Project Proposal

B.Kendrick, A.Syed

Contents

[Introduction 3](#_Toc415768163)

[Goals / Requirements 3](#_Toc415768164)

[Must Have 3](#_Toc415768165)

[Should Have 3](#_Toc415768166)

[Could Have 4](#_Toc415768167)

[Clients & End Users 4](#_Toc415768168)

[Research Topics 4](#_Toc415768169)

[Risk Assessment 4](#_Toc415768170)

[Methodology 5](#_Toc415768171)

[Resources 5](#_Toc415768172)

[Repository (GitHub) 5](#_Toc415768173)

[Eclipse Framework 5](#_Toc415768174)

[BlueJ 5](#_Toc415768175)

[E-BOB 5](#_Toc415768176)

[Project Plan/Time Management 5](#_Toc415768177)

Creating an Eclipse Plugin to Mimic the “Useful” Features of BlueJ

# Introduction

It is known that educating an individual in object oriented programming techniques can be daunting and ultimately challenging task. Whilst there are a variety of techniques to aid in teaching programming there are few software aides available at this time, especially in bridging the gap between beginner and intermediary skill sets. BlueJ is an Integrated Development Environment (**IDE**) that aims to provide a pedagogical environment for programming, especially in an object oriented manner. BlueJ provides a variety of features that are designed to assist a new programmer in understanding the code they produce, features such as: a workbench to instantiate and view objects in real-time, an in-depth object inspector so that the states of variables can be viewed whilst the program is running and a class diagram to view created classes and have a constant visual aid in understanding how classes work together. BlueJ however lacks some functionality and extensibility of other IDEs such as Eclipse and is generally not used often in commercial development as it lacks some of the key functionality required; the most critical of these is a dedicated and sophisticated debugging tool.

This project aims to provide potential students and teachers with the tools found in BlueJ that are beneficial to learning in a more standard environment. We aim to take the most useful features and concepts from BlueJ and re-define them to better serve the education of programming, specifically in an object oriented manner. This project will therefore develop an Eclipse plugin using the Eclipse framework which mimics some of the more useful features of the BlueJ IDE.

# Goals / Requirements

## Must Have

* **Class Diagram –** The plugin will draw a class diagram showing the relation between the various classes.
* **Workbench –** A workbench is required to instantiate instances of individual classes and makes calls to their methods as required.
* **Inspector Functionality –** There must be some form of inspector utility that enables to monitor the variables within a class.

## Should Have

* **Enhanced Inspector Functionality –** As well as being able to view the values of variables it should also be possible to change them as the program is running to see the results.
* **Class Templates –** The user should be able to create a class based on a template. (e.g. *Class, Abstract, Interface, Enum etc.)*
* **Package Management –** Separate class diagrams for individual packages, these should be navigable.
* **Easy to Use UI –** The UI should be simple to use, intuitive and familiar to students learning java through BlueJ.
* **High Performance –** The plugin should be responsive and quick with minimal delay times for actions.

## Could Have

* **Scope Highlighting** – The ability to highlight the background of each code block depending on its depth within the code. For example an if method would be surrounded with its own colour for easier code readability and visual scanning.
* **Syntax Error Highlighting** – Whilst Eclipse has basic syntax highlighting it would be useful to better explain mistakes made by potential students and perhaps have definable colour editing per error to make specific, but basic, errors easier to find.

# Clients & End Users

The clients of the system are to be lecturers, this plugin has to make it easier for them to bridge the gap between basic OO programming and more advanced concepts, specifically when going from an IDE designed to be for beginners to a more professional IDE designed for commercial use.

The end users of the system are intended to be students learning more advanced programming concepts. The system will need to be designed in a manner that is intuitive and straight forward. The user will theoretically be coming from using BlueJ so the interface should be familiar to them.

# Research Topics

Throughout the development of this project a variety of research will be undertaken to better understand the requirements of the tool and to help compare our solution to ones that exist.

The topics we shall research are:

* Software for teaching programming
* Eclipse plug-in framework
* Methods of teaching programming
* Visualising code for learning
* UML standards?

# Risk Assessment

In this section we will quickly outline some risks and the plan to deal with them.

Initially we will need to learn the eclipse framework for creating a plugin, the potential risk here is that we may end up not completely understand the framework to create complex functionality to accomplish our requirements. This could result in delaying of reaching milestones and therefore the plan to deal with this would be to get accomplish ‘must’ and as many as possible ‘should’ requirements which may not involve complex involvement of the eclipse framework.

Another risk is under expectation of workloads from other modules; this again could lead to delaying our milestones of the project. The plan to deal with this would be the same as above.

Illness, family matter and extenuating circumstances are another risk, these again would affect our ability to hit deadlines and accomplish all the requirements. In these cases we would communicate with our supervisor for an appropriate solution.

# Methodology

It was decided that we would follow the Waterfall model in the lifecycle of our project; this has helped us create a rough plan for the year. We decided to go for the waterfall model as it is only two of us and we believed it would be easier to manage the project with this model, rather than going for a strict prototyping methodology such as RAD. There will however be some form of prototyping within certain sections of development; this is where we will be using the reoccurring waterfall model.

# Resources

## Repository (GitHub)

We will be making use of GitHub to store our project and code, this will enable easy sharing of files and allow us to ensure the project is backed up and secure.

## Eclipse Framework

This will be used to develop the plugin. We will need to find a variety of documentation and literature around the subject to enable us to efficiently learn and make use of the framework.

## BlueJ

This will be used as a start point, to get an idea what is required.

## E-BOB

E-BOB is a similar plugin for eclipse that brings the workbench across, it will be worth researching this software to get a good starting point for the project.

# Project Plan/Time Management

Gantt chart attached as separate document.