

# Technical Design Document

---

## Contents

Game Details.....	2
Team Members.....	2
Game Concept .....	2
Technical Goals .....	3
Technical Goal 1 – Enter descriptive name.....	3
Technical Goal 2 – Enter descriptive name.....	3
Technical Goal 3 – Enter descriptive name.....	3
Technical Risks .....	3
Technical Risk 1 – Enter descriptive name.....	3
Technical Risk 2 – Enter descriptive name.....	3
Features/Mechanics/Tasks .....	4
Deliverables .....	4
System Requirements.....	4
Target Device 1 - Enter target platform/device name.....	4
Target Device 2 - Enter target platform/device name.....	5
Third Party Tools.....	5
File Formats .....	5
Coding Conventions.....	5
Source Control .....	5
Game Flow .....	6
Game Objects and Scripts.....	7
Gameplay Systems.....	9
Gameplay System 1 – Enter descriptive name .....	9
Gameplay System 2 – Enter descriptive name .....	9
Input Method(s).....	9
User Interface .....	10

## Game Details

Enter the game & team name, and any other overarching details for the project

- **Game Name:** *Enter-Game-Name*
- **Team Name:** *Enter-Team-Name*

## Team Members

List of technical team members and broad overview of their roles.

Name	Job Title	Responsibilities/Roles
<i>Enter each team member</i>		
e.g. Ian Roberts	Systems Programmer	AI Serialisation System Player Controller
	Gameplay Programmer	
	UI Programmer	

## Game Concept

Describe the game concept here in 2-3 sentences. Focus on what the player can DO.

## Technical Goals

*What are the technical aspects of your game that your team aim to deliver? E.g. Challenging AI, Procedural Generated Levels, Interesting Jetpack mechanics.*

*Who will work on these goals?*

### Technical Goal 1 – Enter descriptive name

**Who's Responsible:** *Enter team member*

**Description:** *Enter tech goal*

### Technical Goal 2 – Enter descriptive name

**Who's Responsible:** *Enter team member*

**Description:** *Enter tech goal*

### Technical Goal 3 – Enter descriptive name

**Who's Responsible:** *Enter team member*

**Description:** *Enter tech goal*

## Technical Risks

*What are the technical (i.e. related to programming) features and ideas most likely to cause problems? E.g A mechanic requires learning new design patterns and 3<sup>rd</sup> party libraries you've never used before. Or none of the programmers have experience with AI.*

*What can you do to reduce the risk? E.g Bob will perform additional research and will spend the first two days making a test project to prove this idea is possible. If not, we will cut the idea.*

### Technical Risk 1 – Enter descriptive name

**What's the risk about:** *Enter information about technical risk*

**How will risk be mitigated:** *Enter information about risk mitigation*

### Technical Risk 2 – Enter descriptive name

**What's the risk about:** *Enter information about technical risk*

**How will risk be mitigated:** *Enter information about risk mitigation*

## Features/Mechanics/Tasks

A list of exactly what systems exist in your game and who is responsible, including scheduled dates for completion.

Feature/Mechanic	Who's responsible	Scheduled Date
<i>Enter each feature/mechanic</i>		
e.g. Player walking	Max	Alpha, 20/11/2019
e.g. Player jumping	Max	Alpha, 22/11/2019
e.g. Loading/Saving game data	Phil	Alpha, 25/11/2019
e.g. Platforms randomly generated	Phil	Alpha, 27/11/2019

## Deliverables

What will you deliver at the end of production?

Deliverable	Who's Responsible	Who's the Owner
<i>Enter each platform/input</i>		
e.g. Executable for Android	Terry	Client
e.g. Completed Admin Forms	Phil	Manager
e.g. PC Installer	Margaret	Team Leader

## System Requirements

What devices is your game targeting? What's the recommended hardware? Portrait or Landscape mode on mobile?

### Target Device 1 - Enter target platform/device name

**Recommended Hardware:** *What's the recommended/minimum hardware expected to run the game*

**Platform Specific Requirements:** *Explain any platform specific considerations (e.g. mobile/console/vr requirements)*

## Target Device 2 - Enter target platform/device name

**Recommended Hardware:** *What's the recommended/minimum hardware expected to run the game*

**Platform Specific Requirements:** *Explain any platform specific considerations (e.g. mobile/console/vr requirements)*

## Third Party Tools

*What third-party tools are you using? List Unity with version number and any other tools you might need. Include any asset packs you plan to use from the asset store.*

## File Formats

*What file formats will be used for models, textures, sounds and other assets.*

*If your project will be using a custom file format, explain in detail the format (eg. Text/Binary, order and size of each element in the file, file/byte offsets (if necessary))*

## Coding Conventions

*What coding conventions will your team use? Everyone on the team should use the same conventions.*

*Use an existing guideline as a template, rather than create one from scratch. One example can be found at <https://csharpcodingguidelines.com> which also includes a Visual Studio plugin to automatically analyse your codebase (The C# Guidelines Analyzer).*

## Source Control

*Which source control will be used? What rules should all team members adhere to when using source control?*

**Source Control Repository:** *Git/Perforce/SVN/CVS/Other*

**Source Control Client Tools:** *SourceTree/Visual Studio (Team Explorer)/TortoiseGit/Other*

**Source Control Remote Repo URL:** *This is your GitHub/BitBucket/Other URL used to clone the repo*

**Ignore/Config file:** *Specify the file format for .gitignore .gitconfig files, or link to a URL of a reference .gitignore to use*

**Commit message formats:** *Include any specific information each team member must include in their commit messages*

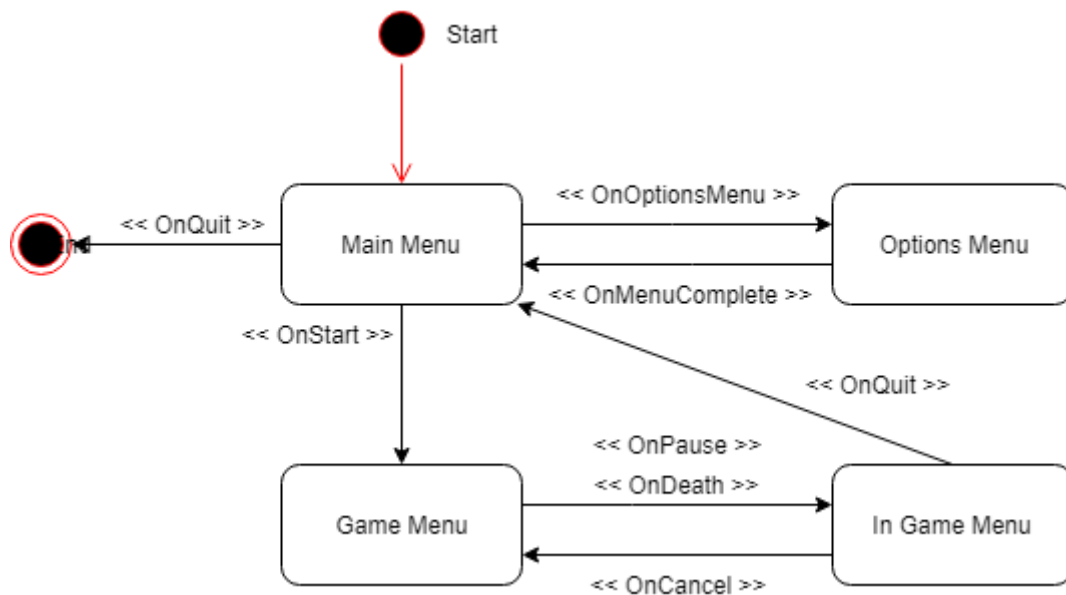
**Other repo notes:** *Include any other considerations eg. Only committing certain folders? How/When Branches are used? Will Pull Requests be used & how? Etc.*

# Game Flow

List the scenes in the game, and a short description of what the scene is responsible for.

Scene	Who's responsible	What is does
<i>Enter each game menu</i>		
e.g. Main Menu	Jenny	Starting menu, takes players to other menus, starts a new game, or exits
e.g. Options Menu	Karen	Main options menu for changing audio/graphics settings
e.g Main Game Level	Phil	Load the main game level and manages the main gamestate. Can be interrupted by the player selecting the "Pause" option to open the In-Game menu

Include State Diagrams to describe menus & basic game flow (example below). Use online tools like <https://www.draw.io>



# Game Objects and Scripts

Provide **necessary technical details** for any technical team member to implement at least the skeleton for each system in your game, including **classes/functions/properties/interfaces/inheritance/relationships**. This can be done with:

- One or more UML Class and/or Collaboration diagrams describing the static architecture and
- One or more Sequence and/or Activity diagrams describing specific parts of the run-time behaviour. These are often in response to a particular use-case
- Supporting text to help explain the diagrams, when and if necessary

For example, include class diagrams for each Game object that requires a custom script, and include information about what it does, what data types it manages, and other notes a programmer might need to write that code

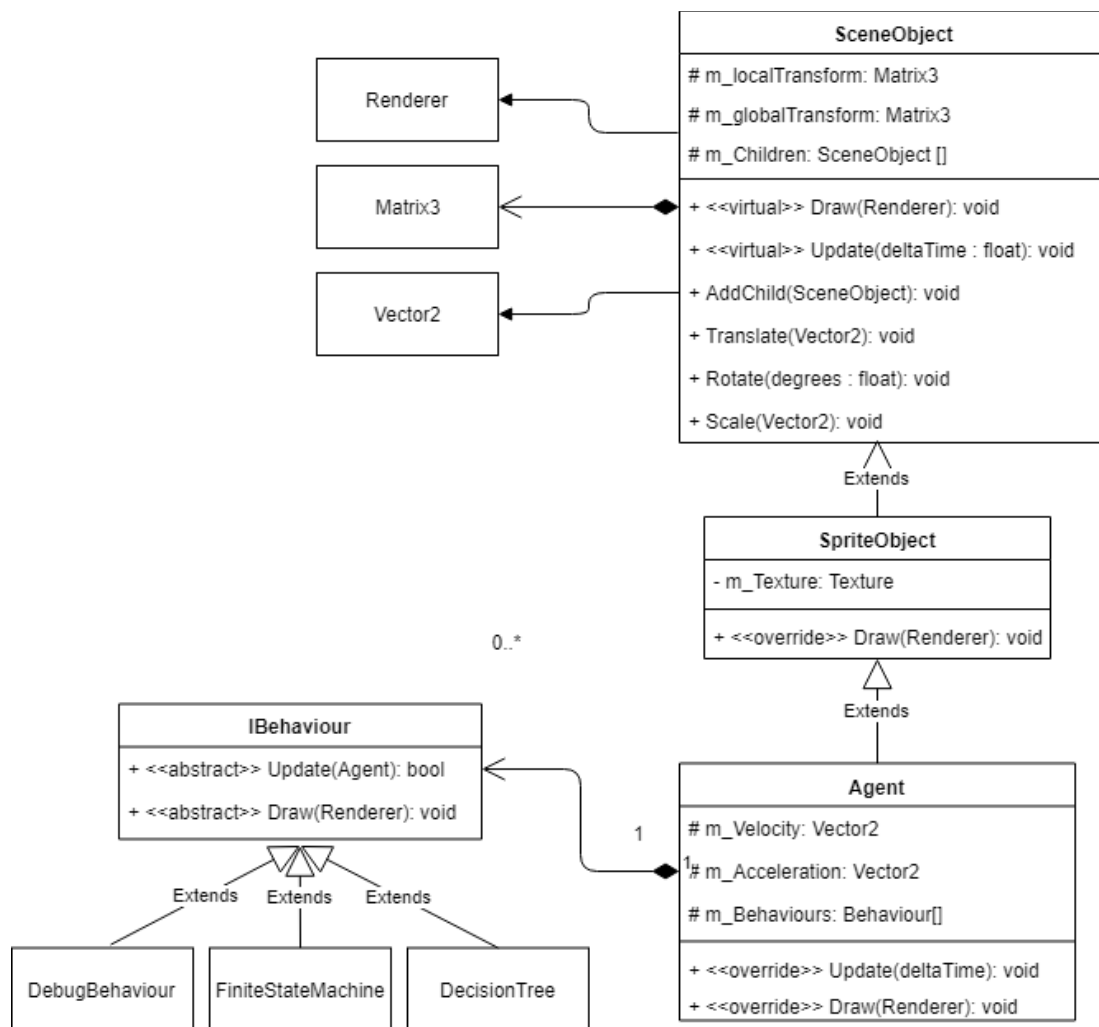


Figure 1 Example UML Class Diagram

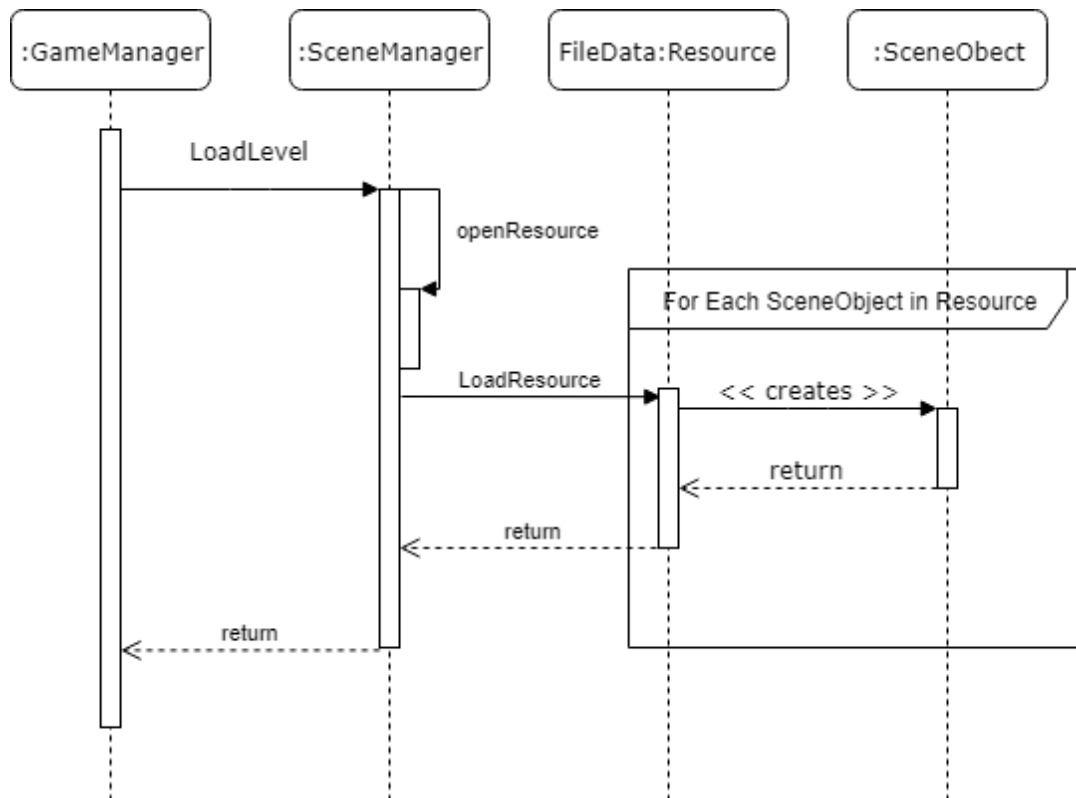


Figure 2 Example UML Sequence Diagram



## Gameplay Systems

Describe in more detail your individual gameplay systems, including how they interact with other systems in your game, what classes are responsible, and using UML diagrams where appropriate. Eg. If you are developing a random generation system, then describe how it works! If you're developing some unique AI or algorithm for a Goblin Sorcerer, then describe the details of how that will be implemented.

### Gameplay System 1 – Enter descriptive name

**Who's Responsible:** Enter team member

**Description:** Enter description of gameplay system

**Diagrams:** Include UML diagrams, if possible

### Gameplay System 2 – Enter descriptive name

**Who's Responsible:** Enter team member

**Description:** Enter description of gameplay system

**Diagrams:** Include UML diagrams, if possible

## Input Method(s)

Describe the Input method for each target platform (e.g PC / VR / Console).

Target Platform	Input System	Who is responsible
<i>Enter each platform/input</i>		
e.g. PC	Mouse/Keyboard	Fred
e.g. PC	Xbox360 Controller	Trevor
e.g. PS4	Playstation 4 Controller	Terry

# User Interface

Add user-interface design mockup. Display any differences between PC and Mobile interfaces.

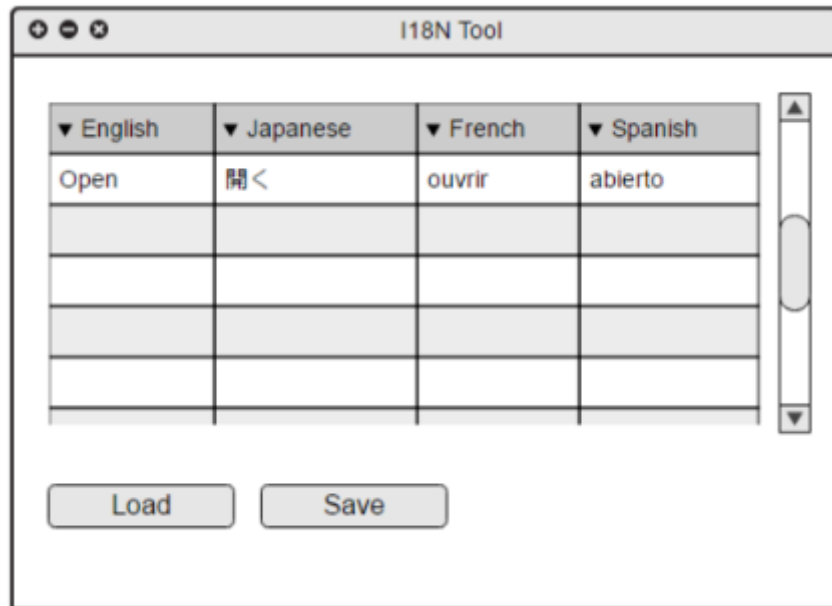


Figure 3 Example Mock-up Main Window

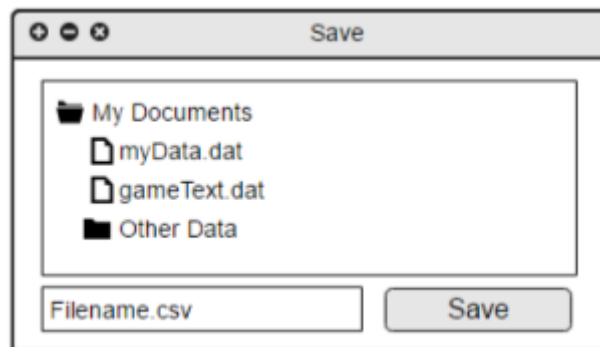


Figure 4 Example Mock-up Save Dialogue

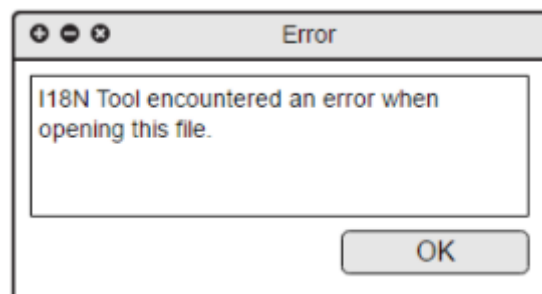


Figure 5 Example Mock-up Error Dialogue

