Interim Report

Group 6
Marcel Pfaffhauser
Elias Farhan
Marko Ristin
Miriam Tschanen

Our Interim report is split into two parts, programming and artwork. This is because these main areas of the project are mostly independent of each other and were worked on by different team members.

Programming

What we have so far

What we have implemented first, was the whole game's logic being identical to the card game.
This means our game is not real-time, but instead state based.
We implemented easy ways to add new moves and tested this by adding our base move set of 2 movements, 2 blocks and 4 attacks.
We used this first move set to test if the behavior, when attacking and being attacked works correctly. It is important, that our delay mechanics as well as our combo system and the cooldowns work, since they are defining the character of our game and are of utmost importance when it comes to the game's balance.
Additional to this, we implemented an attack queue such that we can play 3 cards at a time, the same way we did in our physical prototype.
On top of that we implemented a basic GUI helping the players to understand what is going on.
Additional we use some placeholder animations at the moment, this is also to help understand what is going on, and to help in playtesting the game until our final animations can be fully used.
We implemented the basic input scheme to be pressing buttons on the keyboard, but this will later be changed to inputs on game controllers.
So in short what our game can do at the moment is the following:

- Both players can select 3 moves (in order) at the beginning.
- Whenever both players have 3 moves in the queue the first command they have entered (among those 3) will take place.
- The game's logic then calculate what will happen, and we are shown the placeholder animations.
- The animations include color codes for blocking (2 kinds) and being hit as well as movement and indicators on the kind of attack.
- We have bars representing life and the delay of the player characters, which change accordingly to being hit.
- We have additional an Indicator showing which player has won, when one player’s life drops to 0 or below.
- For our attacks we have indicators for the number of attacks being chosen (in card form) + additional feedback for when choosing an attack, which is on cooldown.

What didn't go as planned

We wanted the game to behave accordingly to the rules of the physical prototype, but to be able to be played in real-time.
It was not hard to implement, BUT it was not satisfying to play with our placeholder animations. This is why we changed our first implementation to be more like our initial card game.
Even the turn based card game was not really clear in our first prototype, since our GUI was just not clear enough. We also saw this problem, but we used more time for the game logic and, hence, did not have enough time to polish the GUI.
We have now tried to improve this further, but we are sure that some additional polishing will need to take place at some point.
We did plan from the beginning, to make the code easy to extend, but, as happens often, we used some "quick and dirty" approaches at some point, since we wanted to get the code running.
Now before we further add complexity to our code we will be going to refactor a lot of our code, to make it easier readable and changeable.
Also we want to make it easier for different people to work on the code at the same time.
This refactoring will take some time, but on the other hand it will also make it easier to later change small things, and, therefore, help to improve the gameplay.
Additional we have implemented a system to easily add more attacks (the advanced attacks we have), but we have not built the evolution phase yet.
This was partly due to us needing more time for debugging some parts of the logic than we anticipated, and partly for us not being sure yet how we want to have the evolution phase exactly.

What is better than anticipated

So even though we used a little bit more time on debugging then we wanted, all in all the programming went quite smoothly.
Also for a lot of things, there are some quite elegant solutions possible in unity and we can more or less do what we want.
Of course finding nice solutions takes some time, even with the help of the internet we still managed to learn a lot and it is interesting/fulfilling to see some of the nice solutions for small problems we came across.
We did all in all not come across any bigger problems, and as far as we see it, everything we want to do is possible to do in not too complicated ways.
And we have a clear vision on what we want to do next and how we can manage our code, we not just want to refactor a lot of things, we actually know how we will and can do it.
Having clear (small and big) goals and no unanswerable questions is what we hoped for, but we were not sure if it will really go this way.

What to do next

- We will replace the placeholder animations with the real animations.
- We will implement another class called deck, for helping to handle all the infos regarding the cards of the players, this will also make it possible to remove a lot of our code duplication we have at the moment, since we needed a lot of functions for both players. This class will help us make our game logic easier to read and easier to change in the future!
- We will try to put the UI stuff into a separate place from the game logic, such that it is easier to split the work.
- We will implement a possibility to add some “goals” and sublevels, in order to implement a tutorial mode, which seems to be quite important for beginners.
- Add an evolution phase to the game to allow character modifications and help the game to get a better replay value.
- Do a lot of small improvements over all. Making the game more accessible and add some more content to the game (tutorial mode, different attacks for evolution phase, more backgrounds etc.)

Artwork

Currently only the base version of the avatar is complete, we haven't started on the assets for the evolution phase yet. We'll outline a bit of the process below, before summarising what we have so far and how we plan to proceed.

Our character model is based on a free human mesh which we downloaded from the following site: http://tf3dm.com/3d-model/high-poly-male-2014.html
It was then heavily modified in Blender to lay the basis for the unique leg and foot structure and to get the basic proportions right. From there it was imported into ZBrush for further detailing.

Sculpting

Working with ZBrush turned out to be surprisingly easy. Once you get used to the convoluted control scheme, the workflow is quite intuitive and it feels much like traditional painting or working with real clay. One difficulty we encountered was that our mesh had very bad edgeflow because of the edits made in Blender. For example, the reduction of five toes to three and other major deformations in the leg resulted in the geometry flowing diagonally across the foot instead of following the general shape. Mesh topology is a lot more important in ZBrush than we anticipated, because unlike real clay the deformations are limited by the available geometry and cannot affect arbitrarily small areas. It took a lot of subdividing and careful smoothing to get rid of the unwanted creases across the toes and foot.

The mesh looks a bit different than the initial concept art, especially in the facial area. On one hand this is simply because it was less work, on the other we decided that a more human appearance might allow players to identify with their avatars a bit better.

The final sculpt was once again exported to Blender for the low poly modelling and normal map baking.

Retopology

Creating the low poly model to use in the actual game is the step that we underestimated most. We expected to just be able to use a fully automated script to reduce polycount and maybe having do a little bit of cleanup after, but the bad edgeflow mentioned earlier made available retopology tools either destroy symmetry between the two body halves (which is very bad for rigging) or produce results that looked quite bad and didn't reduce polygons by enough.

So retopology was done manually by extruding geometry over the surface of the high poly model by hand. This is an incredibly time consuming process and it took us about a week, but the advantage is full control over the resulting mesh and its edge loops. Complexity can be reduce where it is not needed (e.g. the chest and thigh), and additional loops can be placed in areas that are heavily deformed during animation (e.g. around the joints). The final mesh has around 9'000 triangles, which is probably a bit more than necessary especially in the facial area, but it's a sufficient reduction compared to the original sculpt.

UV unwrapping and normal map baking was done in Blender before finally exporting the mesh to Maya for rigging and animation. There still seems to be a slight problem with the normals that causes some sharp edges from the low poly mesh to be baked into the normals, but it's hardly noticeable at the distance the model is viewed ingame, so it will be fixed only if we have time.

Rigging

Creating the skeleton and rigging the character also took a bit longer than planned, because we decided to use a few special features of Maya. The final rig includes a full IK/FK setup for the arms and legs, a spline solver for the tail and set driven keys for certain hand gestures. We probably could have done most of what we need with a simple skeleton as well, but the benefit of a good rig is that it makes posing and manipulating the character much easier, so it will save us much time in the animation phase.

Animation

Animating the character is lots of fun, but turned out to be a bit harder than expected. Blocking in the poses is really simple with our rig, but getting the motion inbetween to look fluid can be quite difficult. Because our avatar looks so human there is a very pronounced uncanny valley where the movements look stiff and mechanical.

Unfortunately our rig does not export to Unity well, so we have to bake our animations, but that doesn't seem to cause any problems.

Summary

The animations for the functional minimum are coming along nicely, but currently we only have a first idle loop ingame. Fortunately the evolution phase doesn't add much new content in terms of animation, but we will need meshes for the different body parts. So in terms of artwork we are a bit behind schedule, we really didn't anticipate how much work creating these assets really is. On the bright side, since artwork can be done completely independently to the main gameplay and programming effort, production can continue well into the playtesting and bugfixing phase. This gives us another four weeks to finish up the assets. We may need to make a few concessions here and there, like just using textures for some evolution parts instead of full blown meshes.