**Software Engineering Testing**

**Interact with Objects/Environment**

**Unit Testing**

**Cian Larkin – x17453136**

**Introduction**

Our game idea is a Battle Royale for Grand Theft Auto 5 and my chosen use case is interacting with objects and the environment. I have decided to go with performance testing to test this section of the game as I feel it is necessary to test how the game handles players maneuvering throughout the environment and using all of the objects available. I hope to find that frame rates don’t drop and that there are no bugs when players try to carry out a certain task. It is also vital that our servers perform as expected whether it is 9am or 9pm.

**Implementation**

There are many different types of performance testing I will talk about some of them below, all of them share a common goal of making sure that the game is going to perform up to standard under it’s expected workload. They all revolve around the speed of the game, the scalability of the environment/player size, and the stability of the game under stressful server situations, such as high player counts.

* I plan to use load testing first as this will show me how the system handles high player counts as the game will hold a maximum of 100 players per game. The objective of this is to test the frame rate for the player if they’re in a dense environment where there is a lot going on. I will need to make sure the system can handle our planned maximum operating capacity which will be 100, I think the easiest way to test this would be to introduce 100 bots to the game and see how it handles them. With successful load testing this will give all of the stakeholders involved confidence in our system and how will perform when it’s made live. I plan to use the LoadRunner software which has the ability to carry out what I have talked about above. (Guru99.com, n.d.)
* I will also look at using endurance testing to see how the environment handles constant player activity. I will first overload the system with objects for example adding lots of vehicles and doubling the amount of loot found by players and testing the frame rate when 100 players are added to the environment. I will have thousands of automated games up and running with 100 players in each to test how our servers handles lots of active players at one time. I hope to find no defects in our system performance after carrying out this test. (Guru99.com, n.d.)
* This leads me to volume testing which will also be testing how our servers handles thousands of players interacting with objects and the environment across many servers, this will give me an insight to any bottlenecks that may occur particularly in the load screen. We will keep adding numbers to our servers to identify any negative impact of our server speeds. The player counts will be staggered and then servers will then be stopped in order to check all of the logs. (Guru99.com, n.d.)

**Where would this be used**

This type of testing is carried out towards the end of the development cycle after the bulk of the game has been developed. In order to carry out a successful performance test we will have to plan out each test, this means we will need to understand the size of our environment in relation to populating it with various elements to test the strength of our servers. We will have to plan out the goals of each test in relation to the kind of frame rates we want to see and also in relation to our servers. The next step will be to engineer our tests, we will carry out the performance tests I have listed above, we will need to have enough developers to carry out the tests effectively and also agree on any software we plan to use. After these steps have been carried out we are ready for performance tests which must be monitored as they occur as small changes in the code could affect the games performance in a negative way. To stop this from happening we can set up monitoring scripts which will detect any issues and they can be quickly be fixed, in situations like these we would have to use debugging techniques such as brute force or backtracking in order to get things back up and running.

**Advantages vs Disadvantages**

* Load Advantage – by carrying this out we will increase our customer satisfaction with the game mode. No one wants to play a battle royale with poor frame rates and long load times due to too much activity being carried out. (Guru99.com, n.d.)
* Load Disadvantage – running thousands of automated servers with automated players can be an expensive section of development but I feel it is necessary as we need to be able to handle thousands of players using objects in the game environment. (Guru99.com, n.d.)
* Endurance Advantage – this is used to make sure there isn’t any degradation of our game over a long period of time, this means after our servers have been running for months and years players shouldn’t have any problems when using the many objects found in the environment. (Guru99.com, n.d.)
* Endurance Disadvantage – it can be hard to predict the popularity of a product which means it will be hard to pinpoint how much stress needs to be applied to the environment. (Guru99.com, n.d.)
* Volume Advantage - The main benefit of this is if we identify potential server lag we will be able to save lots of money that would’ve been spent on maintenance. Also by keeping our servers running smoothly we will be able to hold onto our players and they won’t be tempted to go to another game.
* The overall benefit of performance testing is that it gives you the ability to put as much pressure on your product as you want in order to test its full capability. In relation to this particular test case if we found that a 100 player count slowed down the game and there was no way to fix it well then we would have to make a drastic change to it before release.

**Bibliography**

* (Guru99.com, n.d.)

Guru99.com. (n.d.). [online] Available at: <https://www.guru99.com/load-testing-tutorial.html> [Accessed 27 Mar. 2019].

* (Guru99.com, n.d.)

Guru99.com. (n.d.). [online] Available at: <https://www.guru99.com/endurance-testing.html#4> [Accessed 27 Mar. 2019].

* (Guru99.com, n.d.)

Guru99.com. (n.d.). [online] Available at: <https://www.guru99.com/volume-testing.html> [Accessed 27 Mar. 2019].