Cubecraft Game: A Framework for Mobile Game’s Development

Manuel Romero Rivera, Jorge Aguilar Cisneros

Universidad Popular Autónoma del Estado de Puebla
21 sur 1103 Col. Santiago Puebla, Pue., México. C.P. 72410

manuel.romero@upaep.edu.mx, jorge.aguilar@upaep.mx

ABSTRACT

A video game is a software system destined to entertainment. It can be played in different devices as computers, consoles, mobile devices, and so on, integrating audio and video that allows enjoy experiences that, in many cases, would be difficult to do in the reality. Although the video games now are more than informatics products; it is also a business, information and formation tool, an object of investigation and even a social phenomenon [1].

There are some frameworks used to make video games for different operating systems (Android, IOS and so on) and for different electrical devices (computers, mobile, and so on), but all the time we need new frameworks to help people to develop by themselves their own video games even though They do not know anything about software code. This is the reason that we do a proposal to build a new framework called Cubecraft. It was oriented for kids in elementary school, between 8 to 12 years old. We hope our framework allow them to develop video games for Android mobile devices. They can build their own video games and storage it into their cell phones without write any line of code only dragging and dropping cubes in our framework.

We also propose design and develop a website where students will be allow to show and share their games, in this way we are going to encourage socialization and collaboration among them..

KEYWORDS


1. INTRODUCTION

Over the past 30 years, video games have become an important part of contemporary global entertainment. Today they are a huge business worth billions of dollars [10]. Furthermore, video games have great positive in addition to their entertainment value and there has been considerable success when games are designed to address a specific problem or to teach a certain skill [11]. We think is important to play video games but is more important to build our own video games.

There are a big quantity of tool and framework to allow developing video games, for example, commercial: Abyssal Engine, Alamo, Cafu Engine, and so on, free an open source: 3D rad, Adventure Game Studio, Blender, and so on, Freeware: Real Library, BYOND, Defold, and so on. Anyhow, we want to build a new framework with the paradigm drag and drop because we want to build a framework to our Mexican kids, we want to hear their requirements about our framework, in this sense we developed a
survey where we asked about our framework and our outcomes show us: Cubecraf is easy to use and attractive, user do not have problems to interact with it.

2. PROBLEMATIC

Nowadays, the video games are a ludic phenomenon very important into the childhood and adolescent culture. For example, in a research made with teenager in ages between 14 and 18 years, the percent of persons who play video games were 59%.

Perhaps, these persons could have been asked themselves, at least one time: How a video game is made?, What I have to know? or how do I start?. Perhaps an important quantity of them have tried to make a video game without a successful result, however, others, perhaps, have created their first game with hard working, but there is always questions without answers and a lot to learn [2].

When we talk about video games development, we used to think that just one special person can creates it, but it is a big mistake, because create a video game needs a very different people like designers, musicians, writers, etc.

Once the team is complete, the Software Engineering needs to have a specific knowledge about develop and manage an industrial project software development, only in this way a game software development could be successful.

3. STATE OF THE ART

A framework is a software structure, made by customizable and interchangeable components to develop a software application [3]. Usually it is formed by support systems, compilers, code's libraries and a friendly user interface that help to use the entire component in an easy and fast way [4].

In other words, a framework could be considered as a generic, incomplete and customizable software application, that the user can add the correct pieces to create the application.

Nowadays, there are a lot of frameworks for games development, but they are focused to different purposes. In this paper, we search information of the most used frameworks, based in their advantage and disadvantage. This analysis are going to be used for establish the basis of our framework.

The first framework analyzed was Kodu. This is a visual software tool specialized for Xbox and PC video games, generates by a conceptual idea. This tool is designed to be accessible to kids. Now, Kodu is used in countries as Malaysia and Singapore as an educative tool, because it teach es the way to develop games. The last reports of these schools show that it is a tool that helps to improve the stories creation, helps the kids solving difficult problems using the “one step at the time” methodology, teach concepts as “Cause-Effect” and shows that software development is a way that kids can demonstrate their creativity[5].

Other framework investigated was Scratch, it was developed in the Massachusetts Institute of Technology, and actually it have been translated about 50
languages and it has more than two millions downloads. It success is contributed to the ease for develop animations, games, online news, books reports, greetings cards, and so on [6].

Looking for a framework, that could be a direct competition of Scratch; we found the Greenfoot frameworks. Greenfoot is educational integrated software for software development’s taught and learns to people without previous knowledge. This framework is designed for an audience of minimum age of 14 years and is very common in university introductory courses [7].

Talking specifically about mobiles’ game development, the most used framework is AndEngine. This framework was launched to public at early 2010 by Nicolas Gramlich, based in the availability of power and engines for the 2D games development supported by Android platform [8].

4. CUBECRAF DEVELOPMENT

We create a framework that allows developing video games for mobile devices with Android or Symbian Operative Systems. The objective of Cubecraft is that elementary school’s students can create games just joining cubes. This section shows our framework proposal.

4.1 System Architecture

The project is divided in two parts, the framework developed in C# .NET and a website developed with PHP4, Ajax and MySQL. We choose a .NET platform because is a set of new technologies that Microsoft have been work in the last years with the objective to make a simple and powerful platform for software distribution. The use of PHP4, Ajax and MySQL are based in the compatibility this technologies has with most of website providers.

The technologies used are shown in figure 1, Cubecraf was developed in C# then it can use source code developed for Android (Java+XML) or Symbian (J2ME).

Fig. 1. Technologies used to develop Cubecraft

4.2 User’s Interface

We used some practices established in HCI (Human-Computer Interaction) patterns. These patterns are “General Graphic User Interfaces”, “2D Graphics User Interfaces”, and “use of metaphor” [9].

From “General Graphic User Interfaces”, we use a Shield practice; this way the user cannot accidentally select a function that has irreversible effects. This practice is used for the error’s administration and is usually implemented as a windows or alert where the user is informed about the execution of these processes. This practice shows to user if a task has finished or it still in execution.
We take from “2D Graphics User Interfaces” two practices: Breadcrumbs and Unidirectional input devices. The breadcrumbs practice offers in the user interface, a navigation track from the start window to the window the user wants, and the Unidirectional Input Devices allows the user use the mouse or the keyboard access to the options and make it use easier.

We took from “Use of Metaphor” the combination of communicative codes as visual and verbal, the objective was to get an optimal redundancy in the process of communication with the user.

Fig. 2. Cubecraft Interface

Taking the good practices last mentioned, we design the user interface. The window on the left, allows the user interact with Cubecraft and to develop their video game (fig. 2). We can see four areas: The Option Menu, the Work Cubes, the Workspace and the User Help Console.

In the Menu area, there are a menu bars and quick access that allow the user make some actions for example: create a new project, open an existing project, save a project, and so on.

The second area is the Work cubes, in this section, we identified two areas: Cubes and My Cubes. In the first area, is shown as a list, here we show the system cubes. In the second area are shown the cubes the user could add to the framework.

The third area is the workspace. In this area the user can connect cubes to make the game’s functionality.

The fourth and last area is the user’s help console; this area is divided in two sections: A User’s functionality help and Error Console. The User’s functionality help updates every time the user add a cube to the workspace, showing the name of the cube and the kind of cube used.

The Error Console, lets the user know the errors o potential problem warnings that do not allow the game to compile or do not perform the actions. Every time an error occurs, the user can identified if it is an error or a warning with the icon showed in the left of the error. For an error, we use a red icon with a cross and for the warnings a yellow icon with an admiration mark.

Cubecraft creates its own workspace in the “Documents” folder. Inside this folder the system creates the files of every project creating a specific folder for everyone.

Inside every project’s folder, there are another two folders and a file with .CUB extension. The folders are “Entregables” and “SRC”, the Entregables folder have the application compiled and ready to be download to a mobile device, the extension of this file is defined by the Operative System selected in the project’s creation.

The SRC folder contains all the classes generated by the framework used to
compile the game. As the Entregables’ folder the extensions of the files inside this folder are defined by the Operative System selected.

The file generated in the project’s root folder contains all the configuration information, this file is detected by the framework when the user wants to open it. For safety and integrity of the data, the user must avoid modifying or deleting it, because it may cause the project not run and loses the job done.

Fig. 3. Functionality cubes

The functionality cubes represent an action which can be executed in a game. We chose Cubes because they are ease to manipulate and generate depth effects using two-dimensional images. The perspective used in the cubes (isometric view), can generate an effect of assembly between two or more cubes (figure 3).

They handle three colors, based on traffic light colors. Those colors were chosen because their use in the daily life of people is very common.

Every cube needs the user sets some parameters to work. Usually the root cubes just needs an identification name, while other cubes need more parameters for complete their setting. The cubes have a button in the front with the label “Configurar”. When this button is pressed, shows a window that read the parameters specified in the function’s DLL and configures the parameter’s fields based in the type of data received.

4.3 Website

The website is designed as a way where users can upload their games and discuss their experiences with others developers. In the website, inexperienced users can find important information as user’s manual, API’s documentation, and so on.

As a safety measure, all the actions the user can do in this website, including the use of the Online Score System, must be done as a register user. The user can create their account using a registration form or using a social network as Facebook, Twitter or Google+.

5. FUTURE WORK

As future work, we are going to test the system with students in ages between 8 to 12 years in different schools of the city, applying a survey that could give us a feedback about the usability and acceptance. Also, the survey is going to help us in search of new requirements and functions the final users want to have in their game’s development. After this first survey we are going to modify our framework and will be to apply one more time our survey. This second survey will give new feedback about the viability of the system and if the modifications we did based in the first survey get better the system or we have to update something.

Once the framework fulfill all the requirements and modifications, We are going to distribute Cubecraft game.
6. CONCLUSIONS

Although there is an important quantity of frameworks for the PC or Mobile app development, we want to build one to specific requirements or our Mexican kids. This framework could have a high impact in the way the professors taught the principles of software development to their students.

Regarding the kids skills, this framework could be an extra tool for their parents, teachers and them to improve unconsciously their skills that can be used in the school or later in their jobs.

REFERENCES


