***Name: - Muhammad Abu Bakar Sani***

***Student Number: - 17112044***

***Debugging***

* **Introduction**

Debugging is a cycle activity involving execution testing and code correction. Testing and debugging is the last step of all applications before launching, to make it error free and user friendly. Testing comes first then debugging. Once the testing is finished then developer know all bugs in that application. And debugging Is the process of removing all bugs from that application to make it error free. This process is a complete cycle i.e. TESTING then DEBUGGING then again TESTING and so on. Until all errors will be removed. Debugging is the most difficult part as compare to all others because it requires more time. Firstly, developer need to spend a lot time to do proper testing then recognize what and where these errors are? Sometimes the symptoms are completely irrelevant and It's very hard to figure it out.

Secondly, try to find the solution of all these errors one by one. Sometimes to resolve single error developer need to change whole code, sometimes two errors are linked with each other if one is solved other will solved automatically.

Thirdly, do the testing on that code to check if it works properly. If YES, then move towards the next code. If NO, then stick with that to solve it before he moves on to the next error.

There are many different types of bugs: -

* 1. Function-related bugs
  2. Coding bugs
  3. Documentation bugs
  4. Data bugs etc.
* **Implementation**

Our project is an addon of a mini racing car tournament which takes place within the game of Minecraft.

Here are the bugs and its solutions that we faced: -

Problems we faced

* 1. Case sensitivity in Player class, if player stored his name in small alphabetical letters then every time, he needs to write in small letter otherwise the game will not access it. We solve this issue by using a code **“. toLowerCase()”**. This will automatically convert the user name into small alphabetical letters before it stores into database or before checking that name exists in database or not. It grabs the user name from the textfield convert all capital letters into small letters then store in variables. For Example, **name=nameTF.getText().toLowerCase();**
  2. Application crash due to Player class, if by mistake user enters his name instead of his age in age textField then app will crash. After that user need to open and fill all the info again from the scratch. We solve this issue by using if else statement inside the button “Done” onclick method. Whenever use hit “Done” button if else statement will work to check that is user write his correct age in correct format? if(yes) then move toward the next process. Else, display “Please enter age in correct format”. For Example if (**age=Integer.parseInt(ageTF.getText**)){ “your age is ok”} else{ “Please enter age in correct format”}. Same strategy we apply for Lap class in which use need to enter laps number for race.
  3. We have also faced documentation bugs. We mixed our use case diagram with sequence diagram. Instead of making direct link between player and all main use cases with which it interacts. We make a connection between player and first use case with which it interacts. Then link that use case with the send second use case and so on until it reached to the last one like Sequence diagram.
* **Situations where it would be used**

In debugging process, time is directly proportional to damage an app. To save an app from big damage debugging need to be done quickly otherwise it may crash whole system. It’s not a good practice to leave all the errors to be solved at the end. Because sometimes at the end it may become more complex to solve and can cause more errors then you expect.

Debugging can be used in many situations like: -

1. To remove functional related bugs.
2. To make system error free.
3. Most important situations are to remove code and design related bugs.
4. Data bugs. To check is user entering correct input? Or what will happen when enters incorrect input or data.
5. Debugging can also be used to remove Documentation bugs. Documentation bugs is very important to be removed otherwise it will make everything wrong when developers actually start doing codes.
6. It can also be used to remove System related bugs. If you have perfect code and your system or platform is not running well. Then it’s very hard to trace out where that exact problem is. So good developers always test their system before they actually start coding.

* **Advantages**

Debugging has too many advantages. Here are some of them: -

* 1. It will make App errors free.
  2. Debugging can save any software company from big losses.
  3. It will help developers to fulfil the needs of end user.
  4. Debugging is not just related to App codes, but it also deals with documentation bugs, system related bugs, data bugs etc.
  5. Documentation debugging keep the developers to the right track. And I will save their time and money.
  6. Debugging is most cost affecting activity.
  7. It’s very good practice to enhance your experience and make your mind sharp.
  8. It plays very important role in success of Application.
  9. It makes app more user friendly and secure.
  10. Sometimes It will save developers to start everything from scratch.
  11. It's very good practice for unexperienced developers to get experience.
* **Disadvantages**

Here are some of the disadvantages of debugging: -

1. Sometimes Debugging require well experienced expensive application developers.
2. Debugging is one of the most difficult part of the app development.
3. Firstly, developer need to figure it out what and where that error is? Secondly, he/she need to find the solution of that particular problem. That’s why it takes a lot more time.
4. Sometimes company need to pay huge amount for the debugging process.
5. Sometimes debugging took a lot time that customers start using other software's.
6. Sometimes symptoms and causes may be separate from each other.

* **Bibliography**

Kencoder Blog. (2019). The Four Steps of Debugging. [online] Available at: https://kencoding.wordpress.com/2015/06/06/the-four-steps-of-debugging [Accessed 28 Mar. 2019].

YouTube. (2019). Debugging in Software Engineering | Debugging Approaches | Strategies of Debugging | Art Debugging. [online] Available at: https://www.youtube.com/watch?v=JiWMiAFsMss [Accessed 28 Mar. 2019].