**Planet Defenders**

**Game Design Document**

**Version 0.1**

**08 October 2014**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Status** | **Filename** | **Reason for Amendment** |
| V0.1 | 08/10/2014 | Draft | Game Design Document v0.1 | Document created |
|  |  |  |  |  |

**Document History**

Contents

[Concept 5](#_Toc401045729)

[Elevator Pitch 5](#_Toc401045730)

[Summary Overview 5](#_Toc401045731)

[Plot 5](#_Toc401045732)

[Target Audience 5](#_Toc401045733)

[Concept Art 5](#_Toc401045734)

[Background 6](#_Toc401045735)

[Personal 6](#_Toc401045736)

[Competition / Similar 6](#_Toc401045737)

[Platform 6](#_Toc401045738)

[Features 7](#_Toc401045739)

[Key Features 7](#_Toc401045740)

[Genre / Keywords 7](#_Toc401045741)

[Technical Overview 7](#_Toc401045742)

[Projected Costs 7](#_Toc401045743)

[Development Schedule 8](#_Toc401045744)

[Version 0.1 (Pre-Alpha) 8](#_Toc401045745)

[Key Features 8](#_Toc401045746)

[Version 1.0 (Alpha) 8](#_Toc401045747)

[Key Features 8](#_Toc401045748)

[Gameplay 9](#_Toc401045749)

[First Minute 9](#_Toc401045750)

[Gameflow 10](#_Toc401045751)

[Victory Conditions 11](#_Toc401045752)

[Graphics 11](#_Toc401045753)

[HUDs 11](#_Toc401045754)

[Sounds 12](#_Toc401045755)

[Controls 12](#_Toc401045756)

[Game-Specific Subsections 13](#_Toc401045757)

[Pieces 13](#_Toc401045758)

[Level Design 14](#_Toc401045759)

[Enemies 15](#_Toc401045760)

[Plot 15](#_Toc401045761)

[Characters 15](#_Toc401045762)

[Artificial Intelligence 15](#_Toc401045763)

[Technical Aspects Section 16](#_Toc401045764)

[To Include or Not to Include? When? Why? 17](#_Toc401045765)

A. Small paragraph outlining the goal of the game. What do you want out of it? What should the players get from playing it? The style of game? Possibly a name.

B. A list of components in the game (users characters, enemies, NPCS, items, exits, menus, scoreboards, inventories, gravity, etc).

* Of these components list out their available actions, and values they will hold (health, attack power, defense, jump height, speed, etc)
* After you outline what they can do sketch out what they SHOULDN'T do.
* I would highly suggest grouping similar functionality into an abstract class and having classes inherit from that super class. Write out this class flow here as well.

C. If you have components that interact write out the allowable interactions (including keyboard command and mouse interface with your game).

* For example if you have a player-character and an enemy – what happens when one or the other touch each other? How would either damage the other?
* How does combat work?
* How do the over arching game mechanics work?

D. Level development – what would a level look like? Is this a hugely rendered area? What are the visible render bounds? Is it a tightly focus top down view? What happens when a user transitions between levels?

E. Art and Graphics – is this 2d or 3d? What sort of animations would you need for each object? What sort of animations should exist for the interaction of components (section c)

F. Sound – what points in the game should sound be present? Player-character actions? Interactions of components in section c? What sort of sound will you need to gather up?

G. Rough sketches of the over all game layout. Are you going to have an interactive menu on the left? Where would the player know how much health do they have? Is there going to be a right click short cut for the user?

H. Testing – how would this game be tested? Who will test it? A generic test plan write up would be good here (a more descriptive and exhaustive one should be made up when you know what components you are keeping or not).

I. Time line and division of duties. Do you have a team that is helping you out? What are you goals and milestones? Is this a three day game build? Three months? A time line helps keep you on task and pushes a sense of boundaries on when you should be turning over major parts of the game.

J. Marketing – do you plan on releasing this to XBOX live for sale? If so how are you going to get this on xbox live? How about as download able content from the web – what is needed then? If you just plan on doing this for personal value then not much needs to be outlined here.

# Concept

## Elevator Pitch

*Planet Defenders* is a 2D tower-defence strategy PC game, hurling the player into the role of a battlefield commander, tasked with defending the valuable remaining natural resources from the terrorist androids waging war on mankind.

## Summary Overview

The player takes command of the battlefield, and is tasked with building various strategically-placed defences to protect the valuable storage facilities required to sustain the world. After proving his worth as a battlefield strategist, the player will then be obligated to join the battle himself, helping his troops defeat the android invaders.

## Plot

The year is 2142. Thirty years previously, mankind was invaded aggressively by alien android terrorists known as the Cyenoids, whose superior technology swiftly overpowered most of the human race. Now, the last remnants of civilization exist solely in small pockets of land in North America, surviving on the precious remaining stockpiles of resources. Contact between locations is intermittent, but obtainable, and the groups have developed a loose coalition, informally known as the World Defence Brotherhood (WDB).

These resources are the only thing still sustaining the fragmented WDB; because of this, the android invaders have focused almost exclusively on hunting down the enduring WDB strongholds in order to exterminate the planet entirely.

## Target Audience

*Planet Defenders* is targeted towards male gamers between 14-22 who would typically enjoy playing role-playing and real-time-strategy PC games. Specifically, *Planet Defenders* will appeal to fans of games such as *Command & Conquer*, due to the similar theme and setting.

## Concept Art

**TO-DO**

# Background

## Personal

The background section of your game concept simply expands upon other products, projects, licenses, or other properties that may be mentioned in the introduction; so it's optional. The background section is physically separated from the introduction so that readers can skip it if they already have the information presented. But the background section is important for licensed properties and sequels and concepts with strong influences from previously released titles in the same genre. If you intend to use an existing set of code or tools or to license a game engine, then describe these items and their success stories here.

## Competition / Similar

This is a key subsection of your document. In here you must compare your game to others already developed. It is important to give a small description of the game being compared to, and point the similarities between both. This is an excellent opportunity to expand the comparisons that were already made across the GDD and give the reader a better picture of what the game will actually be.

At the end summarise your product's strong points and convince the reader why would your product sell despite its competitors. This is the trickiest part, because you must pick good opponents, otherwise the reader just won't know what you are talking about, and still keep your game’s image shining; therefore a good writing is crucial. Your ‘adversaries’ also help on the notion of how big your market can be.

## Platform

Extremely straight-forward section. Just enumerate the platforms that your game is being designed for. An estimate of the system requirements are also a good call. If needed, you can comment on porting the game and the difficulties involved.

# Features

## Key Features

* Various land types with different passability levels, requiring considerate planning to defend against enemies.
* Different enemy types with different skills introduced at various stages of the campaign, keeping the player alert and forcing them to change strategy to accommodate them.
* Achievements to encourage players to complete and replay the game.
* Engaging Campaign mode developing the story of the WDB fight against the Cyenoids.
* Replayable missions in campaign mode to attain the highest campaign rankings.
* Skirmish mode to allow players to practise their tactics before applying them in campaign mode.
* Map Editor to allow players to create their own maps for use in Skirmish mode.
* “In-play” mode, where the player controls a vehicle on the map and can use it to attack the oncoming hordes of Cyenoid attackers.

## Genre / Keywords

In a few words, define the game genre and flavor. Use existing games' classifications from magazines and awards as a guide. For example, you could choose one of the following: sports, real-time strategy, first-person shooter, puzzle, racing simulation, adventure, role-playing game, flight simulation, racing shooter, god simulation, strategy, action-strategy, turn-based strategy, side-scrolling shooter, edutainment, or flight shooter. Then you can refine your game's niche genre with these or other words for flavor: modern, WWII, alternate reality, post-apocalyptic, futuristic, sci-fi, fantasy, medieval, ancient, space, cyberpunk, and so on.

## Technical Overview

The game will be written in C++, utilising the SDL (Simple DirectMedia Layer) library for hardware abstraction. The tools used will be Microsoft Visual Studio for programming and version control, and Adobe Photoshop for artwork.

## Projected Costs

N/A

# Development Schedule

The **Milestone Schedule** subsection is where you must define each necessary steps in order to develop the game, which is basically a timeline of the intended completion of phases of your game. Through that, not only you, but also the investors, can have a very rough estimate of the interval of time needed to complete the project.

A more detailed version of the development schedule can be found in the document “Delivery Plan”.

## Version 0.1 (Pre-Alpha)

Target Release Date: **February 28th 2015**

### Features

* Working main menu system with placeholders for future content
* All basic gameplay functionality implemented
  + Mouse input for main menu and sidebar menu
  + Two tower types available for placement
  + Two enemy types available to attack
  + All land types implemented
  + Towers will attack enemies that come within range
  + Enemies can spawn from pre-defined locations
  + Enemies will use A\* pathfinding to locate the player base and move from spawn point to base

## Version 1.0 (Alpha)

Target Release Date: **May 31st 2015**

### Features

* Campaign mode with placeholders for Resume and Replay missions
* One Campaign scenario read in from a file
* Highscores available

# Gameplay

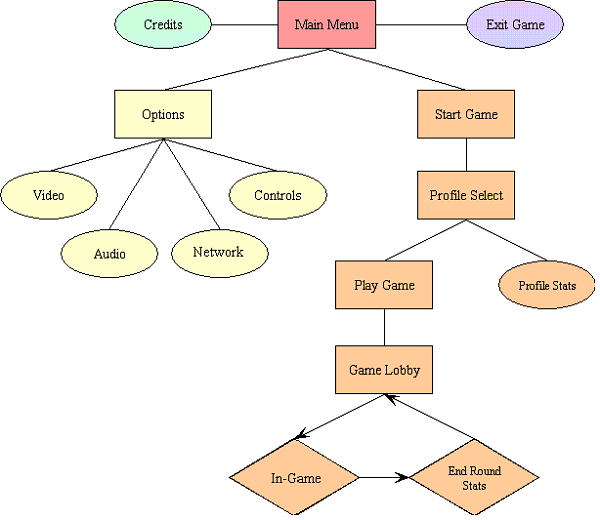
## First Minute

After the game is loaded, the player is presented with a main menu screen consisting of seven buttons overlaid on an artistic background. Two of the buttons allow the player to enter a new game (Campaign and Skirmish), two of the buttons lead to tracking player progress (Achievements and Highscores). The remaining three buttons lead to a Map Builder, Settings, and Exit.

After selecting the Campaign option, the player is presented with three options; New Campaign, Resume Campaign, and Replay Mission. Resume Campaign is greyed out if the player has not previously started a Campaign. Replay Mission will only list missions that have previously been completed by the player.

Having selected New Campaign, the game starts a new Mission One screen, with limited resources available (improved resources will be available as the player progresses) and tutorial text on the screen. The player selects to Continue to start the game and remove the tutorial text. The player then uses the mouse to build towers as desired, and clicks the Launch Attack button to send the first wave of enemies. The towers will attack any enemies that come into the range of the towers the player builds, and when enemies are destroyed, the player can use the bounty collected to build more towers.

## Gameflow



## Victory Conditions

Here you state what is required for the player to clear a stage, win a match, or advance another level, whether your game is a puzzle, where the player advances to the next level when all pieces are combined in a certain way, or a sidescrolling shooter where the player advances a stage when he defeats the boss at the end, or whatever. Obviously, this depends entirely on what kind of game you’re designing.

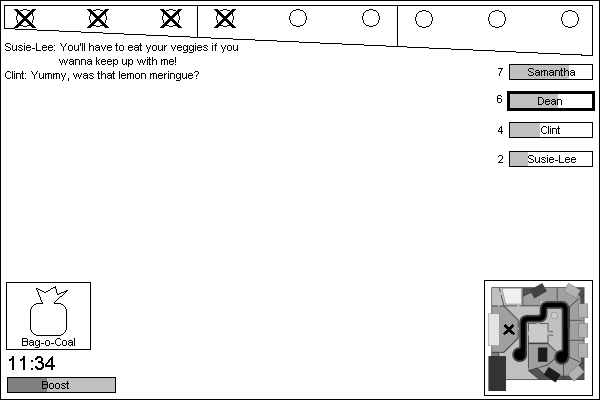
Example:

*In Space Invaders, the player advances to a new wave each time he destroys all enemies from the current wave. Since the waves are endless, the game will keep going until the player runs out of lives.*

### HUDs

The *head-up-display* is the in-game interface the player will have when playing the game. Rather than in-game menus like settings or inventory screens, this refers specifically to the floating windows and bars which don’t normally interact with the game and serve a information-only purpose. This includes health bars, mini-maps, time counters, equipped items and their amounts, money and etc. Although the size of the HUD will vary according to the game type (MMORPGs and RTSs will have big HUDs while sidescrollers and puzzles will have very small ones) keep in mind that a HUD shouldn’t occupy too much of the screen.

Example:

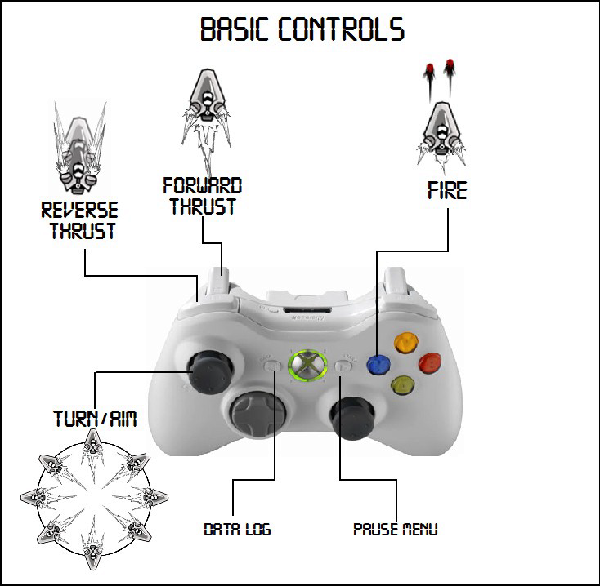


## Sounds

## Controls

Stating which buttons/keys do what can be troublesome in the case where a single button does more than one action (i.e. The ‘A’ button in any 3D Zelda). Start by putting a simple picture of a controller or a keyboard with each button highlighted with their function in a more general sense. After that, if your game has advanced combos or something similar to that, explain them carefully, stating under which conditions each combo is “activated”.

Example:



(Image from CrunchTime Games Inc's [Shred Nedbula document](http://gamecareerguide.com/thesis/080903_ShredNebula_pitch.pdf).)

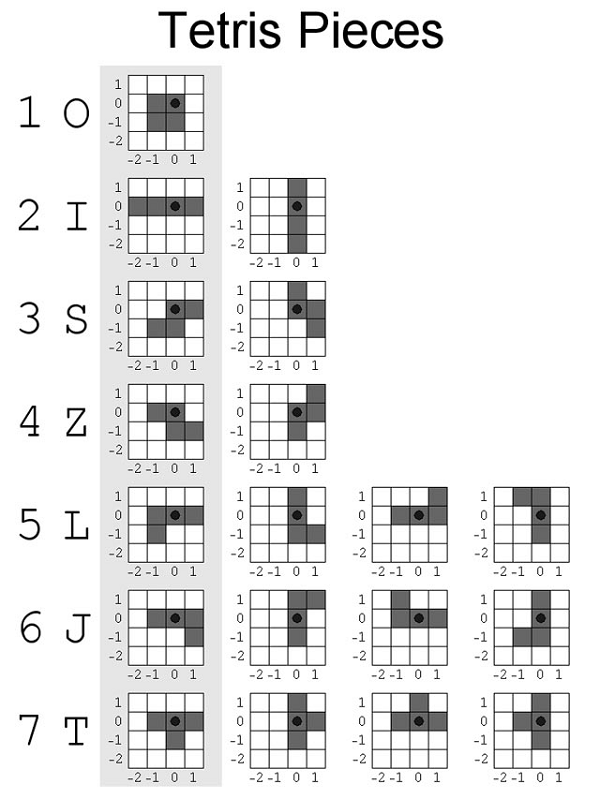
## Game-Specific Subsections

Puzzles could have a “Pieces” subsection, sidescrollers will probably have a “Level Design” one, space shooters may have “Enemies” and so on. As the title in bold above says, each game will have their own specific subsections, and since we can’t compose a subsection for all the possible ones that one GDD can have, we will provide you with the three bold subsections presented here as examples.

### Pieces

Suppose we have a puzzle game, where the player rotates different pieces in order to create a line of matching pieces to gain points. This would be a nice subsection to show some sketches of the many different types of pieces, as well as explaining their rotation pattern, stating their points value, and maybe describe their positioning placement. Pictures are welcome as always!

Example:



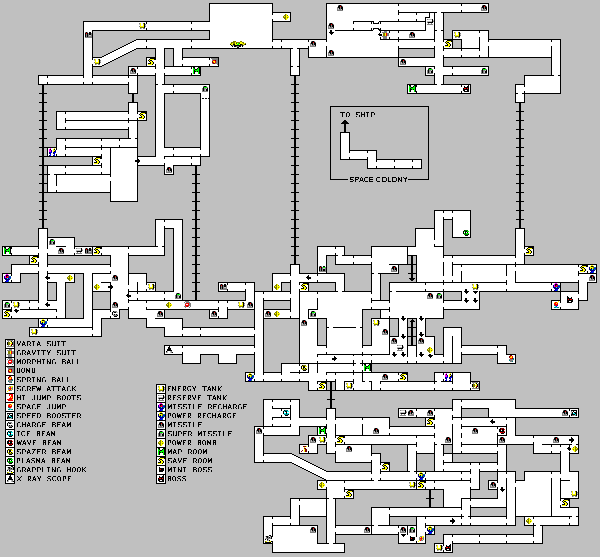
(Image from [Colin Fahey's Tetris article](http://www.colinfahey.com/tetris/tetris_en.html).)

### Level Design

Now let’s pretend we have a typical 2D platformer. One of the core elements of the game is the stages the player has to go through. It’s important that each stage feels unique so the player won’t feel like he’s just repeating the same thing over and over again. On the other hand, the player should still be familiar with the flow of the stage, i.e. if there’s always a checkpoint somewhere halfway through it, or some collectible items along the way.

What are the different types of enemies, terrains, doodads and power ups and do they allow the level designers to come with many different stages? You could present some beta stage diagrams to illustrate how will they be carried out.

Example:

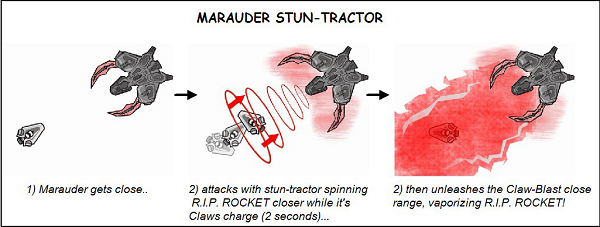


*(Map from Super Metroid; image from* [*jansenprice.com*](http://jansenprice.com/metroid/met3_sm/)*.)*

### Enemies

It’s very popular for space shooters to have many kinds of enemies, each one with different attacks and movement patterns, as well as different values for health, speed and targetable area. As such, it’s no surprise you would need an extra section to present all the game’s foes and their stats. Also, you could state some of their more obscure behaviour like shooting an extra beam when their health is low and so on.

Example:



*(Image from CrunchTime Games Inc's* [*Shred Nedbula document*](http://gamecareerguide.com/thesis/080903_ShredNebula_pitch.pdf)*.)*

## Plot

Many games are set in fictional worlds, each with their own geography, history and characters, in which the player will undoubtedly play a large role as the protagonist. If your game has a particularly interesting setting, it would be interesting to include a little insight on the game’s storyboard, describing the protagonist’s main events during his adventures and details about the lore.

## Characters

Lots of games aren’t made of enemies alone. There may be a protagonist and allies to help him overcome his foes. For example, even a tower-defense game without a controlled character can still have side-characters like a tutorial-NPC giving you tips on how to overcome certain challenges at the beginning of each stage. If you do have a protagonist that the player controls, then what’s he like? Does he have any abilities and powers? Keep in mind that this shouldn’t feel like a “How to Play” subsection.

## Artificial Intelligence

Any game will need a persisting world to handle all the player’s actions to the game and the other way around. That includes enemy movements, player controls, collision handling, time counting, random number generators and many other things one could need in a game. Although people not directly related to the programming may not understand this subsection entirely, they should at least grasp the basic of it. Most of all, keep the coding out of here and simply state the enemies' moving patterns, the chain puzzle piece falling algorithm, maybe illustrate the combat system with a flowchart and so on.

Example:

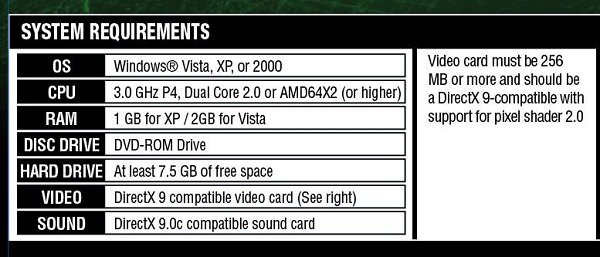
The characters on the board will escape the orbs using simple pathfinding / flocking algorithms. Every level will use up to three different script files to issue commands to the animated characters. (...) Player bots will be used to simulate real players. This will allow any level to be played even if there are more goals than players. The decision process that the A.I. system is trying to solve is this:

* Should I place a new block? If so:
* Where do I place the block?
* What type / material should the block be?

# Technical Aspects Section

The technical aspects consist of a series of game data, such as the system requirements on which it will play and the framework in which it was developed, the method or algorithm it was based on, and the maximum number of elements that can be rendered on screen. The graphical technical aspects consists of software used, modeling type, art style and others according to these topics.

The system requirements are the necessary computer settings for the game to be played, like the size it occupies on the computer’s HD and how much RAM is needed.



Another important technical aspect not to forget is the ESRB (Entertainment Software Rating Board) rating (or similar), already explained earlier. Some of the ratings are shown below.



# To Include or Not to Include? When? Why?

Technical aspects interest the companies that will distribute it or that will use the technology developed in the game, so always add something in it if you're showing this to someone that will approve or disapprove the game. There has to be some care when writing technical aspects. You can write something in the wrong subject. For example: limiting the platform and distribution game mode belongs to Marketing Aspects, not to Technical Aspects.

* Introduction
* Background (optional)
* Description
* Key features
* Genre
* Platform(s)
* Concept art (optional)