Suggestions for Software projects using RAD equipped with fewer resources.

[1]Madan Kapoor, [2]Manit Kapoor, [3] S.Gunasundari
[1, 2] B.E. Student, Department of Computer Science, and engineering, Velammal Engineering College, Chennai
[3]Asst Professor, Department of Computer Science, and engineering, Velammal Engineering College, Chennai
madankapoor10@gmail.com,manitkapoor2@gmail.com,gunapondyuniv@gmail.com

***Abstract*—In software development, sometimes projects started in the Rapid Application Development model have the resources necessary but if the resource is cut down due to various managerial decisions leading to problems. Some suggestion of techniques to deal with projects which are developed in RAD software model with fewer resources is described below with an example project. In the example project, development of a efficient word game and a number game using the RAD process is depicted and the suggestions related to tackling problems are provided to scale to industry level with keen consideration. The C programming language and a GUI toolkit is used to develop the project using RAD tools**

Keywords—Software development; resources; suggestion; RAD; Gtk; Mindstate;

I. Introduction

Let us consider a scenario where the development of a project takes place using RAD process. Now we already know that software development begins with some initial analysis for the process selection and it continues with requirement analysis then followed by series of steps. It is known that some process models like RAD require a lot of resources and are applicable only to low-risk projects. What if we begin a project with RAD but later the resources are cut down and the risks become moderate.The budget of the entire software process may go high and even projects may fail.By cutting down on resources mostly refers to the situation when people with less technical knowledge are working on the project and person month count is also reduced. Bringing forth an example project we will discuss the issues and provide suggestions to undergo a successful RAD process. Consider a project being developed by a few students with basic knowledge of C programming and computers. This project will make things easy to understand the suggested approaches without getting into dept of programming or software's requirements.

II. Example Project

Cows and Bulls is an old code-breaking mind or paper and pencil game. It is a game with numbers or words that may date back a century or more. This game helps us to increase our thinking capabilities and glossary. This game is developed for windows using GIMP toolkit using C language. GUI is also improved using CSS. Words are taken from www.morewords.com. The Game exists in word and number version. This application is created in both number and word. The project details are discussed here.

III. GTK+

GTK+ is a multi-platform toolkit for creating graphical user interfaces. It is also called as GIMP Toolkit. Offering a complete set of widgets; GTK+ is suitable for projects ranging from small one-off tools to complete application suites. GTK+ is written in C but has been designed from the ground up to support a wide range of languages, not only C/C++. Using GTK+ from languages such as Perl and Python (especially in combination with the Glade GUI builder) provides an effective method of rapid application development.

IV. Gtk+ Architecture

GTK+ (previously GIMP Toolkit, sometimes incorrectly referred to as the GNOME Toolkit) is a cross-platform widget toolkit for creating graphical user interfaces. It is licensed under the terms of the LGPL, allowing both free and proprietary software to use it. It is one of the most popular toolkits for the Wayland and X11 windowing systems, along with Qt The GTK+ library contains a set of graphical control elements (widgets); version 3.13.3 contains 203 active and 37 deprecated widgets. GTK+ is an object-oriented widget toolkit written in the C programming language; it uses GObject, which is the gLib object system, for the object orientation. While GTK+ is primarily targeted at windowing systems based upon X11 and Wayland, it works on other platforms, including Microsoft Windows (interfaced with the Windows API), and Mac OS X (interfaced with Quartz). There is also an HTML5 back-end called Broadway. GTK+ can be configured to change the look of the widgets drawn; this is done using different display engines. Several display engines exist which try to emulate the look of the native widgets on the platform in use. Starting with version 2.8, released in 2005, GTK+ began the transition to using Cairo to render the majority of its graphical control elements. Since GTK+ version 3.0, all the rendering is done using Cairo.

1) GUI designers

There are several GUI designers for GTK+. The following projects are active as of July 2011:

- Glade supports GtkBuilder, which is a GTK+ built-in GUI description format.
- Gazpacho, GUI builder for the GTK+ toolkit written in Python[12]
- Crow Designer, relies on its own GuiXml format and GuiLoader library.[13]
- Static, part of MonoDevelop, oriented towards Gtk#.

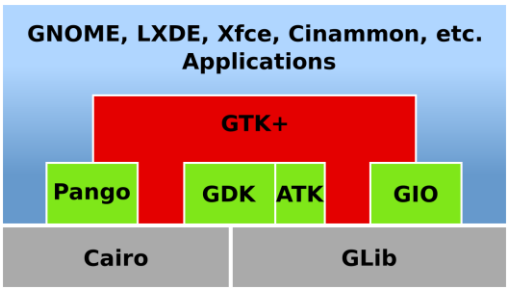
a) GtkBuilder

Gtk Builder offers you the opportunity to design user interfaces without writing a single line of code. This is possible through describing the interface by an XML file and then loading the XML description at runtime and creates the objects automatically, which the Builder class does for you. For the purpose of not needing to write the XML manually, the Glade Interface Designer lets you create the user interface in a WYSIWYG manner.

This method has several advantages:

- Less code needs to be written.
- UI changes can be seen more quickly, so UIs are able to improve.
- Designers without programming skills can create and edit UIs.
- The description of the user interface is independent of the programming language being used.

There is still code required for handling interface changes triggered by the user, but Gtk.Builder allows you to focus on implementing that functionality.



V. GTK+ FEATURES

a) *Stability*

GTK+ has been developed for over a decade to be able to deliver the enticing features and superb performance that it brings to your application development. GTK+ is supported by a large community of developers and has core maintainers from companies such as Red Hat, Novell, Lanedo, Codethink, Endless Mobile and Intel.

b) *Language Bindings*

GTK+ is available in many other programming languages thanks to the language bindings available. This makes GTK+ quite an attractive toolkit for application development.

c) *Interfaces*

GTK+ has a comprehensive collection of core widgets and interfaces for use in your application.

- Windows (normal window or dialog, about and assistant dialogs)
- Displays (label, image, progress bar, status bar)
- Buttons and toggles (check buttons, radio buttons, toggle buttons and link buttons)
- Numerical (horizontal or vertical scales and spin buttons) and text data entry (with or without completion)
- Multi-line text editor
- Tree, list and icon grid viewer (with customizable renderers and model/view separation)
- Combo box (with or without an entry)
- Menus (with images, radio buttons and check items)
- Toolbars (with radio buttons, toggle buttons and menu buttons)
- GtkBuilder (creates your user interface from XML)
- Selectors (color selection, file chooser, font selection)
- Layouts (tabulated widget, table widget, expander widget, frames, separators and more)
- Status icon (notification area on Linux, tray icon on Windows)
- Printing widgets
- Recently used documents (menu, dialog and manager)
- Cross Platform

Originally GTK+ was developed for the X Window System but it has grown over the years to include backend support for other well-known windowing systems. Today you can use GTK+ on:

- GNU/Linux and Unix
- Windows (32-bit) and 64-bit
- Mac OS X
- Accommodating
- GTK+ caters for a number features that today's developers are looking for in a toolkit including:
- Native look and feel
- Theme support
- Thread safety
- Object oriented approach
- Internationalization
- Localization
- Accessibility
- Bidirectional text support (LTR/RTL, Left To Right/Right To Left)
- UTF8 support
- Documentation

- Foundations

GTK+ is built on top of Glib. Glib provides the fundamental algorithmic language constructs commonly duplicated in applications. This library has features such as: (this list is not a comprehensive list)

- Object and type system
- Main loop
- Dynamic loading of modules (i.e. plug-ins)
- Thread support
- Timer support
- Memory allocator
- Threaded Queues (synchronous and asynchronous)
- Lists (singly linked, doubly linked, double ended)
- Hash tables
- Arrays
- Trees (N-ary and binary balanced)
- String utilities and charset handling
- Lexical scanner and XML parser
- Base64 (encoding & decoding)
- Mobile

The GMAE (GNOME Mobile & Embedded) initiative has advanced the use, development and commercialization of GNOME components as a mobile and embedded user experience platform. It has brought together industry leaders, expert consultants, key developers and the community and industry organizations they represent. As a direct result of this, GTK+ has features pertaining to mobile and embedded platform requirements.

GTK+ has been involved in a number of embedded initiatives over the past few years including the development of:

- Nokia 770 / N800 / N810 / N900
- One Laptop Per Child Project
- OpenMoko

VI. GAME DETAILS

How to play?-Word Version

1. Computer will generate a randomized 4 letter word, which will be meaning full
2. The random word selected will have all letters unique.
3. You have to try to Guess that Word within 15 attempts.
4. For Each Try, a window will display the number of tries remaining. It will also show at the left side number of cows and bulls.
5. Cows: Refers to a number of alphabets in your guessed word which are in right place in the computer generated a word.
6. Bulls: Refers to a number of alphabets in your guessed word which are present in the computer generated a word, but which are not in the right place.
7. All your attempts are displayed with a number of cows and bulls for that attempt. Try to use them for analysis.
8. Remember that each letter is unique.

VII. HOW TO PLAY?-NUMBER VERSION

1. The computer will generate a randomized 4 digit number with all digits unique.
2. You have to try to guess that number within 15 attempts.
3. For Each Try, a window will display the number of tries remaining.
4. It will also show at the left side number of cows and bulls.
5. Cows: Refers to a number of digits in your guessed number which is in the right place in the computer generated number.
6. Bulls: Refers to a number of digits in your guessed word which is present in the computer generated a number.
7. All your attempts are displayed with a number of cows and bulls for that attempt. Try to use them for analysis.
8. Remember that each digit is unique**.**

VIII. REQUIREMENTS:

Recommended Requirements

Operating System	Windows(any)
Hard Disk Space	86 Mb
Free Hard Disk Space	500 Mb
RAM	128 Mb

IX. LIMITATIONS

- Only 4 letter words are used.
- Some of the words are quite difficult to find out.
- Words from other languages which are now the part of English are also included which makes users difficult to guess.

X. Implementation AND RESULT:

a) ***UI Implementation:*** GUI is implemented using a GTK+ design software glade. Glade supports Gtk+

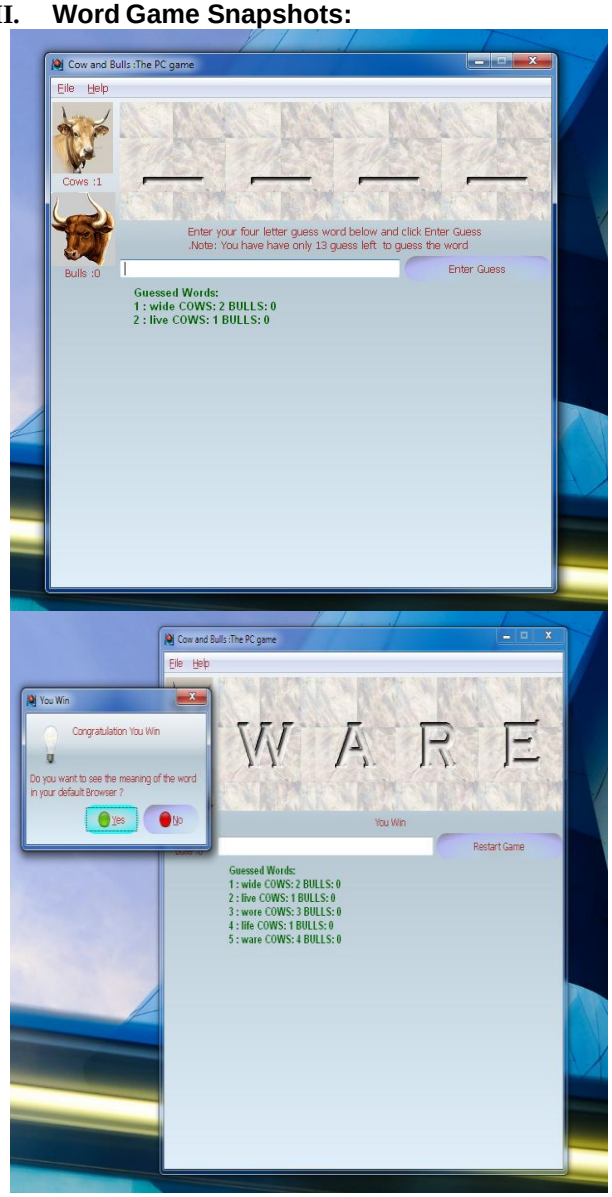
Builder, which is a GTK+ built-in GUI description format. Special effects for button and background are given using CSS. Images for letters are taken from coolart.com. Also, the menu bar is implemented to provide information about the application and instructions to play. Various internal functions are connected using features provided by glade.

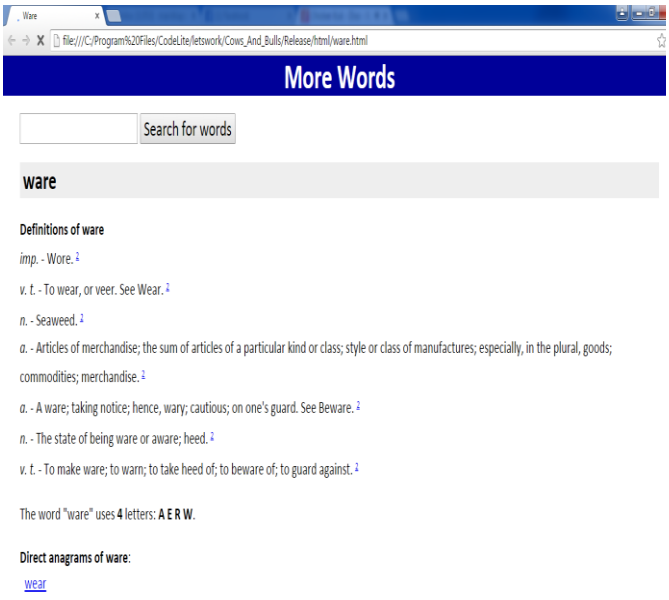
b) Internal *Implementation:* Internal implementation is done through C language using codelite IDE. The gtk+ environment, which is integrated with codelite, is used to connect GUI and internal implementation. Shell commands are also used to access various internal files. An algorithm for selection of random words is implemented. A physical data model is used for the management of words. An Indexed system is used to access words. Various constraints are also coded using C language. The C compiler used is MinGW C compiler**.**

XI. Suggestions followed while developing

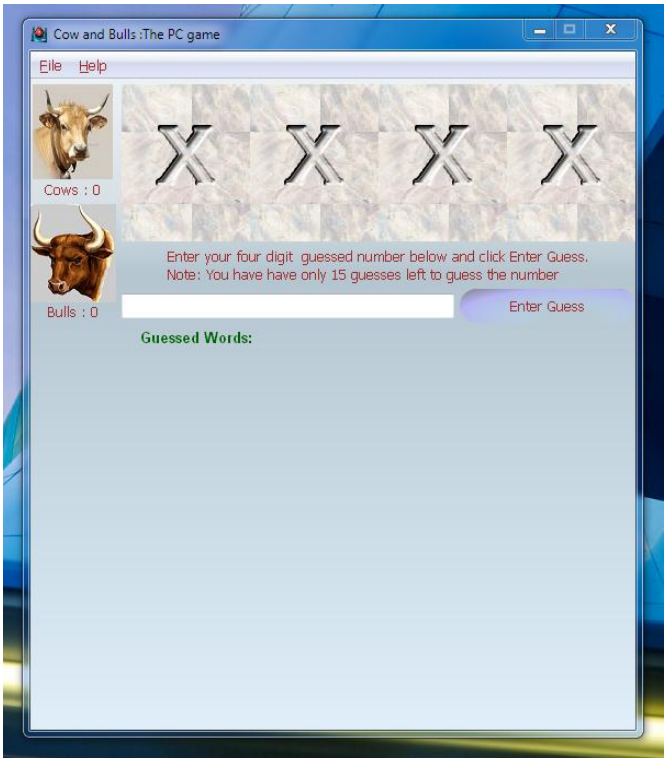
While development it is very handy to use mind states to tackle problems and find around the clock solutions, it is already known that mind has two states focused and diffuse modes. A person in diffused mode can tackle problems easily and find round the clock solutions. To bring the change in the mind state, open and free communication helped students to tackle problems and identify the equirements. Development of Open source software and open source tools have supported portability and quick project creation thus when resource are less open source software help a lot. The probability that students stay in that mode as enhanced by short-term goals and motivation theoretically. Adaptability to new tools requires good IQ and quickness in mind so students were trained in IQ before getting into the development process.Exchange of people in the parallel process to work quicker to tackle problems, initially understanding the problems is difficult but when the flow ideas start there is a boost in productivity.GUI design was carried out with background analysis of the stakeholders mind and the software currently used software. Most people used windows XP and windows 7 in bluish theme. Use of bluish gradients related to weather gave a good GUI to the software. The above Ideas can be scaled and applied to many industry projects.

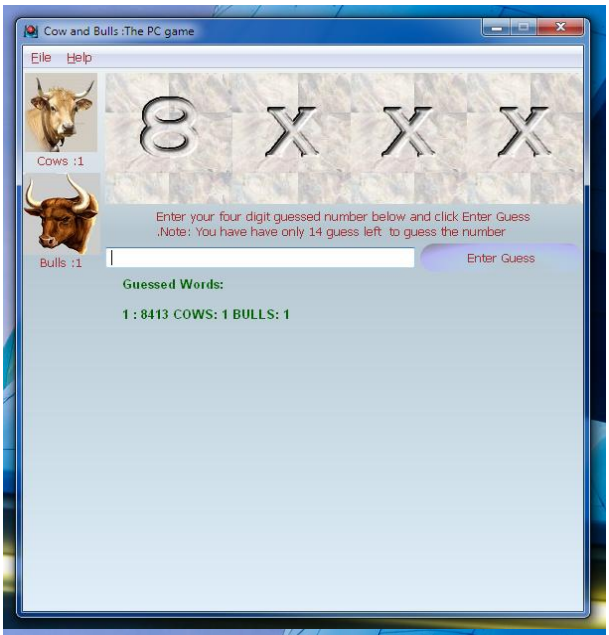
XII. Word Game Snapshots:

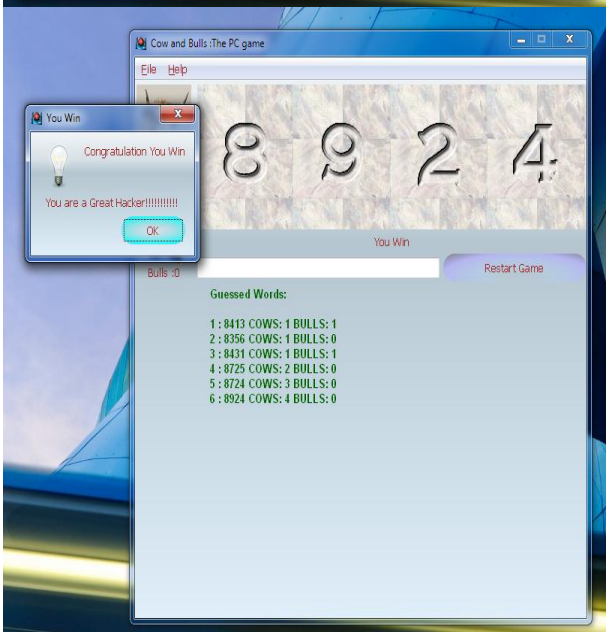




XIII. NUMBER GAME SNAPSHOTS:







XIV. REFERENCES

[1] GTK+ 3 Reference Manual - The GNOME Development Site https://developer.gnome.org/**gtk3**/**3**.6.5/, www.gtk.org/

[2] https://www.coursera.org/learn/learning-how-to-learn/lecture/75EsZ/introduction-to-the-focused-and-diffuse-modes

[3] https://developer.gnome.org/**gtk3**/

[4] GTK+ - Wikipedia, the free encyclopedia *https://en.wikipedia.org/wiki/**GTK**%2B*

[5] User interface design - Wikipedia, the free encyclopedia *https://en.wikipedia.org/wiki/**User_interface_design***

[6] Dynamic-Link Libraries (Windows) - MSDN - Microsoft

https://msdn.microsoft.com/en-us/library/windows/desktop/ms682589(v=vs.85).aspx
http://community.wikidot.com/howto:design-your-own-css-theme

[7] Windows Installer (Windows) - MSDN - Microsoft
https://msdn.microsoft.com/en-us/library/cc185688(v=vs.85).aspx