Egg Runner

# CS 476 Software Development Project

Brandon Huzil……….ba.huzil@sasktel.net

Viktor Fries……………..Vafries@gmail.com

# The problem and our motivations

When coming up with our project idea for our software development project, we had two main goal categories in mind that we wanted our chosen project to help us achieve. First, we recognised that we needed to decide on a project that is in some way meaningful to us. Whether through a project we recognize as an important tool for the world to have or a project that we both knew we would enjoy making. Software engineering and software development can be a difficult and trying task. Road bumps occur and new requirements can come down the pipeline at any time. These trials can be difficult on team moral and the day you don’t want to work on your team’s project is the day your project has failed. If we came up with a project that interested us or we thought was important for the world to have it makes losing moral and motivation a much less likely occurrence. Our second goal was to create a project that challenged us in unique ways other than to just pick a difficult project. For the sake of future workplace success, we need to learn how to make use of new technology, learn how to communicate effectively to reduce conflicts between each others contributions, challenge ourselves in ways that lie outside the realm of programming such to diversify our skillsets, and finally learn how to maximise our success within the confines provided.

To come up with a project that would fully satisfy our goals we needed to look for inspiration. What has been done already that we can improve in some way or what has been attempted to do that we could offer our own solution to. When looking at what’s been popular in the gaming industry lately, we came across two unique examples from Nintendo. Super Mario Maker released in 2015 and Super Mario Maker 2 released in 2019 were both massively popular and enjoyed Nintendo experiences. They took the tried and true formula of the 2-Dimensional Mario platformer and added to it to appeal to players creative side. Both games see players building levels from a set of themes and tile sets and placing classic Mario objects, obstacles, mechanics and enemies. Players then submit their levels and then go on to play any of the myriad of other players levels. Its no wonder its successful, Mario mechanics are a foundation for success, give the players the ability to play the creative architect, now you have endless content using solid mechanics as a base. It’s a brilliant system. The only problem is do you have the game? Do you have the console? That’s an expensive road block making Super Mario Makers one flaw being the accessibility factor. This was our problem to solve. Make a platforming game with level editor that relied on player content that is more accessible then Super Mario Maker or Super Mario Maker 2.

But how would making a 2D platformer with level editor component challenge our goals and fit the requirements of this project. First is it meaningful to us. Both of us play games, both of us take part in groups and communities that encourage the creating of ideas in ways that are practical to entertainment. The chance to produce our own piece of entertainment media is exciting and its that excitement that is going to keep us motivated to complete this project regardless of the difficulties and set backs we may face. Now what about challenging us in unique ways. We would make use of the Unity development platform which is an environment that was new to us both this semester. This project would require each game feature to be double implemented. The in game logic needs to exist for a workable mechanic but we also need to include this mechanic in the editor and figure out how to include it in the database. This means that we need to communicate what is mechanics are key to create a minimum viable product and what mechanics are key to create an engaging game that is fun to play and allows creative initiative. This also means we need to communicate in how we convert a stored level into an actual game logic with all mechanics working in the numbers and locations they need to be. We would need to create art and assets for our game which lie outside the realm of programming, an we need to ensure proper player feedback for their in game actions which is in the realm of communication. And lastly, we would needed to incorporate a development model that ensures we have a finished product at the end but is flexible in that we can seamlessly add new mechanics while having a potential end product before and after. Our project needs to be web deployed which is exactly how we will make our game more accessible. Our Problem was decided and it fit our goals and constraints.

# Feasibility

Egg Runner will be a web deployed platforming game where users make use of the level editor to create their own levels that others can play. As it is web deployed more people will have access to our game then have access to Nintendo’s Super Mario Maker’s. This satisfies our problem to make a more accessible version of the platforming game with editor design. But is it feasible?

First and foremost, our project is economically feasible. We can produce Egg Runner off of a free version of unity and we have access to a free database as University of Regina Computer Science students. This project is economically feasible as it will not cost anything to produce.

Unity is a powerful tool that is time tested as a standard in game design at the level of our project. Unity will have no difficulty handling what we want out of Egg Runner. The databases we can create as University of Regina students should also have no foreseeable problem holding our levels. In terms of the software tools at our disposal they are fully technically feasible. Although our knowledge and skills in unity are limited, we have enough prior knowledge and access to enough resources to help us through. Unity makes use of C#, a language that share much akin to languages such as C and Java, both are languages we both have experience using in academic settings. The stretch to code in C# should not be an overwhelming hurdle, in fact we suspect it will be a surprisingly comfortable learning curve. Our knowledge in web-based languages such as JavaScript or PHP presents a greater anticipated difficulty then C#. We are yet unsure how easy it is to connect a unity project to a database or how much web-based code will be needed but despite this we are confident in our abilities to overcome this challenge. To remedy this possible difficulty, we can scale our use of databases accordingly. Players don’t necessarily need an account. Simply storing levels and retrieving them is well enough. As time permits, we could build off of that base to add things such as high scores or accounts. The resources online for help in using unity and C# scripting in unity is vast and of a very high quality, in addition there are even larger amounts of community resources, help, and tutorials online. We anticipate few difficulties in using unity as we will make use of the vast quantity of resources at our disposal. All together we see this project as being technically feasible thanks to the reputation unity has, the online resources available to us, and the fact that difficult problems can be scaled back and we will still solve our problem and exist within the confines of the project requirements.

As Egg Runner is a game produced for our software development project class the user requirements are very loose. We can consider the project requirements and guidelines to act similarly with user requirements. We know the project requirements will not change half way through development and so we don’t have to consider changing requirements when planning development of our game. As our game involves a level editor for players to play each others levels, online deployment is a must in order to store databases, this in turn satisfies both our requirements to have our project web deployed and use 3 tier architecture. All of our requirements are met with our plans and no changing requirements will come down the pipeline meaning Egg Runner is operationally feasible.

The type and style of game we are making is not unique. Of course, our main inspiration is of the same style game. But because our problem is to develop the same type of experience yet more accessible, we can consider our game Egg Runner to be a modification on that formula. Our problem is not to reinvent this style of game or to improve on its core gameplay, it is to modify the accessibility of it. It is in this way an improvement over our influence of Mario Maker. Similar style games have also existed before 2015s Super Mario Maker. In 2008, created by Media Molecule and published by Sony Computer Entertainment was Little Big Planet. Little Big Planet and its subsequent games were in the same style of Super Mario Maker although they seems to focus more on original platforming and puzzle mechanics that users could put in their levels for others to play. However it too could be considered to be lowly accessible as not everyone had both the game and a PlayStation 3 just like not everyone had The Nintendo games and a Wii U or Switch. Once again our game Egg Runner will be superior in terms of accessibility.

With the Super Mario Maker series and Little Big Planet series we know this style of game his well enjoyed. We do not need to do any acceptance testing for game style enjoyment thanks to these acceptable meaning our game will be feasible as an entertaining experience. We will improve on these formulas through accessibility which is the major drawback or both of these influencing series.