ICS - BIOSHOOK™ PORTFOLIO

Team Leader
Philip-Nicolas Varga

Team Member
Elen Misura

1 - Proposal
Done by Elen

2 - Contract
Done by Phil-Nic

3 - Storyboard
4 - Fulfillment
The original proposition was only fulfilled to a medium extent. Core game mechanics are indeed present in the “final” product, but the game is far from complete. For example, in the storyboard tutorial and debrief screens cover educational aspects and gameplay teaching aspects of the game. However, in the final version, they are missing, and the levels are launched from separate JARs.

5 - Skills / Logs
During the course of creating this project, the team leader, Philip-Nicolas, and the team member, Elen Misura, worked on separate parts of the project hoping to complete each and put them together to achieve the final product. The team leader, Phil-Nic, focused on the game play and the levels, whereas the team member, Elen, focused on creating the menus, instructions, and visual aspects.

When creating this project, there were a few major challenges that we encountered. The first one of them was the lack of a plan on how to achieve the final product that we were going for. When we began, we were very excited about the idea and what the final project might look like that we did not have a clear plan and view as to how we were going to actually achieve that final project. Another problem was lack of communication. This proved to be a very big problem because even though we thought we understood the same thing about our situation and where we were with the project, we weren’t understanding the same thing. I believe we both thought that the other was further along than we actually were. And the last major problem we encountered was not working efficiently. We did not use our time as wisely as we should have.

We learned a lot about libGDX. libGDX is a great game building environment that has really great graphics. Because libGDX is made specifically for games, it was easier to make our project look visually appealing and work smoother. The way a real game should.

Phil-Nic

I encountered challenges in mapping. Creating textures and laying them out sounds simple, but turned out to be more time-consuming than I thought it would be. As well, trying to accurately represent fluid dynamics without simulating large amounts of particles was an interesting coding problem, that was fun to solve.
Regarding Java, I deepened my knowledge of object-oriented programming. Concretely, I learned about the useful instanceof operator.
Introspectively, I learned that the quantity and quality of my work is very much correlated to my motivation to do the work itself. Also, I learned that I become less efficient under extreme stress.

Elen

I learned that communication is something that you need constantly. I didn't really know this before. I thought that splitting the work the way we did would be okay and that would be the end of it. But it wasn't. You and your team members need constant communication and you need to be on the same page all the time to be able to succeed. I learned that I am not very engaging when it comes to communication, which is not a good thing.

Together, the team spent about 220 - 250 hours on the project. Philip-Nicolas worked on this project for 120-150 hours, and Elen worked on it for about 100 hours.
The unfinished programs that were handed in were 100% my work.

6 - Conflict Resolution

When things did not go well, the team leader and team member tried to get together to talk through what was happening and try to figure out how to fix it. Unfortunately, these talks that were had once something did go wrong or there was a problem, did not last long and were mostly only about that specific topic, rather than including other topics and making sure that everybody was on the same page.

Philip-Nicolas

As the team leader, I did not notice that things were not going well until late in the project. There were little interpersonal issues to deal with. However, if I had pushed harder, small problems would have been an indication of work. My fear of seeming "too bossy" caused me to stand still while small issues piled up.

7 - Technical Manual

The game is quite simple. The player has four gauges to keep track of: Health, Oxygen, ATP, and Protein.

The player “dies” when the health gauge is empty, so it is important to keep track of it at all times. ATP is used for movement and healing. Once the player runs out of ATP, their only movement comes from brownian motion, bumping into other cells, and currents if applicable. Healing takes protein and ATP. Running out of oxygen should cause the player to rapidly lose health.

Movement in the game is done through the arrow keys, or the WASD keys. For long journeys, time may be sped up using the C key, slowed using the Z key, and reset using the X key. In order to change the view based on surroundings, the player may zoom in and out using the L and O keys, respectively.

Resources are gathered by contacting other cells. Red blood cells contain oxygen, protein bundles contain protein, and glucose transporters contain glucose. Every collision with these types of cell transfers some resources to the player.



The player accumulates points by colliding with round viruses.

The player loses health and does not gain points when they collide with pointy viruses.

The game should end when the player reaches 20 points, or when the player dies. A win allows the player to advance to the next, more difficult level. An infinite mode should exist in order to test high scores.

8 - Biggest Challenges

The following challenges were faced, but were overcome with minimal success.

Lack of plan - The lack of a coherent plan understood by both group members decreased motivation, caused confusion, and made progress stagnant over the first two weeks of the project.
Communication - Communication was sparse within the group, and caused lack of clarity on requirements, slowing down the project.

Programming challenge: Putting the pieces of the project together, we could not figure out how to do that. Aside from that, we couldn't really help each other out much because we did two completely different pieces of the project. So we didn't know much about the others piece of the project.

9 - Biggest Success

The biggest success that each one of us individually finished our part of the game. The team leader, Phil-Nic, finished all three levels of the game and the team member, Elen, finished all the menus, instructions, and visual aspects. We were able to finish the game, however, not put our parts together. This resulted in our semi-finished game. Another one of our biggest successes were when we actually handed something in rather than having nothing to hand in and getting a zero on this project, we managed to hand something in and salvage some marks.

Programming success, Philip-Nicolas: Fluid dynamics system. This was a captivating engineering problem. Realistic representation of flow without simulating all the particles. It was solved by manually adding vector regions to the map, and moving cells along those vectors. Afterwards, damping, brownian motion, and masses were tuned for realism.

10 - Regrets

I regret not having argued more for my the original grouping, which (if successful) would have kept me motivated for the duration of the project, leading to better work, as I had been looking forward to making a big, awesome project for most of the semester. As well, approximately a week's worth of work had already been done, so I would not have wasted that work, and I would have had time to add more features.
Within the actual ISP group, I regret not taking charge enough, in order to fix group issues. Additionally, I regret having let my lack of motivation stop me from being effective early in the project. A final regret I have is that I spent a lot of time working on features that did not end up being implemented due to the time management issues our group faced. They are just sitting there, in the code, unused, and it breaks my heart.

11 - Self Evaluation

Phil-Nic

As a group project, the final version that was handed in was not satisfactory. Based on the fact that the game is unfinished, and not tuned properly, I would rate it very badly. I would expect to be paid money to sit for an hour playing it, and there is absolutely no

chance I would buy it. However, this was not a group project. The version that was handed in was 100% my work. In addition to working on what I was assigned in the group, I also picked up a lot of the rest of my group's slack. For example, maps that were reported to be completed, simply weren't. And, since as written in the challenges section, the mapping took longer than usual, I spent so much time making more maps that I was not able to add them to the game. Code that was sent over, that was supposed to be working, didn't. As communication was sparse, these issues took extra time to resolve. Given the fact that I took up a large portion of extra work, I would give a better score to this project.
Finally, as a group project, I would give it a 4.5 - 5 out of ten, and as an individual project, I would give it a 7.5 - 8 out of ten.

Elen

When working on this project, I wished that I had more time work on it because I was not familiar with the program that we were using and did not know many commands and tricks around it. I wish I had more time to explore libGDX and get to know it better so that I was able to achieve a better final product. I also wished that we had more time in general so that we could of had time to get together and explain the code that we wrote to one another and so that it would have been easier to put together. Had we done that, we would have not had that much trouble putting it together because we would of understood the final program better.

12 - Project Idea

If we had to come up with our own idea for a project instead of being assigned one, it would be making something not so much educational but rather more adventurous and interesting to us as opposed to focusing on making it educational. Another idea for a project that we would like is to make something that will help us on later, maybe in life, maybe in school. Something that we can use. For example, a graphing calculator. Or any other type of calculator that will come in handy in math so that we do not have to search for it online.