**Recommendation Report on Debugging and Error Checking video game program code**

**in the more efficient and timely manner**

***Cameron Windsor Fox***

**EGR3350:01**

**Fall 2017**

**Mr. Pestak**

**Introduction**

*Purpose*:

The purpose of this report is to recommend a new debugging process for Company X which is failing to meet its development timelines due to limitations with software debuggers.

*Problem*:

Company X is a new Indie Video Game company working to build a new RPG (Role Playing Game) video game using a video game development framework. They have started working with python source coding along with using a program called pygame. This is a program that works with python source code to create a functioning game with this specific language.. They plan to have the game finished and ready to release in the next two years. With hitting this deadline, they will be able to show of this new game along with other big Indie game titles to ensure that games will give this s try. If they miss it, then they may not be able to get the needed feedback to make any sales or profit from the game. As the company goes to work and each team within the company works on their share of the game, the program developers are having issues with error checking the source code being created. If the developers are unable to find and if all the errors and problems within the code of the game, every error can add up and the worst case is the game could be unplayable as a result.

*Scope*:

This report will compare the use of Unity and the built-in software Mono-Develop and Microsoft Visual Studios to see which software is better suited for the developers using three criteria: reliability, education of software uses and cost.

**Discussion**

*Criterion x3*

**Reliability**

Explanation:

The use of reliability for this report will be dealing with comparing both software and by using the same source code, then testing the error checking for both. When each software detects an error, this will see how the program will alert the developer and how the error checking changes before and while in debugging mode and during test runs. This will show which software is more efficient with detecting problems within the source code and letting the developer know accordingly. If it takes developers five to 10 minutes spent checking for errors on their own with a debugger or software not able to either detect them or alert the developer where the error is, this will use up time for working on the game.

Data:

By examining the software, when error checking with Mono-Develop before entering Debug mode, if any grammatical or syntax errors occur Mono-Develop will show a little red line under the location where the problem was created. Aside from that, this is all that mono-Develop will do before entering Debug Mode. If a developer is using this process when checking for syntax mistakes, it could use up a well five minutes or longer if the software can’t easily inform the developer right away. When debug mode is activated, the developer is brought right to the red line location in the code along with a note outlined in red showing where the error is and what the program expects to be there to allow the error to be corrected. This will reduce the time needed to go searching for the error yourself, taking only a minute or two to make the correction without delay before moving on to the rest of your code. There is one main problem with the debugger that developers use with Mono-Develop. Developers can only test one script or one written group of code at a time. Meaning that to do a full debug test will mean take 15 minutes at best to run the debugger for each script individually. After Mono-Develop has gone through debug mode, if any problems are left unchecked, when developers go to do any test run of the source code, the program will automatically stop and tell the developers, all errors must be resolved before testing can begin. This right there makes sure the developers have all the source code corrected, only then will Unity allow the code to be tested and progress can continue. Along with this process, Mono-Develop Built-in Unity software is setting the developer up for a major time loss here. If the developer couldn’t find an error or problem before entering debug mode or while debug mode was running, this means they will have to spend an unknown number of minutes or longer to go back through the code and find what is going wrong and where the issue is originating from. This will mean going back and starting the debugger for each set of source code for the game, taking from 15 to even 30 minutes of searching before an error is discovered.

In the case of using the same source code and using Microsoft Visual Studios, when the same type of error occurs, the software will also draw a little red line under the error but then visual studios goes one step further and at the bottom of the computer screen an error list box is shown and will automatically update when an error is created. This perfect because it catches syntax and variable building errors right away and this is easy for the developer to notice, so correcting the mistakes takes little time. This has a good reduction in minutes per error correction, for searching before testing the debugger. Developers are right away notified where an error was made and gives the location directly to it. Shortening the time needed to traverse the code to find the location themselves. While in debug mode, this can take up a good amount of time for error checking. Depending how big your source code is and how much has already been built, this can make the use of debugging very time-consuming depending on the amount of source code being tested and debugged. By watching how the variables change from line to line can help identify where a problem started. Cutting the time to find the rough location to minutes instead of hours of checking. From there it makes the debugger’s job much easier. If there are any problems with how the source code should be implementing the game, this means that instead of an error with the code being syntax issue or variables are not being created properly, the program will see everything is correct, but the developer may have made an error with writing a specified line in the code. This is very difficult for developers, because the error in this case is completely invisible and the debugger can’t detect it. Using up more and more time to go through each line of code until the developer finds what was done wrong.

Interpretation:

By looking at this information from start to finish regarding reliability with error checking and notifying the developer. Including the checking before Debug mode, In Debug mode, and after Debug mode; Unity takes the win. With error checking before debug mode Visual Studios is more capable at finding errors and notifying the developer allowing them to be fixed sooner, while in debug mode, both software was about the same stance. But Unity takes the win at the end. When trying to do any code testing Unity will not allow the developer to run the code until all errors are fixed. Telling the Developer problems are still not fixe in the code and need to be corrected for testing to start.

**Education of Software Use**

Explanation:

For Video Game Developers to use a game development program they must first learn how it works. Does this program have a website to teach a new programmer how it works? What kinds of functions or source code is used? Can the developer learn how to use this program and its software easily or will it take time by not having the information to research?

Data:

With using Unity and its software for building source code and debugging, it is very easy to learn what everything does. Unity’s website has information for every function and use of source code that the software offers. It is very well constructed, and everything is easy to follow along. One plus for coding in Mono-Develop, if a developer is having trouble thinking of what function to use or how to apply a certain aspect for a section of code or a script for the game, there is a handy short cut to take you directly to the website, more precisely right to the function type you want to work with and then lists all the different uses this certain function or variable has. The developer clicks on the link that is needed and here, the information is describing what this variable or this source code will do. At the end giving an example of how it would look like.

When working with Microsoft Visual Studios, if the developer needs to look up how the language works, for the most part, this software doesn’t have the same type of help as Unity has for Mono-Develop. If someone needs help on how something works in visual studios the developer needs to search other sites for where other programmers have asked similar questions and they learn how the code works, this way.

Interpretation:

By looking at how the developers can learn both working with Mono-Develop with Unity versus using Microsoft Visual Studios, by using Mono-Develop the built-In software, you will have a much easier time learning how the code will work, run, and how to build it. With using visual studios, it would be more challenging to go online and search sites to find an answer to a question on how a function or section of code is supposed to work. Unity built-In Mono-Develop wins this criterion.

**Cost**

Explanation:

With this criterion, cost will be comparing the price that developers would need to pay to be able to use the two programs but also the software that is needed for video game development that the programs supply. The data will be a table showing the price of each program and the software used for coding and to do any debugging.

Data:

|  |  |  |
| --- | --- | --- |
| Program: | Unity [1] | Microsoft Visual Studios [2] |
| Program Type #1: | Personal: $Free | Visual Studio Professional Subscription: $799.00 |
| Program Type #2: | Plus: $35.00 per month | Visual Studio Enterprise Subscription: $2,569.00 |
| Program Type #3: | Pro: $125.00 per month | Visual Studio Professional 2017: $499.00 |
| Program Type #4: | N/A | Visual Studio Test Professional Subscription: $899.00 |
| Source Code Software: | Mono-Develop | Visual Studios |

These are the different prices for the different choices of software for Unity and Microsoft Visual Studios. Prices checked on their main websites.

Interpretation:

From looking at the costs for each software version listed from each of the two-program’s main website, Unity would be the cheapest to buy compared to the software choices with Microsoft Visual Studios.

**Conclusion**

*Summary*

This report will compare the use of Unity and the built-in software Mono-Develop and Microsoft Visual Studios to see which software is better suited for the developers using three criteria: reliability, educating the software uses, and cost. With Unity’s Built-In Software Mono-Develop being the better choice.

*Conclusions*

After going through all the data and information, Unity is in the lead with Microsoft Visual Studios, in second place. With Unity alone being a cheaper deal, as well as having a website to clearly explain every part of source code that can be used making everything easy to follow and understand. This is a very good choice for a new game development company. Even though Visual Studios might come out on top with some uses with making debugging easier, Unity’s Built-In software has so much more potential and uses that visual studios doesn’t have the resources to support the same way Unity’s software Mono-Develop can.

*Recommendation*

Based on these conclusions, I have found that with using Unity, to pick from either using the Built-In software for developing source code and debugging Mono-Develop or using Visual Studios, I recommend using Mono-Develop. Even though Mono-Develop has its points where Visual Studios wins, it is a better choice overall it is going to be cheaper than having both the subscription with Unity and the payment for Microsoft Visual Studios. Using the built-in software Mono-Develop when using the video game development program Unity will be the best choice for the Indie Gaming Company X to work and get their new video game built and finished by the due date.

**Point of Contact (POC)**:

Cameron Windsor Fox

Junior College Student at Wright State University

Majoring in Computer Science

(419-429-9817)

cameron1w1fox@gmail.com**References**

[1] Prices for different Unity Program Choices:

Unity – Store

<https://store.unity.com/?_ga=2.169341409.135285862.1508798032-1477483054.1504639806>

Searched Oct. 24, 2017

[2] Prices for Microsoft Visual Studios Program Choices:

*Microsoft*

<https://www.microsoft.com/en-us/store/b/visualstudio?cl_vend=bing&cl_ch=sem&cl_camp=290157475&cl_adg=1260040365319056&cl_crtv=78752537509183&cl_kw=visual%20studios%20used%20for%20development%20companies%20price&cl_pub=bing.com&cl_dvt=c&cl_mt=b&cl_gtid=kwd-78752546351751:loc-71167&cl_dim0=WaxUpwAAA2-62wL3:20171024210320:s&OCID=AID620866_SEM_WaxUpwAAA2-62wL3:20171024210320:s&s_kwcid=AL!4249!10!78752537509183!78752546351751&ef_id=WaxUpwAAA2-62wL3:20171024210320:s>

Searched Oct. 24, 2017

**Appendix**