**Note:** Sections highlighted in Red are options for this assignment!

# Technical Design Document (TDD) (Template)

## **COVER**

(Insert evocative cover image here)

## **YOUR GAME’S TITLE**

Document version number (keep this current!)

Written by (your name/team name here)

Point of contact (producer or lead designer with contact info.)

Date of publishing

Version number (This is the software version number of the game).

Footer should always have:

Copyright © Team name Date Page Number Current Date

**TDD Outline**

**Table of contents**—Remember to keep this current.

**Development Requirements** — An annotated list of key software versions developed during the production process.

* **Development** — IDE, test suites, editors etc.
* **Game Engine**— The selected engine for the game including version and tools.
* or **API Versions** — A list of APIs and version numbers, used in the game.
* **2D/3D Software** — A list of APIs and version numbers, used in the game.
* **Project Management** — Project management tools used.
* **Source Control** — Details of the server and client software used.
* **Sound Software** — Sound recording and editing software.

**Asset Specifications** — Details of supported asset formats and restrictions, for example model file formats, vertex winding, polygon counts or sprite file formats, width and height. Be sure to consider all required assets as appropriate to your game.

**Project Structure** —Details of the directory structure.

**File Naming Convention** —Convention for naming code and asset files.

**Level / World Details** — Layout and asset list including appropriate size limitations.

**Development Plan** —Outline of the production plan from pre-production to delivery.

|  |  |  |  |
| --- | --- | --- | --- |
| Milestones | Date | Deliverable | Approval |
| Pre-Production End |  | TDD  Prototypes |  |
| Milestone 1 |  |  |  |
| Milestone n |  |  |  |
| Alpha |  |  |  |
| Beta |  |  |  |
| Final |  |  |  |
| Pitch and Play |  |  |  |

|  |
| --- |
| **Terminology:**  The terminology here is based on that used by Douglass in [Doing Hard Time](https://www.amazon.co.uk/Doing-Hard-Time-Developing-Addison-Wesley/dp/0321774930). Architectural design refers to the structural relationships between classes represented using Class and Package diagrams in UML. Mechanistic design refers to the interaction of runtime entities in performing particular behaviours represented using sequence diagrams. Finally detailed design is reserved for complex sequences of actions (Activity Diagrams) or state changes (State Chart Diagrams) performed by individual objects. |

**Coding Conventions** — conventions for headers, variable and method names etc. these will be specific to the language being used for the game.

**Testing Plan** — Details of testing including unit, integration and system tests and results from play-testing undertaken.

* **Automated Tests** — Details of automated test frameworks, mock frameworks etc. and their coverage.
* **Performance Tests** - Tests on the rendering, memory usage, reliability etc.
* **Playtesting** — Overview of the kinds of tests, e.g. player control, combat, puzzle etc. and individuals responsible.
* **Testing Schedule** — Schedule of testing to be undertaken at each milestone, may include both unit tests suites to be complete and early playtesting of mechanics / levels.

# Examples

There are few examples of TDDs online, possibly as these are often included in a single document (bible,) including both GDD and TDD.

1. [Unholy Roses](https://hell-and-back.fandom.com/wiki/Technical_Design_Document)
2. [Unity Tutorial Example](http://game-lab.alliance-artem.fr/wp-content/uploads/2021/01/Zombie-Toys-Technical-Design-Document.pdf)