

Localization Unity Tool

Nvriezen & Bob Jeldes

Revision: 1.0.0

GDD Template Written by: Benjamin “HeadClot” Stanley

Special thanks to Alec Markarian
Otherwise this would not have happened

Overview

Genre

Features

Targeted platforms

Project Scope

Influences

- Wonder Boy: The Dragon's Trap

The elevator Pitch

What sets this project apart?

Technical Description

How does it work?

Priority

Code Structure

Overview

Genre

- Unity Localization Tool

Features

- Edit Game Text
- Dedicated Window inside Editor
- Screenshots for Context
- Loading Dedicated Scenes and Sequences

Targeted platforms

- Unity 2018.2.1f1

Project Scope

- Tool Time Scale
 - 8 weeks
- Team Size
 - Niels Vriezen
 - Dialog System
 - File Loading System
 - Bob Jeltens
 - User Interface
 - Unity Editor Window
- Software / Hardware
 - Engine
 - Unity 3D Personal
 - Software
 - MonoDevelop
 - Hardware
 - Laptop capable of running above software
- Total Costs
 - No Costs

Influences

- Wonder Boy: The Dragon's Trap

- The game came shipped with the engine used to create the game (Desktop releases only). Niels having used the tool and wanting a way to easily edit and translate text in his own games, he got inspired to create his own.

The elevator Pitch

A dedicated Unity Localization Tool specifically targeted to the translators instead of the developers. Focusing on User Interface for the tool itself, together with packing features like real-time editing and screenshots for context.

What sets this project apart?

- User Interface specifically targeted to translators
- Real-time editing
- Dedicated Unity Editor Window for easy access.
- Screenshot feature

Technical Description

How does it work?

- Bob is focusing on the user interface while Niels is focusing on the file loading system.
- We plan to first create a rough prototype where we load in text from a JSON or regular text file. This text will be viewable through a tool window inside Unity. This way we get to know the technologies we want to use a little bit more, before diving into the deeper layer of our tool.
- After the first prototype we slowly add more features and try to focus on efficiency. Choosing the final approach to load files, together with finalizing our design for the user interface. This does not lock everything down. As we always try to find a better way to do things. This means the design, even though finalized, can always change if we think we can do it better.
- The last one or two weeks will be used to iron out most bugs and be sure we got what we wanted. Maybe there is even some time to polish things up a bit.

Priority

- In short
 - Managing huge amount of text
 - User Interface
 - Editing feature
 - Screenshot feature
 - Scene loading feature

Code Structure

- We will keep the user interface and loading of all the data separated. These two systems will talk through one manager script. This manager script for example, simply calls some functions to load data if the editor window requests it.