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

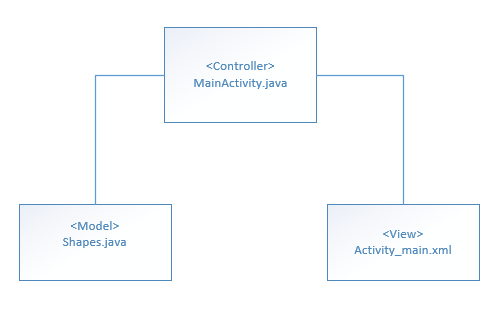
### Design patterns

According to (Bautista, 2010):

* + - 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 consists 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.
    - In my application I reused multiple codes (classes, methods) to provide the same solutions to the same problems.



Following, code:



This is my Controller, it talks to the model and send data to it.



This is my model (Player.java). Most of the processing is done on this class.



This is my view. The view follows this layout.

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

(wikipedia, 2019)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.

A programming idiom is a pattern, algorithm or way of structuring code. To talk about programming idioms is to talk about those patterns that recur frequently in code or to propose new ones.

The benefits of being familiar with idioms, particularly the larger ones, is that when looking at code you can see several lines of code but because it is familiar as a particular idiom you can mentally represent and think about the code as that single idiom instead of having to necessarily read and comprehend each line individually.

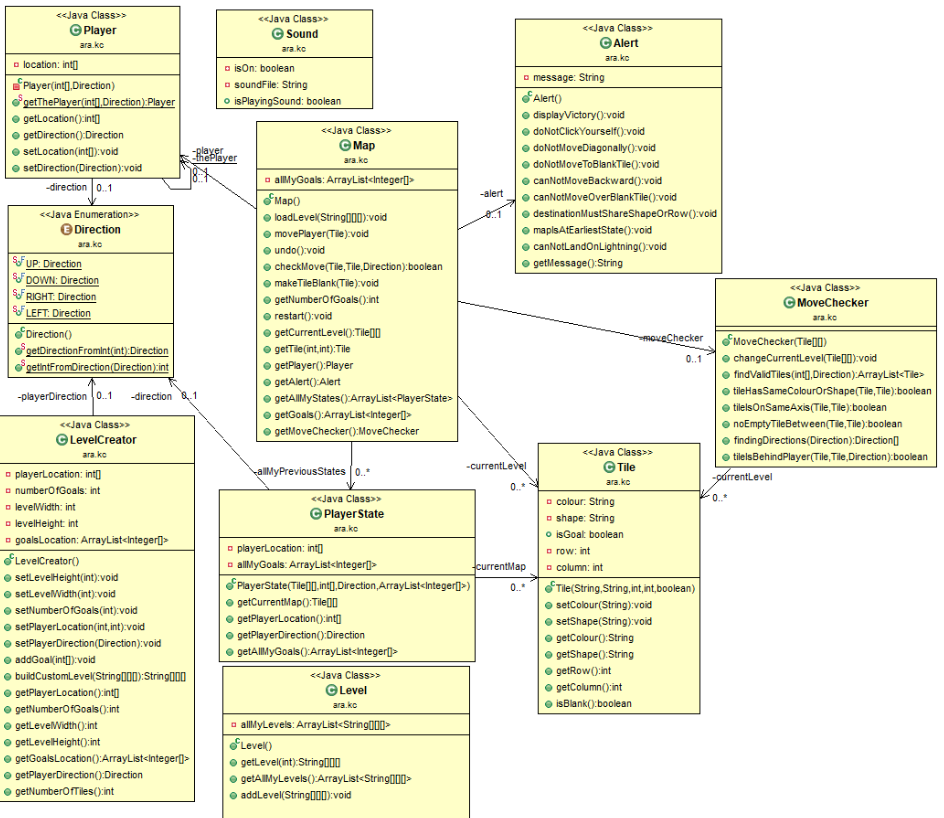
To say that code isn't idiomatic is to say that it doesn't structure itself in ways that allow human readers to think about the code effectively.



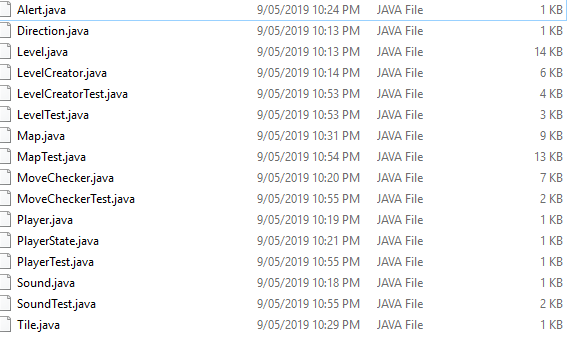
A programming idiom is the usual way to code a task in a specific language. For example, a loop as shown above.

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

* (Tutorials Point, 2019)Suggested that SDLC, Software Development Life Cycle is a process used by software industry to design, develop and test high quality software’s. 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 very complex, and design is simple (game is simple to play but hard to code).
* The main process is:
  + - **Requirements Analysis**- Here I made a list of features that the application needs for the level designer.
      1. Read from String.
      2. Compute the width and height.
      3. Create a new level (width, height).
      4. Construct the map from String.
      5. addBlock on the map.
      6. addTile on the map.
      7. addColor on the Blocks.
      8. addPlayer on the map.
      9. addTarget on the map.
      10. addScore on the map.
      11. addPlayerDirection on the map.
      12. Edit Cell to Block on the map
      13. Edit Cell to Floor on the map.
      14. Edit Cell to Crate on the map.
      15. Edit Cell to Player on the map.
      16. Edit Cell to Target on the map.
      17. Edit Cell to Player on target on the map.
      18. Edit Cell to Crate on target on the map.
      19. Undo action.
      20. Redo action.
      21. Error handler for invalid undo action.
      22. Error handler for invalid redo action.
      23. Load the map.
      24. Display the map with cells represented as characters.
    - **Design**- here I made a design level diagram to create a picture of what needs to be done.

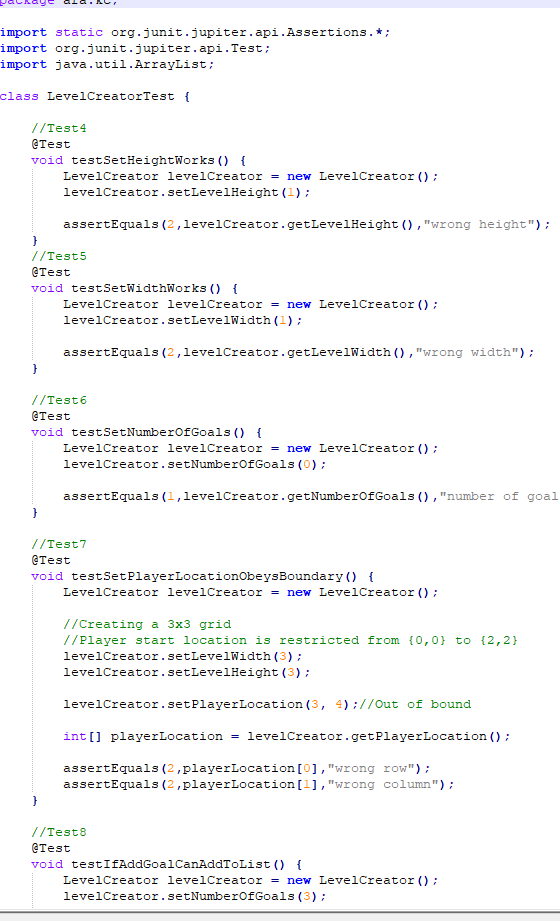


* **Implementation-** here I implemented the design and build the game in java.



* **Testing -** here I tested the game several times to get it right. I tested:
  + If the player moves up, down, left, right.
  + If the boxes up, down, left, or right with the player direction.
  + If the map is built, edit, correctly, centred even if the screen is rotated.

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 requires 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.**

(Venners, 1998) 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.

### **Class Diagram**

### **Activity Diagram**

### **Wireframe**

### **Sequence Diagram**

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

* According to (NTU, 2018), 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.

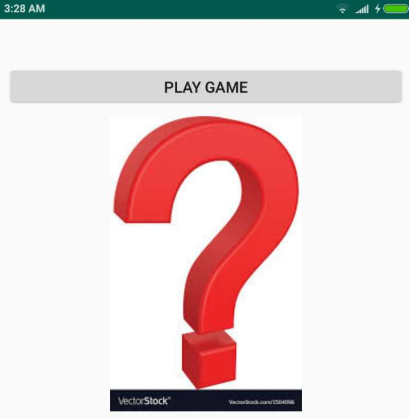
* + **Consistency (consistency in location, format, syntax, and naming)**

The way the system looks, and works should always be consistent.

* + **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 app more usable by creating swipe features, allows user to drag objects on the screen when creating a map (wall, boxes). Also, I should have built an error handler which tells user what they need to do when they put wrong input. Should provide a guide or help option.

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

(Digital Technologies Hub, 2019)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.

## Java Coding Standard

**Coding Standards for Components:**It is recommended to write components name by its purpose. This approach improves the readability and maintainability of code.

**Coding Standards for Classes:**Usually class name should be noun starting with uppercase letter. If it contains multiple word than every inner word should start with uppercase.

E.g.: String, StringBuffer, Dog

**Coding Standards for Interface:** Usually interface name should be adjective starting with uppercase letter. If it contains multiple word than every inner word should start with uppercase.

E.g.: Runnable, Serializable, Comparable

**Coding Standards for Methods:**Usually method name should either be verb or verb noun combination starting with lower letter. If it contains multiple word than every inner word should start with uppercase.

E.g.: print(), sleep(), setSalary()

**Coding Standards for Variables:** Usually variable name should be noun starting with lowercase letter. If it contains multiple word than every inner word should start with uppercase.

E.g.: name, age. mobileNumber

**Coding Standards for Constants:**Usually constant name should be noun. It should contain only uppercase If it contains multiple word than words are separated with (\_) underscore symbol. Usually we declare constants with public static and final modifiers.

**Java Bean Coding Standards:** A Java Bean is a simple java class with private properties and public getter and setter methods

**Getter Methods:**

1. It should be public method
2. Method name should be prefixed with “get”
3. It should not take any argument

**Setter Methods:**

1. It should be public method
2. Return Type should be void
3. Method name should be prefixed with “set”
4. It should take some argument

public class StudentBean{

private String name;

public void setName(String name){

this.name=name;

}

public String getName(){

return name;

}

}

Note: For boolean properties getter method can be prefixed with “get” or “is”

**Coding convention for Listners:**

* To register a Listner method name should prefixed with add

Eg: public void addMyAccountListner( MyActionListner);

* To unregister a Listner method name should prefixed with remove

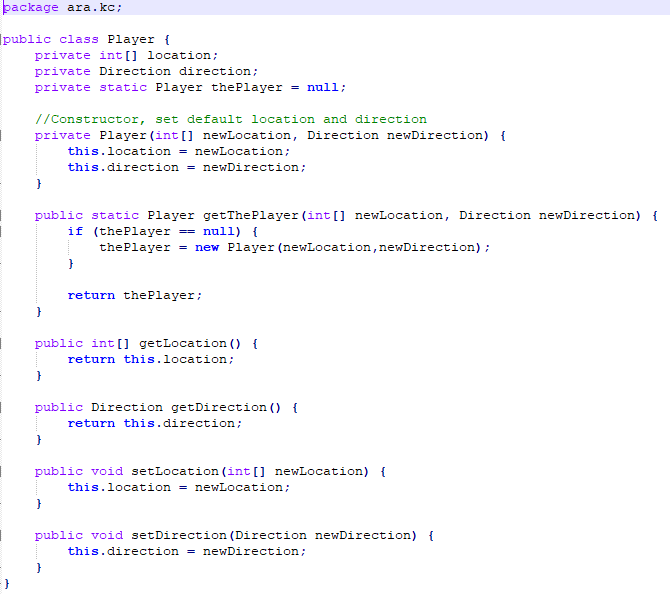
Eg: public void removeMyAccountListner( MyActionListner);

In my application I followed Java coding standard.

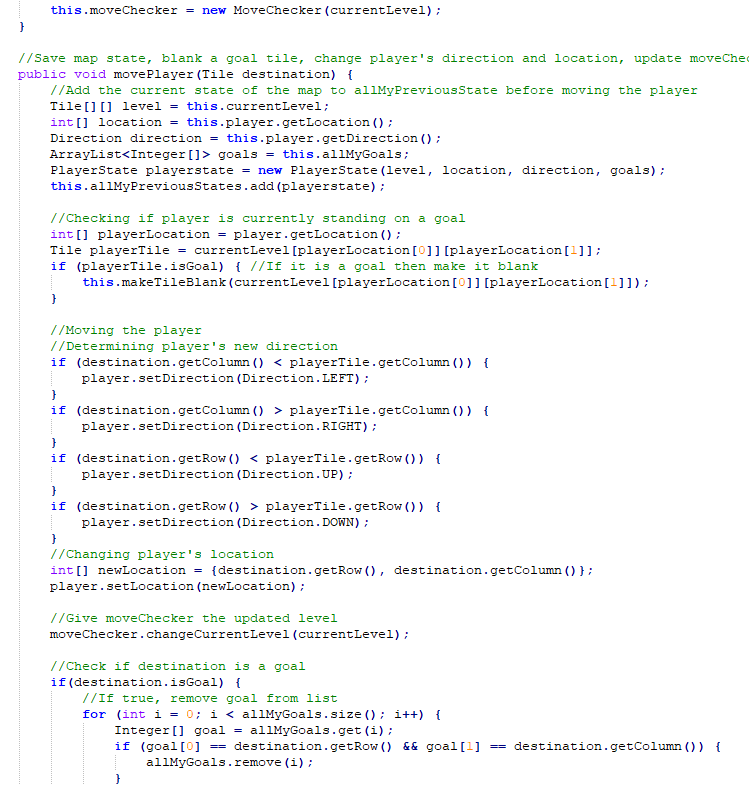
### **Code Example 1**

These codes coding standard are effective 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**



### **Code Example 3**



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 (guru99, 2019) 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.

In SDLC, STLC, V Model, Unit testing is first level of testing done before integration testing. Unit testing is a WhiteBox testing technique that is usually performed by the developer. Though, in a practical world due to time crunch or reluctance of developers to tests, QA engineers also do unit testing.

**These are my OWN CODES**

### **Example 1**

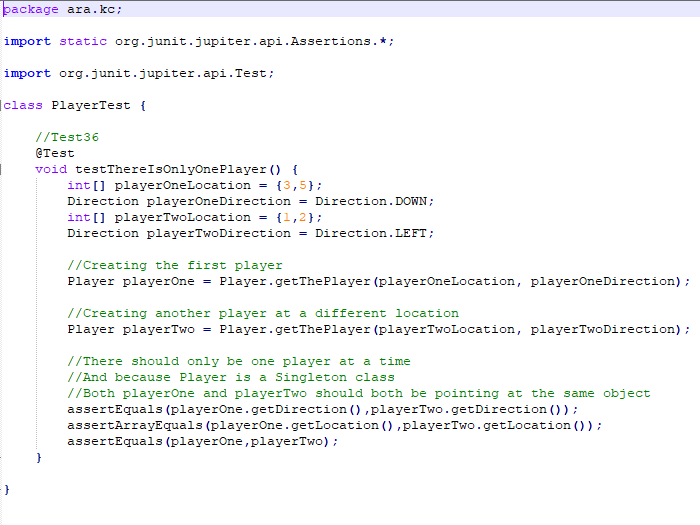
This is the testing that I implemented as an Eyeball Maze model designer in eclipse.



### **Example 2**



### **Example 3**



My approach is simple and easy to implement. JUnit testing allows me to use class inheritance which helps me automate the testing easily.

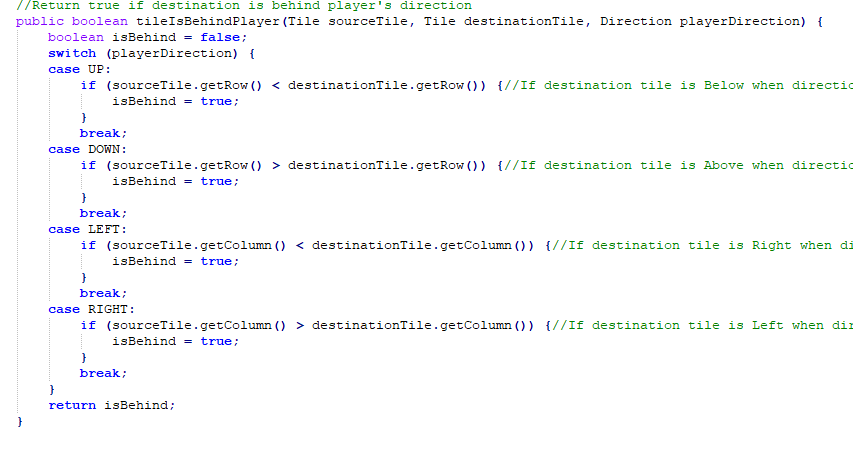
## **Debug and test programs to the systems level.**

(GShute, 2018) 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.

**Difference between Testing and Debugging**

|  |  |
| --- | --- |
| **Testing** | **Debugging** |
| The purpose of testing is to find bugs and errors. | The purpose of debugging is to correct those bugs found during testing. |
| Testing is done by tester. | Debugging is done by programmer or developer. |
| It can be automated. | It can’t be automated. |
| It can be done by outsider like client. | It must be done only by insider i.e. programmer. |
| Most of the testing can be done without design knowledge. | Debugging can’t be done without proper design knowledge. |

### **Example 1**



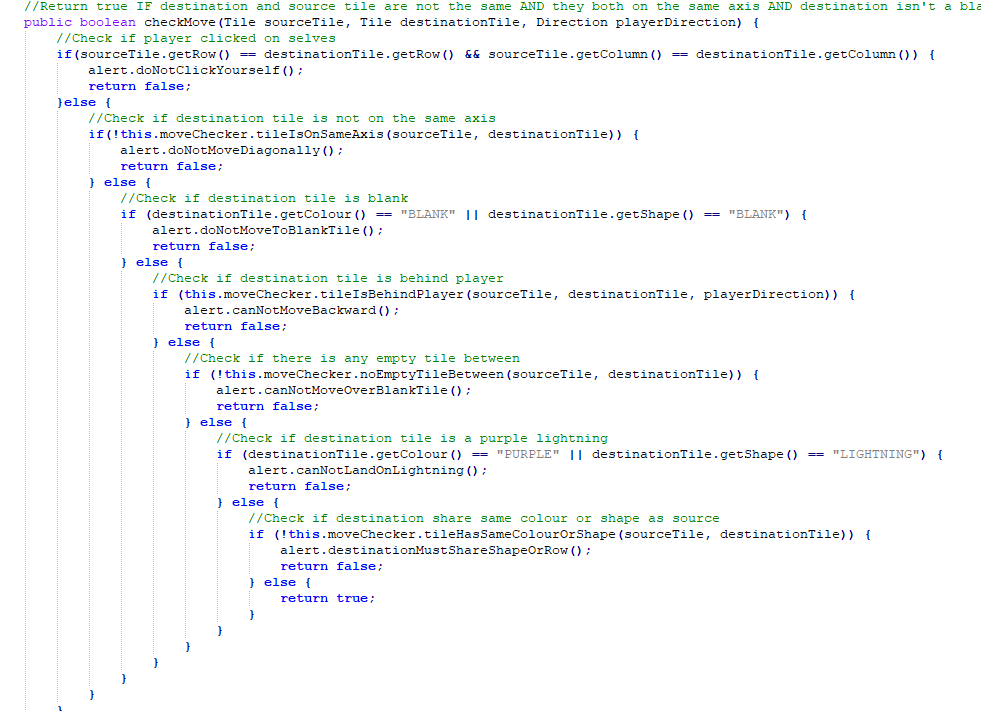
This is an example of debugging technique that I used. To be able to understand what causing the error, you need to output the value of variables that you are using inside the method.

On my examples, I am trying to build to map using nested loop and nested if, if else conditions.

Using system.out.println to output the value of the variables, for example int I, I was able to detect the problem that causing my method to build the map incorrectly.

I learned that the character “|” was causing the bug and it should not be included in the count.

### **Example 2**



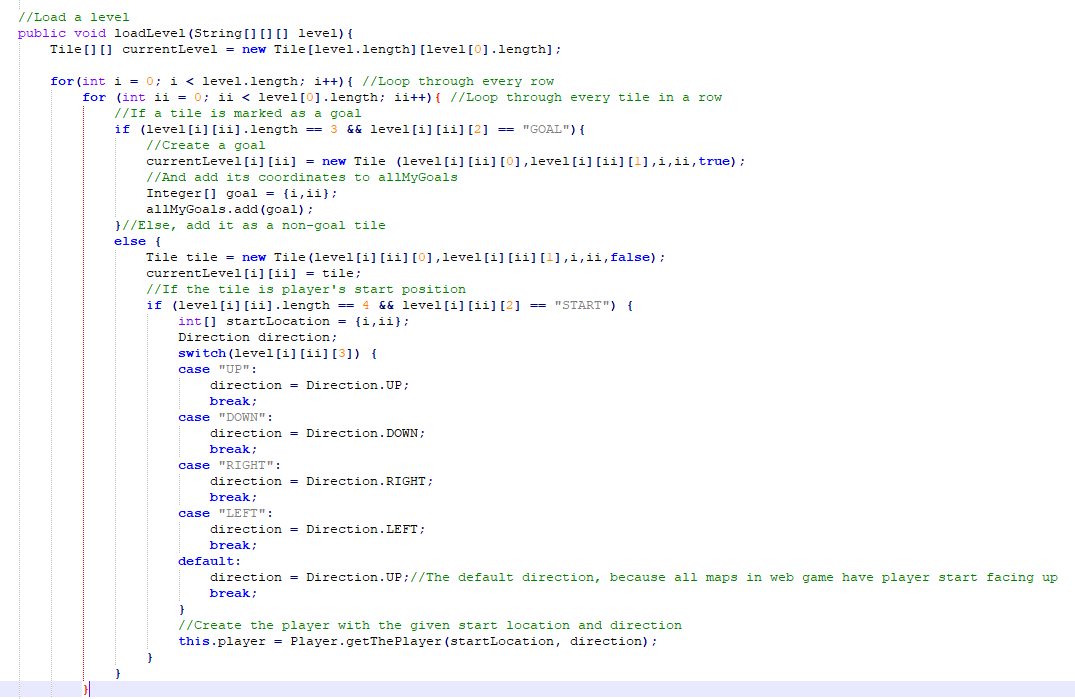
I should try and avoid complicated loops and nested loops if possible, should find a simpler way to do things.

## **Provide all appropriate systems level documentation.**

As we have been taught that “The important Systems level documentation is COMMENTS in the code! You could extend the comments by using JAVADOC format, and then automatically generate documentation about the code. Other tools within Eclipse or Android studio that help generate documentation are also useful”.

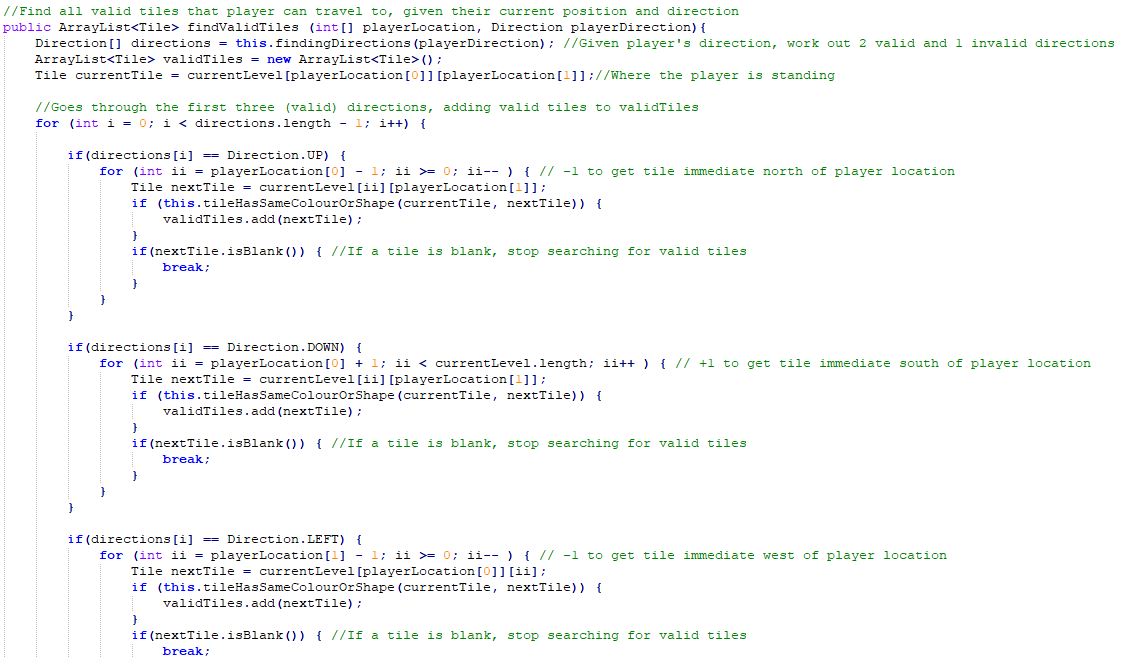
“A class diagram of the final build or a UML dynamic diagram which explains how you <<MODELL>> and <<VIEW>> and <<OTHER\_STUFF>> would be useful systems level documentation”.

### **Example 1**



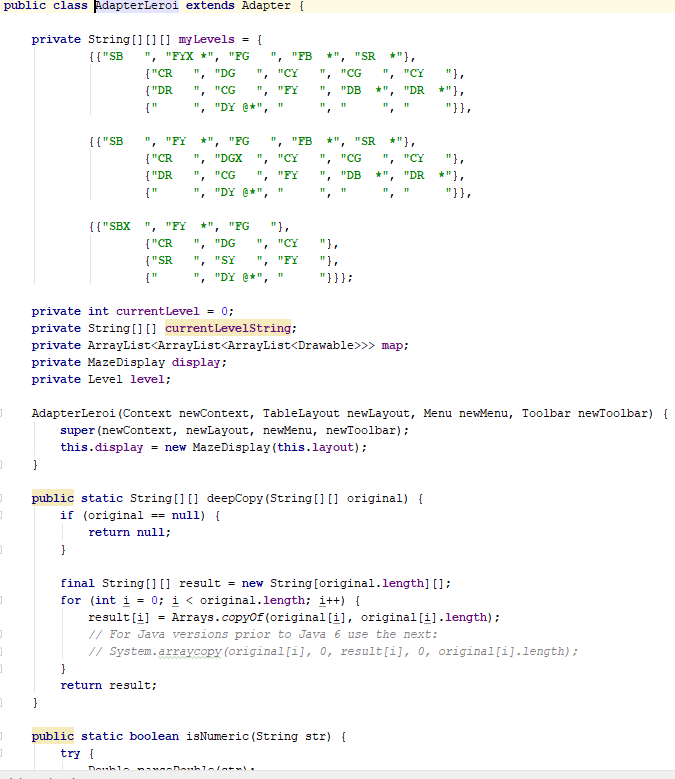
On my examples I put comments on almost every line of code, so when I come and look at this after 2 months, I think I would still be able to work on this code again easily.

### **Example 2**



## **Maintain existing programs and update documentation.**

As a developer, it’s always important to have reliable documentation to guide your work. If you are like most developers, you can be supporting multiple applications at the same time, which means that it is even more crucial to have documentation in place to help track all aspects of each application. It’s also helpful for development, maintenance, and knowledge transfer to other developers. The following are some areas in which documentation is especially valuable for a developer and some of my thoughts on what should be included. If possible, the documentation should be focused into the various components that make up an application.

**Example1** 

Here I had to make modification, to make get the data and pass it to another activity, so that the map can be built using The String map.

### **Example 2**

Here I had to modify some codes to give the maze display.

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

### **Java**

According to (codec ademy, 2019), JAVA is a programming language expressly designed for use in the distributed environment of the Internet. It was designed to have the "look and feel" of the C++ language, but it is simpler to use than C++ and enforces an object-oriented programming model. Java can be used to create complete applications that may run on a single computer or be distributed among servers and clients in a network. It can also be used to build a small application module or applet for use as part of a Web page.

### **Advantages of JAVA**

* Object Oriented
* Platform independence
* Automatic import class feature (alt + enter) in android studio
* Java is platform independent
* Java is distributed
* Java is simpler to learn

### **Uses of JAVA**

* Android applications
* Server Apps at financial services Industry
* JAVA web applications
* Software tools
* Big data technologies
* 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 help 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 depend 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 mobile 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. (2010, July 7). *envatotuts - Code*. Retrieved from envatotuts website: https://code.tutsplus.com/articles/a-beginners-guide-to-design-patterns--net-12752

codec ademy. (2019). *codecademy - Java*. Retrieved from codecademy website: https://www.codecademy.com/catalog/language/java

Digital Technologies Hub. (2019). *Digital Technologies Hub - Coding*. Retrieved from Digital Technologies Hub website: https://www.digitaltechnologieshub.edu.au/teachers/topics/general-purpose-programming-coding

GShute. (2018). *Debugging and Testing*. Retrieved from GShute Website: https://www.d.umn.edu/~gshute/softeng/testing.html

guru99. (2019). *guru99 - unit testing*. Retrieved from guru 99 website: https://www.guru99.com/unit-testing-guide.html

NTU. (2018, April). *Java Programming Tutorial*. Retrieved from NTU website: https://www.ntu.edu.sg/home/ehchua/programming/java/j4a\_gui.html

Tutorials Point. (2019). *Tutorials Point - Overview*. Retrieved from Tutorials Point Website: https://www.tutorialspoint.com/sdlc/sdlc\_overview.htm

Venners, B. (1998, February 01). *Javaworld: Design techniques*. Retrieved from Javaworld website: https://www.javaworld.com/article/2076601/introduction-to--design-techniques-.html

wikipedia. (2019, Januaury 10). *Wikipedia - Programming Idioms*. Retrieved from Wikipedia Website: https://en.wikipedia.org/wiki/Programming\_idiom