## **Describe and use common Design Patterns, Algorithms and programming language Idioms.**

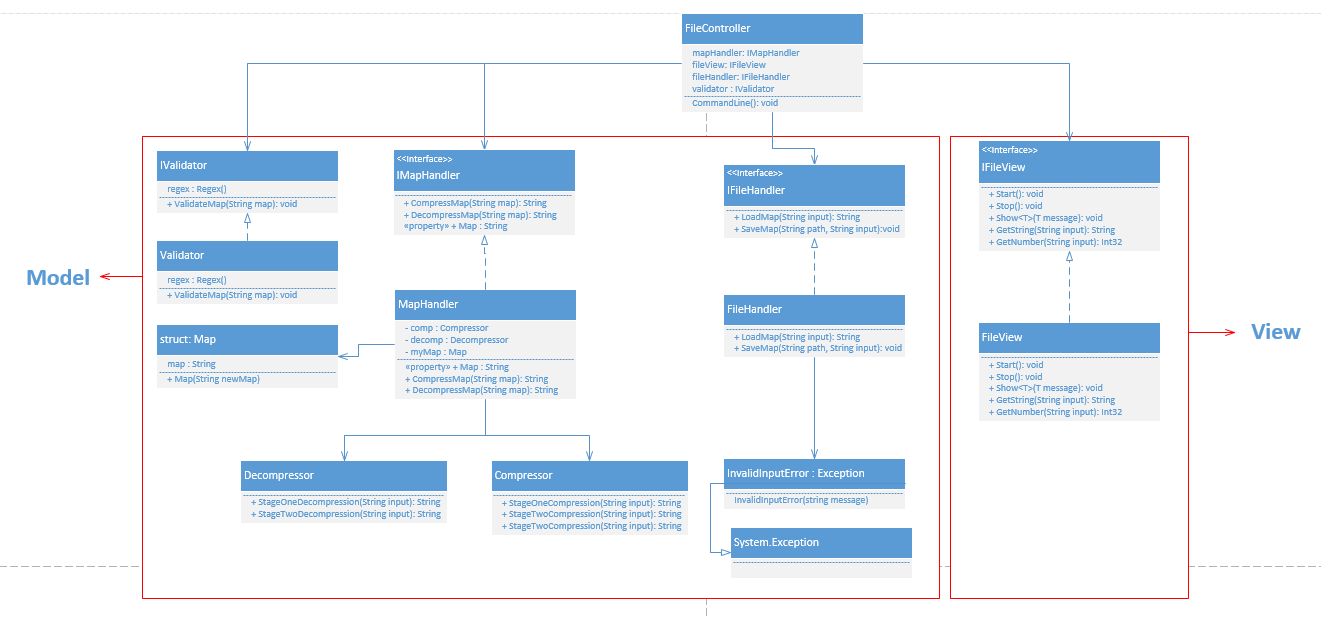
### Design patterns

According to (Bautista, n.d.)

* + - Design patterns are optimized and reusable solutions to common problems that developers often deal with. If used correctly it can be very effective but if used in wrong situation it could be very divesting (double edge sword, swing it to the right direction).
    - It consist of 3 types of design patterns. Structural creational, and behavioural.
    - Structural - the relationships of entities to make them work together.
    - Creational - how objects are created for more effective process.
    - Behavioural -how communications between entities to create simple and more flexible usability.

### Use Design Pattern

* + - I implemented the model, view, and controller.
    - This makes my code easier to understand and maintain.
    - Decouples objects.
    - In my application I reused multiple codes (classes, methods) to provide the same solutions to the same problems.



### **Algorithms and Programming language Idioms.**

(UNM, n.d.) suggested that Idioms are a form and structure for knowledge that helps us bridge the differences between patterns as abstract descriptions of a problem and its solutions and an understanding of how best to implement that solution in a given programming language. A language idiom is the expression of a design pattern in a given language. In this sense, design patterns + idioms = quality programs.

## **Use appropriate software life-cycle models and software construction steps.**

* (tutorialspoint, 2015) Suggested that SDLC, Software Development Life Cycle is a process used by software industry to design, develop and test high quality softwares. The SDLC aims to produce a high quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.
* The SDLC model that I used in this project is the waterfall model. I used waterfall model because the project is small, requirements are not complex and design is simple.
* The main process are:
  + - **Requirements Analysis**- Here I made a list of features that the application needs for the filer, and the Moscow that the other group posted to the Moodle for Level Designer and Game. Other documentations such as Use Cases, pseudocode.
      * + Sample of requirements I have done:

Read a text file

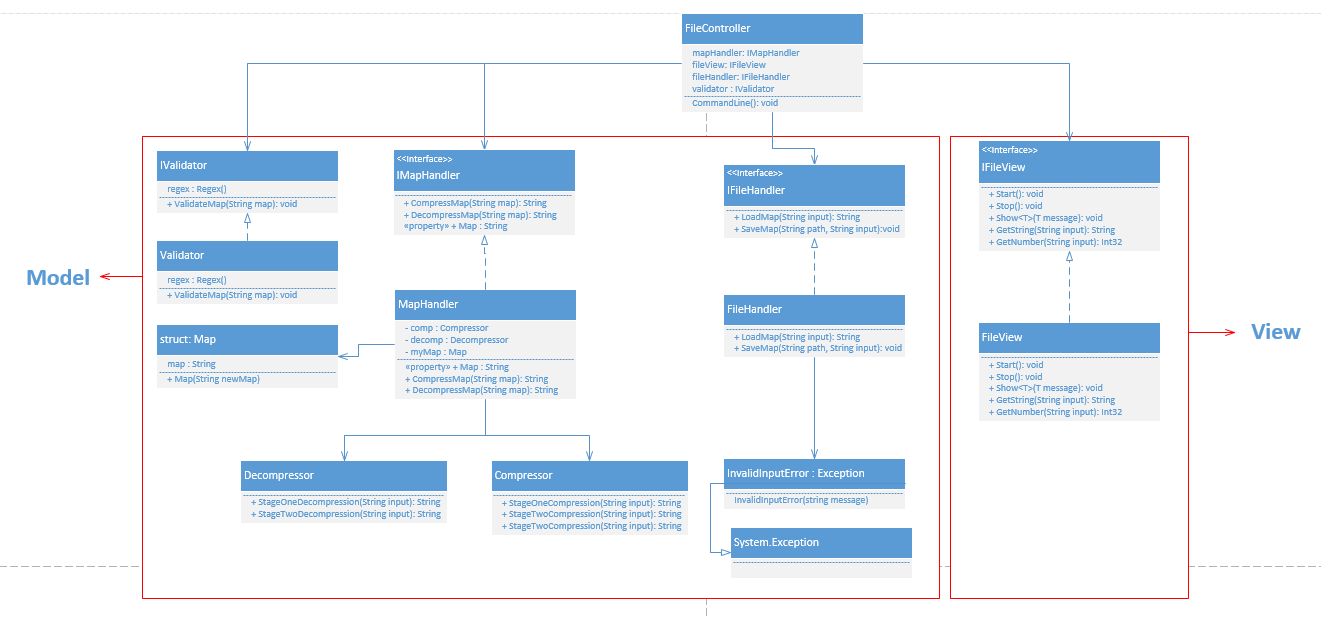
Save a text file

Compression

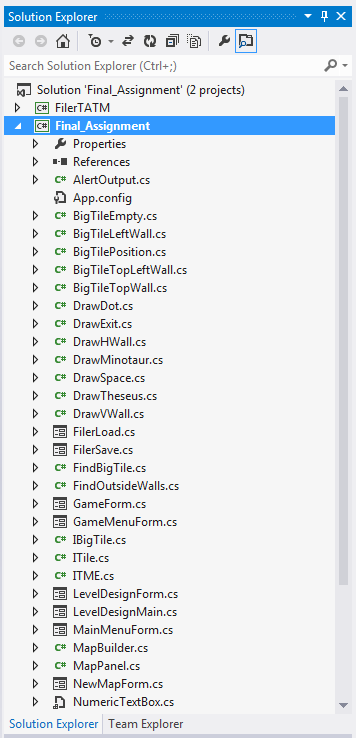
Expansion

Exception handling

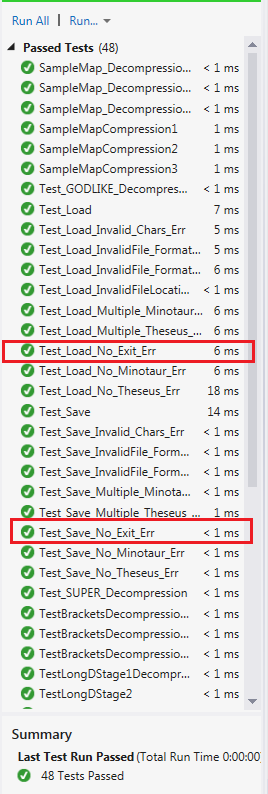
* + - **Design**- here I made a design level diagram to create a picture of what needs to be done, activity diagram, structure diagram.
      * + Sample of diagrams that I have done:



* + - **Implementation-** here I implemented the design and build the game in C# using the visual studio IDE. Game, Level Designer and Filer.



* + - **Testing –** The method I used to test my programs is unit testing.
      * + Sample of unit testing I have done:



Using waterfall approach on my project is the best approach compared to others because the project is small, requirements are clear and simple. If I used other model such as the iterative and incremental approach, then it require more unnecessary work.

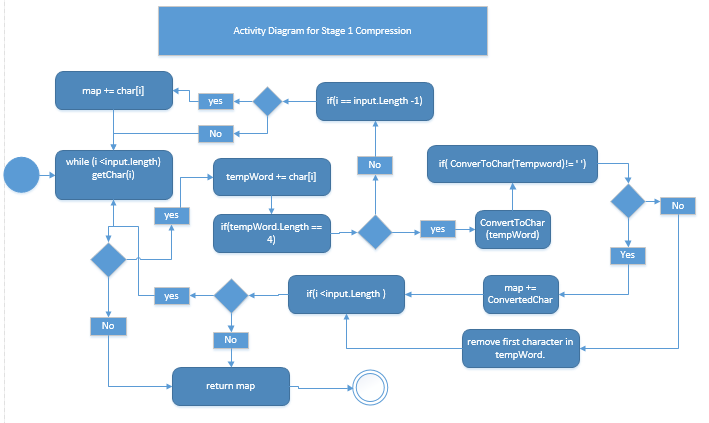
This approach is effective because it requires less management works, less time, and simple to implement.

I could have improved by creating more documentations, manage time more effectively.

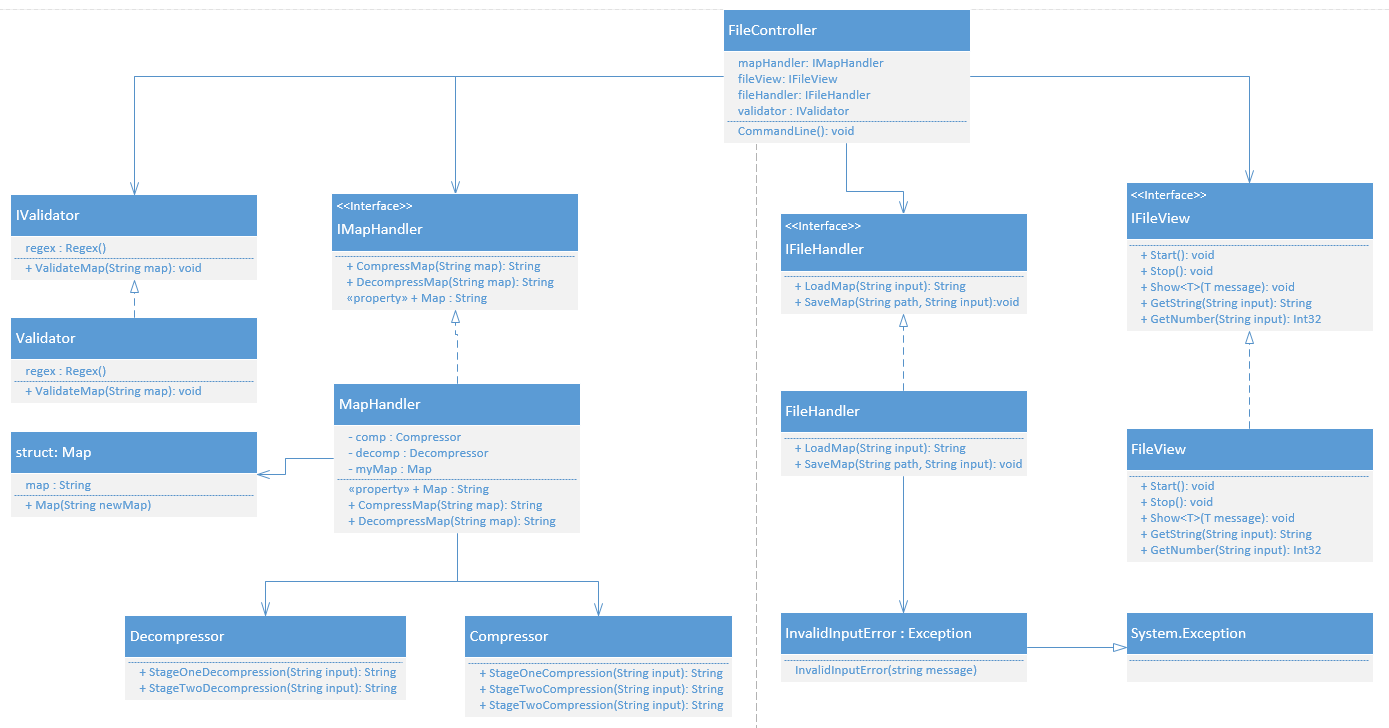
## **Design programs.**

(Garner, 2012) Suggested that in design, you need to think about the algorithms you will use, the program design, record design time in the time recording log. UML modelling is performed which is called the Unified modelling language.

### **Activity Diagram**



### **Design Level Diagram**



## **Design user interfaces which conform to recognised usability criteria.**

* According to (usability.gov, 2015), User Interface (UI) Design focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate those actions. UI brings together concepts from interaction design, visual design, and information architecture.
  + **Suitability for the Task (appropriate functionality)**

The system should meet the needs and requirements of users when carrying out tasks.

* + **User Control (controllability, user explicit control)**

The system should be designed to let users control the interface as much as possible.

* + **Flexibility (suitability for individualization, adaptability)**

The interface should be sufficiently flexible in structure, in the way information is presented and in terms of what the user can do, to suit the needs and requirements of all users.

* + **Error Management (error tolerance, error prevention and correction)**

The system should be designed to minimize the possibility of user errors, with built-in facilities for detecting and handling those which do occur.

* + **Compatibility (conformity with user expectations, natural and intuitive)**

The way the system looks and works should be compatible with user conventions and expectations.

* + **Self-descriptiveness (information feedback, user guidance and support)**

Information feedback, guidance and support should be provided to help the user understand and use the system.

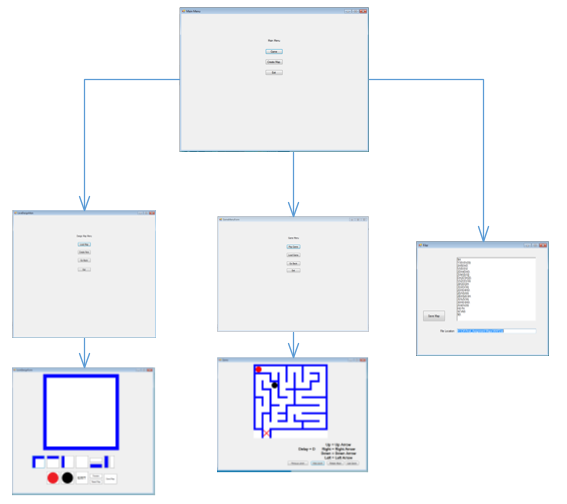
* + **Consistency (consistency in location, format, syntax, and naming)**

The way the system looks and works should be consistent at all times.

* + **User Workload (minimal memorization, brevity, mental load)**

The system should be designed to keep the user's mental load (particularly memory load) within acceptable limits and to increase the speed of interaction by increasing brevity.

### **What I implemented.**



* I could have added more features to make my program to be more **error tolerance, error prevention and correction** by implementing an error handler for the game when moving against a wall, should have some trigger which alerts the user that you can’t go that way.

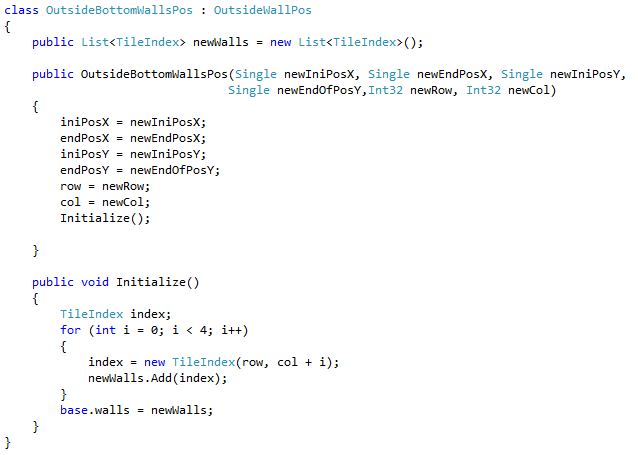
## **Code programs in the specified language using the prescribed standards.**

(Mytton, 2004) Suggested that a coding standards document tells developers how they must write their code. Instead of each developer coding in their own preferred style, they will write all code to the standards outlined in the document. This makes sure that a large project is coded in a consistent style — parts are not written differently by different programmers. Not only does this solution make the code easier to understand, it also ensures that any developer who looks at the code will know what to expect throughout the entire application.

In my application I followed JSlint coding standard. Example of coding standards below (my own code).

**THESE ARE MY OWN CODE**

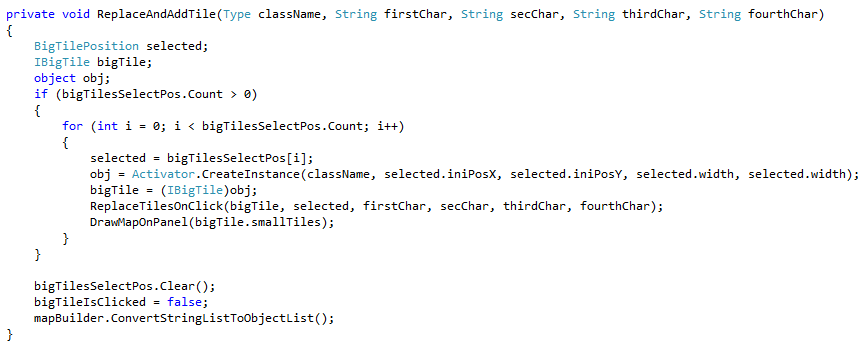
**Code Example 1**



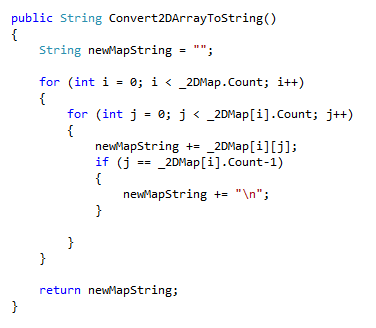
These codes coding standard are eff **Code Example 1**ective because the variables names are related to what they do, indentation are stated and clear, easy to read because of good spacing.

These codes coding standard are eff **Code Example 1**ective because the variables names are related to what they do, indentation are stated and clear, easy to read because of good spacing.

### **Code Example 2(These are my own code)**



### **Code Example 3(These are my own code)**

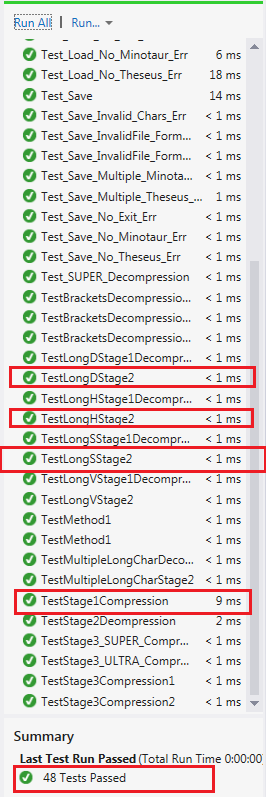
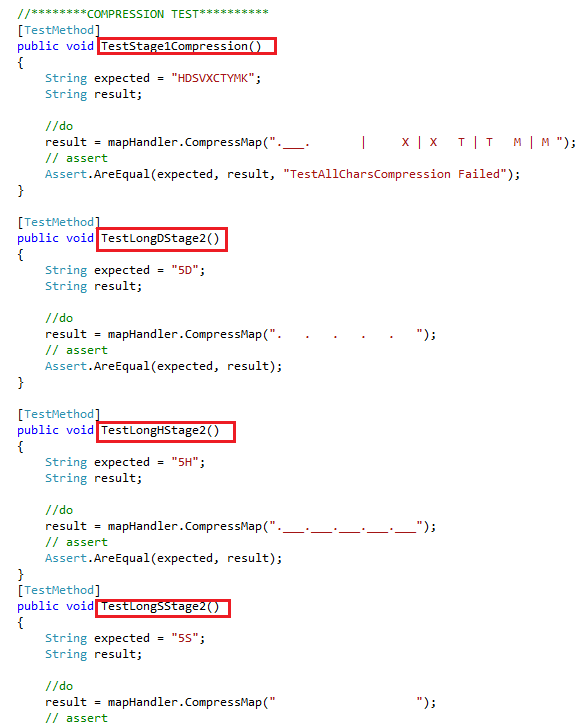


Improvements:

I could have added more comments on the code to make it easier to maintain.

## **Produce and execute testing strategies at the systems level using a unit testing framework.**

According to (tutorialspoint, 2015) Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

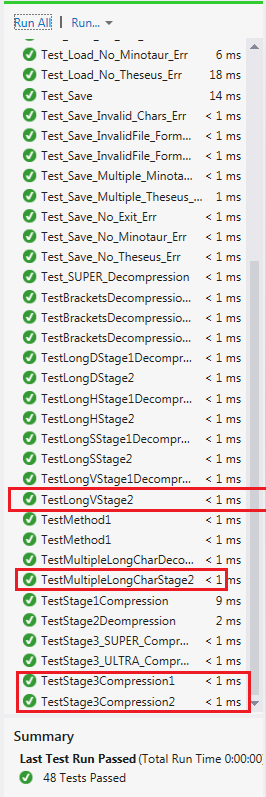
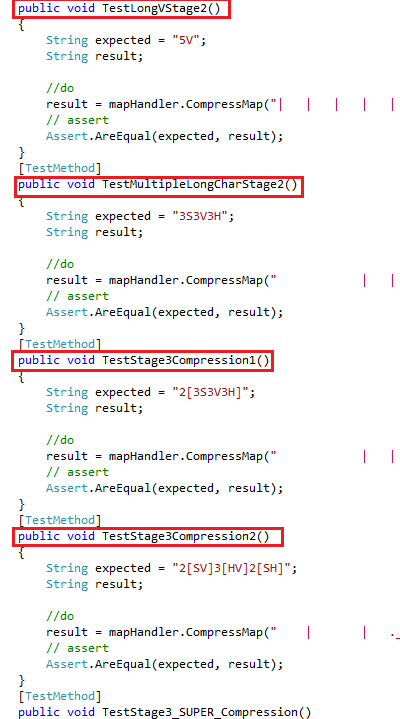
****The Testing framework that I used in C# unit testing.

**These are my OWN CODES**

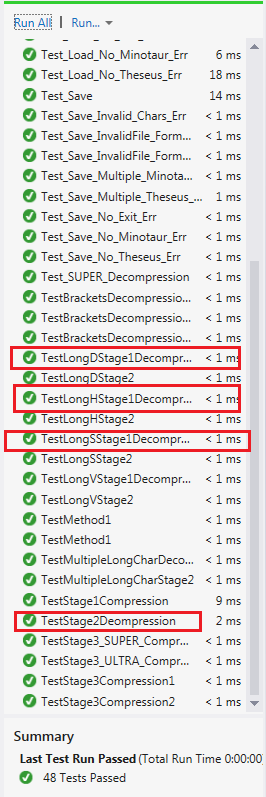
### **Example 1**

This is the testing that I implemented as a Filer

### **Example 2**

****

### **Example 3**



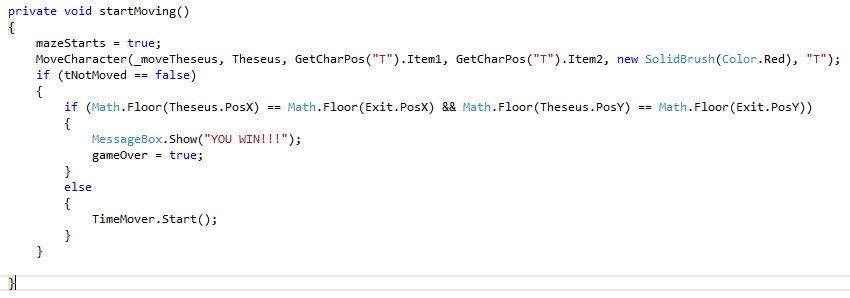
My approach is simple and easy to implement. Unit testing allows me to use built in tools from visual studio which helps me automate the testing easily.

## **Debug and test programs to the systems level.**

(seas, n.d.) Suggested that debugging is the activity of finding the causes that produced undesirable program effects. Two basic techniques are used to debug a program. In one technique, called tracing, a program can be instrumented to produce a trace. In tracing, a program runs to completion without external intervention and information about several selected states of the program are saved. After a program is executed, this information is studied to find the causes of a program's malfunction. A program malfunction is called a bug.

**These are my own code**

### **Example 1**



This is an example of debugging technique that I used. To be able to understand what causing the error, you need to output the value of variables that you are using inside the method.

On my examples, I am trying to show that the players has either win or lose the game.

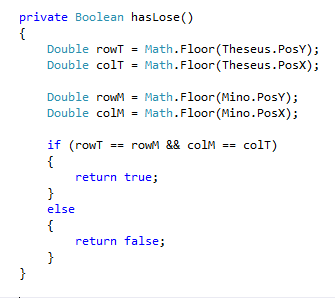
Using MessageBox.Show() to output the value of the variables, for example , I was able to detect the problem that causing my program not to show the message..

That the coordinates does not match exactly when Theseus is either caught or win.

### **Example 2**



### **Example 3**



## **9. Compare and contrast the features and uses of different programming languages.**

### **C#**

C# is a multi-paradigm, object-oriented programming language encompassing strong typing, essential, declarative, efficient, class-based, and component-oriented programming disciplines. C# is one of the programming languages designed for the Common Language Infrastructure.

C# is a well-designed and type-safe that allows C Sharp developers to build a wide array of secure and robust applications that run on the .NET Framework.

### **Advantages of C#**

* C# language is projected to be a simple, modern, general-purpose, object-oriented programming language.
* C# helps you to import a namespace and use the classes in a component—no COM plumbing or registry lookups required.
* C# supports XML comments to add comments to code. The comments are placed into XML format and can then be used as needed to document your code.
* C# simplifies the syntax to be more constant and more rational while also eliminating some of the more difficult features of C++.
* C# provides operator overloading which is not available in VB.NET
* C# allows you to access memory directly using unsafe code blocks
* C# allows you to implement an interface in a base class and re-implement the interface in the derived class and provide a new definition for it C# permits nested classes which are not allowed in C and C++
* C# establishes better event management using delegates and supports conditional compilation & cross-language interoperability with any .NET language
* C# is based on reflection mechanism which is biggest advantage of C#.

### **Uses of C#**

* Windows Applications
* Mobile Applications
* Server Apps at financial services Industry
* Web applications
* Software tools
* Scientific Applications

### **Disadvantages**

* More memory required for processing instruction.
* A bit slower than other languages.

### **PHP**

PHP is a programming language that can do all sorts of things: evaluate form data sent from a browser, build custom web content to serve the browser, talk to a database, and even send and receive cookies (little packets of data that your browser uses to remember things, like if you're logged in to a website.

### **Advantages of PHP**

* Open source: It is developed and maintained by a large group of PHP developers, this will helps in creating a support community, abundant extension library.
* Speed: It is relative fast since it uses much system resource.
* Easy to use: It uses C like syntax, so for those who are familiar with C, it’s very easy for them to pick up and it is very easy to create website scripts.
* Stable: Since it is maintained by many developers, so when bugs are found, it can be quickly fixed.
* Powerful library support: You can easily find functional modules you need such as PDF, Graph etc.
* Built-in database connection modules: You can connect to database easily using PHP, since many websites are data/content driven, so we will use database frequently, this will largely reduce the development time of web apps.
* Can be run on many platforms, including Windows, Linux and Mac, it’s easy for users to find hosting service providers.

### **Disadvantages of PHP**

* Security : Since it is open sourced, so all people can see the source code, if there are bugs in the source code, it can be used by people to explore the weakness of PHP
* Not suitable for large applications: Hard to maintain since it is not very modular.
* Weak type: Implicit conversion may surprise unwary programmers and lead to unexpected bugs. For example, the strings “1000” and “1e3” compare equal because they are implicitly cast to floating point numbers.

### **Conclusion**

PHP and JAVA are both great programming languages, however both has its own advantages and disadvantages. For me application created by using either two languages depends on how the developer code the program, if the app is insecure then it’s not just the language, it’s mainly on how you built/created the code.

However, for me JAVA is the best language for creating windows form applications. It’s simpler and the error report it gives you is detailed enough to debug a problem. PHP is more on the web applications, queries and data exchange.

# Bibliography

Bautista, N. (n.d.). Retrieved from Code.tutsplus.com: http://code.tutsplus.com/articles/a-beginners-guide-to-design-patterns--net-12752

Garner, S. (2012). Retrieved from dragonpoint.com: http://www.dragonpoint.com/whats-the-design-phase-of-a-software-development-project/

Mytton, D. (2004, September 23). Retrieved from http://www.sitepoint.com/coding-standards/

Rouse, M. (2014). Retrieved from http://searchsoa.techtarget.com/definition/Java

seas. (n.d.). Retrieved from www.seas.gwu.edu: http://www.seas.gwu.edu/~adagroup/concurrency/lopes/2softwar.htm

Toyotaro Suzumura, S. T. (2008). *Performance Comparison of Web Service Engines in PHP, Java, and C.* Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.508.3995&rep=rep1&type=pdf

tutorialspoint. (2015). Retrieved from www.tutorialspoint.com: http://www.tutorialspoint.com/sdlc/sdlc\_overview.htm

UNM. (n.d.). *www.cs.unm.edu*. Retrieved from https://www.cs.unm.edu/~luger/ai-final/supplement-ch1final.pdf

usability.gov. (2015).