PROJECT-SCOPES TECHNICAL REALISATION

by MicroScopes

Contents

1 Introduction

This document contains information concerning the technical aspects of the project Project-Scopes. The document aims to ensure the consistent and concrete vision of the project's technical implementation, of which the development team should follow. Any issues not covered in this document remain open to individual interpretation, however, it is recommended to maintain the inside-team-cohesion, especially in terms of tools and concepts used.

1.1 EPICs

Development process of Project-Scopes is divided into two EPICs:

- **EPIC 1 -** fully implemented local mode. In EPIC 1 the aim is to provide playable version for max 6 players on one machine

- **EPIC 2 -** fully implemented LAN mode. In EPIC 2 the aim is to allow players to connect to the game from their own machines

2 Technology and tools

2.1 Engine

unity

Unity 5.4.1f1 Personal is the main engine of Project-Scopes. Personal license provides full-featured Unity version free to use for private and commercial projects as long as the revenue is less than 100,000$. Unity's Scripting API allows to use programing languages such as Javascript and C#. The programming language for Project-Scopes scripts is **C#**. Scritps can be created in **MonoDevelop**, the *integrated development environment* (IDE) supplied with Unity, or other free licensed editor. MonoDevelop combines the familiar operation of a text editor with additional features for debugging and other project management tasks.

2.2 Version control

git

Git is the main version control system of Project-Scopes. It is free and open source distributed. All versioned files of the ProjectScopes are stored in free and public remote repository on **GitHub**. The main tool for verification and reviewing of commited changes is **GerritHub** integrated with GitHub.

2.3 Documentation

2.3.1 Project Documentation

LaTeX

LaTeX is a document preparation system in which the technical documentation of Project-Scopes is created. LaTeX is distributed as part of many TeX implementations. To create and edit LaTeX documents the TeX editor is necessary. For the Project-Scopes it is recommended to use **MiKTeX**, the free and open source distribution of LaTeX. For documents editing it is recommended to use **Texmaker**, the free and GNU General Public Licensed LaTeX editor.

2.3.2 Code Documentation

Doxygen

Doxygen is the tool for generating documentation from annotated C# sources. It is distributed under the terms of the GNU General Public License. for the Project-Scopes it is recommended to use LaTeX output.

2.4 Project Management

TAIGA

Taiga is a project management platform for Project-Scopes. Project is created on Taiga.io servers which allows to create one private project and no limited public projects.

3 Structure of the code

3.1 Architecture diagram

4 Game mechanics

4.1 Players

4.2 Arena

4.3 Bonuses

5 Graphical User Interface

The Graphical User Interface is fully designed and implemented in Unity game engine version 5.4.1f1 Personal. Only standard unity assets are used, no additional elements are required. All of the implementation is written in C# programming language.

5.1 EPIC 1 Specific Info

In EPIC 1 all of the users are gathered on the same device and all of the user actions are performed with this device's mouse and keyboard.

5.2 Palette of colors

The following picture shows the full spectrum of colors that are used in the project.

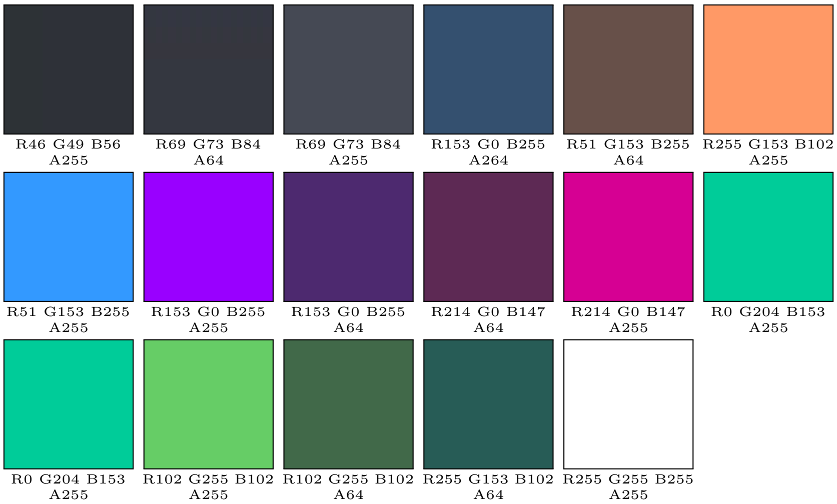

Figure 1: Pallete of colors

5.3 Components

The GUI is build with standart unity UI objects. These objects have been built into components that make up the user interface.

5.3.1 PlayersSettingsPanel

The main GUI canvas object contains one `Panel` object named `PlayerSettingsPanel`. It is the user interface background on which all other components are placed. The `R46 G49 B56 A255` value is used as its color. The panel size is `650x457` pixels.

5.3.2 PlayerDisabledPanel

One of the fundamental elements that is a part of GUI is `PlayerDisabledPanel`. It's the unity `Button` object which is used to add a player to the game. The component color is of value `R69 G73 B84 A64)`. It consist of two subcomponents. The first one is an inactive `InputField` unity object which indicates what will be the player color after the game start. There are six player colors available: `R214 G0 B147 A64)`, `R153 G0 B255 A64)`, `R51 G153 B255 A64)`,`R255 G153 B102 A64)`, `R0 G204 B153 A64)`, `R102 G255 B102 A64)`. The second one is `Text` object with default `+` value which indicates that PlayerDisabledPanel is to be pressed. Below pictures shows the panel details.

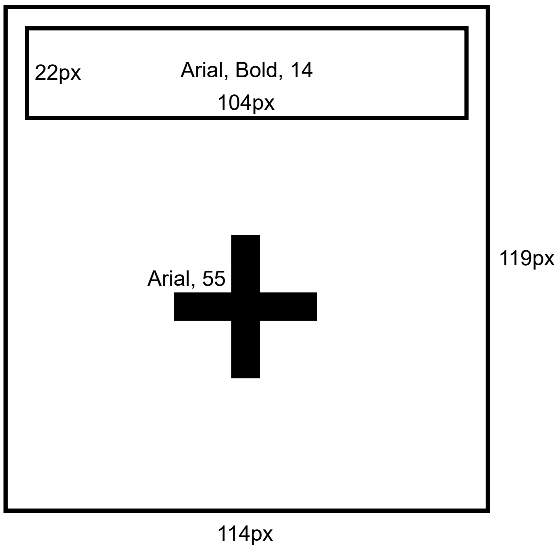

Figure 2: PlayerDisabledPanel size

The ingame pictures of all possible PlayerDisabledPanels look as follows:

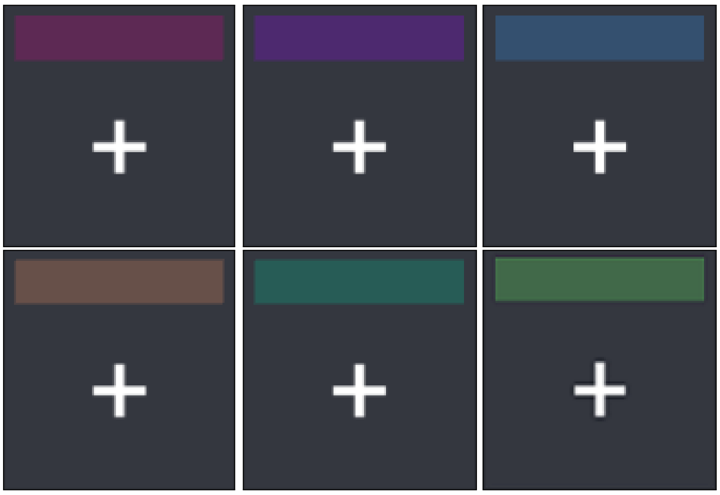

Figure 3: InGame PlayerDisabledPanels

5.3.3 PlayerEnabledPanel

The `PlayerEnabledPanel` is activated version of the `PlayerDisabledPanel`. The only difference in colors is the transparency, i.e. PlayerEnabledPanel is opaque. There is on more `Button` object located on InputFiled which is used to remove player from the game. Its constat `X` text has the same color as the InputField. Instead of Text `+` object the panel contains two more `Button` objets. The buttons are used to set players movenet keys. Both Buttons are the same and have `(R46 G49 B56 A255)` color.

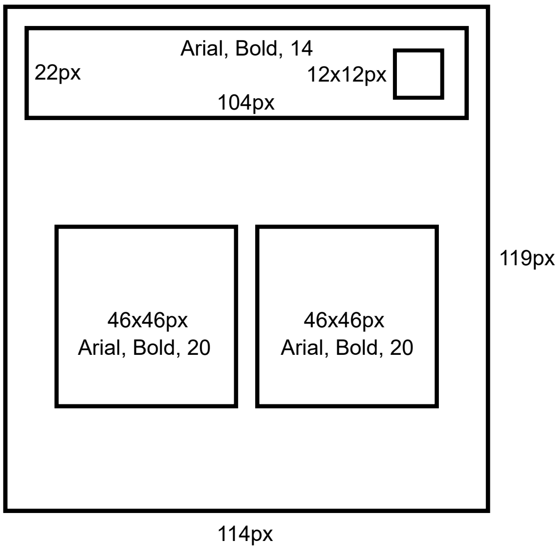

Figure 4: PlayerEnabledPanel size

The ingame pictures of all possible PlayerDisabledPanels look as follows:

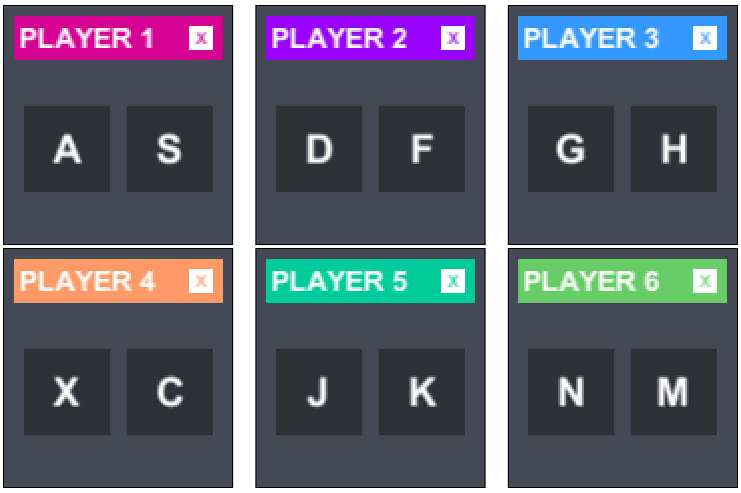

Figure 5: InGame PlayerEnabledPanels

All of the componetns has its default values hardcoded. All of them are explained in the implementation section.

5.3.4 ArenaSizePanel

The `ArenaSizePanel` background color is exactly the same as the color of the `PlayerEnabledPanel`. The panel itself contains two main components. The first one is a `Panel` with the background color the same as the color of the `PlayersSettingsPanel`. This panel contains a ”ARENA SIZE” `Text` of `(R255 G255 B255 A255)` color written in capitals letters only. The second element of the panel is a `Slider` of the following two possible colors depeneding on status: `(R255 G255 B255 A255)` in case it is not filled and `(R0 G255 B153 A255)` otherwise.

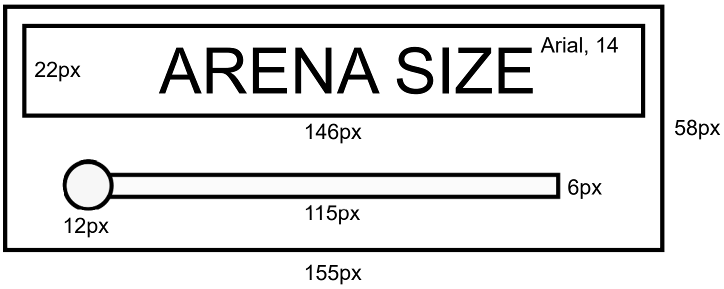

Figure 6: ArenaSizePanel size

The ingame pictures of the component looks as follows:

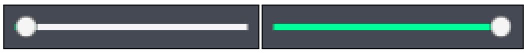

Figure 7: InGame ArenaSizePanel Slider

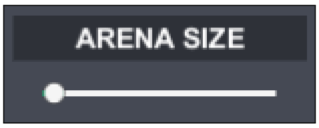

Figure 8: InGame ArenaSizePanel

The functionality and implemetation of `ArenaSizePanel` is described in implementation section.

5.3.5 InitialSpeedPanel

The only differece between `InitialSpeedPanel` and `ArenaSizePanel` is the text displayed on the panel. In this case it is ”PLAYERS SPEED”. For detailed information about the colors and components please refer to `ArenaSizePanel` section.

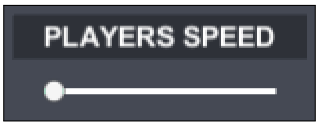

Figure 9: InGame InitialSpeedPanel

5.3.6 InitialSizePanel

The only differece between `InitialSpeedPanel` and `ArenaSizePanel` is the text displayed on the panel. In this case it is ”PLAYERS SIZE”. For detailed info about the colors and components please refer to `ArenaSizePanel` section.

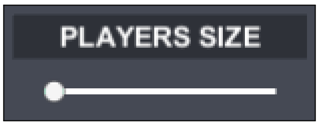

Figure 10: InGame InitialSizePanel

5.3.7 StartButton

The color of the ”START” `Button` is the same as the color of the ’slider panels’ text panel background. The `Text` ”START” has a pure white color.

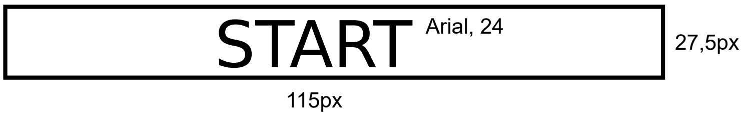

Figure 11: StartButton size

The ingame pictures of the component looks as follows:

Figure 12: InGame StartButton

5.3.8 Startup GUI

Here is an example of a GUI just after the game starts:

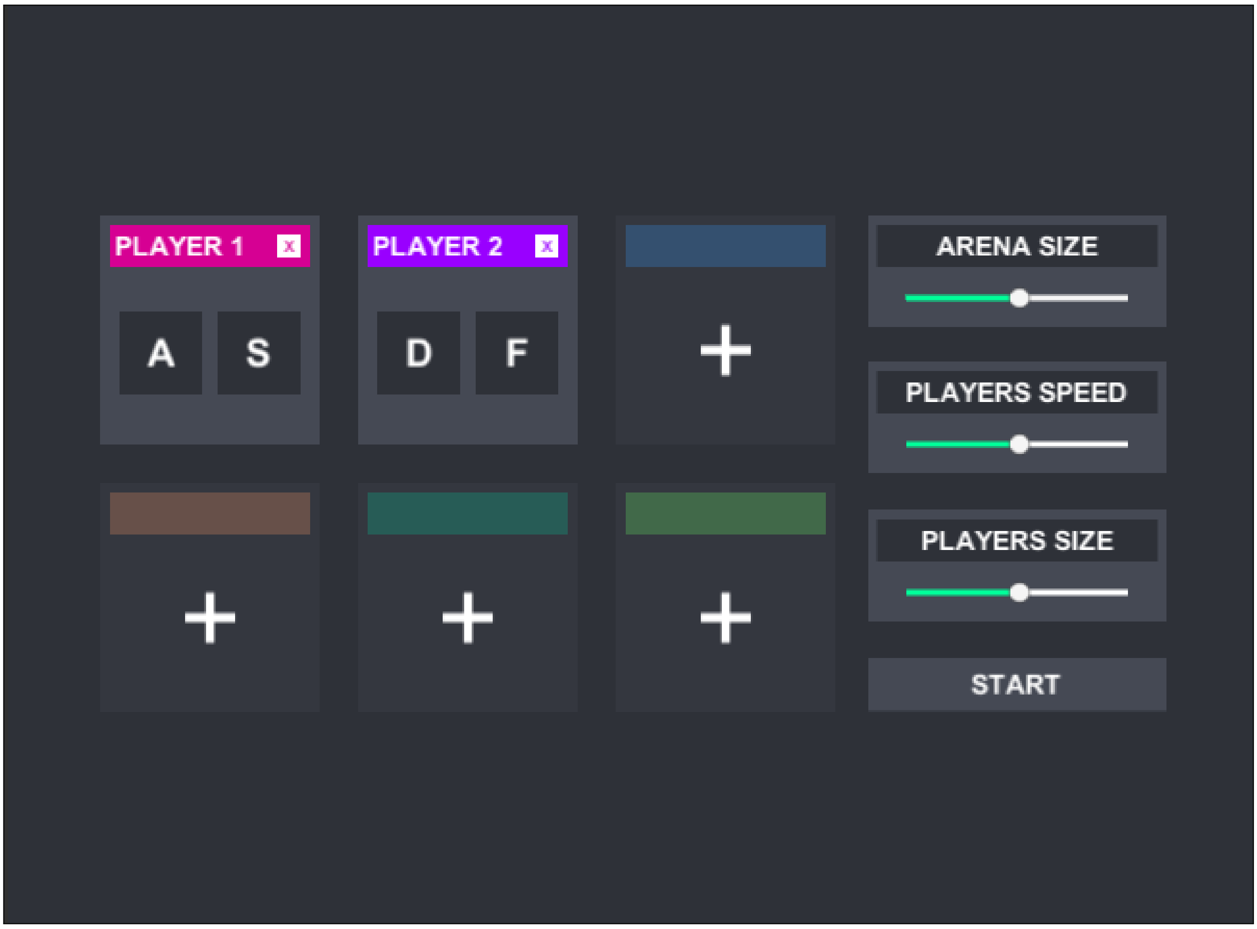

Figure 13: Startup GUI

5.4 Implemetation

The GUI implementation is located in *GUIManager.cs* script which uses *GUIHelper.cs* that contains helpful methods. The file is using *Configurator.cs* script to write the user settings before game starts. The following functionalities are implemented:

- Reading initial game configuration. It is stored in 'default.cfg' file.
- Adding and Removing player. There is a minimum of two players that must participate the game. There is no possibility to lower the value from the GUI. A user can manipulate the number of players from two to six. It is also impossible to have more than six players in the game.

- Setting the nickname of the player. On each `PlayerEnabledPanel` there is an `InputField` by witch user can set player unique name. The nickname is limited by 9 characters and may contain only english alphabet letters and digit.

- Setting the player movement keys. Each player must have its own movement keys. There is no possibility that two playes has the Setting key set. There is also no possibility that the player has the same key set for both directions.

- Changing the initial arena size. The `ArenaSizePanel` slider allows user to set the initial arena size. There are three possible sizes of the arena: small, normal and big.

- Setting the initial speed of all players. The `InitialSpeedPanel` slider allows user to set the initial speed value. There are three possible speeds to be set: slow, normal and fast.

- Setting the initial size of all players. The `InitialSizePanel` slider allows user to set the initial size value. There are three possible sizes to be set: thin, normal and fat.

- Starting the game. The `StartButton` loads a new scene with the game itself.

5.5 Sounds

There are no sounds implemented on the GUI yet.

6 Tests

6.1 Requirements

6.2 Types

6.3 Report

List of Figures

List of Tables