

REQUIREMENT ANALYSIS OF USER INTERFACE COMPONENTS FRAMEWORK FOR MOBILE DEVICES

Yurii Milovidov

Abstract. *The article suggests approaches to simplify the design of a user-friendly and modern graphical user interface for mobile devices. Comparative analysis of mobile application development environments is represented. The selection of the development environment for creating the user interface is justified. Conclusions are made regarding the choice of the Unity environment, which supports the compilation of applications for various platforms, as well as the developer-friendly mechanism for creating composite controls.*

Key words: *graphical user interface, framework, mobile development, cross-platform development, user control.*

ITHEA keywords: *D.2 Software engineering, D.2.1 Requirement/Specification, D.2.13 Reusable Software.*

Introduction

Designing convenient and effective graphical user interface (GUI) is one of the most crucial aspects of mobile application development. It is a precondition for creating various GUI components for simplifying of user interface planning. But limitations of development environment may cause extra difficulties while implementing the GUI. There are three most common ways to surpass such difficulties: creating self-developed solutions, paying for ready-to-use non-free libraries or looking for some options on the marketplace.

In any case, stakeholders should set up the process of reusing GUI libraries.

Preparing new GUI libraries is performed by combining default user interface (UI) components into certain composite objects. Such objects, possessing needed functionality, are close to what developer needs in the majority of cases. Such approach is time-consuming - developer has to join components and set up all the interactions and visual parts. So, it is a proper solution to create a library of assets that are reusable and cover common cases and demands.

This grounds the actuality of the problem of implementing a library for designing composite user elements in game development engines. This article is devoted to solving this task for Unity3D environment.

Other widespread effective requirement analysis techniques are based on Model-Driven Development Approach using visual modeling languages as UML [Chebanyuk, 2014a], [Chebanyuk, 2014b], [Chebanyuk and Shestakov, 2017].

Related work

While development frameworks for mobile devices provide rich support for sophisticated input mechanisms like gestures, etc., they lack support for graphical editors. In the paper [Buchmann and Pezoldt, 2014] authors present a lightweight framework for graphical editors which empower the user to easily build touch-enable graphical editors for android devices.

The development and maintenance of mobile applications for multiple platforms is expensive. One approach to reducing this cost is model-driven engineering. In the paper [Jia and Jones, 2012] authors present a novel model-driven approach to cross-platform mobile application development using a Domain Specific Language (DSL), called AXIOM (Agile eXecutable and Incremental Object-oriented Modeling). This approach could significantly reduce the development cost and increase the product quality of mobile applications.

Unity is a feature-rich, fully-integrated development engine that provides out-of-the-box functionality for the creation of interactive 3D content. The book [Thorn, 2015] shares extensive and useful insights to create animations using a professional grade workflow.

The book [Smith and Queiroz, 2013] helps build successful games with the Unity game development platform using the powerful C# language, Unity's intuitive workflow tools, and a state-of-the-art rendering engine to build and deploy mobile, desktop, and console games.

Task

Design a framework of graphical user interface components that is answer to the next requirements:

- convenient for reuse;
 - containing elements that are not present in standard UI framework.
-

Grounding of choice game engine for user interface designing

Currently there are several environments for Android applications development, and they vary in many ways. The majority of such environments require purchasing a specific license for commercial usage. However, there are some non-proprietary solutions with open source code. The most widespread are:

- android studio;
- unity.

The major advantages of these environments are:

- presents of debugging tools;
- out-of-box software for hardware emulation;
- integration with modern integrated development tools.

Peculiarity of Unity3D is a possibility of creating cross-platform mobile applications, while Android Studio is designed only for implementing Android apps.

Asset store libraries overview

Represent an analysis of free libraries that are represented on Asset Store market in "Scripting/GUI" category.

Consider the next uGUI Windows Extension, Skill UI, NGUI Infinite Pickers and GS Custom Multipurpose Dynamic Listview.

uGUI Windows Extension is a library by Motion Entertainment that is designed to use modal and dialog windows in a Unity application. This library includes customizable templates and ready-to-use features like window transitions. The appearance of components in uGUI Windows Extension is highly customizable.

Skill UI by Hamed's Games is a set of components for designing user interfaces includes the following:

- grid for displaying text, images and other interface elements in a tabular view in Unity3D;
- uniform grid for displaying the very same information as grid when all components, and grid cells have the same size;
- dock panel - interface area used for placing child elements in horizontal or vertical rows;
- wrap panel - interface area that places child elements horizontally from left to right;
- stack panel - interface area that places child elements in a line that may be aligned horizontally and vertically.
-

NGUI Infinite Pickers (developer Cregzo) is a big collection of GUI widgets:

- date picker - interactive scrollable widget for selecting date and time values;
- image picker - interactive widget for selecting an image (i.e. game avatar);
- item picker - interactive widget for selecting one of text field options (i.e. character class selecting).

Requirement specification for GUI components library

A requirement specification for graphic elements' user library is presented in the table 1.

Table 1. Requirement specification for graphic elements user library

Requirement code	Requirement description
Functional requirements	
F1	Tabular display of text and numeric data;
F2	Tabular data scrolling
F3	Table is updated as soon as the data are changed. After this, the table is available for new review.
F4	Vertical and horizontal content alignment
F5	Text areas display depending on state of other widgets
F6	Text areas displaying and collapsing data
Non-functional requirements	
NF1	To be well documented using behavioral UML diagrams
NF2	To be designed for easy reuse
NF3	To be extensible
NF4	Adapting UI components for different resolutions of mobile screens
NF5	Portability

Consider process of software requirement elicitation taking an example functionality of UI component supporting representation of table supporting scrolling. Activity diagram for UI table with scrolling is shown in the figure 1. Sequence diagram is shown in the figure 1.

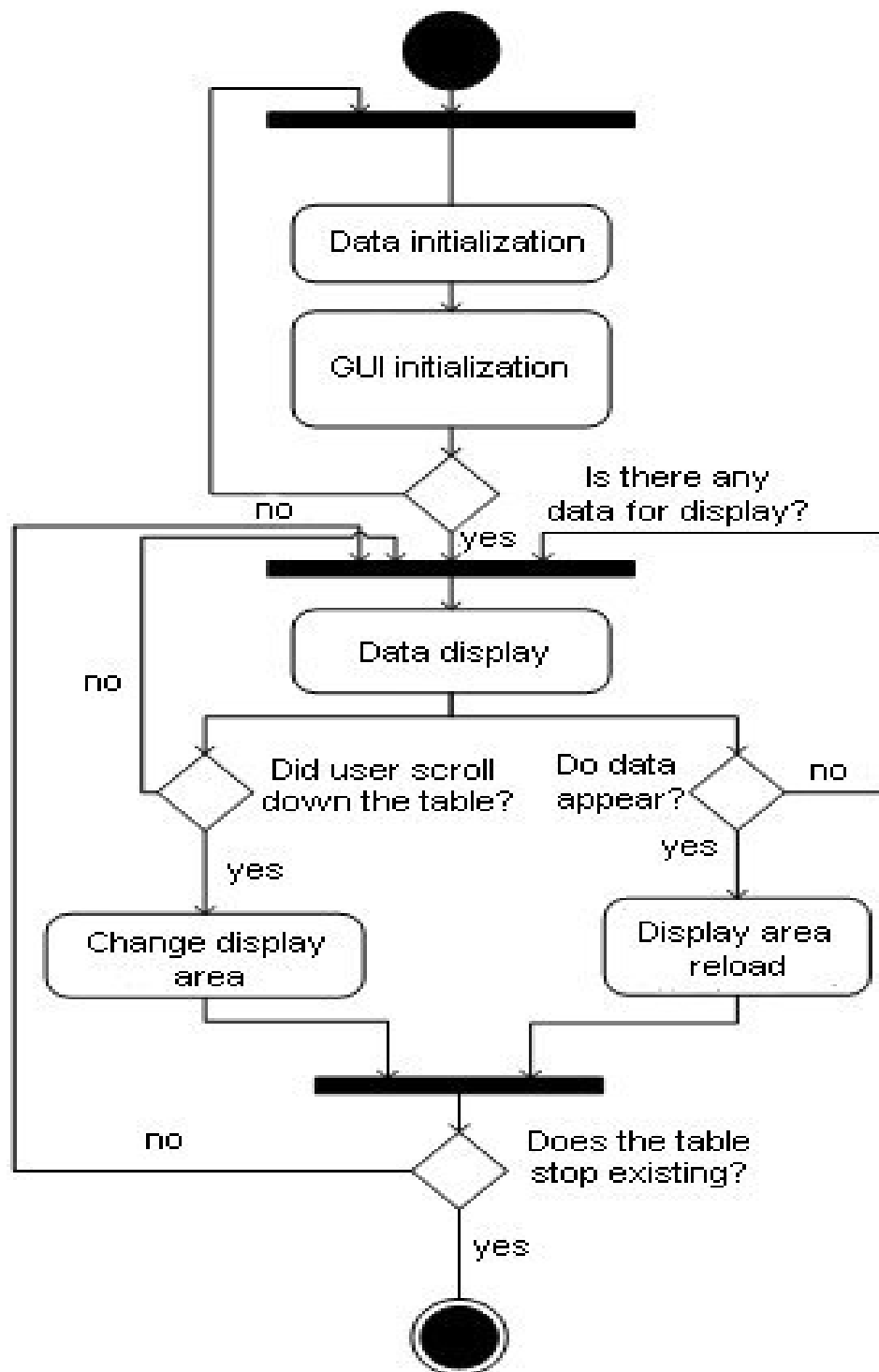


Figure 1. Activity diagram for UI table with scrolling

During table draw the table data are initialized and displayed. Table is bind to database, file or other data structure defined by developer.

During the data display, user is able to scroll through the component, is the amount of records is bigger than the display area. For the detailed analysis of algorithm for user interface working sequence diagram is designed. User passes data inside the table, which should check the data and display it in a tabular representation with indents and cell alignments.

All the alignment and indentation parameters are set up by the component developer. Upon the update, data is rewritten and resent to the table. After this, the table is filled with up-to-date information. As soon as user leaves the table it is destroyed. However, all the data are saved and renewed table have to be redrawn further.

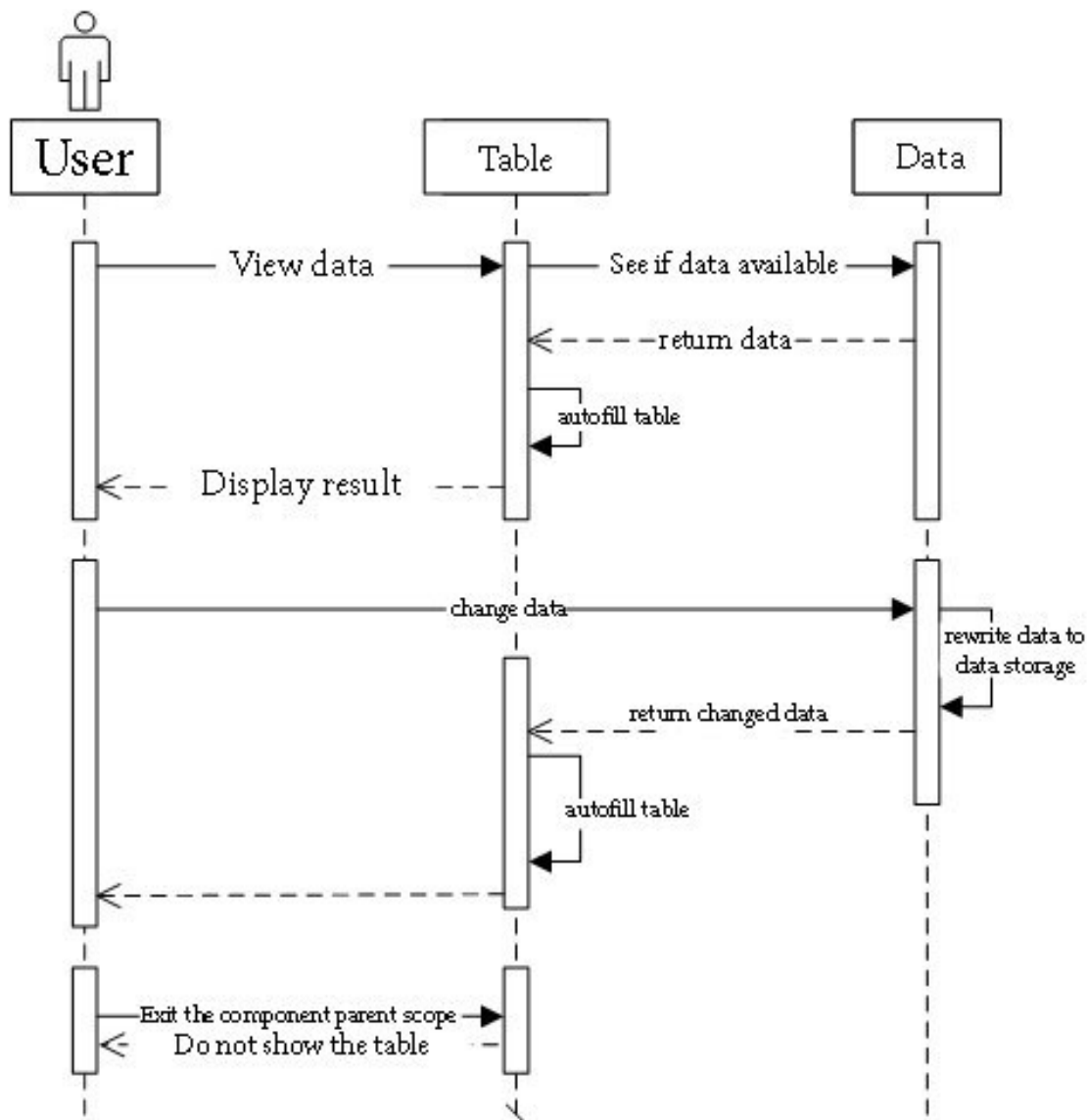


Figure 2. Sequence diagram for UI table with scrolling

Elicitation of non functional requirements

Convenient using

All the library elements are composed from the basic Unity GUI elements. It has a simple-to-understand API for developers acquainted with the Unity GUI system. In terms of ergonomics, library components can match the default GUI system components.

Portability

Library components may be used in any Unity project, which implies it can be used on various platforms and multiple operating systems.

Unity v5.3 is available on PC, Mac, Linux, Android, IOS, Tizen, PS3, PS4, Xbox 360, Web GL, Unity Web Player, Apple TV, Samsung TV, which gives an opportunity to present the product to a large audience.

Conclusions

In scope of a given paper, the following results have been accomplished:

1. a specific development environment for library creation has been chosen;
2. functional requirements for library components have been defined;
3. a fundamental library structure has been established;
4. the dependencies and interconnections between libraries have been set up;
5. prototypes of user components have been designed according to the functional requirements.

Bibliography

[Buchmann and Pezoldt, 2014] Thomas Buchmann, Patrick Pezoldt. A Lightweight Framework for Graphical Editors on Android Devices, Proceedings of the 9th International Conference on Software Engineering and Applications, August 29-31, 2014, in Vienna, Austria, University of Bayreuth, Germany ISBN: 978-989-758-036-9 – Access mode:

<http://scitepress.org/PublicationsDetail.aspx?ID=5zulhFNKzes=&t=1>

[Chebanyuk, 2014a] Olena Chebanyuk Method of behavioral software models synchronization International journal Informational models and analysis. 2014, Vol. 3, № 2, pp. 147-163

[Chebanyuk, 2014b] Olena Chebanyuk. Method of domain models designing. International Journal Informational Models and Analysis, 2014, Vol. 3, № 3. pp. 233-245

[Chebanyuk and Shestakov, 2017] Olena Chebanyuk, Kyrylo Shestakov. An Approach for Design of Architectural Solutions Based on Software Model-To-Model Transformation "International Journal Informational Theories and Applications", Vol 24, № 1 – 2017, p.60-84

[Jia and Jones, 2012] Xiaoping Jia, Christopher Jones. A Model-driven Approach to Cross-platform Application Development, Proceedings of the 7th International Conference on Software Paradigm Trends - Volume 1: ICSOFT, 24-33, 2012, Rome, Italy. ISBN: 978-989-8565-19-8 – Access mode: <http://scitepress.org/PublicationsDetail.aspx?ID=sbAVxycyWJ0=&t=1>

[Smith and Queiroz, 2013] Matt Smith, Chico Queiroz. Unity 4.x Cookbook. – UK: Packt Publishing, 2013. – 386p.

[Hocking, 2015] Joe Hocking. Unity in Action: Multiplatform game development in C# with Unity 5, ISBN: 9781617292323, 2015. – 352 p. – Access mode: <https://www.manning.com/books/unity-in-action>

[Thorn, 2015] Alan Thorn. Unity Animation Essentials, ISBN113: 9781782174813, 2015. – 200 p. – Access mode: <https://www.packtpub.com/game-development/unity-animation-essentials>

[Lukosek, 2016] Greg Lukosek. Learning C# by Developing Games with Unity 5.x, ISBN13: 9781785287596, 2016, 230 p. – Access mode: <https://www.packtpub.com/game-development/learning-c-developing-games-unity-5x-second-edition>

Authors' Information



Yurii Milovidov - National University of Life and Environmental Sciences of Ukraine, Kyiv, scientific Department of Programming Technologies. Senior Lecturer.

e-mail: milovidov@email.ua

Major Fields of Scientific Research: Programming technologies, web-design, Internet – technologies.