Specification

EditKong Team

March 14, 2014

Abstract

This project is a 2D map editor, which allow everyone to create his own map. To start playing with the new map, we will provide a game example. Of course anyone can also develop a game to be able to make the generated map playable.

Description

Title	Specification « EditKong »
Created at	March 9, 2014
Last update	March 14, 2014
Author	EditKong Team
Manager	Clément
Version	1.0
Member	Clément - Alexandre - Jordan - Adrien
Student Number	13129091 - 13129085 - 13129038 - 13129054

Reviews

Date	Author	Version	Section(s)	Comment
03/09	Jordan Kergoat	0.3	Specification	
03/10	Adrien Mille	0.5	Component	
03/12	Alexandre Cloquet	0.8	Interaction	
03/12	Clément Lebossé	0.9	Specification	MileStone
03/14	Work group	1.0	Interaction	

Contents

List of Figures

Chapter 1

Specifications

1.1 People definition

This project goal is to target community of gamer, to provide a 2D game with a long life time thanks to the map editor. By making custom maps, people will be able to play as long as their imagination let them create maps.

To make sure that the created maps are fun to play, we want that the solution be used by people loving 2D games, so they have enough experience to make nice custom maps.

1.2 Pain points

A recurent problem in editing software is difficulty to use it. So for our software we will make a major point to develop a software easy to use. We want our software usable at the first sight.

Gamers have no time to practice or learn for developpment skills, that's why we want that the editor can be use without seeing programming language, only use with graphic interface in a drag and drog software type.

1.3 Scenarios

1.3.1 Using EditKong from scratch

- Create a new map
  - Define map name
  - Define map size
- Draw your map
  - Drag and drop blocks
  - Insert logic in blocks
- Place your objective
  - Define the start point
  - Define the end point
  - Define the target point
- Save you map
- Start playing

1.3.2 Using EditKong from existing map

- Open existing map
- Apply any modification
- Save you map
- Start playing again

1.4 Goals / Non Goals

Goals	Non Goals
Map editor for 2D platform	Networking feature
Easy to use	Background story creation

1.5 Priority

For this project, we are going to start by making the editor, and we will finish by making the game example. Indeed, the game is an example, not the finality.

1.6 MileStone

Map editor:

Create new map / Load existing map	03/09
Manage block type	03/16
Drag and drop blocks	03/23
Writing JSON file	03/30

Starting Beta

Game example:

JSON reading	04/06
Displaying map	04/13
Game mecanic implementation	04/20
Play	04/27

Chapter 2

Interactions

2.1 Introdution

Here is a list of all interactions that user can have with the map editor :

Interaction	User	System
Click for new map	X	
Click for load map	X	
Click for save map	X	
Create new block type	X	
Edit block type	X	
Delete block from map	X	
Put block in map	X	
Move block from map	X	
Write Json file		X
Display map		X
Display tools box		X

2.2 Detailed interactions

2.2.1 User : Create new map

- Inputs:
  - Click on creation button
  - Fill map name in dialog box
  - Fill map size in dialog box
- Outputs:

  - If success:
    * Diplaying the new map in the editor
  - If fail:
    * Displaying error messages in alert box
      · Wrong size error
      · Empty name error
      · Invalid name map error

2.2.2 User : Load map

- Inputs:
  - Click on load button
  - Choose the map to open
- Outputs:
  - If success:
    * Diplaying the map in the editor
  - If fail:
    * Displaying error messages in alert box
      · Wrong type of file error
      · Wrong Json format error

2.2.3 User : Save map

- Inputs:
  - Click on save button
- Outputs:
  - If success:
    * Diplaying the success message in alert box
  - If fail:
    * Displaying error messages in alert box
      · File access error

2.2.4 User : Create new block type

- Inputs:
  - Click on '+' button in the block type dock
  - Fill block name in dialog box
  - Fill sprite in browser box
  - Customize your block actions using checkbox
- Outputs:
  - If success:
    - Diplaying the new block in the block dock
  - If fail:
    - Displaying error messages in alert box
      - Wrong sprite error
      - Conflicting action error
      - Block name empty error

2.2.5 User : Edit block type

- Inputs:
  - Select a block and click edit button
  - Edit the block
  - Click save button
- Outputs:
  - If success:
    - Diplaying the modified block in the block dock
  - If fail:
    - Displaying error messages in alert box
      - Wrong sprite error
      - Conflicting action error
      - Block name empty error

2.2.6 User : Delete block from map

- Inputs:
  - Click on the delete button mode
  - Confirm your choice by clicking Yes button
  - Click on the block to delete
  - Click back on the delete button mode to exit delete mode
- Outputs:
  - If success:
    - Delete the block from the map
  - If fail:
    - No failure case

2.2.7 User : Put block in map

- Inputs:
  - Select a block from the block dock
  - Click on the wanted cell in the map
- Outputs:
  - If success:
    - Diplaying the block in the map
  - If fail:
    - Displaying error messages in alert box
      - Cell already taken error

2.2.8 User : Move block from map

- Inputs:
  - Right click on the wanted cell in the map
  - Choose "Move" in the option
  - Click on the wanted cell in the map to move
- Outputs:

- If success:
    - Block display in the wanted cell
- If fail:
    - Displaying error messages in alert box
        - Cell already taken error

Chapter 3

Components

Here is the general component diagram, later in the document each component description will be detailed.

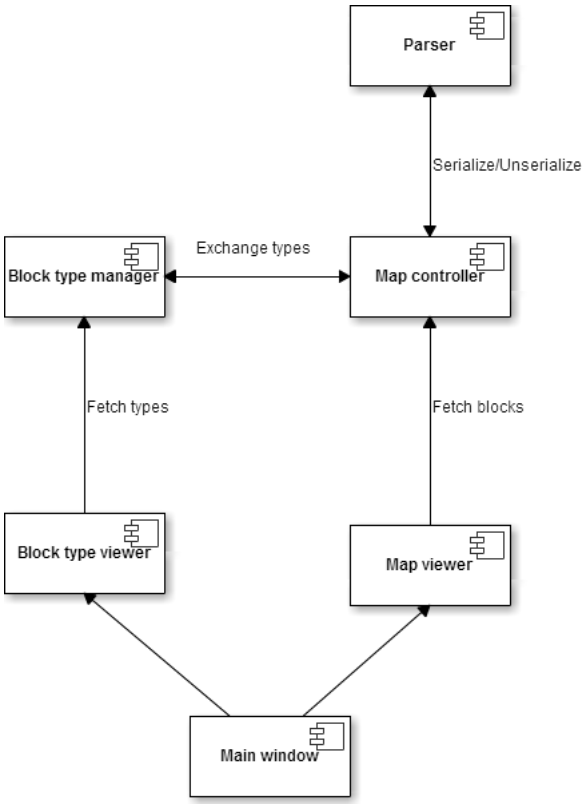

Figure 3.1: Component diagram

3.1 Parser

The parser has two gaols, first load a map from JSON file, then write JSON file from existing map.

3.2 Block type manager

The block type manager manage every types of block and save them to the map controller.

3.3 Block type viewer

The block type viewer will display the block types according to the block type manager and provide interections with it. It allows to create a new type, to edit or to delete an existing type.

3.4 Map controller

The map controller manage the map storage.

3.5 Map viewer

The map viewer will display the map according to the map controller and provide interections with it.

3.6 Main window

The main window provide interections between all the modules.