Game Concept & Design Document

**Version 1**

Dauntlet



Crystal Methods

Braun, Eric

Hart, Adam

Keller, Colton

Leatherman, Garrett

Losinski, Tyler

Submission: Monday, February 23

Game Concept & Design Document Template

**Document Contents**

[1 Template Introduction 1](#_Toc346192018)

[2 Concept Document 2](#_Toc346192019)

[2.1 Title Page 2](#_Toc346192020)

[2.2 Introduction 3](#_Toc346192021)

[2.3 Game Analysis 3](#_Toc346192022)

[2.4 Game Atmosphere 5](#_Toc346192023)

[2.5 Game Play 5](#_Toc346192024)

[2.6 Key Features 6](#_Toc346192025)

[2.7 Selling Features 6](#_Toc346192026)

[3 Design Document 7](#_Toc346192027)

[3.1 Design Guidelines 7](#_Toc346192028)

[3.2 Game Design Definitions 7](#_Toc346192029)

[3.3 Game Matrix 7](#_Toc346192030)

[3.4 Game Flow Chart 8](#_Toc346192031)

[3.5 Player Elements 8](#_Toc346192032)

[3.5.1 Player Definition 8](#_Toc346192033)

[3.5.2 Player Properties 8](#_Toc346192034)

[3.5.3 Player Rewards (Power-ups & Pick-ups) 9](#_Toc346192035)

[3.5.4 User Interface (UI) 9](#_Toc346192036)

[3.5.5 Heads up Display (HUD) 9](#_Toc346192037)

[3.5.6 Player View 9](#_Toc346192038)

[3.6 Antagonistic Elements 9](#_Toc346192039)

[3.6.1 Antagonistic Definitions 10](#_Toc346192040)

[3.6.2 Antagonistic Properties 10](#_Toc346192041)

[3.6.3 Antagonistic List 10](#_Toc346192042)

[3.6.4 Artificial Intelligence (AI) 10](#_Toc346192043)

[3.7 Global Game Elements 10](#_Toc346192044)

[3.8 The Story 10](#_Toc346192045)

[3.8.1 The Story Copy 10](#_Toc346192046)

[3.9 Concept Art 10](#_Toc346192047)

[3.10 Level Design 11](#_Toc346192048)

[3.10.1 Level Copy 11](#_Toc346192049)

[3.11 Audio & Sound F/X 11](#_Toc346192050)

[3.12 Game Architecture 11](#_Toc346192051)

[3.12.1 Game Architecture Overview 12](#_Toc346192052)

[3.12.2 Architecture Copy 12](#_Toc346192053)

[3.12.3 How to play Copy 12](#_Toc346192054)

[4 Technical Document 13](#_Toc346192055)

[4.1 System Requirements 13](#_Toc346192056)

[4.2 Visual Content 13](#_Toc346192057)

[4.3 Audio Content 14](#_Toc346192058)

[4.4 Programming Content 14](#_Toc346192059)

[4.5 Code Structure 14](#_Toc346192060)

[4.6 Concerns and Alternatives 15](#_Toc346192061)

[4.7 Resources 15](#_Toc346192062)

[4.8 Technical Matrix 15](#_Toc346192063)

[5 Appendix A 16](#_Toc346192064)

[5.1 Taxonomy 16](#_Toc346192065)

[5.2 Genre 16](#_Toc346192066)

[5.3 Game Elements 17](#_Toc346192067)

[5.4 Content 17](#_Toc346192068)

[5.5 Theme 17](#_Toc346192069)

[5.6 Style 17](#_Toc346192070)

[6 Appendix B 18](#_Toc346192071)

[7 Appendix C 1](#_Toc346192072)

[8 Appendix D 1](#_Toc346192073)

# Template Introduction

This document is based on the research done during an internship at Overloaded Pocket Media. The document it’s self has been updated several based on new experiences and insight into game design. In this version it is my intension to describe a broader range of game than just mobile games, as well as provide a document for fellow students and game designers to use.

This document has been created with the intention of making a clear game design document. To do this the document has been divided into 3 different documents, the game concept document, game mechanics document and game tech document. This is done with the intention of making a modular document that can be updated and split into different versions easily.

Documents created with this game design template will focus upon creation of a well devised game. What this template does not do is predict the playability of the game and define a plan for the overall game development. It is there for important to consider this is only one phase in the Game Development process. I recommend that any game design be tested with tangible means such as pencil and paper, cards, board, etc, before any major resources are allotted to it. It is also important to understand that this document will act as a starting point and can not replace hands on interactivity and game testing.

Much of the contents of the game design will be influence by factors such as the game being licensed, the type of game and the game designer himself. In the end this template is to act as a guide to concise documentation allowing the designer to give more attention to creativity.

# Concept Document

The concept document serves the purpose as a way to present a game concept.

A general overview of the game, with the idea anyone can read and understand what the game is like.

This part of the document is one that will change very little once the concept is accepted.

## Title Page

Game Name, Game Logo, Game Catch Phrase, Document Type, Document Version.

These need to be clear so that when used as a document everyone immediately recognizes it.

## 

## Introduction

Dauntlet is a top down rogue-like comedy thriller game. There is free movement, enemies, and progression in Dauntlet.

## Game Analysis

This is a general overview of the game.

|  |  |
| --- | --- |
| **Game Description** |  |
| Genre: | * The type of genre for our game will be a top down style game with rogue like game play. * Genre types*:*   + Top Down   + Rogue   + Beat-em up |
| Game Elements: | * Allow for a ghost to beat up enemies using his awesome gauntlet. * There will be progression using point you receive either at the end of each level or by killing enemies. * Enemies should drop some items that will help you better defeat them for a short period of time. * There should also be special moves such as:   + Spin attack   + Possibly a ranged attack   + Knockback |

|  |  |
| --- | --- |
| Game Content: | * Short story about ghost who finds a gauntlet * The game should be upbeat and will include some of the following aspects   + Humor   + Cartoon Graphics   + Funny story |
| Theme: | * The theme of the game can best be described as a comedy thriller. * Some of the follow points also describe the game’s theme   + Cartoon style   + Funny   + Fantasy |
| Style: | * The style of this game will be very cartoonish making the mood a little lighter.   + Cartoonish   + Happy suspense |
| Game Sequence: | * The game will have a rather shot linear story that can’t really be changed   + Linear- Storylines |
| Player: | * The game will only support one player |
| **Game Reference** |  |
| Game Taxonomy: | Dauntlet is first and foremost designed around fun gameplay mechanics and the story of the player’s character. As such, it is classified as a fictional game/narrative. Since Crystal Methods collectively came up with the story for Dauntlet, it is a fictional work. However, the narrative story of how the main character returns the discovered gauntlet is not the primary driver of the game; the gameplay is. Therefore, Dauntlet is classified first as a game and secondly as a narrative. |
| Player Immersion: | While playing Dauntlet, the player will experience tactical, narrative, and physical immersion. The tactical nature of the game comes from maneuvering around the rooms and fighting the enemies present. The player will have to choose how to defeat the enemies, between where to go in the room, which move to use, and what enemies to kill and what ones to avoid. All of this provides tactical entertainment and immersion. Also, as the player plays through the game, they will become invested in the story of the game, both in the explicit telling, and in the implicit creation that is each play through. The player will also experience physical immersion during gameplay. One of the mechanics of the game will be knockback from player attacks. This will give a sense of physical presence and power to the player, providing a deeper level of physical immersion to the player. |
| Reference: | * The rough gameplay concept came from the arcade game gauntlet, in which the player will traverse through levels while attacking enemies and finding the exit. * The inspiration for different attacks and knockback came from the 2d Legend of Zelda games, in which the player obtains new methods of attack as the game progresses and many items have a factor of knockback. * Luigi’s Mansion was the inspiration for the multiple rooms/levels in Dauntlet with differing themes throughout, as well as the tone of mild, cutesy spookiness we wish to convey in our game. |

|  |  |
| --- | --- |
| **Game Technical** |  |
| Technical From: | This will be a 2D, top-down game with sprite-based graphics. |
| View: | The view of the game will be top-down in a legend of Zelda style in which the camera will be fixated in each room and will transition to the next at the appropriate times. |
| Platform: | This game will be written in C# with Microsoft’s XNA development software suite in Visual Studio 2013. |
| Device: | This game will be designed for play on the PC. |
| **Game Sales** |  |
| Consumer Group: | This game will be geared towards people who enjoy indie games with a shorter but still fun experience. |
| Payment: | The payment for the game will depend on which platform it is released on (Steam, Origin, Independently, etc). Payment will most likely be via paypal or credit card. |
| Estimated Price: | The price of this game will be around $1-3. |
| Device Support List | This game will be supported on all machines that can run games written with XNA. If the computer supports DX10 and Shader model 2.0, it will support Dauntlet. |

## Game Atmosphere

Dauntlet is a colorful and upbeat game. There are some sound effects for enemies and the player. There are two types of characters in Dauntlet, player and enemies. Both characters have the ability to attack and take damage. Dauntlet will have multiple rooms that you may progress through by defeating enemies.

## Game Play

Opening the game, the player should be met with a simple title screen depicting the game’s logo. There will be a small prompt “Press any button to start!” that will immediately initiate the game. The game will have a single mode of play with no planned customizable options.

The game revolves around small cartoonish ghost character named Dante, who resides in a haunted mansion and acts as the game’s controllable protagonist. In a short opening presentation, he is seen perusing the courtyard of the mansion when he discovers a single humorously large gauntlet. Other ghouls and undead creatures approach Dante and threaten to take it. In self-defense, Dante equips the gauntlet and fights his way out. Now a rogue of the mansion, Dante must fight his way through the many rooms and corridors within to find out who is the true owner of this spectral weapon.

The game is played in a top-down free-roam perspective. The player controls Dante across the labyrinth of rooms within the mansion using WASD keyboard controls and the mouse on PC or an analog stick on Xbox 360. Along the way, hostile enemies like zombies, skeletons, and other spooky-scary things will appear in each room, and the player will have to engage them in combat using the Xbox ABXY buttons (PC controls are still undecided).

Dante will have four simple RPG-like stats, three of which define his ability in combat: Speed, which determines how quickly Dante can move across the map, Power, which determines the strength of his attacks, and Stamina, which determines how many hits he can take. Stamina will act as the player’s health and will decrease as Dante is attacked by enemies, and will result in a lose condition when Dante’s stamina falls to zero. Likewise, enemies have their own health that Dante must decrease to zero to defeat them. Enemy health is not shown to the player. As Dante defeats enemies, he will accrue experience points that can increase his overall Level, the fourth stat. Level itself has no direct effect on the gameplay, but with each level-up, the player will receive an arbitrary number of points that can be allocated to Dante’s combat stats. This allows the player to fine-tune and customize the play experience, as well as adds incentive to the objective of combat. As the player progresses further into the game, the enemies will become tougher to defeat and deal more damage to Dante, meaning the player must strategize Dante’s combat and work to keep his Level up to par.

The gauntlet will have a small variety of attack actions all based on the basic act of punching – a standard forward jab that can attack enemies within immediate striking distance, a long-range jab that can strike enemies in a single direction at a distance, and a spin attack that can attack all foes in a short 360-degree range at once. Dante will only have a single basic attack at the beginning of the game and acquire further abilities as the game progresses as permanent skills. There will also be temporary consumable item buffs that will appear throughout the mansion, such as items that restore lost stamina or provide defensive boosts that decrease damage taken from enemies.

The game will be a single open-roam world divided into individual fixed-perspective rooms that are represented by rooms in the mansion. Many of the paths in the mansion will be locked or otherwise inaccessible and must be unlocked through progression of the game. If Dante loses all of his stamina, the player will be returned to an arbitrary checkpoint prior to the “death” with little or no lost progress, up to an arbitrary number of “deaths”, after which the player will receive a Game Over and must start the game anew. The game is won when Dante makes it to the final room of the mansion and defeats the final boss character.

The “fun” in the game stems from the constant real-time combat with AI-controlled opponents, where the player must make decisions on how to survive the onslaught of foes while traversing the overworld looking for a way to progress. The levelling and stat allocation system makes each play of the game more distinct and allows the player a sense of agency and fine-tuned control over the combat system. The figure that ties this all together is the cartoonish fantasy that allows players to suspend their disbelief with the story and immerse themselves in the game.

## Key Features

* Multiple rooms to explore
* Real-time overhead combat
* Several enemies with varying behaviors
* Consumables and perks
* Rpg leveling to improve your player stats​

## Selling Features

* Focus on humor in any text within the game
* Cartoon style graphics
* Satisfying punch mechanics
* All custom sounds, music, and graphics

# Design Document

This document describes how game objects behave, controlled and properties they have.

This documentation is primarily concerned with the game it’s self.

This part of the document is meant to be modular. Meaning you could have several Game Mechanic documents attached to the Concept Document.

## Design Guidelines

This is an important statement about any creative restrictions that need to be regarded and a general over all goal of the design.

## Game Design Definitions

This is a section where the definition of the game play is established.

Definitions should include how a player wins, loses, passes levels and the main focus of the game play.

Issues that should be addressed here are:

* Menu
* Synopsis
* Game Play
* Player Control
* Game Over (Winning & Losing)

## Game Matrix

This is a spreadsheet containing the generic names of the player and antagonistic elements and their game properties.

This should allow an easy cross reference for an elements in the game that a value.

Consult with the programmer about the properties a game object.

*See Appendix B*

## Game Flow Chart

This is where a visual of how the different game elements and their properties interact.

Game Flow chart should represent Objects, Properties and Actions that are present in the game.

Flow chart objects, properties and actions should have a number reference to where they exsist with in the game mechanics document.

*See Appendix C*

## Player Elements

List all the elements that are directly related to or to the benefit of the player.

Devise two sets of names for player elements. One set is a generic name (or code) and the other is its game name.

Describe the terminology that you use to describe the player’s properties.

This is a good place to interact with a graphic designer to get graphics to match to game names.

Graphics that will be seen during game play should be exhibited here.

Multi-player issues should also be mentioned here.

### Player Definition

Make quick descriptions that define the player.

What are the default settings for the player at the beginning of the game or level.

A suggested list of player definitions:

* Actions: What can the player do?
* Information (Status): What information about the game is available for the player?
* Default Properties: How does the player begin the game?
* Winning: How can the player win?
* Loosing: How does the player lose?

### Player Properties

Make a list that defines the properties that a player has.

Player properties can be affected by player’s action or interaction with other game elements.

Define the properties and how they affect the player’s current game.

A suggested list of player definitions:

* Health
* Weapons
* Actions
* Etc.

Each property should mention a feedback as a result of the property changing

### Player Rewards (Power-ups & Pick-ups)

Make a list of all objects that affect the player in a positive way. (i.e. health replenished)

Define these objects by describing what affect they cause and how the player can use the object.

### User Interface (UI)

This is where a description of the user’s control of the game can be placed.

It is also recommended to think about which buttons on a device would be best suited for the game.

Consider what the worst layout is, then ask you self if your UI is it still playable?

A visual representation can be added, where we relate the physical controls to the actions in the game.

When designing the UI, make use of the expertise of someone from quality control.

### Heads up Display (HUD)

This is where a description of any graphics that will represent information during game play should be described.

A visual representation (mock-up screenshot) here would be useful.

This is a good place to interact with a graphic designer.

### Player View

A screen shot is very necessary here.

A definition of how the camera moves for the player.

A mock-up of an overview of the level relative to the screen size will help create a perspective of a levels size compared to what is actually seen.

## Antagonistic Elements

This is where a list of antagonistic (enemies, opponent) objects should be listed with graphics and written description.

Describe the terminology that you used to describe antagonistic properties.

Devise two sets of names for player elements. One set is a generic name (or code) and the other is its game name.

This is a good place to interact with a graphic designer to get graphics to match to game names.

### Antagonistic Definitions

This where a description of what makes a antagonistic element.

### Antagonistic Properties

This is a list of properties that antagonistic elements have in common.

### Antagonistic List

This is where a list of all the antagonistic elements goes.

### Artificial Intelligence (AI)

This is where visuals and written description(s) of the antagonistic element’s behaviors.

These should be labeled in such a way that they can be used in level design with out having to describe them again.

Devise generic names for repetitive behaviors.

This is how an AI action could be broken down:

* Normal State: What is the object doing if it has not come in contact with the player?
* Detection State: What does it take for this object to detect the player?
* Reaction State: What does the object do as an action after passing the reaction state?
* End State: What happens to the object after player has reacted correctly or incorrectly to object?

## Global Game Elements

In this section it is important to describe the boundaries, neutral objects, camera views and scale of the world.

Neutral game world objects can be things like a static background, objects that do not interact with the player or antagonistic elements.

## The Story

This is where the story can be described in detail.

A story board can be used to tie in graphics to the text.

This can later be used for splash screen concepts.

### The Story Copy

A shorter version of the story (The in game version) should also be written here.

This is where the script for in game characters or story information during the cut scenes would be placed.

This category does not always pertain to the current Game Design.

## Concept Art

Sketches that are used for the concept can go into this section as visual reference.

In the case of a brand, certain creative restrictions should be noted here.

This is a good place to interact with a graphic designer to get graphics to match to game names.

## Level Design

This is where information pertaining to level design and visuals of the level design goes.

Level design can best be shown as a flow chart.

Use generic names to create level design.

### Level Copy

This is where the script for in game characters or story information during the cut scenes would be placed.

## Audio & Sound F/X

This is where game ambient and Sound F/X should be listed with generic names then described.

This section also includes deciding if you will use a device’s vibration ring mode.

## [Game Architecture](file:///C:\Documents%20and%20Settings\Giel%20Claessens\Local%20Settings\Temporary%20Internet%20Files\Current%20Game%20Ideas%20&%20Designs\Game%20Doc%20Template\options.htm)

This is best done by a flow chart to represent the overall game.

Number each screen.

* Title Screen
* Option Screens
* Game Modes
* End Screens

*See Appendix D*

### Game Architecture Overview

The splash screens or video clip need to be in accordance to game story and style.

If cut scenes use video then story boards should be created.

This will need to be created with the graphic designer.

Menu should be designed with the most important options easily accessible.

Be aware how many clicks it takes to accomplish a task.

It would be wise to get together with someone from quality control and a programmer.

The Game Instructions should be written so that the player understands how to play the game.

Mock-ups should be made so that the programmers get the correct layout of the menu.

Mention and describe high score screen here as well.

### Architecture Copy

All text fro the game can be complied here.

Review Game Architecture Overview

### How to play Copy

This will be game copy.

Information for the player, clearly describing how to play the game.

# Technical Document

The information concerning the technical aspects of the game should be placed here.

The technical document is best achieved with consensus from the people responsible for the Visual, Programming and Audio aspects.

This part of the document is meant to be modular. Meaning you could have several Game Technical documents attached to the Concept Document and Game Mechanic Document.

## System Requirements

This is a list of system requirements that a device will have to meet to run the game.

This also represents the restrictions that may apply to the end product.

## Visual Content

A list of technical requirements from those in concerned with the visual aspects of the game.

This is a section that will require extensive meetings with a graphic designer.

All objects should be listed with their generic names.

* General
  + File Size Restrictions
  + File Format Type
  + File Quality Type
  + Visual Scale
* Player Elements
  + Type of States (Default, Damage, Destroyed, ect.)
    - Amount Animation Frames
* Heads Up Display (HUD)
  + Type Icons
  + States
  + Font Type
* Antagonistic Elements
  + Type of States (Default, Damage, Destroyed, ect.)
  + Amount Animation Frames
* Global Elements
  + Background/Texture/Tiles
  + Font Type

## Audio Content

It is very important to communicate with the audio designer before and while the audio content is being developed.

* General
  + File Size Restrictions
  + File Format Type
  + File Quality Type
* Player Elements
  + Type of Sound f/x
  + Device Vibration
* Antagonistic Elements
  + Type of Sound f/x
  + Device Vibration
* Global Elements
  + Ambient Music
* Splash Screens
  + Ambient Music
* Menus
  + Type of Sound f/x

## Programming Content

Contents should be collaboration with the programmer.

The object here is to try to organize and modulate as much as possible.

* General
  + Requirements
  + File Size Restrictions
  + File Format Type
  + Specify Coding Conventions
  + Language/Device Restrictions
  + Screen Type (Small, Medium, Large)
* Player Elements
  + Type of Event
* Antagonistic Elements
  + Type Event
* Global Elements
  + Type of Event
* Splash Screens
  + Type of Event
* Menus
  + Type of Event
  + Type of Options

## Code Structure

This is where an overview of how objects/functions/data interact, a list of what specified functions/routines do and a list of what order modules will be written.

## Concerns and Alternatives

If there are concerns about something technical they should be stated here and what will be alternatives to the concern.

## Resources

A list of applications and equipment, that is acceptable for use on development of this game.

This is mostly a legal issue that development members must be aware of.

## Technical Matrix

The Matrix will be split into the different device series for each content category.

Technical Matrix includes the content lists of Audio, Visual and Programming.

# Appendix A

In this appendix there are lists for a games Taxonomy, Genre, Elements, Contents, Theme and Style. These lists are far from complete but are useful for further examples.

## Taxonomy

* Simulation
  + Story
  + Play
  + Chance
    - Fiction
    - Non Fiction
* Story
  + Simulation
  + Play
  + Chance
    - Fiction
    - Non Fiction
* Play
  + Story
  + Chance
  + Simulation
    - Fiction
    - Non Fiction

## Genre

* Basic
  + Adventure
  + Arcade (any "twitch" element)
  + Construction & Management
  + Puzzle
  + Role-play
  + Simulator
  + Strategy
* More Specific
  + Abstract (i.e., Arkanoid)
  + Adventure
  + Arcade (very generalized category)
  + Beat-em-up
  + Flight Sim/Space Sim
  + FPS (first-person shooter)
  + MMOG(Massive Multi-player Online)
    - MMORPG (role playing game)
    - MMORTS (real time strategy)
    - MMOTBS (turn base strategy)
  + Platform
  + Puzzle
  + Racing
  + RPG (role-playing game)
  + RTS (real-time strategy)
  + Shoot-em-up (scrolling shooter)
  + TBS (turn based strategy)
  + Trading

## Game Elements

* Alignment
* Catch
* Chase
* Collecting
* Combat
* Cooperation
* Dodging
* Escape
* Fighting
* Forbidden Act
* Hiding
* Jump
* Luck
* Maze
* Nursing
* Obstacles
* Puzzle
* React
* Resource Management
* Seeking
* Shooting
* Story Comprehension
* Target
* Trading
* Trivia
* Timing

## Content

* Action
* Drama
* Erotic
* Horror
* Humor
* Pure Play
* Realism
* Thriller

## Theme

* Abstract
* Crime
* Fantasy
* Noir
* Porn
* Sci-Fi
* Spy
* War
* Western

## Style

* Abstract
* Cartoon
* Manga
* Old School
* Realism

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Appendix B  |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Object** | **Properties** | |  |  | | *Unit type* | *Hit Points* | *Damage* | *Bullet speed* | *Movement* | | *Basic Enemies* |  |  |  |  | | Cannon -01 | 30 | 100 | Medium | NA | | Cannon -02 | 60 | 150 | Slow | NA | | Static rocket launcher | 70 | 700 | Medium | NA | | Flock 1 | 15 | 500 | NA | Medium | | Flock 2 | 15 | 35 | Slow | Medium | | Medium 1 | 80 | 350 | Fast | Slow | | Medium 2 | 90 | 400 | Medium | Slow | | Rail | 120 | 600 | NA | Slow | | Small 1 | 25 | 200 | Fast | Fast | | Small 2 | 30 | 210 | Fast | Fast | |  |

|  |
| --- |
| Appendix C image004 image003 |

# Appendix D

