Evolution Party

Alpha Release

Jascha Grübel, Renzo Roth, Philomena Schwab, Isabel Schacher

Current Status

We could fully implement the functional minimum, the low target despite of the particle system, most of the desirable target and parts of the high target. We had to decide against implementing a particle system that allows for full interaction with particles, because its computation had side-effects with player actions (more details in the Progress Chapter).

Implemented Features

Functional minimum ✔

- Single Player ✔
- One playable avatar ✔
- Basic player controls (jump / move) ✔
- Basic Rigid Body Physics ✔
- Simple Collision Detection ✔
- Simple catastrophe scenario ✔
    - Meteorite (RB) falling from the sky ✔

Low target

- Multiplayer ✔
- Player characters evolve ✔
- Particle System ✔
- 2 More catastrophe scenarios, for example:
    - Blizzard ✔
    - Food ✔

Desirable target

- Multiple start avatars to choose from ✔
- Every start avatar has a different special ability ✔
- Every start avatars has different evolution possibilities ✔
- Add 2 more catastrophe scenarios, for example:
    - see-saw (✔)
    - Overpopulation

High target

- More catastrophe scenarios
- Random avatar generation
- Evolution of animal is decided by player game style ✔
- Add more player interactions, for example:
    - punch
    - climb
    - grab

Extras

- Additional Game Modes (2 vs. 2)
- Manually configurable animals

Progress

Graphics

Since our last demo we replaced the highpoly avatars with lowpoly avatars (see below) and inserted them into the levels. We also refined the gameplay and added sounds and music. In addition we created a game over screen as well as nicer GUI Elements.

Engine

Behind the scenes we integrated a library for physical water simulation called liquidfun. The library is based on box2d - a 2D physics engine. The engine was incompatible with the unity 2D physics engine and in an effort to integrate it we replaced it. First tests gave a positive feedback for the interaction with the water and we continued integration.

When we introduced the complex characters we ran into a computational bottleneck that had been there before but so far remained undetected. The fixture polygon of the characters would jam and box2d would solve the interaction physically. As the displacement causes another collision box2d needs more computational power and the framerate drops below 1 fps. We tested again with simpler fixtures, but the problem remained. This meant especially that the “hugging” to keep warm in the Blizzard phase would be prone to cause massive framerate drops.

Lamentably, this brought as into the position where we had to revert to the unity physics engine and drop our approach to fully integrate water into our levels. We are currently looking into using liquidfun for purely aesthetic reasons.

Outlook

We are preparing for playtesting as the core mechanics of our game need a lot of fine-tuning. We will split the work between optimizing the current game flow and trying to implement the high target points.