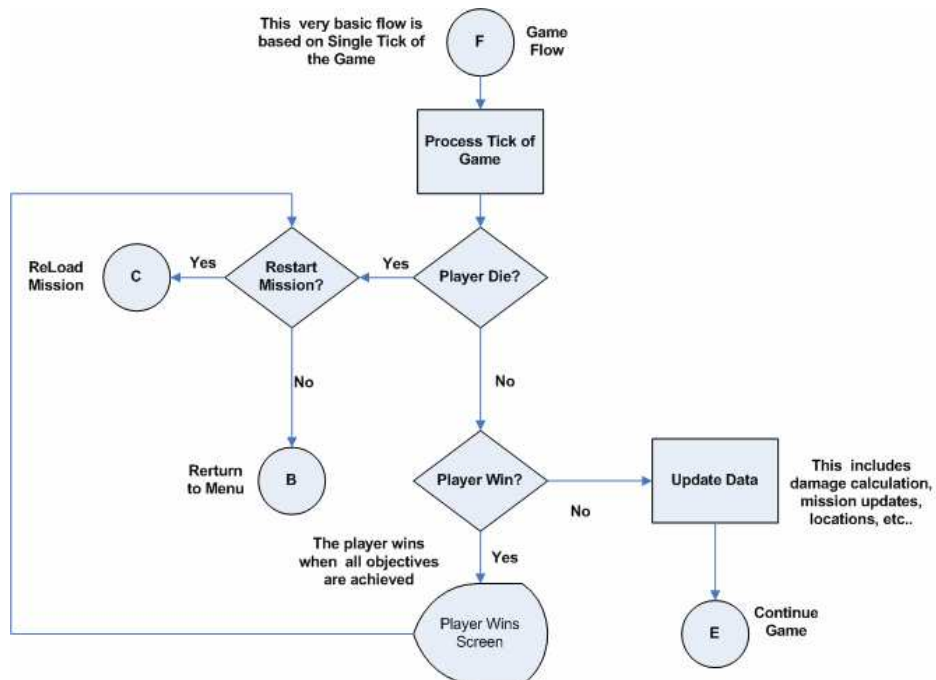


Figure 13 – Defender 2 Game Flow (Continued)

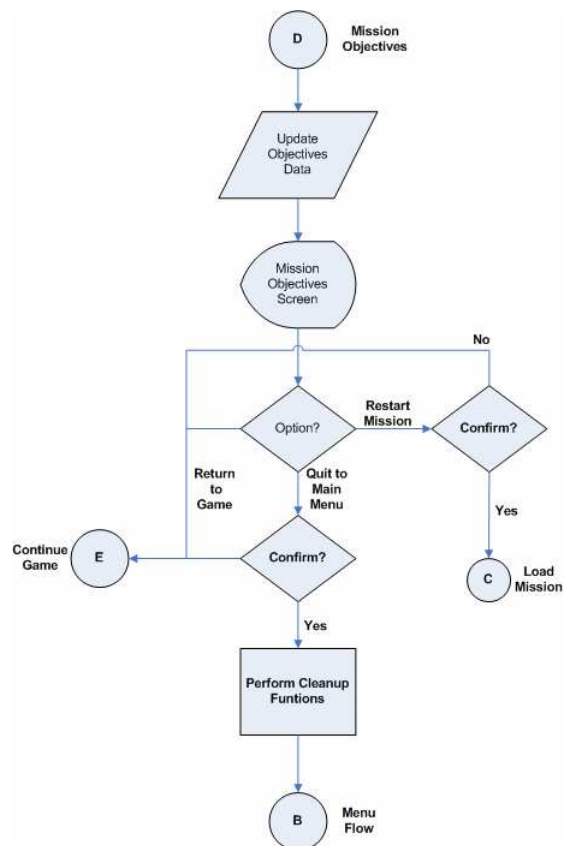


Figure 14 – Mission Objectives

7 Conceptual Level Design

7.1 Location

The location of Defender as described earlier is Xanthu VII, and Insectoid colony world the outermost planet on the Insectoid worlds. Xanthu VII has a sun consisting of enough potassium that is puts out a lilac to violet flame, filling the skies with an eerie mix of red and purple. There are high velocity winds in the upper atmosphere so the clouds move by with amazing speeds in stark contrast to the calm Mars-like surface.

Since the Xanthu Sun does not provide a strong light so the planet surface is a constant dusk view. This provides perfect condition for Insectoids, who evolved with large round eyes to absorb more of the light. This will be advantageous to Xophad since humans will only have limited vision.

7.2 Player Information

The mission takes place in the early morning hours as Xophad makes contact with Xent to receive the final briefing on his mission. At a pass just outside of the insecticide factory complex Xophad prepares for take revenge on the Humans that have ravaged his world. Xophad will not require much in the way of inventory, given the stealthy nature of the mission. Xophad will only be armed with the neutralizer, the natural Insectoid Mind Push ability, and his wits. It is here where the player will receive the mission objectives.

7.3 Mission Objectives

7.3.1 Mission Objectives

- Infiltrate the Humane outpost (factory complex)
- Find the hidden chemical agent hidden in the research facility
- Sabotage the Master Vat of Insecticide by combining the neutralizer with the chemical agent
- Get out of the complex and meet up with Xent (the mission starting point)

7.3.2 Mission Area Breakdown

The game consists of one level that is broken into multiple scenes:

- **Outdoor starting area**
 - Player meets NPC who relates the story.
- **Outside Factory**
 - Evade guards and find way into factory.
- **Shipping**
 - Area with a desk, barrels, and some guards.
- **Agent Building**
 - Player must avoid insecticide pools and retrieve the chemical agent from this building.
- **Fenced in area between buildings**
 - Area with a guard and a puzzle in how to enter Production building.

- **Production Building**
 - Contains Master Goo Vat and some guards.
- **Escape**
 - Run across top of building and jump to safety.

7.4 Challenge Highlights

Xophad is not equipped for combat, but will defend himself as best as he can. However his main weapons are stealth and cunning. As described above the main puzzle is to devise a strategy to accomplish the mission. There are no boss battles, just mission objectives (listed earlier). Basically the only objective is to destroy the insecticide or die trying.

7.4.1 Cut Scenes

There are no cut scenes per say in Defender 2, although there will be a brief explanation of the mission objectives at the beginning of the game level.

7.4.2 Storytelling

The story will be revealed by means of an Insectoid NPC, which will brief the player on the mission at the beginning of the game. Objectives will appear in an in-game menu to remind the player of the storyline.

7.5 Wow Factor

The player should be fascinated with the look and feel of the alien worlds and the visual/audio effects in the game. The main character, Xophad, should stick in the players mind, since there was considerable effort to make him as much an insect as possible with audio sounds that are samples of cicada's throughout the world.

7.6 Map Description

The figure below is a conceptual overview of the mission level map main area.

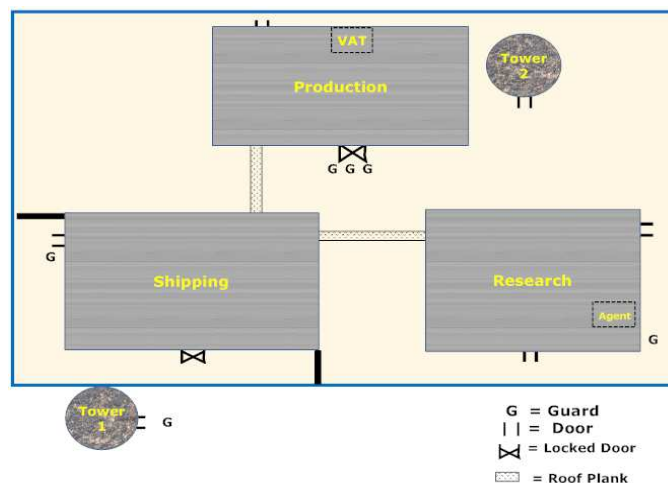


Figure 15 – Conceptual Bio-Weapon Production Facility

Although not shown in the diagram, there will be areas with bio-pools of the toxin that if stepped into, will instantly kill the player, so it should be avoided at all costs. In addition, this map is not all-inclusive.

The key objectives are to retrieve the agent from within the Research facility, combine the agent with the neutralizer at the Vat in the Production building, then get back to the drop point (The starting point of the mission).

The main points are guarded by patrols that move along a predetermined path. The facility itself can be seen and walked around by the player to get a visual layout of the place so a strategy can be formed to meet the objectives and to beat the game. The strategy used will vary somewhat by the player, but the plot was design to be linear and meant to move the player along a pre-determined plan.

The factories have metal floors and concrete walls with vats of green bubbling gas and goo that is harmful to Xophad, or any Insectoid for that matter, on contact.

7.7 Assets and Requirements

Most buildings will be created either from scratch. However, some existing models may be converted from a different format to the Torque dif standard. Most dts shapes will be gathered from other games, like Noon of the Dead and the FPS starter Kits. Many of the sounds will come from existing Torque games or be sampled from sounds from the Internet.

Much programming will be required to develop Defender 2. The Torsion IDE will be used to aid in the development of TorqueScript files. Quark and the Torque Constructor will be used with the map2dif utility to create the necessary .dif files for the buildings. Any .dts exports will be done with Maya, though most .dts files will be “borrowed” from other games. Image editing programs such as paint, The GIMP, and Photoshop will be used to re-texture or re-skin objects in the game.

The bulk of the relighting is handled by the Torque Game Engine, although it will be our responsibility to set the lighting and particle effect settings in the game.

8 Lessons Learned

The Torque Game Engine provides a lot functionality that allows the game developer to visually develop a game prototype quickly and efficiently. However, the main lesson learned is that it takes a lot of art to develop any quality game and this project was no exception.

Additionally, planning and communication is key to developing a game that is not only fun to develop, but is also consistent with the proposed game plan. Communication is the key component that must be mastered early on in the project if it going to succeed.

With this in mind it was decided to set up a Google group to provide a focal point for communication and to have a place to upload files.

Another lesson learned is realizing our project scope was not realistic early on in what we initially wanted to do given the available time frame. It was decided early on to trim the game down to a more manageable level and to use what “freebie” artwork we can to produce the game while not knowingly violating any license agreements.

8.1 Post Mortem Evaluations

8.1.1 Barry Belcher Post Mortem

Name: Barry Belcher

Instructor MAXIM

Project: Defender 2_TGE GAME

Course CIS 487/587

What went right?

1. The Torque Game Engine provides many features that are readily available.
2. The original storyline remained intact.
3. The stealth nature of the game that's gets away from the usual FPS.
4. TorqueScript allowed us to develop a terrific game in a relatively short amount of time.
5. I was lucky to have very dedicated teammates that embraced the idea of the game enthusiastically
6. There were many tutorials available to use as reference.
7. The Googlegroups allowed us to communicate rather effectively.
8. There were no arguments over the project. Some minor disagreement on game direction, but all in the best interest on the game.

What went wrong?

1. Content creation is not covered very well, if at all. This forced to custom creation that took an excessive amount of time.
2. The scripting language leaves a lot to be desired. I have worked with several scripting languages and this language allows for some very dangerous coding concepts.
3. Scope resolution is too liberal. For example if I have files A, B and C and file C refers to function x(), function x could be in either A or B or C, but there is no easy way to tell. In other words C should have been forced to call A.x() so the user can tell where the function call is from.
4. Not easy to tell what files are used by another.
5. No real script compiler, although you can buy Torsion, but this still has problems.
6. Documentation is very sporadic and there was no real example where enemy AI fired back or anything.
7. There were scheduling conflicts between team members that sometimes led to some minor difficulties.
8. My system crashed that led to some very stressful moments until I got a replacement.

Lessons learned/process improvement suggestions:

1. Need better documentation for TGE– Perhaps Torque is best, perhaps not.
2. Engine scripting is helpful, but Torsion compiler (although not perfect) definitely helped.
3. Content creation tasks should be done by external forces early on, perhaps CCS students or several art collections available for purchase/use through the university.

8.1.2 Aaron Curley Post MortemName: Aaron CurleyInstructor: Bruce MaximProject: Defender2Course CIS 487 Fall 07**What went right?**

We were rather ambiguous in our goals. For a rather complex project, we did very well to get it completed on schedule. Though we did have to “cut out” some features, we did implement most of the features we initially planned on in the design document.

I was especially happy that I was able to get the neutralizer sprayer and insecticide puddles working. I initially knew that these two features would be difficult, but when I went to begin work on them, I realized that I had underestimated the difficulties in getting them to work properly. I feel that these items turned out fairly well, given the lack of native engine support.

Thankfully, Mike immediately set up a Google group for our project. This was extremely helpful for distributing project files. Slightly annoying was the 10 MB file size limitation, but we quickly resolved the problem by uploading new versions of the project in multiple zip files.

What went wrong?

First and foremost, the lack of complete and accurate documentation was a great annoyance in this project. Although Torque supposedly has a lot of documentation, I quickly found that most of it is disorganized and incomplete. Often, the documentation is out-of-date and misleading. Many high-quality examples for how to do things could be found online or in the text; however, when trying to implement a feature with no example code similar to it (such as the insecticide puddles or automated camera movement), I wasted numerous hours searching for the necessary information. Often, one set of documentation would only contain half the puzzle, with the other half in an entirely different set of documentation, or forum post.

As an example, one of the problems I faced was that the `onCollision()` callback of ShapeBase derived objects would not be called when a bullet (from the sprayer) collided with it, even though the documentation stated that `onCollision()` was called for ANY colliding object. Changing the “collidable” property of the object did not seem to help. Only after much searching did I find a post mentioning that the `onCollision()` callback was never called by ShapeBase objects. The only reason player and terrain collisions

generate this event is that the Player/Terrain object calls the ShapeBase's onCollision() method.

We also encountered numerous problems creating artwork for the project. Although we could use media from other projects, we still lacked the custom .dif files necessary to create the factory buildings. We encountered difficulties with exporting to .dif from the Torque Constructor, and much time was spent learning how to set up the alternate tool QuArK and debugging the MAP2DIF utility (which we initially had the older version of, due to out-of-date links in the Torque documentation). However, in the end, we were able to get both QuArK and the Torque Constructor working well, once we were able to understand how to workaround the bugs.

Lessons learned/process improvement suggestions:

Because of the rather rushed development process, we did not have an opportunity to set certain standards and always inform other team members of the design of all the components. For instance, when implementing code that handled storyline progression, I designed a rudimentary messaging system. However, when another team member expanded it, he instead wrote his code directly into the trigger event handler. This was not a major problem, but it illustrates that it would be well-advised to hold more meetings and better communicate our technical design decisions.

Although I already had known this, I again learned that it is not always advisable to treat system documentation as accurate. It is best to always verify that information is correct by testing it out using a short test program.

I will be interested to see how other game engines are organized. As for the Torque engine, I was rather disappointed in the “messiness” of the Torquescript code. My overall feeling of Torquescript was that it was a “hack”, with very little to no structure at all. I don't see how professional games could possibly be maintainable if their scripting is organized in a fashion similar to the starter kits. Perhaps, if all of the Torquescript had been written from scratch, we could have had good, consistent, and object oriented coding standards throughout the project.

8.1.3 Mike Hardy Post Mortem

Name <u>Michael Hardy</u>	Instructor <u>MAXIM</u>
Project <u>Torque Game Defender 2</u>	Course <u>CIS 487/587</u>

What went right?

1. Final game was quite large with many features
2. The mission stayed true to the design document, with tweaking of locations.
3. An interesting idea of an Insectoid with special weapons (not the standard range or melee items) was successfully made.
4. Regular testing has squashed many bugs (but not the two green insectoid ones).
5. AI can limit LOS through vector calculations and ray casting in scripts for obstructions allowed for AIs that didn't act stupid.

6. Plenty of source code online to use as references where the Torque documentation failed.
7. Torsion editor demo helped track down errors sooner than just using Torque command line.
8. Game can be freely distributed.
9. Had fun lending my voice to the Soldier's pain, death, and alert sounds.

What went wrong?

1. Content creation was difficult. Editing the textures was a bit haphazard. Tools that were supposed to export models only worked with the demo files.
2. The Torque engine had a less rigorous data structure design, allowing accidentally overriding methods. Parameters that were invalid or mistyped didn't always give a warning while others gave warnings when everything was reasonable.
3. Torque documentation often gave examples from previous versions that have since changed.

Lessons learned/process improvement suggestions:

1. Pick the engine based on the documentation – Torque could be the best?!?!?
2. Engine scripting was indicative of real life game programming.
3. Content creation tasks could rely on more external libraries instead of making in class to put more focus on programming.

8.1.4 Evan Musu Post Mortem

Name Evan Musu Instructor MAXIM
 Project Defender 2(Torque Game) Course CIS 487/587

What went right?

- The project did get all the main features promised on time.
- There was easy communication through our Google group page.
- The community on the garage games website is very robust and most of the time is very accurate.
- The Torque Engine is user-friendly and is not user intimidating.

What went wrong?

- Not too much easy access documentation on the Torque Engine (but you can say that about a lot of programming languages too).
- The schedule function explained in the Maurina textbook does not explain all the variations of the function. I found there about 4 variations of using that function from Aaron's research and reading post from the garage games website.
- As for the Maurina, it is a quick guide to the Torque Engine but doing the Code Sampler's examples were much more useful (maybe requiring the Finney books would be better).

Lessons learned/process improvement suggestions:

- Give out this website <http://www.garagegames.com/docs/tge/general> as a starting point to figure out what all the variables mean in each of the basic data blocks that go along with the Torque Game Engine. The website was a very good starting point for me. Also the code sampler website was very helpful too (everything was accurate for those tutorials).
- Giving out Torque Engine License Number to the Garage Games website might be useful to access some of the restricted area of the garage games website (I know Maxim offered us the license key but I still manage to figure out parts of the scripting).
- Torsion and Torque Constructor are two useful tools to use with the Torque Engine. I highly recommend getting a license for the school for Torsion.
- Overall, using the Torque Engine was an interesting dive into 3D game engine scripting.

9 Bibliography

- Belcher, Barry and Curley, Aaron, **Defender**, Created 10/2007
- CCS and UM-D, **Noon of the Dead**, Creators listed below

University of Michigan – Dearborn

John Andzelik	Dustin Kelley	Eric Nelson
Todd Pugh	Robert Rak	Brian Schroeder

Center for Creative Studies

Kevin Bindschadler	Jason Briney	Rex Hamilton
Justin Harrison	Adrian Haynes	Carl Vogler

- Finney, Kenneth C., **Advanced 3D Game Programming All in One**, ISBN-10: 1592007333, ISBN-13: 978-1592007332, Printed 2004
- GarageGames.com, **FPS Starter Kit**, Copyright © 2000-2007
- **Insectoid Sounds Modified for use in Defender 2 (Authors Unknown)**
 - oookame.mp3, <http://mushinone.cool.ne.jp/English/ENGitiran.htm>
 - cicada1.wav , <http://www.naturesongs.com/insects.html>
 - merula22-8.wav and trill22-8.wav, http://www2.arnes.si/~ljprirod3/asian_cicadas.html
- Harris, Kevin R., **Torque Game Engine Tutorials**, <http://www.codesampler.com/torque.htm>, Copyright © 2005
- Maxis, **Sim Copter Sound Effects**, Copyright © 1996
- Maurina III, Edward F, **The Game Programmer's Guide to Torque**, ISBN-10: 1568812841, ISBN: 978-1568812847, Printed 2006
- RokasK and Turbo Squid, Inc, **Neutralizer and Agent Models: Nitric Oxide System**, <http://www.turbosquid.com/FullPreview/Index.cfm/ID/270144>, Copyright © 2007