Lucky 3 Studios

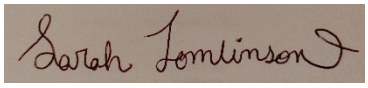
“Virus Hunter” Postmortem Report

8/13/2023

Team Lead: Sarah | Level Designer: Michael | UI Programmer/Designer: Amber | Programmers: George, Alina | Artist: Raz

Project Management

A. Project Planning and Scope – Sarah Tomlinson



Planning and scope are always important considerations in a project, and ours was no different. The biggest hurdle we had was in our first week and then our Alpha stage of development. We had a list of goals we wanted to achieve in the game by this time, but quickly got derailed by technical difficulties with Bitbucket and Git. And of course, being down one of our team members for a whole cost valuable time that could have been spent adding those initial ideas and would have resulted in more time at the end of the project to add additional elements, but ultimately, we ended up with a project we are all proud of. The initial scope was very ambitious and resulted in us basically having the entire project complete by the end of the Beta Release, so that the final week could be spent on testing, bug fixing, and implementing extra features and ideas that went beyond the original requirements.

Our life cycle was affected in that we didn’t have time to do a full-fledged test at each phase and spent far too much time in the “Pre-production” phase of the life cycle at the Pre-alpha and Alpha stage. It took almost a full week to diagnose and resolve the merge conflicts and begin our ability to update the project without any conflicts. Our foundation was delayed, which meant we had to readjust and push back some of our

goals to the following week. These issues were mitigated, however, before they even arose. That's to say that because the project was initially scoped to be almost fully completed by the time of the Beta release, we had some flexibility built into the schedule, already. Any other issues that came up were mitigated by reviewing the planned schedule and choosing what items needed to be completed immediately and what could be pushed to the next week or to the final release stage. For example, we originally wanted to have all three pickups in the game and functioning by the Beta Phase, but as we got closer to the day, it became clear we couldn't achieve it in time, so I reviewed the specs, decided on 2 that *needed* to be finished that week, then pushed the third to the following week to take a little pressure off the current week and ensure that the functionality of each was given proper time and attention.

This is also an example of a planning and scope decision that worked well for us: flexibility. We made sure to acknowledge early on that there would likely be things we had to push back and re-scope, which allowed us to approach the project with an open-mindedness that helped us quickly make any changes and not get stuck too far into one direction, causing too much wasted time. We also had an itemized checklist and timeline for each item that needed to be completed and added to the game, which helped us stay on task and know exactly who was responsible for what and when. This was created and maintained on Monday.com, allowing everyone to see what was being worked on and update tasks as they were in progress and completed. This way, instead of team members accidentally working on overlapping work or the same files was largely avoided, except in a few instances where two tasks required the same file to be edited (The character blueprint, for example). These decisions were made in a

dedicated effort to reduce any parts of the project needing to be re-done or two people spending time on the same aspect while no one is working on another important asset or element.

B. Development – RazaDon Willrich 

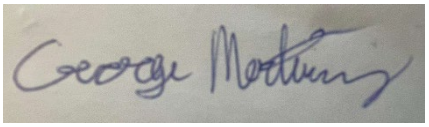
The development process of “Virus Hunter” went extraordinarily well, and the team was greatly impressed with the way it turned out. Many of our major decisions were made early in the development cycle to establish a clear vision and to avoid any sudden shifts. It’s generally good practice to tackle these decisions as soon as possible because they can pose complications in the long run if they’re postponed. In some cases, decisions may not be made immediately, but it’s important to take large ideas into consideration as they arise and address them thoroughly, as a team, in a timely fashion. One of our larger decisions was based around a theme for the project. Our overarching theme consists of a hunter that hunts infectious viruses, as the title implies. From there, we compared two potential sub-themes and discussed other ideas that would help make the product more unique. Initially, we explored a biohazardous theme that involves a player exterminating viruses inside of an infected life form/organism. On the other hand, we discussed a cyber security theme, where a player eliminates active viruses from a computer system. After deep consideration, we chose the latter and continued to build upon that idea. Making these types of decisions in advance really helped set a directional path for development.

For the most part, our technical programming aligned with the planning/documentation that we’ve created. In our weekly meetings, we planned out the methods and techniques we’d use to achieve what we were going for. Then, our

programmers would implement the features we went over in each following week. As an Art Director, I was mainly responsible for creating various assets to be used by the programmers and placed into the game. In early builds, some assets, like the Power-Ups, would utilize temporary placeholders to ensure that the code was behaving properly. In later builds, these placeholders would be replaced with their intended asset or mesh. This has shown itself to be an efficient way of maintaining a solid workflow.

In this project, everyone was a Designer because we all contributed to the way that the game plays/feels and provided some level of creative input. This type of thing may vary from project to project, it all just depends on the size of the project and team. However, one of the first steps to making a game is deciding who is doing what so that the project stays on rail, and that's something we all seem to have done very well.

Quality Assurance and Testing – George Martinez

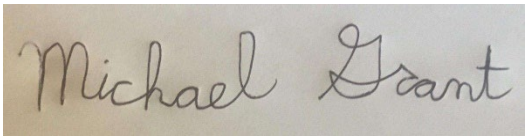


There is a laundry list of various processes we tested during the development of the project; most issues were found in-house by different members of the team discovering a variety of minor issues during their individual play tests. We also had outside testers, friends, or family, that provided feedback on certain phases of the project, as well as the instructor. For a small project and a small team like this, the testing process worked well because everything had a solid grasp on how it should work or appear. However, if this were a larger project with grander scope, this method of testing would likely reveal holes due to bias of team members or family, requiring a more thorough testing environment.

We rarely had any bugs that caused issues with the game design specifications, rather most of the issues involved additional mechanics or visual material that were added to improve the game project beyond the required scope. To help mitigate bugs, we organized blueprints to be as easily readable for other team members as possible, this helped others identify issues behind mechanics without having to rely on a particular individual that created the blueprint. Furthermore, accessible, and open communication through a shared application such as WhatsApp and Discord allowed for each team member to aid in solving problems by pooling everyone's expertise.

Although the current testing methodology worked for a small, 7-week-long project, it had fundamental flaws that would appear for projects larger than a school assignment. A dedicated playtesting team would help pinpoint issues that the biased viewpoint of the other development team members would overlook. As their job would be to look for flaws, and not just particular aspects related to a newly designed blueprint or level design adjustment. This would greatly increase the accuracy and amount of feedback in the project's development.

Tools and Practice – Michael Grant



I believe that many of the tools and techniques planned for the project proved quite useful throughout the development process. The first of these was the schedule in Week 2. The designation of key milestones for us to meet over the weeks helped us to meet each deadline with efficiency and maintain a structure for the work we were doing. But much more than our structured schedule, I believe ongoing communication was essential to our productivity. This was made possible using *Monday.com* and *What's*

App? The former of which allowed us to remain aware of how well were progressing through our self-appointed objectives, and the latter of which allowed us to provide one another with real-time updates and recommendations throughout our work. On top of that, our weekly meetings have proved to be considerably helpful. The routine discussions we held gave us the chance to plot out the important tasks that needed to be accomplished each week and touch base on any emerging concerns.

I believe that, of all the tools and techniques employed throughout the development process, none of them lacked usefulness. All of them were able to help us with the project in one way or another. Thanks to our constant updates and communication throughout the development process, we were able to tackle numerous roadblocks and complications we encountered, as well as remaining on schedule, and continuing to accomplish the goals we set for ourselves. I am confident that without the approach we planned or the resources we implemented; we wouldn't have been able to be so productive throughout the project's development. Overall, none of the things we had plotted out for our work on the project proved not to be useful.

I don't believe the design documentation itself was really the inspiration for many of the tools or techniques we chose to employ for this project aside from the timetable we established and the elements we included in gameplay. I believe that the choices to use *What's App?* and *Monday.com* were solely owed to my teammates' own intent for communication and updates. However, I feel that the design documentation gave us a very clear idea of what needed to be done. Our early discussion of the prioritized goals for the project allowed us to get a very detailed idea of what the product would need to look like and what needed to be accomplished. This provided us with the means to

establish a timeline of the objectives for each week, progressively building up to the ideal complete product. This entailed milestones like the number of rooms present, the viruses fought, U.I. components, items, and functionality. Thanks to our adherence to such a timeline, we were able to achieve a final, working game by Week 6 through determination and teamwork.

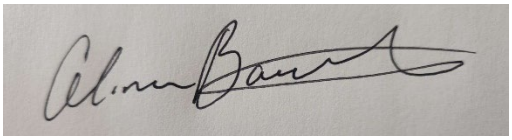
Communications – Amber Fondren 

The team had excellent communication throughout the project. Most people were around or checked in often. At each phase, there was always an update on what was going on or what someone was working on. It was very rare that the team was not aware of what was happening with the project. We also used Monday to keep track of what needed to be done, and that made it easy to keep one another accountable and check off our tasks as they were completed. Regular communication was kept through WhatsApp. It was because of the organization of tasks on Monday, and the consistent check-ins on WhatsApp that we were able to successfully fulfill our roles.

The part that did not function well with the team was the way our application choice WhatsApp works. If someone had a question or concern, it would sometimes be buried beneath the other conversations happening, as there is a single chat window. This made it cluttered and unorganized. I saw it mentioned, and am inclined to agree with the group leader, that a chat application like Discord would have solved this issue, as it allows a different channel for a different topic. It also allows for tagging, which highlights a message for whomever the question is intended for, or if it is more urgent. When the questions were lost in the discussion, it slowed production slightly, because in

some cases an answer was needed to continue. Luckily, the team was great at repeating the question if it had been missed, so everything was resolved soon after.

Something that helped throughout the project was the team leader checking in on where everyone was with their work. This usually prompted, if no one had spoken up before, someone to voice a concern or a question. Another thing was everyone's willingness to help. When someone was stuck, everyone would jump in and offer advice or solutions. There was very much a 'no one left behind' kind of feeling with the group. Lastly, there was always a weekly meeting in which we would go over our tasks and make sure everyone was on the same page. Aside from unavoidable technical problems, the way the team helped keep one another accountable, leaned on one another, and kept in touch solved any challenges that arose during production.

Conclusions – Alina Bautista 

Despite some technical difficulties, our project came together well in the allotted time given. Because of the team lead's helpful organization, everything was clearly structured and scheduled every week on Monday.com without becoming too overwhelming, which gave us ample time to work on everything that was to be added to the project. For any future game development teams, organization and structure are of utmost importance. It is all too easy for a project to lack quality and specific implementations and/or features because there is no clear goal or confusion amongst team members. Apart from weekly postings, daily communication was also important to keep up to date with what everyone was doing, and what everyone's status on certain assignments was. It was also a very effective way of providing help should another

team member need it at any time given that our method of communication was through a messenger on our phones.

Communication methods aside, another vital aspect of understanding and properly merging the things we work on separately without any misunderstandings was our decision to go with blueprints instead of C++, which ultimately would have made the project a bit more confusing to work on as a team. Towards the end of our project, when we were polishing things and making sure that everything coincided properly, we often shared blueprints to double check if everything was arranged properly. While obviously possible with C++, it would have made things much more confusing. The communication process alone was made simpler thanks to our decision to go with blueprints.

In conclusion, establishing a good foundation of communication and order is important for any team's success. Our team did well in all aspects and maintained good communication and quality throughout our time working on the project. While we had a handful of difficulties, thanks to both leadership and effective communication, as well as around the clock help from the team, we managed to create an amazing game that covered all requirements and then some.