Tongji - Development

**Technical Design Document**

Table of Contents

[1. Overview 3](#_Toc260387709)

[2. Requirements 3](#_Toc260387710)

[2.1. Brief 3](#_Toc260387711)

[2.2. Reference 3](#_Toc260387712)

[3. Dependencies 3](#_Toc260387713)

[4. Existing Technology 4](#_Toc260387714)

[4.1. Features 4](#_Toc260387715)

[4.2. Reference 4](#_Toc260387716)

[5. Implementation details 4](#_Toc260387717)

[6. Proof 5](#_Toc260387718)

[7. Issues 5](#_Toc260387719)

[8. Risks 6](#_Toc260387720)

[9. Estimates 6](#_Toc260387721)

# Overview

This is a video game running on PC under Windows. The game is a 3D version of the classical “Bomberman” video game.

* There are not just a kind of AI.
* Player can edit the new level or rewrite the old level in this new version game. Because the level editor is not just made for developer, the player also can use it to edit their maps.

# Requirements

## Brief

1. Level Map editor Part:
   1. It is a 3D version editor. Player can use W-S-A-D keys to control the Camera rotate around the map.
   2. Player can choose the map file which will be rewrote, or just save current map as a new map file.
   3. When the game is start, load the correct map file. And create the 3D map.
   4. The map data include all the elements except the environmental.
2. AI Part:

## Reference

# Dependencies



# Existing Technology

## Features

现有技术的使用

<Feature>

* <Feature bullet points, what it is in this feature we’ll be using>

<Feature 2>

* <Feature bullet points, what it is in this feature we’ll be using>

unitypackage - NGUI

UI

* NGUI is a very mature unitypackage for UI.
* It accord with KISS (Keep it Simple and Stupid) principle.
* In UI part, NGUI is really useful.

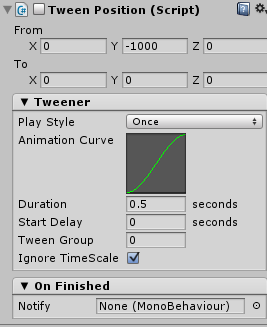
## Reference

* <bullet points of technique references, for example related technique document>

# Implementation details

UI Part:

* + 1. Tween part: Tween Position, Tween Rotation, Tween Scale;(NGUI)



* + 1. Unity function OnGUI:

void OnGUI()

{

if (showWindow)

{

GUI.Window(0, new Rect(110, 10, 200, 60), DoWindow, "Basic Window");

}

}

Map Editor Part:

1. Create the map data.

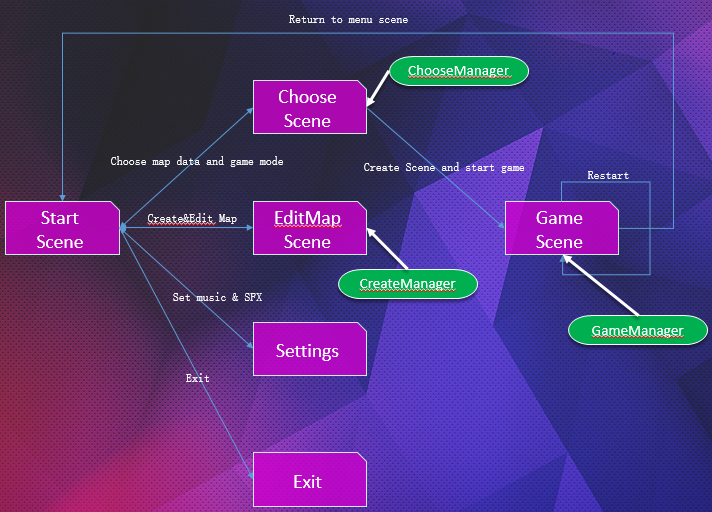
Create Manager Script:

1. Save the map data.

Save the map data as “ xxx .map ”, in the “ Setting\Levels ” folder.

1. Load the map data and build the 3D map by the map data.

Before the game scene, player can choose the level number, actually the level number is the level map name. Example: level 5, the map file name is “ 5.map ”.



# Proof

* The map editor save map file correctly as “ xxx .map ”: it can be found in the folder “ Setting/Levels ”
* Before the game scene, there is a level scene. Player can choose level. If the map created correctly, the map data works.

# Issues

* Map editor Part:
* AI part:

# Risks

* In the map editor, player may cannot use it well if they haven’t read the tips about how to use this editor.
* In the map editor, player may rewrite a great map by a not good map. And it cannot be recovered.
* In the map editor, player could not fellow the tips. And may made the map load part become wrong.

# Estimates

|  |  |
| --- | --- |
| Tasks | Estimate in days/person |
|  |  |
|  |  |
|  |  |
| Total |  |