The **BITMAP BROTHERS**

PC Format

Xenon 2000: Project PCF

© 2000 The Bitmap Brothers

Contents

Section 1: Overview

Introduction.....5
Background.....5
Overview.....5
Best Hi-score.....5
Best Level Design.....5
Best Mod.....5

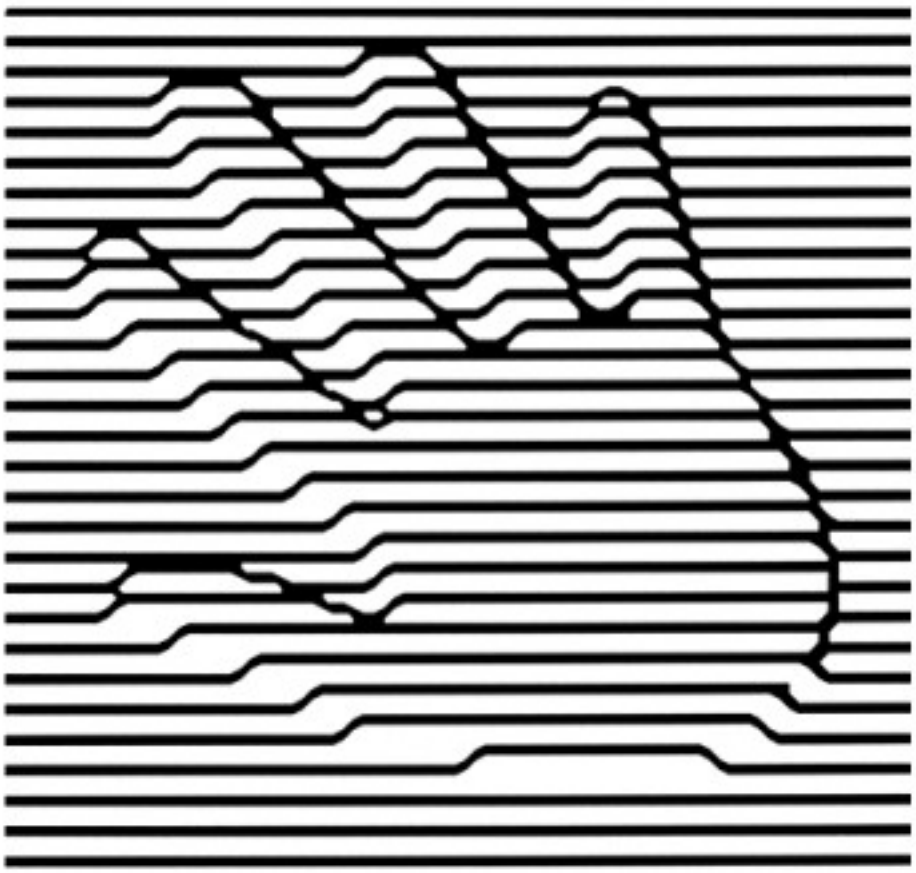
Section 2: Game features

Installing the Game.....7
Attract Mode.....7
Menu System.....7
Start Game.....7
Options.....7
High-score Table.....7
Credits.....7
Quit.....7
Game Environment.....8
Space.....8
Asteroid Belt.....8
Planet.....8
Controls.....9
Basic Play Format.....9
Checkpoints.....9
2 Player Mode.....9
HUD.....10
Pickups.....11
Hardware.....11
Clone.....11
Wing-tips.....11
Homing Missile.....11
Laser.....11
Weapon Up.....11
Software.....12
Energy.....12
Speed Up.....12
Cloak.....12
Dive.....12
Score Bonus.....12
Alien Types.....13
Static Wall Hugger.....13
Moving Wall Hugger.....13
Asteroid.....13
Rusher.....13
Pod/Spores.....13
Homing.....13

Drones....14
Loner....14
Organic Gun....14
Organic Projectiles....14
End of Game Boss....14
Alien Shield Values....14
Scoring....15
Alien Score Values....15
Hi-score Table....15

Section 3: Technical Design

Consumer Requirements....17
Resources....17
Compiler....17
Text Editor....17
Map Editor....17
Music Playback....17
Graphics Editor....17
Technical Overview....18
Game Library....18
Main Game Code....18
Graphics Requirements....19
Sprites....19
Backgrounds....19
Sprite Requirements....20
Background Blocks....21
Sound Requirements....22
Music....22
Menu Samples....22
Game Samples....22
Article Structure....23
Part 1 - Introduction....23
Part 2 - Programming....23
Part 3 - Design....23
Part 4 - Graphics....24
Part 5 - Sound....24
Part 6 - The Finished Game....24
Key....24



Overview

Xenon 2000: Project PCF

© 2000 The Bitmap Brothers

Introduction

The idea behind the project is to create a simplistic, yet feature-packed and above all fun game, which can be produced in a very short space of time and then split into six cohesive sections in order to show **PC Format** readers the methodology behind creating a computer game.

The first issue will come complete with all the necessary development tools and utilities on the cover mount disc with all necessary documentation and explanations of how to set up a basic development environment. The subsequent issues will then show the reader in easily digestible chunks how to put the various aspects of the game together. The focus will be on coding, with additional sections each month relating to design and graphical issues.

Due to the fact that this feature is aimed at a very wide audience base we need to ensure that it is not over-complicated, and as such we have decided on creating a 2D game as the added complications of a third dimension make staying within the project boundaries unfeasible.

Having looked at our back-catalogue of titles and toying with a number of new concepts the most popular idea has been to create a modern interpretation of the classic **Xenon** series.

Background

Xenon was The Bitmap Brothers' first game, released in January 1988 to critical acclaim. An arcade-style scrolling shoot-em-up, it rocketed to fame after being used as a player challenge on ITV's Saturday morning kids show Get Fresh. It went on to make history as the first Amiga title to enter the UK top 40, selling over 100 000 units.

18 months later The Bitmap Brothers did the impossible by improving on the original in every area. **Xenon 2** is regarded by many as the best shoot-em-up of its era, and continued to create history and set trends by being the first game to feature a licensed soundtrack (Megablaster by Bomb the Bass). It also won numerous awards including Best Shoot-em-up and Best Music & Graphics and sold nearly a quarter of a million units worldwide. To this day it is still referred to as an influential, genre-busting title and is one of the most requested games for a 'true' update.

Overview

As with the previous **Xenon** titles the game will be a 2D vertically-scrolling shoot-em-up, specially constructed to give the player maximum variety, enjoyment and longevity whilst also allowing us to teach the reader as much about the basics of designing and coding a game as is possible within the limited confines of the project.

Upon completing the project, it will be a nice idea to maintain the interest of both people that have completed the game and those yet to even start or finish coding it, by running various competitions judged both by PC Format and us. Examples are as follows:

Best Hi-score: Which player can achieve the highest score in the game? (Possibly too easy to cheat!)
Best Level Design: Which player can create the best new level?
Best 'Mod': Which player can modify the standard code to create the most interesting and innovative new game features.

Possible prizes could range from a PC with a full C++ programming kit to a week working at Bitmap HQ testing Z2, although this is something that needs to be discussed.

Game Features

Xenon 2000: Project PCF

© 2000 The Bitmap Brothers

Installing the Game

For people who are following the project, they will be running the game through the compiler so installation is not an issue. For those who are not interested in following the project but still want to play the game, we will supply a full version with a minimal install shield or self-extracting executable on the final cover disc of the project.

Attract Mode

If the game does not sense any peripheral input for a preset period whilst in the Main Menu, it will enter the 'attract mode'. The idea of the attract mode is to stop screen burn-in and also to attract peoples attention towards the game.

The attract mode will cycle the following features until such time as the game senses input from the player, whereby it will return to the Main Menu:

Main Menu
Hi-score Table
Main Menu
Recorded Demo
Repeat

Menu System

The menu system will consist of an animated star field backdrop with the various options printed as text in the centre of the screen. The player will be able to highlight and select the various options using their preferred control method, which by default will be the cursor keys for selection and the fire key to confirm. The Main Menu will consist of the following options:

Start Game

1 Player / 2 Player

Options

Video:	*Resolution:*	*320x240 / 640x480*
	Screen Mode:	*Full Screen / Windowed*
	Colour Depth:	*8-bit / 16-bit / 24-bit / 32-bit*
	Particle FX:	*On / Off*
Audio:	*Music:*	*0 - 10*
	SFX:	*0 – 10*
Control:	*Keyboard / Joystick*	

Hi-score Table

View Hi-scores

Credits

View Credits

Quit

Quit Game: Yes / No

Game Environment

The game will contain a single level consisting of three distinctly different linked stages. The journey through the level will take the player from deep space, through an asteroid belt and then into the core of a planet where they must face an end of game ‘boss’.

This is enough to be able to give the player a varied game experience whilst also allowing us to show them a broad range of different programming and design techniques.

Space

The first section of the game will be set in space and will feature a scrolling parallax star field background. The player will be eased gently into the action, with early chances to increase their crafts speed and firepower, and also some decent combination score opportunities from the snake-like waves of enemy attack.

Asteroid Belt

The three sections of the game will blend seamlessly into each other, and when the player reaches the end of the initial space section they will see their reflexes and trigger speed thoroughly tested as they pass through an asteroid belt.

There will be various sizes and types of asteroid, with some requiring more hits than others to destroy, some which break into smaller asteroids when shot and some indestructible ones that are best avoided altogether!

Planet

At the end of the asteroid section of the game, the player will move through the outer atmosphere and inside the planet itself. This section will involve the player dealing with all sorts of enemy types and configurations along with dangerous environmental features, culminating in a face off with the final end of game ‘boss’.

The planet interior is also split into 3 sections:

Rock: When the player first enters the planet, the environment will consist of barren rock.

Installation: As they move deeper into the planet, the rock walls will gradually change to hi-tech alien alloys as the player enters the start of the alien installation.

Core: As the player nears the very centre of the planet, the installation walls will be covered with organic alien flesh, culminating with the end-of-game boss’ lair.

Xenon 2000: Project PCF

© 2000 The Bitmap Brothers

Controls

The player will be able to control the craft using either the keyboard or a compatible joypad.

The control configurations will be preset, and will consist of the following:

8-way directional movement
Fire (hold for auto-fire)
Dive
Fire-direction toggle (forwards / backwards)

The player will also be given the choice of two different control configurations.

Basic Play Format

- The player will start the game with a quota of three ships, and the game ends upon losing their final ship.
- A ship is lost when its shield energy level falls below zero.
- Craft add-ons have independent energy values.
- Energy is lost through the player colliding with various objects, be it an enemy projectile, an enemy or the environment itself. The value of energy lost will depend on the values of the object the player collides with.
- When the player loses a craft they will respawn at the last checkpoint they passed, with the full default health/ship values but without any of the ship enhancements they had previously acquired.
- The player will be awarded an extra craft upon reaching 4000000 points.
- Upon losing their final craft, the game will end. If the player has attained a high enough score they will enter the name entry screen, otherwise they will be returned to the Main Menu.

Checkpoints

Checkpoints are invisible markers at set points through the level. As a player passes a checkpoint it will be triggered in the game code, with the player's ship spawning at the last-triggered checkpoint upon losing a life.

2 Player Mode

When the player chooses to start a new game from the Main Menu they will have the option to choose from a solo game or a two-player game. The two-player mode works in exactly the same way as the solo version, except the players take it in turns to use their quota of ships.

Neither player can affect the outcome of the second party in any way; the winner is simply the player with the highest score at the end of the game.

When a player dies the screen will clear, with the text 'Player X Ready' (where x is the number of the player who's turn it is). The next peripheral input will then automatically trigger the game to start, with the player continuing from the last checkpoint they passed.

HUD

On-screen information whilst playing the game should consist of the following information:

Whilst in two-player mode, we will use coloured text to indicate which player is currently active.

Pickups

As the player progresses through the game they will be rewarded with various pickups, either randomly or for destroying certain enemies. The player must physically collect these pickups by flying the craft over them in order for them to take effect.

Hardware

'Hardware' pickups are:

Clone

The clone is a small pod-like craft that follows alongside the players ship. Unlike **Wing-tips**, which are physically attached to the ship, clones hover a set distance to the left and right, moving forwards 45 degrees when the player moves backwards, backwards 45 degrees when the player moves forwards and firing when the player fires. Each clone will have its own independent shield value.

Configuration: 1 or 2, positioned to the left and right of the players' ship.
Shield: 60 units (50% of the ship value)
Ammo: Can fire standard blaster bullets and lasers in a forward direction only.

Wing-tips

These hardware add-ons sit on each of the crafts wing tips, increasing the players' firepower. Each wing tip will have its own independent shield value.

Configuration: 1 or 2, attached directly to the left and right of the player's ship.
Shield: 30 units (50% of the Clone value)
Ammo: Can fire standard blaster bullets and homing missiles both forwards and backwards.

Homing Missile

A self-explanatory weapon, the homing missiles will lock on to whichever hostile is nearest to the player when fired, although it should prioritise wall-mounted or rear-attacking enemies if there are any on-screen.

Laser

The most powerful weapon, the laser is relatively thin and so needs an accurate aim to be fully effective. Best applied in conjunction with a full quota of Clones!

Weapon Up

As with the Speed Up, the Weapon Up increases the potency of the ships standard blaster in terms of speed, size and strength. Also effective for Wing tips and Clones.

The player can collect a maximum of three of these simultaneously.

*See **Hit Point** table (pg.) for weapon values.*

Software

'Software' pickups are:

Energy

The energy pickup will replenish 50% of the players' shield value.

Speed Up

This handy pickup increases the response and movement speed of the craft. The player can collect a maximum of three of these simultaneously. The first will increase the movement speed by 25%, the second by another 15% and finally the third by 10%.

Cloak

This enhancement renders the players' craft indestructible for 5 seconds. Whilst the pickup is in effect, the players' craft should be translucent, and when the pickup is about to wear off the craft should flicker/flash to warn the player that they need to be in 'clear air'.

Dive

When the player uses a 'dive' pickup, the ship will appear to drop into the background, where it will remain for a period of 4 seconds. Whilst in this state the player cannot attack or collide with any foreground or background objects, they are technically indestructible. However, upon returning to the surface the player must make sure that they do not reappear beneath an active foreground layer or they will be killed instantly.

Upon diving into the background, the players' ship and any add-ons (wingtips/clones) should simultaneously scale down in size and darken in colour to give the appearance of dropping into the background.

Score Bonus

Upon defeating an entire 'batch' of certain enemy types, the player will be awarded with a score bonus. This will not automatically be added to the players' score, instead appearing as a pickup in the location of the last enemy killed. The player is then left with a tricky decision to make when confronted by a fresh wave of enemies or a difficult environment feature en-route to collecting their hard-earned bonus points.

Note that the amount awarded may differ depending upon the enemy type and number defeated.

*See **Scoring** table (pg.) for score values.*

Alien Types

A – Static Wall Hugger

As the name suggests this is a static, wall-mounted enemy, which can be positioned in some awkward nooks and crannies that can make it difficult to kill without homing missiles. It fires standard, fairly fast-moving poisonous organic projectiles directly across the screen.

B – Moving Wall Hugger

Same as **A**, except it can follow the contours of the walls.

C – Asteroid

There are three types of asteroid:

1 – Indestructible
2 – High-density
3 – Standard

Each asteroid will also come in three sizes:

a – Small
b – Medium
c – Large

The different sizes and types of asteroid determine how many hits each will require to destroy.

Note that the largest of the asteroids will break into a number of small asteroids.

D – Rusher

This fast-moving enemy does not fire projectiles. Instead they will 'rush' across the screen in formations.

Rushers can travel in straight lines only from top to bottom, bottom to top, left to right and right to left. Vertically travelling Rushers can move individually, but horizontally moving ones should appear in packs of 6.

E – Pod/Spores

This large and slow-moving enemy slowly homes in on the players' craft. When the player shoots it the outer skin will be destroyed, releasing a large number of very small and weak 'spores' that will home in on the player if not destroyed quickly.

1 – Pod
2 – Spores

F – Homing

This mid-sized enemy homes in on the players' craft very quickly. When the player shoots it, it will explode releasing fast moving projectiles in 8 directions.

G – Drones

Drones are small, fairly fast enemies that move in a snake-like formation of 8, each mimicking the actions of the one in front.

H – Loner

These enemies move from top to bottom, firing fast moving poisonous projectiles directly at the players' craft.

There will be three types of Loner:

1 – Standard
2 – Medium
3 - Armoured

The different types simply define the movement speed and hit points of the enemy, with the Standard being fast moving but easy to kill and the Armoured, slower moving but with more hit points.

J – Organic Gun

The Organic Gun is a fixed, organic turret that fires poisonous projectiles horizontally across the screen.

K – Organic Projectiles

A number of the alien enemies including the end of game boss will fire poisonous organic projectiles as weapons.

Although tough, these projectiles can be destroyed before hitting the players' craft or the player can attempt to dodge them, as they have no homing capabilities.

L – End of Game Boss

When the player reaches the end of the level, the screen will cease its inexorable forward scroll to allow the player to attack the boss. The boss will consist of a large creature with 6 eyes and a mouth and to destroy it the player must shoot out each of its eyes.

The boss will alternate between being passive, at which point its eyes will be open and the player can attack, and being aggressive, at which point the eyes will be closed and it will fire a random spread of poisonous organic projectiles towards the player.

Alien Shield Values

Enemy Type	Hit Points	Enemy Type	Hit Points	Enemy Type	Hit Points
A	5	**B**	5	**C1a**	N/A
C1b	N/A	**C1c**	N/A	**C2a**	2
C2b	4	**C2c**	4	**C3a**	1
C3b	2	**C3c**	3	**D**	5
E1	10	**E2**	1	**F**	5
G	1	**H1**	3	**H2**	5
H3	8	**J**	5	**K**	10
L	125				

Scoring

Due to the fact that the game only contains a single level, the emphasis in Xenon will be very much biased towards point scoring, encouraging players to return to the game in an attempt to better the scores achieved by themselves or their friends.

The main element of scoring will be the combo multiplier. When the player attacks batch-type enemies such as drones, each member of the batch the player destroys in succession will multiply its score. For example if the first enemy scores the player 500 points, the second and third will score 1000 and 2000 respectively.

If the player manages to destroy an entire batch of aliens, they will be awarded a bonus score pickup. (Note that the player must physically collect this pickup in order for the score to be added.) The player **must** kill the enemies in a batch sequentially to obtain the combo **and** batch bonus scores.

Alien Score Values

Enemy Type	Score	Batch Bonus
A	30000	N/A
B	30000	N/A
C1a	-	-
C1b	-	-
C1c	-	-
C2a	10000	N/A
C2b	15000	N/A
C2c	25000	N/A
C3a	5000	N/A
C3b	10000	N/A
C3c	15000	N/A
D	10000	50000
E1	20000	N/A
E2	5000	20000
F	50000	N/A
G	5000	50000
H1	15000	N/A
H2	25000	N/A
H3	40000	N/A
J	30000	N/A
K	5000	N/A
L	75000	N/A

Hi-score Table

As with all shoot-em-ups, Xenon will include a hi-score table to complement its scoring system.

When a player loses their final ship, if their score is high enough to warrant entry to the table, they will be prompted to enter three letters of their choice. This will then update the hi-score table, which can be viewed during the games attract mode or by selecting the hi-score table from the Main Menu.

Technical Design

Xenon 2000: Project PCF

© 2000 The Bitmap Brothers

Consumer Requirements

In order to complete the project the consumer will require the following:

Hardware	Software
Pentium P166	Windows 95/98
32Mb system memory	Direct X 7 (supplied)
4Mb graphics card	Borland C++ 5.0 (supplied)
SoundBlaster or compatible	Text editor (supplied)
	Map editor (supplied)
	2D graphics package (supplied)

Resources

This is a list of programming tools required to allow the readers to edit and compile the game and the various tutorial programs. Also included are a 2D graphics package and a map editor to allow editing of the games' background tiles.

Compiler

Borland C++ 5.5 Command Line Tools

Status: Free
Website: http://community.borland.com

Text Editor

VIDE – a freeware development environment

Status: Freeware
Website: http://www.objectcentral.com
Contact: bruce@objectcentral.com

Map Editor

Mappy – a tile-based 2d map editor

Status: Freeware
Website: http://www.rbsite.freeserve.co.uk/robmpy.htm
Contact: rburrows@bigfoot.com

Music Playback Library

BassMod

Status: Freeware
Website: http://www.un4seen.com/music/
Contact: bassmod@un4seen.com

Graphics Editor

Paint Shop Pro or even Windows Paint

Technical Overview

The entire project will be written in C++ as this is the language of choice for most developers nowadays and makes it much easier to introduce the basic concepts of writing a game.

The project is split into two parts:

- a simple game library containing the basic functions needed for writing a game
- the game code itself

The game will use DirectX to access the graphics and sound hardware, although the low-level details of using DirectX will be hidden inside the game library.

Game library

The game library contains the code for the following basic functions:

- setting up the screen
- loading graphics from the hard drive
- drawing a scrolling parallax background
- drawing sprites
- reading the keyboard, mouse and joystick inputs
- printing text etc.

This library is designed to hide the details of using DirectX since we don't really want the user to get into too many low-level details. We can provide documented source code for the game library on the cover disc for those that want to delve into the inner workings themselves.

Main Game Code

This is the code for the game itself and will be the main focus of the articles. It uses the functions of the game library and is concerned purely with the functions of the game:

- displaying menu screens and high score tables
- the main game loop
- scrolling the background and star field
- controlling the ship
- firing bullets
- moving the alien sprites
- checking for collisions between the player / aliens / bullets and the background etc.

Graphics Requirements

The game is designed to run on a 640x480 screen (windowed or full-screen).

Only True Colour display modes are supported (i.e. 8,16,24 or 32-bit displays)

All source graphics should be 24-bit BMP files.

The engine supports colour keying for masking, so use something like RGB (255,0,255) for transparency.

Sprites

All animation frames for a single game object (e.g. player, alien, weapon) should be tiled across a single BMP file e.g. if you've got a 64x64 pixel sprite with 8 frames of animation this could be stored as a BMP of size 128x256 (or even 64x512), just as long as successive frames are arranged next to each other going left to right, top to bottom.

Backgrounds

The background consists of three scrolling layers.

Backdrop Layer

Fairly sparse parallax star field layer on a single bitmap – distant galaxies / nebulae's / planets

Back Layer

This will contain the main background structures, consisting of rock-like formations.

The player cannot collide with the Back Layer, even when in dive mode.

Front Layer

This scrolls at twice the speed of the Back Layer allowing extra features to be overlaid on top of the background. The player will collide with anything in this layer (except in dive mode).

The background is made up from 32x32 pixel tiles. This means the 640x480 screen is 20 tiles width and any number of tiles high.

The engine supports a total of 1024 different tiles for the background and each individual position on the background can be animated using a series of tiles. The tile set is shared amongst the back and front layers.

The background blocks can be stored in as many BMP files as you like, since the map editor let's you import multiple BMP files. So for example if you've got a large feature made from multiple blocks, store it in a separate BMP file.

Xenon 2000: Project PCF

© 2000 The Bitmap Brothers

Sprite Requirements

Sprite	File Name	Size (Horiz x Vert)	Animation
Player's ship:			
Spaceship	Ship2	64x64	Left/right/dive/climb
Clone	Clone	32x32	Rotate/dive/climb
Wing Tips	Wingtip	32x32	Rotate/dive/climb
Ship Thruster	Burner1	16x32	N/A
Clone Thruster	Burner2	32x32	N/A
Ship Retro Boosters	Retros	32x32	4 directions
Pickups:			
Shield	PUShield	32x32	Rotate Left
Speed Up	PUSpeed	32x32	Rotate Left
Weapon Up	PUWeapon	32x32	Rotate Left
Cloak	PUInvuln	32x32	Rotate Left
Dive	PUDive	32x32	Rotate Left
Score Bonus	PUScore	32x32	Rotate Left
Homing Missile	PUMissil	32x32	Rotate Left
Laser	PULaser	32x32	Rotate Left
Player's Weapons:			
Missile	Missile	16x16	3 types
Homing missile	HMissile	16x16	8 directions
Laser	N/A	N/A	Programmed effect
Enemies:			
Wall Hugger	Wallhugger	64x64	Left/right facing/fire
Indestructible Asteroid	Master32/64/96	96x96/64x64/32x32	Spin
High-density Asteroid	Gaster32/64/96	96x96/64x64/32x32	Spin
Standard Asteroid	Saster32/64/96	96x96/64x64/32x32	Spin
Rusher	Rusher	64x32	Flap wings
Pod	Pod	96x96	Pulsate
Spores	SporesA	16x16	Spin
Homing	Homing	64x64	Spin
Homing Projectiles	HomProjc	16x16	8 Directions
Drones	Drone	32x32	Rotate
Loner Standard	LonerA	64x64	Tentacles
Loner Medium	LonerB	64x64	Tentacles
Loner Armoured	LonerC	64x64	Tentacles
Organic Gun	GShoot	64x64	Left/right facing/fire
Boss	Bosseyes2	32x32	Open/close
Enemy Weapons:			
Bullet	Spinners	16x16	3 Types/spin
Explosions:			
Small	Explode16	16x16	
Medium	Explode32	32x32	
Large	Explode64	64x64	
Smoke	Smoke	32x32	

Background Blocks

Distant Parallax layer

Galaxies / Nebulaes / Planets
Starfield / Space debris (programmed effect)

Back layer

Rock structures, which can be joined together in various formations.

Front layer

Various rock formations for first section.

Various hi-tech structures etc. for second section. (Perhaps with bridges for the player to dive underneath.)

Organic, fleshy-looking walls and objects for the final section. Make this visually in-keeping with the theme of the end of game boss.

Sound Requirements

Music needs to be in MP3 format.

Sound effects need to be 16-bit mono samples in WAV format.

Music

File Name	Description
MUSIC_TITLE	Ambient music for menu screens
MUSIC_INTRO	Short and snappy fanfare for level intro
MUSIC_HISCORE	Ambient music for high score entry
MUSIC_GAME	1-2 minute looping music for in-game
MUSIC_BOSS	Short, tense looping music for end of game boss
MUSIC_OUTRO	Music for immediately after defeating boss

Menu Samples

File Name	Description
SAMPLE_MENU_SELECTION	Cursor up/down on menus
SAMPLE_MENU_OPTION	Cursor left/right to change option
SAMPLE_MENU_CLICK	Option selection
SAMPLE_MENU_BACK	Go back/cancel

Game Samples

File Name	Description
SAMPLE_PLAYER_CREATED	Player ship spawning on-screen
SAMPLE_PLAYER_DESTROYED	Player ship exploding
SAMPLE_FIRE_MISSILE	Simple shot effect
SAMPLE_FIRE_HOMING_MISSILE	Shot with a trailing whoosh
SAMPLE_FIRE_LASER	Generic laser zap effect
SAMPLE_SMALL_EXPLOSION	Generic small explosion effect
SAMPLE_MEDIUM_EXPLOSION	Generic medium explosion effect
SAMPLE_BIG_EXPLOSION	Generic large explosion effect
SAMPLE_ASTEROID_BREAKUP	Rock cracking up

SAMPLE_PICKUP	Player collecting a pickup
SAMPLE_BONUS	Player collecting a score bonus
SAMPLE_CHECKPOINT	Player passing a position checkpoint
SAMPLE_DIVE_DOWN	Player ship diving into screen
SAMPLE_DIVE_UP	Player ship returning to top level
SAMPLE_HIT_BACKGROUND	Player colliding with Front Layer

Xenon 2000: Project PCF

© 2000 The Bitmap Brothers

Article Structure

This serves as an outline of the structure, content and chronology of the articles.
All grabs should be 300-dpi jpeg's or standard Windows screen grabs.

Part 1 – Introduction

Double page Bitmap Brothers profile
Double page Game Design article
Single page project introduction

Section	Words	Content	Who
Main	25 strap 900 copy 3 grabs	Overview of directory structure of project Compiling and running the demo programs	JMP
6 step walkthrough	60 words 6 grabs	Editing Demo Program to display different sprites	JMP
Game Boxout	300 words 1 grab caption	Setting up compiler and other tools Description of tools	JMP
Personal Boxout	300 words	Introduction Editorial	PCF
Demo Program	N/A	Moving sprites around the screen	JMP

Part 2 – Programming

Section	Words	Content	Who
Main	25 strap 900 copy 3 grabs	Overview of game structure	JMP
6 step walkthrough	60 words 6 grabs	Changing the type of weapon the ship fires	JMP
Game Boxout	300 words 1 grab caption	Describe programming role	JMP
Personal Boxout	300 words photo	Profile of team member including their role on the project/at Bitmaps. How got into industry etc.	JMP
Demo Program	N/A	Moving players ship and firing bullets	JMP

Part 3 – Design

Section	Words	Content	Who
Main	25 strap 900 copy 3 grabs	Overview of the map editor Mappy Structure of the map	JMP
6 step walkthrough	60 words 6 grabs	Editing the map using Mappy	JMP
Game Boxout	300 words 1 grab caption	Describe level design Why chose particular structure etc.	EJB
Personal Boxout	300 words photo	Profile of team member including their role on the project/at Bitmaps. How got into industry etc.	EJB
Demo Program	N/A	A scrolling background	JMP

Xenon 2000: Project PCF

© 2000 The Bitmap Brothers

Part 4 - Graphics

Section	Words	Content	Who
Main	25 strap 900 copy 3 grabs	Structure of BMP files for animated aliens Discussion of alien movement	JMP
6 step walkthrough	60 words 6 grabs	Editing the map to add alien start points	JMP
Game Boxout	300 words 1 grab caption	Describe graphics production e.g. 3ds Max Why chose particular style for game etc.	MC
Personal Boxout	300 words photo	Profile of team member including their role on the project/at Bitmaps. How got into industry etc.	MC
Demo Program	N/A	Scrolling background with aliens	JMP

Part 5 – Sound

Section	Words	Content	Who
Main	25 strap 900 copy 3 grabs	Discussion of weapons and bullets Triggering sound effects	JMP
6 step walkthrough	60 words 6 grabs	Editing code to change sound fx	JMP
Game Boxout	300 words 1 grab caption	Describe music and sound production Why chose particular style for game etc.	CM
Personal Boxout	300 words photo	Profile of team member including their role on the project/at Bitmaps. How got into industry etc.	CM
Demo Program	N/A	Ship firing bullets at asteroids with sound	JMP

Part 6 – The Finished Game

Section	Words	Content	Who
Main	25 strap 900 copy 3 grabs	Adding a main menu and high score table End of level boss	JMP
6 step walkthrough	60 words 6 grabs	Tweaking game parameters	JMP
Game Boxout	300 words 1 grab caption	Gameplay balancing	EJB
Personal Boxout	300 words	Closing Editorial	PCF
Demo Program	N/A	The finished game	JMP

Key

JMP – John Phillips (Programming)
EJB – Ed Bartlett (Design)
CM – Chris Maule (Sound)
MC – Mark Coleman (Graphics)
PCF – PC Format (Dan Hutchinson)