**Design Specification – Jbrick**

**Release 2.0**

**Version: 2.0**

**Prepared by: Group 1**

**Date: February 16, 2010**

**Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Name** | **Version** | **Comment** | **Section** |
| **11-12-2009** | **Spencer & Priya** | **1.0** | **Initial Version** | **All** |
| **2-16-10** | **All** | **2.0** | **End of Q2** | **All** |
|  |  |  |  |  |

Table of Contents

[Introduction 3](#_Toc245814205)

[Background 3](#_Toc245814206)

[System Overview 4](#_Toc245814207)

[Design Considerations 4](#_Toc245814208)

[Assumptions and Dependencies 4](#_Toc245814209)

[General Constraints 4](#_Toc245814210)

[Goals and Guidelines 4](#_Toc245814211)

[Development Methods 5](#_Toc245814212)

[Architectural Strategies 5](#_Toc245814213)

[System Architecture 5](#_Toc245814214)

[Policies and Tactics 5](#_Toc245814215)

[Detailed System Design 6](#_Toc245814216)

[Bibliography 7](#_Toc245814217)

# Introduction

This document elaborates on the different architecture and design specification for JBrick editor. JBrick editor is the application that is being built as a part of Software Practicum I course to assist in the development of .nxc programs for Lego Mindstorm.

The scope of the document is to give details on the design of the functionalities that are included as a part of Iterations in this Fall 09 quarter.

This document is intended to serve as a guide for individuals to understand in depth about how the editor has been constructed so as to add more functionality into it.

Information related to the features included as a part of the application under consideration could be obtained from CoreFeatures.xls and the SRS documents.

# Background

This application is built so as to replace the existing Bricx Command Center for development of .nxc programs for Lego Mindstorm robots. The aim is to incorporate the relevant features from the existing application and further improve its performance with respect to certain limitations identified as a part of the initial analysis (listed as a part of the core features for JBrick editor in the CoreFeatures.xls document). Here is the general summary of the core features of JBrick:

|  |  |
| --- | --- |
| **Name** | **Description** |
| Graphical User Interface | A clean, user friendly interface is a must for our application. |
| Editor | This provides the basic functionality for saving, opening, line numbering, syntax highlighting. |
| Compiler | The compiler was necessary to compile the code created in the editor. The compiler also provides the results of the compiling. Errors provide a line number and simple explanation of what happened. |
| Communications | Communicating with the Lego Mindstorms brick is a key feature to implement in iteration one. This feature will allow for but not limited to, transmitting and compiling program, telling the brick to run a program, and retrieving information from the brick. |
| Short Cut Keys | Short Cut Keys are one of the accessibility considerations that we have taken an active role in implementing. Our implementation of this strives to have the optimal short cuts in place to provide an intuitive user experience. |

# System Overview

The JBrick Editor is a Java GUI based application constructed with the aim to support the operation of Lego Mindstorm robot.

Parameters that were considered while deciding to build it using Java include the following

* Language should be platform independent
* Support accessibility
* Resources in the team had the technical knowledge required for building this application

The technology involved includes, JFace SWT for GUI built on a basic Java application project.

# Design Considerations

## Assumptions and Dependencies

In order to finalize on the design, we had to consider the following items

1. Accessibility – the GUI should be helpful for all types of users
2. Cross platform – Windows, Linux, Mac OS

## General Constraints

Limited knowledge on the technology with which BricxCC application was built. As it is written in an old version of Delphi and the code itself was not documented well, general code reuse was not an option.

## Goals and Guidelines

Our aim was to build an application using a high level programming language that would be platform independent and at the same time cater to all the functionality that are a part of the existing application.

## Development Methods

The development methodology followed is agile; we are more or less doing incremental development. As and when we understand more about the existing features and get the requirements clarified by the client we update the application.

# Architectural Strategies

* Reuse compilers
* Download and run functionality
* Future plans for extending the software – iteration 2 and 3
* Editor communicates with the robot both through USB and Bluetooth. Current implementation is tested for USB and Bluetooth
* Overall, the application needs to mimic BricxCC's functionality but in a more efficient and user friendly way.

# System Architecture

The application has been constructed based on the MVC pattern. SWT JFace APIs used gives us the compatibility required.

# Policies and Tactics

One way that we are attacking this large task is that we reusing the Next Byte Codes (NBC) compiler. As writing a compiler would take an inordinate amount of time, and the NBC compiler is already written for Windows, Linux, and MacOS.

We will also take advantage of the NeXtTool application that comes packaged with BricxCC. We have also chosen to use the Fantom library provided by LEGO. It is PC and Mac compatible and allows us to communicate with the Lego brick via USB and Bluetooth. The tool provides the functionality to start and stop programs, view system files, and directly interface with the bricks sensors and output motors. Currently, we have only developed and tested in the Windows environment. However, future versions and iterations of JBrick will test the Mac operating system.

# Detailed System Design

The application has been structured in such a way that all the operations required as a part of the editor are defined by extending the action class of the JFace API.

The flow is as follows:

# 

# Bibliography

1. <http://bricxcc.sourceforge.net/>
2. <http://lejos.sourceforge.net/nxj-downloads.php>
3. <http://lejos.sourceforge.net/nxt/nxj/tutorial/index.htm>
4. <http://www.mathworks.se/matlabcentral/fx_files/18646/2/content/RWTHMindstormsNXT/doc/communication/communication.html>
5. <http://mindstorms.lego.com/Overview/NXTreme.aspx>