UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Centre de la Imatge i la Tecnologia Multimèdia

Tool for Data Collection and Analysis in Unity

Bachelor's Degree Final Project

Rúbrica 1

Bachelor's Degree in Video Game Design and Development

Surname: Portella Name: Lorién

Pla: 2014 Director: Löpfe, Lasse

Index

Summary

This document elaborates on the analysis of data collection and visualization tools for videogames, and follows the development of a tool for the Game engine Unity that allows the users to collect, and then visualize said data.

The first section of the document narrates the motivation behind this project, and the problem it is trying to solve. It also covers more specific objectives, related to the final product, that the author also wants to approach.

Then, it elaborates on the State of the Art of the tools that already exist, both for data collection and data visualization, and how can they affect the outcome of the final product. It also analyses the current state of the software that is going to be used, and what would be the best version.

The following section, explains how the project is going to be done. The milestones to be accomplished, the time they should be accomplished, and the different phases of the project, and its description. It also explains the tools that are going to be used to keep track of the progress, like Trello, for short term tasks, Clockify, to keep track of the hours spent, and two GitHub repositories, for the tool and the documents. The validation techniques will enumerate the various methods that are going to be used during the whole production process, to make sure that the tool is up to standards, and specially at the end of the project, to evaluate the final product.

The methodology section will elaborate on the methodology that is going to be used during the development of the tool, the AGILE methodology, and how does it match and affect this specific project.

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Centre de la Imatge i la Tecnologia Multimèdia

Keywords

Data, Collection, Visualization, Unity, Tool

Links

Documents: https://github.com/Witiza/-Tool-for-Data-Collection-Analysis-and-Visualization-in-Unity

Tool: https://github.com/Witiza/DataVizTool

Table Index

Glossary

Dataviz: Short for Data visualization

UML: Unified Modelling Language is a general-purpose, developmental, modelling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system

CSV: file format that consists of a text file that uses commas to separate data values.

Game Engine: Software designed for the production of video games.

Unity: Multi-Platform game engine that can be used to create 3D and 2D games.

Scene: Environment in Unity where all the agents and props are placed.

GameObject: Entity that represent characters, props and scenery in Unity.

Component: Element that can't be attached to a GameObject for different types of functionality.

1. Introduction

1.1 Motivation

I have been working with Unity for the past four years, both at university and with personal projects, and I love how the engine manages to merge accessibility and immediateness with complexity, allowing to do from quick prototypes to fully fledged videogames. This last semester, I got introduced to data analysis and visualization, and its practical use in videogame development.

I did a bit of research and realized that, despite how practical a data collection tool for a big game engine like Unity, would be, there were not many options outside of Unity's own Game Analytics.

1.2 Problem Formulation

Few data collection tools are available to the public, and many developers end up producing their own tools for their company/games. This can be an issue for Indie studios who have a more limited budget, and creating a data collection and visualization tool can be too time consuming. This can also lead to studios straight up ignoring feedback they could get using data from their players.

Data visualization tools are more common, and available to the general public, but without a way to collect said data, free visualization tools become useless. Also, visualization functionality that involves a part of the game like a level become harder to pull off.

1.3 General Objectives of the TFG

The main objective of this TFG is to develop a tool encapsulated in a Unity Asset Package, that provides users with data from their games with minimal intrusion, and then allows for the visualization of said data in the Unity Engine.

- Allow for Data Collection
  The tool should be able to let the user configure which data they want to track, with the tool being the less intrusive possible, while giving the user the biggest amount of customization available.
- Allow for Data Visualization
  The visualization of data should go further than showing raw data, obviously. The basic graph types need to be supported, as it will be expanded in the State of the Art. The graphics should be interactable to some degree, letting viewer hover over data to highlight that section. Finally, the user will be able to visualize the data inside the game itself, without having to actually run the game, using a heatmap.

1.4 Specific Objectives of the TFG

- Develop a tool in Unity that uses its editor UI: Familiarize with Unity's GUI system and develop a tool that is perfectly integrated into unity and its UI.
- Render over a Unity Scene: Reduce the amount of loading time that the tool needs by rendering the heatmap over the Unity Scene using Shaders.
- Put it in the Unity Asset Store: When the tool is finished, publish it to the Unity Asset Store.
- Database Functionality: Set up a local database to connect the data collection and visualization, with options to improve to a remote database if there is enough time/resources.

1.5 Reach of the Project

The Tool will work with Unity, one of the most used Game Engines in the market, that has a free license with all the functionality. This means, that anyone or any studio that uses Unity to produce their games. But inside this big audience, the project is intended to be put in the Unity Asset store for free and to be used primary by indie or hobby developers. If the end result proves to be useful enough, bigger studios would probably use the tool too.

2. State of the Art

2.1 Unity

Before getting into data collection and visualization, it is important to analyse the environment we are going to be working in. In this case, we are talking about the **Game Engine** Unity. This software will serve as a framework to both develop our tool, and then implement it. That is why we need to look into Unity versioning. Unity is a finished product, but it is also one constantly evolving. It does this by releasing new versions each year, and inside those versions, patches. We are going to use one of the latest stable versions, **Unity 2020.1.16f1**. Despite this, unless we use a feature added in this version, our tool will have backwards compatibility with versions form 2018 upwards, due to Unity guaranteeing Long Term Support of this versions. By developing our tool in a rather new version, there are less chances of incompatibility once new Unity patches are released, and thus, we will need to update our tool less frequently.

2.2 Market Study

The following tools are the direct competitors, and references, of the tool in development. I will analyse each one, and elaborate on what do they do right, what do they do wrong, what does our tool provide that they don't, and what can we use from this software in our tool

2.2.1 Unity Analytics

Unity already has its own analytics system, Unity Analytics. It provides both data collection from your product, to data visualization in their webpage. Users can use the default events, without the need for implementation on their games, as with only activating Unity Analytics in your Unity project, it will automatically track:

- New installs
- Daily active users (DAU)
- Monthly active users (MAU)
- Total sessions
- Sessions per user
- Time spent in app
- User Segments for Country and Platform

Then, it offers default plots like Active Players, New Users, or the Sticky Factor (how compelling a game is for new users) Average Revenue per User, and more, using the data gathered.

Unity Analytics also offers the option to program custom events from the project itself, allowing the user to track custom variables, and then plot them in the webpage.

Finally, if the user is subscribed to Unity Pro, they will be able to visualize analytic events in a Heatmap.

Issues found in Unity Analytics

- For some features, the users need to be subscribed to Unity Pro, which is not common in smaller development teams.
- During this process, Unity has access to your game's data.
- Despite being customizable, setting up custom events can be restrictive.
- The documentation, despite being produced by Unity, focuses more on theoretical aspects than the practical use of the tool.
- The graph options are limited, as only funnel, scatter plots an bar graphics are available.

What can we use from Unity Analytics

- Template Collection: Allowing the user to just drop our tool in their game, and already track basic in game events can be key to help the users get the hang of the tool, while they also have data to start using the visualization tools.
- Template Visualization: In the same way we have default data collection, we can use this data to plot visualization to the user.

Conclusion

Despite Unity Analytics being very similar to our tool, we can find enough differences to develop a different product. First of all, our tool is completely free, and will let the user do visualization, heatmaps, and even download the Raw Data to use in other programs. Another strong point for our tool is that it will have more visualization options. Graphics like Sunburst, Bubble charts or geographic graphs can be added to the tool, in order to compete with Unity Analytics. Finally, the tool will be open source, which will let users to modify and adapt its code to suit their needs. With a better documentation and accessibility, it can easily become a more usable tool than Unity Analytics.

3. Project Management

3.1 Procedure and tools for the tracking of the

3.1.1 GANTT

		Start	End	Weeks	R1							R2							R3							R4						Defensa
					1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21							
					05/03/20	12/03/20	19/03/20	26/03/20	02/04/20	09/04/20	16/04/20	23/04/20	30/04/20	07/05/20	14/05/20	21/05/20	28/05/20	04/06/20	11/06/20	18/06/20	25/06/20	02/07/20	09/07/20	16/07/20	23/07/20							
Documentation			1	16																											
	Prepare Workspace	1	1																												
	Sections Draft	1	1																												
	Introduction	1	2																												
	Project Management	1	2																												
	Methodology	1	2																												
	State of the Art	2	3																												
	Review Documentation	3	4																												
	Development	4	17																												
	Conclusions	15	17																												
		15	17																												
	Future of the Project	15	17																												
Data Collection																															
	UML and Structure	4	4																												
	Basic Component	4	4																												
	Component UI	4	5																												
	Component Funcionality	5	6																												
	Data Saving	6	6																												
	Testing	6	6																												
Data visualization																															
	UML and Structure	6	7																												
	Unity Class	7	7																												
	Visualization GUI	7	8																												
	Visualization Rendering	8	8																												
	Data Visualization	8	9																												
	Testing	9	9																												
Database																															
	Database Research	3	3																												
	Database UML	10	10																												
	Write to Database	10	11																												
	Get Data from Database	11	12																												
	Local Database	12	13																												
	Remote Database	13	14																												
QA																															
	Project Simulation	14	14																												
Publishing to Unity																															
	Look into Asset Store Publishing	15	15																												
	Prepare Assets	15	15																												
	Publish	15	15																												

Table 1, GANTT Chart

3.1.2 Trello

For keeping track of more immediate tasks, Trello will be used. For the practical parts of the project, the plan is to first do the UML for each section, and then divide the resulting scheme in tasks to be done in the Trello board, also following the sections and tasks in the GANTT chart.

3.1.3 Clockify

An online clock that allows the user to keep track of the time spent working, and in which tasks is the time being spent. Additionally, there is a section to analyse the hours logged and see the total hours spent, order them by tasks or projects, and more.

3.1.4 GitHub Repository

For this project, two GitHub repositories have been created. One will hold the Unity Project witch the scripts for the tool and the QA, and the other will keep track of the project document. The reason behind this is that having the Unity Project in a separate repository draws the line between the TFG and the actual tool, in order to facilitate forking of the repository and having the commits more ordered.

3.2 Validation Tools

During the project, various validation tools will be used to evaluate the project at certain stages and dimensions.

- Inside Validation: When finishing each of the three modules, data collection, visualization, and databases, a testing session will be done. To test if data collection is working properly, a unity demo project will be used, where the collection tool will be implemented, and after play sessions, we can see if the data is being collected properly. For the visualization, we can either use the data generated in the collection evaluation to then try and visualize said data, or artificially generate our own data, with a script for example. Finally, to test the databases, play sessions can be done again, and then visualize the data generated automatically.
- Implementation in a new project: For this validation, I will use a current videogame that I published to Steam working with a team. I will implement the tool in this videogame, and then have play sessions in order to generate data. If remote databases are implemented, I can play the game from other computers or users, to check if it works properly.
- Usability validation: Finally, to test if the tool is clear enough, I will ask various classmates and colleagues, to run a test where they need to implement the tool in a demo project, and set it up so it collects basic data, like position, attacks and deaths.

3.3. SWOT

	Helpful	Harmful
Internal	**Strengths** I am very familiar with the platform (Unity) where the tool will be developed and used. I have done a data analysis subject, and I am currently studying Data Visualization	**Weaknesses** I cannot dedicate 100% of my academic time to the project, as I am also studying two subjects. I have a lack of knowledge of databases, and is an important part of the project.
External	**Opportunities** The platform of the tool (Unity) is widely used by our primary target audience, hobbyists and small indie studios.	**Threats** In spite of having differences, Unity's own data analysis tool and the tool developed share similarities.

Table 2, SWOT analysis

3.4 Risks and contingency plan

Risk	Solution
Unity rendering being too difficult to implement smoothly, resulting in a more simple heatmap.	To compensate, more data visualization functionality should be added, like more graph variants or more interactability.
Unity GUI not being flexible enough, the customization level of the tool will suffer.	The code of the tool will be structured and commented enough to facilitate direct editions. Other aspects such as data recollection speed or size can be reinforced to compensate.
Databases being too complicated	Albeit being a key component, the complexity of the database section of the tool can be reduced, and even removed, without the other aspects being affected.
Lack of time to finish the tool.	The tool is modular, and sections can be reduced without affecting the rest. Data collection can be made less customizable and data viewing can have less visualization options.

Table 3, Risks and contingency plan

3.5. Initial costs analysis

To calculate the costs of the project, I will assume I am the sole developer for this tool, and I have a Junior level.

Type	Subject	Price			
Salary	Junior	€	5,040.00	Salary per hour:	21 €
Equipment	PC	€	900.00	Working hours per week	15
	Peripherals	€	60.00	Development months	4
Marketing	Logo	€	200.00		
		Total:	€ 6,200.00		

Table 4, Costs

The price of the logo is what a Junior Artist would ask for a simple logo, including small size, bigger size, and banner for the Asset Store

4. Methodology

AGILE methodology is the one I have chosen for this project, as it is a methodology, I am familiar with, and it suits the development of a software like this project.

Looking through the agile manifesto, the pillars that I will pay more attention and elaborate my plan around are:

- Individuals and interactions over processes and tools: The tool will be user oriented, and this means that it needs to be clear and concise enough for anyone with minimal knowledge to use it.
- Working software over comprehensive documentation: Despite having a documentation and references of sorts, the tool should be able to be operated without the need to constantly refer to a manual
- Responding to change over the following a plan: Being the biggest project I have done yet, changes are bound to happen. The tool will be open to modification during the whole production process, even during the final weeks, as long as it does not conflict with the planning and time available.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale: working software is the primary measure of progress.

Documentation

The first phase of the project is to research and write down how the project is going to get done. To do this, the necessary workspaces have been created, like the Trello Board, the Clockify project, the Unity Project, and most important, both, the GitHub repository that is going to store the Unity Project, and the repository that is going to store and keep track of this document and additional resources that are made, like excel sheets.

The practical part of the project has been divided in three main sections, as they are independent from each other, and by doing this, the progress for each section is independent, despite being ordered in the GANTT chart. What this means is that, in order for a section to be worked on, the previous section in the chart does not need to be completed, and testing can be done for each section independently.

Data Collection

The first functionality that is going to be done, is the collection of data from the game the tool is being used on. The specific objectives for this section follow the procedure to develop a software. First, and **UML** will be done to have a scheme of how will the scrip be structured. The objective is to have a **GameObject Component** in order to collect the data from the game, so the next step is to create the base class and structure of the component. After that, both the UI of the component and its functionality will be programmed. The UI is important, as it is the element that the tool users will interact with. Finally, the component will need to save the data generated as **CSV** files.

Data visualization

The next section in the development will be Data Visualization. The steps will be the same as in Data Collection, starting with an **UML** to define the structure, and then base structure that evolves into both functionality and UI, with the difference that, instead of a Component, a C# class will be used. In this case, functionality refers to the ability to receive **CSV** files and visualize them. To modify how data is visualized, the user will be able to modify settings and choose options in a UI. Additional sections in the GANTT like Visualization GUI and Rendering have been added, as both are tasks that can prove to be difficult. GUI refers to the creation of a complex UI Window resembling the ones used in visualization software, using Unity's GUI system. And Rendering is the process of displaying the data over the Unity Scene, via Heatmaps and event timelines that are rendered directly into the scene.

Database

The third main section of the project will be the one that ties the previous two together. It is also the one that is most subject to change, as the project progresses, either due to a lack of time, or an adjustment in the previous sections or the project itself. The first goal is to have a local database we can write data and read data from. Then, the objective is to move the local database to a remote one, so in a practical use of the tool, the users would be able to gather data from various instances of the tool. This can prove challenging, as the knowledge of this section is inferior to the previous ones.

QA

After finishing the tool, and testing of each individual section, an overall QA is required. To do so, a dummy project will be created, and the Tool implemented. For further testing, external users can be recruited as QA testers of the tool.

Publishing to Unity

Finally, once the project is completed, it needs to be uploaded to Unity's Asset Store, and assets like screenshots, readmes and instructions will also be created.

7. Webography and Bibliography

- Agile Manifesto. *https://agilemanifesto.org/*