Maintenance Plan

Holy Guacacode

Overview:

Our deployment plan goal is a full release on the Google Play and App Store. Maintenance costs include items like app store costs, keeping the apps updated, getting customer feedback, and collecting data on app performance. We don't currently have any online support within the app, but a future update could possibly introduce leaderboards for fastest level completion. That would come with additional costs for databases. We would also need to pay developers to continue to update the game to keep customers happy.

App store costs:

The Google Play store only has a one time fee of $25. They take a 30% cut from any money made. They pay out once a month.

The App Store has a yearly cost of $99. Apple also takes a 30% cut. They also pay out around the beginning of each month.

Customer feedback:

There are a lot of options for collecting meaningful feedback. Doorbell seemed like a good fit for our game. Feedback can be submitted with screenshots included. This would be important for anyone experiencing bugs within the game that are easier to show than explain. The cost is $25/month for startups.

Collecting Data:

Continuous collection of performance data is important to predict potential problems and get to solutions faster. Even more so if databases are involved. If databases are implemented, we would invest in a resource like New Relic. New Relic manages and displays all database data and helps manage app performance. New Relic is $66/month.

Databases:

Powerful and reliable servers are important for a successful app. Most of the costs of a database are developing them. We could reserve a DB instance from Amazon for a year for $13/month. This is for the bare minimum and might need expansion in the future.

Developer Costs:

It’s not very practical to keep five people on a full time developer salary. We would have to cut down to one or two developers to fix bugs and implement new features. We likely wouldn’t have the profits to pay these developers a guaranteed salary. It would be more of an hourly basis when fixes are needed or a new feature is being implemented. $20-30 an hour/developer. This would have to be freelance developers who can work on a need based schedule. There would likely be a few hundred hours worth of work over the year between developing new features and bug fixing. 30 * 200 = $6000/year.

Cost Table

Item	Cost
App Store	$99/month $1188/year
Customer Feedback	$25/month $300/year
Data Collection	$66/month $792/year
Databases	$13/month $156/year
Developers	$30/hour ~200 hours of development ~$5000/year
Total one year cost	1188 + 300 + 792 + 156 + 5000 = ~ $7436

Moving Forward:

Depending on the popularity of the game, costs could ramp up with increased demand. Estimated database costs were calculated using the smallest database available. Larger ones get much more expensive, up to $200+/month. Additional developers would also be needed to sustain the growing population and needs. Customer feedback will be crucial to knowing what features are most in demand and what bugs are top priority. Wasted time means wasted money, so it’s important to only work on features that users will appreciate.

Other Costs to consider:

We could possibly put out ads for our game in order to accelerate popularity growth. This is a hard one to estimate due to the wide variety of possible advertising platforms. Most costs for mobile advertisements are put in terms of a cost per click or cost per install. Cost per install is usually $1-2. We could advertise on other mobile games in video or banner form.

It would get extremely costly to advertise on any widely popular games, so smaller games would have to be used. Larger companies spent up to $100,000 - $200,000 on advertising. Those kinds of expenses are not reasonable for a developing team releasing their first game.

Conclusion:

Keeping developers on staff is by far the biggest component of maintenance price. It shows how important easily extensible code is. With a poor foundation, we could end up doubling or tripling the amount we spend on developing because of the massive time sink. Taking a little extra time at the beginning saves a massive amount of time and money down the road.

Many of these price estimates were calculated assuming a one year subscription, but for some there were three and five year subscriptions available at a lower price/month. A maintenance plan looking more than one year out might find a lower cost estimate.

Sources:

https://newrelic.com/application-monitoring/pricing

https://aws.amazon.com/rds/mysql/pricing/

https://doorbell.io/

https://developer.apple.com/app-store/subscriptions/