

REYKJAVÍK UNIVERSITY

FINAL PROJECT

Final Report

Group 19



Franklin Þór Vale
Jón Atli Guðlaugsson
Ólafur Barkarson
Snorri Vignisson

May 14, 2021

Instructor: Haukur Steinn Logason

Examiner: Torfi Ásgeirsson

Client: Flow Education

Contents

1	Introduction	3
1.1	Introduction	3
1.2	The Team	3
2	Project Organization	3
2.1	Methodology	3
2.1.1	Team Roles	3
2.2	Working Agreement	3
2.2.1	Changes in the working agreement in the past 3 weeks	3
2.3	Communication	3
2.3.1	Changes in communication during the last 3 weeks	4
2.4	Coding Rules	4
2.5	Project Scope	4
2.5.1	Estimated Project Scope Chart	4
2.5.2	Estimated Scope and Current Time Spent	5
2.5.3	Time Distribution Between Team Members	5
2.5.4	Burndown Chart of Completed Sprints	5
3	Product	6
3.1	Product Backlog	6
3.2	Evolyte Overview	11
3.3	Final Product	11
4	Progress Overview	12
4.1	Sprint Overview	12
4.2	Animations	12
4.2.1	DoTween Animator	12
4.2.2	DoTweenRepeat Animator	12
4.3	User Interfaces	13
4.3.1	User Interface breakdown	13
4.4	Database	15
4.5	Builds	15
5	Risk Analysis	15
5.1	Risks We Encountered	17
5.1.1	Git Merge Conflicts	17
5.1.2	Final Projects and Papers in the Same Day or Week	17
5.1.3	Final Exams	17
5.1.4	Not Getting Evolytes Assets on time	17
5.1.5	Working Together during Covid-19 times	17
5.1.6	QA taking more time than Expected to Review	17
5.1.7	Competitive Athletes in Group	17
5.2	How our Planning Reduced Risks	17
6	Development	18
7	Work during Covid-19	18
8	Final Words	18

1 Introduction

1.1 Introduction

Evolytes is a platform that teaches elementary level math faster and more effectively in a unique, fun and engaging way. Evolytes is a platform that combines book, mobile game, individualized content and an online web monitor. The idea is that playing a game makes math more fun and engaging for children. The game revolves around a character that the students can choose from premade characters with different story arches. Then they will go into the Evolytes universe. Their goal is to capture other Evolytes, battle Evolytes, collect trophies and chests and so forth. To be able to move forward in the game students have to answer math questions correctly.

Making the game is our final project for a BSc. in Computer Science at Reykjavík University. This game has been made before and our mission is to make the game in a new programming language with a new Game engine to make it more scalable, easier to run and add in some extra features to improve user experience.

1.2 The Team

The team consists of 3 computer science students and 1 software engineering student. The team in its whole had never worked together before this project as it was initially two groups of two members joining forces through their passion in video games.

2 Project Organization

2.1 Methodology

The Kanban methodology was chosen because Flow Education was already using the Kanban methodology and we wanted to stay consistent with their ideology.

2.1.1 Team Roles

Owner: Mathieu Grettir Skúlason

Quality Assurance: Sigurður Gunnar Magnússon

Team: Franklin Þór Vale

Jón Atli Guðlaugsson

Ólafur Barkarson

Snorri Vignisson

2.2 Working Agreement

Due to the Covid-19 pandemic we mostly work on the assignment at home using Slack and Discord for our communication. Team members mostly work on the assignment in pairs of two or alone depending on what is being added. All team members are very active on Discord and Slack and that is where most of our communication takes place. We have biweekly sprint meetings with the product owner where we go over what we have accomplished and what did not go as planned and decide what we are going to be working on in the next sprint. We have weekly meetings among ourselves every Tuesday at 15:00 where we discuss what we are working on and can ask each other for help. Team members also use this time to look at upcoming assignments and discuss availability where all team members are available to work together at two more specific days during the week 4 hours at a time.

2.2.1 Changes in the working agreement in the past 3 weeks

For the remaining 3 weeks the group decided to meet everyday and work together for at least 8 hours a day. Every member should be fully available for this remainder of the course since all of us had finished our exams, projects and assignments. We also agreed on a working location at Snorri's apartment. Our days started at 9 am and we would always work until at least 5 pm. All team members were always prepared to put in extra hours if needed and be easily reachable.

2.3 Communication

We have biweekly sprint meetings at 14:00 - 15:00 where we discuss the last sprint, problems we had during the sprint, how the sprint went and we decide on what to accomplish on the next sprint. We also have weekly meetings where we discuss the current sprint, make sure we're on schedule

and get help from team members if needed. Meetings take place on Discord voice calls or in face-to-face meetings. Overall communication with the company takes place on Slack. Team members communicate through our Discord server, Slack and messenger. People generally work from home but that might change depending on the Covid-19 restrictions. We also have weekly meetings with our instructor. In those meetings we discuss our progress and next assignments within the course.

2.3.1 Changes in communication during the last 3 weeks

For the remaining 3 weeks the group met up everyday and that improved our communication drastically. Interacting in person everyday was much easier, getting help from fellow team members became less of a nuisance than always sending messages back and forth and meeting on discord and screen sharing. Having all members sitting next to you helped improve our workflow, problem solving and kept us more focused on the task at hand.

2.4 Coding Rules

Coding conduct rules were established to make the code more stylishly consistent and easier to navigate.

- All code should be in English.
- We will keep global variables to a minimum.
- We will add space between operators and commas.
- Variable names will be written using camel casing.
- Function names will be written using camel casing.
- Class names will be written using camel casing.
- File names will be written using camel casing.
- Code within braces shall be indented by one TAB (no spaces).

2.5 Project Scope

The group along with the product owner laid down the requirements for the project, after this the group used planning poker to better estimate the time each requirement would take. Then the group took the time of all the requirements and used that as an estimate for the implementation and testing part of the project. For the other categories we used the time we currently had put in for each category as an estimate and took a rough estimate on how much extra time would be put into these categories.

2.5.1 Estimated Project Scope Chart

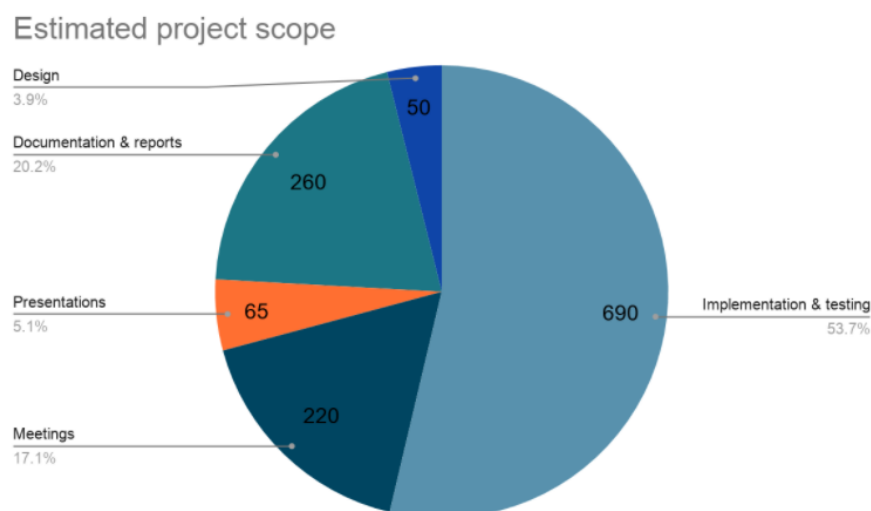


Figure 1: Estimated Project Scope Chart

2.5.2 Estimated Scope and Current Time Spent

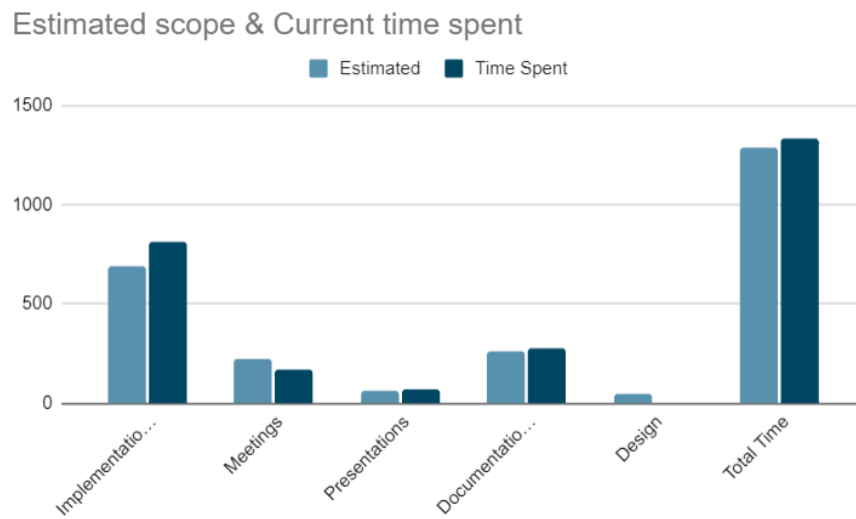


Figure 2: Estimated Scope and Current Time Spent

2.5.3 Time Distribution Between Team Members

This is the current time each team member has spent in hours. As you can see it is distributed pretty evenly between all group members.

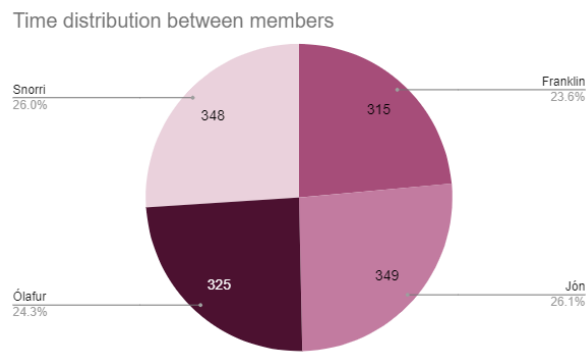


Figure 3: Time Distribution Between Team Members

2.5.4 Burndown Chart of Completed Sprints

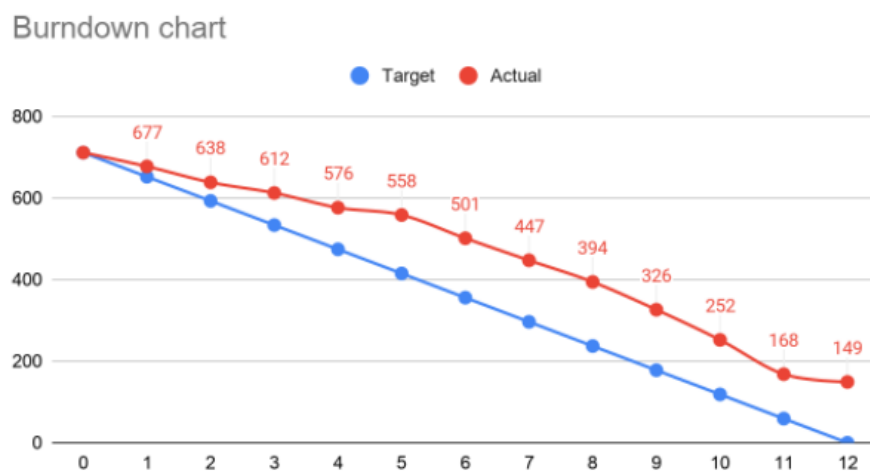


Figure 4: Burndown Chart of Completed Sprints

3 Product

3.1 Product Backlog

The product backlog is where we keep our tasks. The tasks in the backlog are all given value for priority, estimated time and how far along we are with each task. Prioritization is ranked on a scale from A to D.

A: Essential
B: Important
C: Desirable
D: Extra

Estimated time in the backlog is used by assigning story points where we used the numbers in the Fibonacci sequence (excluding 2) where the numbers map to the values.

1: About 1 hour
3: Less than half a workday
5: More than half a workday
8: A full workday
13: More than a full workday
21: Two full workdays

We have four different states for tasks, which are the following.

Planned
In progress
Testing
Done

Lastly we colored the category and requirement **orange** for those items that are new additions to the game and were not in the older version. All requirements that have been added after the list was first made are color coded with **green**. For this report only one requirement has been added at the request of the examiner. Here is the table sorted by priority first, estimated time then and status last.

Category	Requirement	Priority	Time	Status
Communication with backend	Implementation of general end points after demand to communicate with the back-end.	A	21	Done
Communication with backend	Implementation of an communication layer at fitting positions in game.	A	21	Done
Interface/Crystal selected	Player sees the animals he has on hand and what Evolytes he has to put into his lineup. We want to update this view and make it easier to navigate and use.	A	21	Done
Interface/General Design	Implementation of general code to adjust the size of display to different screen ratios to use in different devices in this project.	A	21	Done

Category	Requirement	Priority	Time	Status
Skeletal animation for Evolytes	Implementation on animations for Lighter Evolyte.	A	21	Done
Battle interface	Implementations of multiple abilities which have multiple effects on the animals strength, life and energy.	A	13	Done
Battle interface	Implementation of the use of food in the battle interface.	A	13	Done
Interface/Bag	Ability to use an item within the bag.	A	13	Done
Battle interface	Implementation of reusable code for abilities that give an Evolyte a shield which spans over a few turns.	A	8	Done
Battle interface	Implementation of reusable code for abilities that put opponents to sleep.	A	8	Done
Environment	Implementation of a component that can hide and appear according to what quests player has finished.	A	8	Done
Interface/Animal details/Abilities	Ability to see and upgrade an Evolyte ability if the Evolyte has gained enough XP. - To upgrade an ability we use an "ability token" which you need to collect and then spend for a desired trick.	A	8	Done
Interface/Evolytes overview	The ability to select an animal to get further information.	A	8	Done
Interface/Bag	Select a monster to use an item on.	A	8	Done
Interface/Quest interface from in-game character	Ability to see what items character wants.	A	8	Done
Running game	Controller that gets data about the player when the software has gone through running in the background.	A	8	Done
Skeletal Animation for Evolytes	Implementation of animations for Bloom Evolyte.	A	8	Done
Skeletal Animation for Evolytes	Implementation of animations for Dart animal	A	8	Done
Battle interface	Update the interfaces for animals that inflicted damage or have used up some energy.	A	5	Done
Battle interface	Implementation of the use of crystals to capture other animals in the battle interface with the corresponding interfaces	A	5	Done
BattleInterface/Using bag items	Implementation of a screenshot with an animation that implies what the effect of the item is and updates animal accordingly.	A	5	Done
Environment	Implementation of a component so player can interact with an item	A	5	Done
Interface/Interface after victory	Screenshot that shows the victorious results and overview.	A	5	Done
Interface/Interface after victory	Screenshot that shows the results of the loss and an overview.	A	5	Done
Interface/Questinterface from in-game character	Ability to see if player has all items to finish a quest.	A	5	Done

Category	Requirement	Priority	Time	Status
Interface/Receiving gift	Overview of what you receive for winning a battle.	A	5	Done
Interface/Select student	User has the ability to see all characters accounts on that email.	A	5	Done
Running game	Controller that verifies if user has a valid log in and shifts interface accordingly.	A	5	Done
Running game	Controller that loads in environment accordingly	A	5	Done
Skeletal Animation for Evolytes	Implementation of animations for Ava Evolyte	A	5	Done
Skeletal Animation for Evolytes	Implementation of animations for Fieree Evolyte	A	5	Done
Skeletal Animation for Evolytes	Implementation of animations for Kindle Evolyte	A	5	Done
Skeletal Animation for Evolytes	Implementation of animations for Fyrbat Evolyte	A	5	Done
Skeletal Animation for Evolytes	Implementation of animations for Cinder Evolyte	A	5	Done
Skeletal Animation for Evolytes	Implementation of animations for Badger Evolyte	A	5	Done
Skeletal Animation for Evolytes	Implementation of animations for Bran Evolyte	A	5	Done
Skeletal Animation for Evolytes	Implementation of animations for Fable Evolyte	A	5	Done
Skeletal Animation for Evolytes	Implementation of animations for Kayla Evolyte	A	5	Done
Skeletal Animation for Evolytes	Implementation of animations for Nala Evolyte	A	5	Done
Skeletal Animation for Evolytes	Implementation of animations for Swirly Evolyte	A	5	Done
Skeletal Animation for Evolytes	Implementation of animations for Leela Evolyte	A	5	Done
Skeletal Animation for Evolytes	Implementation of animations for Tremor Evolyte	A	5	Done
Skeletal Animation for Evolytes	Implementation of animations for Storm Evolyte	A	5	Done
Skeletal Animation for Evolytes	Implementation of animations for Stampy Evolyte	A	5	Done
Skeletal Animation for Evolytes	Implementation of animations for Slender Evolyte	A	5	Done
Battle interface	Implementation of reusable code for abilities which span over a few turns and they either help or hurt the animal.	A	3	Done
Battle interface	Ability to run away during a battle.	A	3	Done
Environment	Implementation of a component that allows player to press on an item or person in the environment according to their distance	A	3	Done
Interface/Animals overview	Player sees the animals he has on hand.	A	3	Done
Interface/Bag	Show a description of an item within the bag.	A	3	Done
Interface/Battle interface from in game character	Ability to "Accept" or "Decline" battle.	A	3	Done
Interface/Login	Ability to type in email address and password and a button to log in.	A	3	Done

Category	Requirement	Priority	Time	Status
Interface/User information	Button to switch user and log out.	A	3	Done
Interface/When animal is captured	Screenshot with a image of the animal just caught.	A	3	Done
Skeletal Animation for Evolytes	Implementation on animations for Leva Evolyte	A	3	Done
Skeletal Animation for Evolytes	Implementation on animations for Spring Evolyte	A	3	Done
Interface/Wait-Interface	Wait Interface for in game character. Displays how much time has to pass before being able to accept a quest, battle, reward or opening a chest.	B	21	Done
Interface/Upgrading an abilities	Screenshot that displays that player has got a new ability.	B	13	Done
Skeletal Animations for Evolytes	Implementations of animations for Sprinter Animal	B	13	Done
Environment	Implement zones where wild opponents can appear for battle	B	8	Done
Interface/Animal details/Animal evolution	Display of different evolutionary branches according to the selected animals.	B	8	Done
Store/UI	Ability to purchase items in game (Store UI)	B	8	Done
Interface/Questinterface from in-game character	Ability to see what reward you will get for finishing quest.	B	5	Done
Other	Implementation of a loading screen, which shows the user the progress of the data being loaded in.	B	5	Done
Skeletal Animations for Evolytes	Implementations of animations for Buster Animal	B	5	Done
Skeletal Animations for Evolytes	Implementation of animations for Leya Animal	B	5	Done
Environment	Implementation of portals that load a scene.	B	5	Done
Interface/Animal Evolution	Ability to select to not evolve animal.	C	8	Done
Interface/Bag	Implementation of 5 different types of filters so you can filter by object type	C	8	Done
Environment	Implementation of a component that starts animations when called when player is within some radius.	C	5	Done
Interface/Gift interface from in-game character	Ability to see a reward that a character wants to give player.	C	5	Done
Interface/Animal Evolution	Ability to select and deselect an animal.	C	3	Done
Interface/Battle interface from in-game character	Ability to display animal that in game character will use in battle against player.	C	3	Done
Interface/Battle interface from in-game character	Ability to see rewards for winning an in game character.	C	3	Done
Other	Implementation of error handling screens.	A	13	In Progress
Interface/User information	Ability to change the music and sound in the game.	B	5	In Progress
Interface/Details about abilities	The ability to see an abilities properties for a particular animal.	B	13	Planned

Category	Requirement	Priority	Time	Status
Environment	Implementation of a component that changes its colliders so that characters can cross over and go under a bridge or other similar items.	B	5	Planned
Character driven storytelling in game	Implementation of different scenes that tell the games plot	C	21	Planned
Character driven storytelling in game	Implementation of different scenes that tell the games plot	C	21	Planned
Animation and skinning	Implementation of skinning controllers on animals.	C	13	Planned
Character driven storytelling in game	Implementation of components that initiate a scene according to determined conditions that move the story forward.	C	13	Planned
Character driven storytelling in game	Implementation of the environment and backend communication that indicate the forward progress of the player.	C	13	Planned
Animation Skinning	Implementation of various loading for skeletal animation in regards to skinning.	C	8	Planned
Environment	Implementation of a component that displays a circle under that item which the component is connected to. This would indicate if an item is threatening or not.	C	8	Planned
Environment	Implementation of speech bubbles with an image when player is with in some radius.	C	8	Planned
Interface/Comic book interface	General Interface that takes in information about what comic strip to display.	C	8	Planned
Environment	Implementation of a component which would display or hide items in world according to animal level.	C	5	Planned
Environment	Implementation of a component that starts animations when called when player is within some radius.	C	5	Planned
Environment	Implementation of a component that displays or hides an item in the game world which the component is connected to in regards to how many animals the player has. This is used to limit the area for players that have at least 2-3 animals on hand.	C	5	Planned
Animation Skinning	Implementation of various abilities in regards to skinning.	C	3	Planned
Interface/Chest Interface	The ability to choose to open a chest.	C	3	Planned
Widgets	Animation widget that makes a non-Computer programmer capable of editing and setting up animations	D	21	Trashed
Interface/Bag	Implement a slider for when the bag has more items than it is able to show in one screen	D	13	Trashed
Interface/Upgrading an ability	Ability to see a video with the new animation.	D	8	Trashed
Interface/Upgrading an ability	Ability to see the change to the ability's icon and the new name of the ability.	D	5	Trashed
Interface/Animal details/Information	Ability to see the height, weight and date of the acquisition of the animal.	D	3	Trashed
Interface/Bag	Ability to remove an item from the bag and send a corresponding call to the backend.	D	3	Trashed
Interface/Details about abilities	The ability to see the abilities for a particular animal by opening a video player that plays its animation.	D	3	Trashed

Table 1: Product Backlog

3.2 Evolyte Overview

Accompanying this report, in Appendix II, we included a detailed description of every Evolyte that has been implemented excluding a few new ones. There you can see what abilities we have made for each Evolyte and their families which give a better perspective on the game in its whole.

3.3 Final Product

- One in game island ready for use
- Wide range of Evolytes with unique abilities??⁰
- UI scalable to different screen proportions
- The game should run faster than previous iteration
- Game resolution will be same as screen resolution
- Improved user experience
- Game should be playable on more devices
- Extras that we think are essential to improve the game
- It should be easier to add and improve things in the game later on

4 Progress Overview

This project is split into 12 different sprints where the first three and the last three were week long sprints, and the middle 6 sprints being biweekly sprints. The reason being that during the first three weeks we wanted to get off to a quick start and accomplish as much as we could. Then during the 6 middle sprints we came to the conclusion that due to other coursework we had to change to biweekly sprints which allowed us to plan more ahead and adapt according to our coursework. However, we still held onto the meetings every week and used one as a quick status meeting to help team members with any problems they might have encountered and the other as a sprint reviewer and planner.

When the sprints were two week sprints they started on Tuesdays at 15:00 and ended on Tuesday at 14:00 two weeks later. During the first hour we all go over what we have completed and what problems we encountered. Then we show our work and get feedback on improvements from our quality assurance representative (in regards to animations), after concluding that we set tasks and goals on what we wish to accomplish at the end of the next sprint. Then we switched back to weekly sprints because other assignments, exams and projects were finished and we could put all our attention on finishing our final project.

4.1 Sprint Overview

We created a sprint overview which contains detailed information about each sprint, the product backlog, sprint retrospective and our sprint burndown chart, that accompanies this report in Appendix I.

4.2 Animations

During our work on the animations and each animal controller we found ways to trim our process. That was Done by making Reusable controllers. These controllers were kept under Assets/Resources/Abilities/ReusableControllers.

4.2.1 DoTween Animator

We used DoTween when we were creating Evolyte abilities that only played once and used our reusable animator controller so we did not have create a new one for each animation which saved us time.

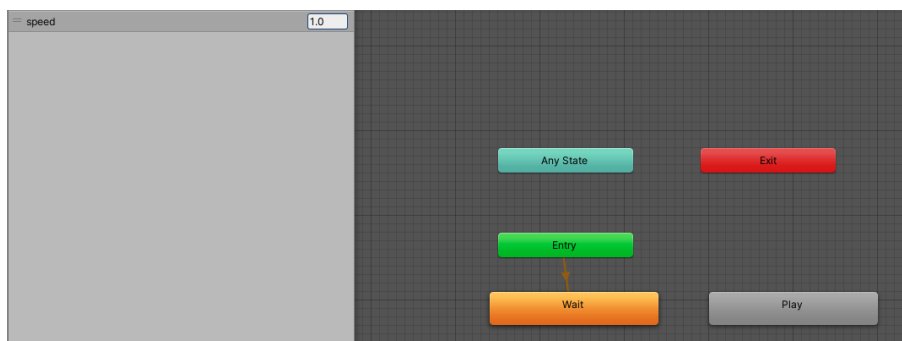


Figure 5: DoTween Animator

We implemented a variable controller that allowed us to easily alter the speed of our animations in the DoTweenAnimator. This made creating animations easier since we did not have to change every key frame to change animation speed, we could simply speed it up.

4.2.2 DoTweenRepeat Animator

When we were creating abilities that span multiple turns, are damage based or need to be removed when an Evolyte is too fatigued to fight we needed to make some changes. We had 3 different states. The start state played once and then went straight to the repeat state which was on a loop until the finish state was called under the above mentioned circumstances.

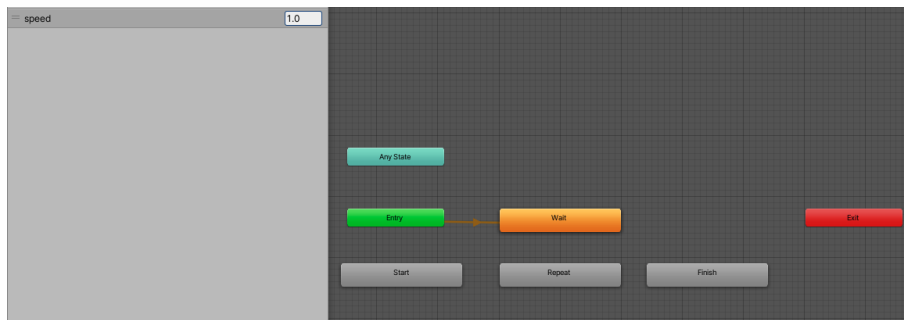


Figure 6: DoTweenRepeat Animator

As you can see we still have the speed variable to manipulate the animations which allows for easier adjustment. When creating the DoTween Repeater we had to create 3 separate animations for all the different states.

4.3 User Interfaces

User interfaces were a major part of the project. Most of the interactions the student has with the game are through these user interfaces (UIs). To minimize time spent on creating these interfaces we focused on creating reusable components with generalized code, that way we were able to reuse the same components on many different UIs which drastically reduced the time spent as well as the complexity of creating these UIs

4.3.1 User Interface breakdown

The user interface that is displayed in *figure 7* shows multiple reusable components that are puzzled together to produce a fully functional UI. Most of the parts that make up the UI are reusable.



Figure 7: Battle UI

On *figure 8* you can see the top panel of the Battle UI which is also used as the top panel on multiple other UIs. The panel features an image displaying the character that the student is interacting with, as well as the type of interaction that the character offers.



Figure 8: Top Panel

On *figure 9* you can see which Evolytes the character you are battling against has on hand. Each crystal uses the same component yet dynamically loads each monster. This is a perfect example of the reusability of our UI components.



Figure 9: Monster Crystal

On *figure 10* is the rewards view which displays what you gain from battling and winning against the opponent. This component also displays items that the battle can reward you, however this particular battle only awards you gold coins. This component is used in all of the person interaction UIs, the only change is the title name.

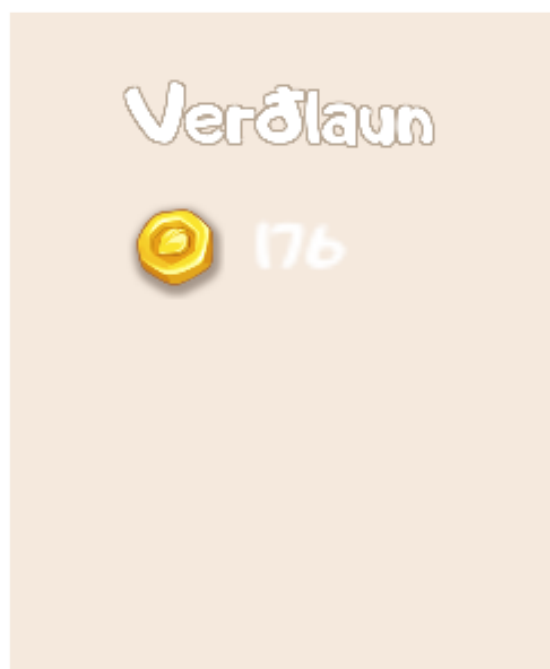


Figure 10: Rewards view

4.4 Database

The company decided since the database was a critical component for the entire system they wanted to handle it themselves. However we were able to add information to it using the Heroku interface when creating new Evolytes for example. They taught us how to use the database but wanted to handle that side of the system entirely on their own.

4.5 Builds

We decided to test a few builds out to see if our project was deployable on different devices. We started with a Windows build since most of us are working on PC's with Windows. We encountered a few bugs, aspect ratio was not correctly set for every scene. Custom inspectors were troublesome because it was still set to active and had to be set to inactive. However, subsequently we got the build up and running without major trouble.

Then we decided to make an iOS build since the game was originally created for Ipads. Since we had fixed the bugs from the other build we got no build errors. Lastly, we tried an android build, which happened to be more difficult then the previous two. We did not end with a final version of an android build working, due to running out of time working on it.

5 Risk Analysis

Here is an overview of the risk analysis for which the group felt could have had an impact on the project. The risks were evaluated based on impact and probability. These factors were scaled from one to five. A priority of a risk was evaluated of their risk score, which was *impact * probability*. Since the first status meeting we have added one risk that was unaccounted for which was how long it takes for our QA to finish his review of animations. It does not really affect our work unless he considers the work not up to standard. Then we will have to fix that. We only encountered mild versions of risks. A few days went into waiting for documentation from the product owner in order to start our new requirements but nothing that caused major disrupts.

Description	Impact x Odds	Risk factor	Obligation	Mitigation	Solution
Git merge Conflicts	3 x 5	15	Group	Use branches as much as we can	Work with the P.O. to solve the problem.
School workload is overwhelming	3 x 4	12	Group	Plan projects in advance	Sacrifice less important workload.
Final projects and papers in the same day	3 x 4	12	Group	Be prepared	Same as mitigation.
Final exams	2 x 5	10	Jón	Use spare time to study	Team members not taking exams will carry extra workload.
Evolyte assets don't exist or need to be made	4 x 2	8	Snorri	Ask in advance for assets or make them in time	Same as mitigation.
Harder to work together in person (Covid-19)	2 x 3	6	Group	Set up home working environment	Work from home. Use discord, slack and messenger for communication.
Too much time spent on documentation	2 x 3	6	Group	Prepare ahead	We will work longer hours to stay on schedule.
Q.A taking longer time than expected to review animations	2 x 3	6	Snorri	Makes closing a sprint more difficult	The responsible group member will contact the Q.A. and see if there is anything that can be done to speed the process.
Not getting assets on time from Evolytes	5 x 1	5	Snorri	Put a deadline on when the assets need to be ready	Remind Flow Education that assets are missing every day until it gets delivered
Competitive athletes in the group (2/4) competitions are time consuming	1 x 3	3	Franklin and Snorri	Plan ahead before competitions	Other team members carry extra workload during matches

5.1 Risks We Encountered

We encountered some of the risks on the list along the way and we will talk about how we handled them here below.

5.1.1 Git Merge Conflicts

We encountered some git merge conflicts but since everyone was always working on their own branches they were rather minor and did not take much time to resolve. In the latter stages however when we were merging bigger features more conflicts arose however none of them put the project at a major risk and were not too time consuming.

5.1.2 Final Projects and Papers in the Same Day or Week

Final projects and papers in the same day or week was a regular thing during this semester but we planned a head and took longer days when needed. This was a problem but this is something we have encountered since we started our studies in Reykjavík University and we are all rather experienced in dealing with extreme workload so it was nothing we were not used to.

5.1.3 Final Exams

Final exams were a problem and group members had less time to work on the assignment but we used every extra ounce of time we could to keep the project progressing.

5.1.4 Not Getting Evolytes Assets on time

If we did not have Evolytes assets in time we asked for them regularly until we got them. They usually responded fast so we seldom had to ask a second time. In the case they had not been made we would make them. There was however the case where there were changes in the character based story line and the company needed designs from the illustrator. Those assets were however not ready on time since the illustrator had other contracts they were working and the comics would not be ready until June. The company asked us to add more animals into the game instead, especially since customers requested it.

5.1.5 Working Together during Covid-19 times

We have a designated chapter on this topic down below *Work during Covid*.

5.1.6 QA taking more time than Expected to Review

When this problem came up we sent messages regularly to get feedback and kept on working on other things since our time was too valuable to waste on waiting. When we got feedback that was always our top priority to fix if there was something that needed tweaking.

5.1.7 Competitive Athletes in Group

It happened regularly that some members were busy with competitions but every member always put in their hours they just started earlier than other group members if they could not stay as long. The other group members worked longer and assignments were always finished on time. Planning ahead and managing time accordingly really made this work.

5.2 How our Planning Reduced Risks

The risks we encountered all impacted our workload but since we had prepared for them ahead of time it drastically reduced their impact and the teamwork really shined through when problems arose.

6 Development

We work in Unity where we code in C# and use the Visual Studio 2019 development environment. Github is used for version control. We were be using the Kanban methodology for project management which is represented by a visual board of tasks that are either planned, in progress, in testing and Done. We will use the digital board on HacknPlan. Our reason for choosing the Kanban methodology rather than, for example, Scrum is that it's more flexible when it comes to finishing a task and it cuts down the overall time of finishing a task from start to end. Team members will not own any part or rights to Evolytes after the assignment is finished.

7 Work during Covid-19

This is a group project that spans 15 weeks. When we were starting this project it was a strange experience since we were 2 pairs forming a group. We had no prior work experience together and were not meeting regularly in person to work on this, which was strange. Our form of communication was mostly online for 12 weeks. For the remaining 3 weeks we finally had the time to come together and work on this project which was far more productive and enjoyable. Even though group members tried to be as active as they could online it was harder to get in contact with each other and always took longer than in person. Each group member had other classes, exams and projects during the 12 week process which added extra strain on the project's progress. At times some group members were swamped with other classes which kept them from their own work in the project and assisting other group members. It was also harder to schedule time to work together since we had different classes on different times which made our time management not as effective as we would have wanted.

8 Final Words

All members are happy with how our final project turned out. Covid-19 was a big challenge especially during a project of this magnitude. We had no working space except for our at home work spaces where most of our work was done. Being able to meet in person for the last 3 weeks and working solely on this project together was a big factor for the final outcome of this project.

The groups teamwork went really well, everybody tried to be readily available, when group members needed help with problems they encountered team members were always eager to help each other. The teams communications with the company was really good. The Product owner tried to be as available as his schedule allowed and that was good when we needed feedback from the company.

This project was a great learning curve, no team member had ever taken part of a project of this magnitude before and working with a start up company really put more pressure than any school work had done. We learned a lot during this final project and gained a lot of insight on how our work flow will look like in the future.