Maintenance Plan

Wizards

Maintenance Plan

What to Maintain

To say that Wizards was a perfect product that could not be improved would be a lie. Some of the features in the backlog were either not as fleshed out as they could have been, or they were not included at all. Regardless of that, there are many improvements to the core of the gameplay that could be made to improve the product. Here are some of the changes that would be ideal to implement:

- More weapons. Currently, there are only two weapons that don't mechanically differ that much. Some examples of weapons could be a weapon that doesn't cause an explosion, but it has an area of effect, or a weapon that causes low amounts of damage each turn to a player.
- More maps. Having procedural generation is nice and all, but maybe some different material would be nice. Maps could also be made larger, even if the game window was the same size. This could be achieved by showing the player only a portion of the map, but drawing more than they can see. Environmental effects could also be utilized and dependent on the map.
- A "movement bar" to restrict the amount of movement a player can do before they have to fire a shot. Currently, there is no reason or incapability to not run right next to your target, and guarantee a hit. By limiting the amount of time a player has to execute his or her turn could add a level of strategy and balance to the game.
- Online multiplayer was the only feature in the backlog that was not implemented at all. Implementing game logic towards a server/client relationship proved challenging given the existing codebase, and with many other features still not yet implemented that would also have to be put towards this relationship, it was decided to focus more on the other portions of the project, and try to make those as good as we could.
- Unique identifiers for the players, so that they can be differentiated from other members of the teams. Maybe, they could even have statistics maintained on their profiles, such as wins, losses, and damage given to other players.

This is by no means an exhaustive list, but it represents some ideas that were not in this first rotation of the product.

How to Maintain It

Of course, all of these changes could not be made without people writing these changes. There are different kinds of ways that this can be done.

- We continue to maintain it
- The open-source community takes off with the project
- We pay someone to maintain it for us.

<u>We Continue to Maintain It</u>

It would seem fitting to present this as an option, given we were the ones to create the project in the first place. It would probably be the case that we would all finish school, and then attempt to make this program economically viable. The hard part is determining what is economically viable. There is of course the potential profit to look at, but there is also the opportunity cost associated with going into this situation. What sort of job prospects do we have coming out of college? If these are greater than the opportunity to work on Wizards, then it would make sense to abandon the Wizards program. Let's say that working on Wizards is the best solution. What sort of profit could we expect from releasing this product? It might be beneficial to look at what similar games are priced for a rough estimate.

- Current Price on Steam: $14.99

(Source: http://licensingbook.com/wp-content/uploads/2012/03/WormsRevolutionLogo-copy-300x292.jpg DoA: December 7th, 2017)

- Free, but with in-app purchases (Source: http://2.bp.blogspot.com/-CBvHmQ4uzw4/UYtggQ8q4LI/AAAAAAAAGlk/GDd1jeppR_c/s1600/Angry-Birds-Logo.png DoA: December 6th, 2017)

Not too many games exist of this category that have become mainstream. This may explain why a game like Angry Birds could take off so well during the infancy of the phone gaming market. Many less professional versions of the artillery genre exist in collections on websites where ads are placed on them. This is another way to gain revenue and charge for playing the game, but this has a varied cost that depends on active viewership in order to gain profits. Armor games is an example of such a platform.

(Source: http://3.bp.blogspot.com/_plp4iaKyw9A/S_bXxSmdwqI/AAAAAAAAAF4/z-bm7F-_BMg/s1600/Armor-Games-juegos.jpg DoA: December 7th, 2017)

After all of the features were implemented, this would seem to be the price bracket we could put into the game. Sales would depend more so on the platform we chose to put it on, which is more of a discussion on deployment tactics. As for what amount is required, it depends on the lifestyle we wish to have. It is possible that we could desire nothing more than to live together in a cramped apartment, in which case our expenses would be much lower than under other more expensive circumstances

The Open-Source Community

The cheapest solution would definitely be to allow others to contribute to the project. The application has no proprietary requirements, and could easily be licensed to allow for open contributions. A game by the name of *Hedgewars* has gone down such a route, to moderate success.

Source (http://www.pingusman.com/wp-content/uploads/2013/04/Hedgewars-Logo.png DoA: December 7th, 2017)

This also carries a couple of requirements depending on desires for flexibility and pace. If we still want some control over what is put into the application, we will still have to monitor contributions. This can be done on GitHub using pull requests for temporary contributors, and full access to more trusted individuals. Not everyone has the best intentions at heart, and it would be important to cater the talent that pushes up commits. The last component to this is generating interest. If nobody knows our project exists, then how could we ever hope to attract developers? Making use of social media at this point would be crucial, especially for the more tech oriented platforms.

Paying Someone Else

Within this idea, there are really two different ways to approach this. One, we can continue to own the intellectual property, and hire others to work for us, or two, we sell the property to someone else. First according to the Bureau of Labor Statistics, Programmers make an average of around $79,840 per year. One programmer will not be sufficient by himself, so at least three would optimal. This would mean that we would need to pay them $239,520 to work on the software. In order to make this viable for us to pay someone as well, we would of course have to make more than this amount of money to make this venture worthwhile. As said previously though, it is hard to say what amount of money would be sufficient.

U.S. BUREAU OF LABOR STATISTICS

(Source: http://www.whorunsgov.com/wp-content/uploads/2015/08/us-bureau-of-labor-statistics.jpg DoA: December 7th, 2017)

The other option is to sell our IP to someone who would then take care of the project for us at zero cost to us. The profit would then be determined by the contract as to whether we would get paid in royalties per game sale, or if we would get all the payments in one lump sum. There may be some fees regarding hiring lawyers to deal with the contracts, but those are one-time costs at the most.