Caviators

A game by Arbitrary Team Name Productions

Markus Webel
Tim Türk
Domenik Popfinger
Daniel Hook

For

TUM - Computer Games Laboratory
Winter Term 2017/2018

Game Proposal

1. Game Description

"Caviators" is a 2D singleplayer arcade game where the player controls two cavemen, gliding down a cliff to catch birds. Since they cannot stretch the net open by themselves, they have to do it "Together", which was also the course theme for this game. The difficulty of the game lies within controlling two characters at once. Every move changes speed, direction and the opening and shape of the net, which makes it a necessity let both characters to work together to achieve the higher goal: Catching birds and keeping them.

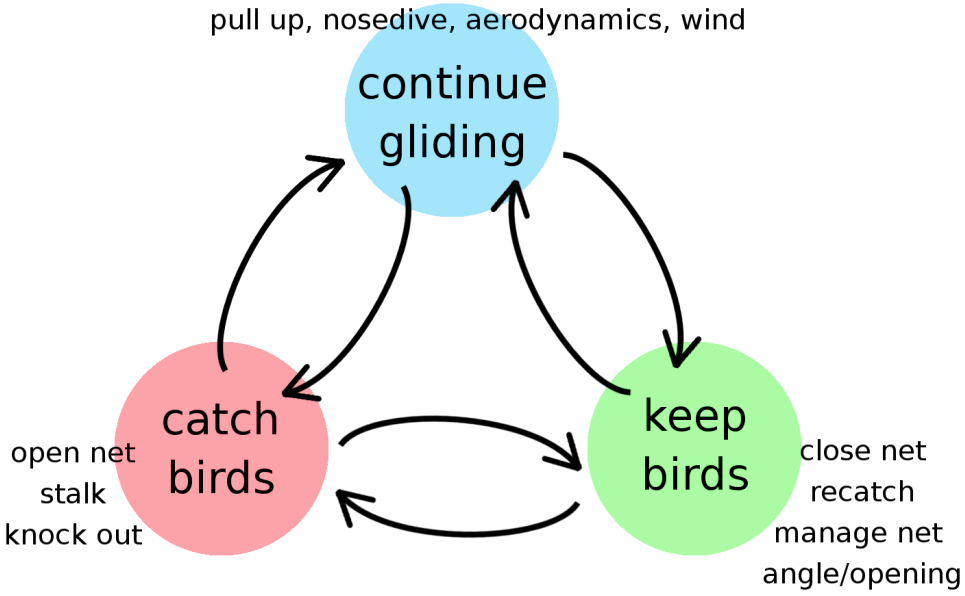
It is inspired by the arcade gameplay of games like "Luftrausers" and the controlling of two characters as in "Brothers – A Tale of Two Sons".

2. Gameplay

The game is split into two phases: First, the gliding phase which lasts about 1-2 minutes after each jump. Second, there is an upgrade phase afterwards, where players can buy enhancements to catch even more birds in the next round.

2.1 Gliding phase

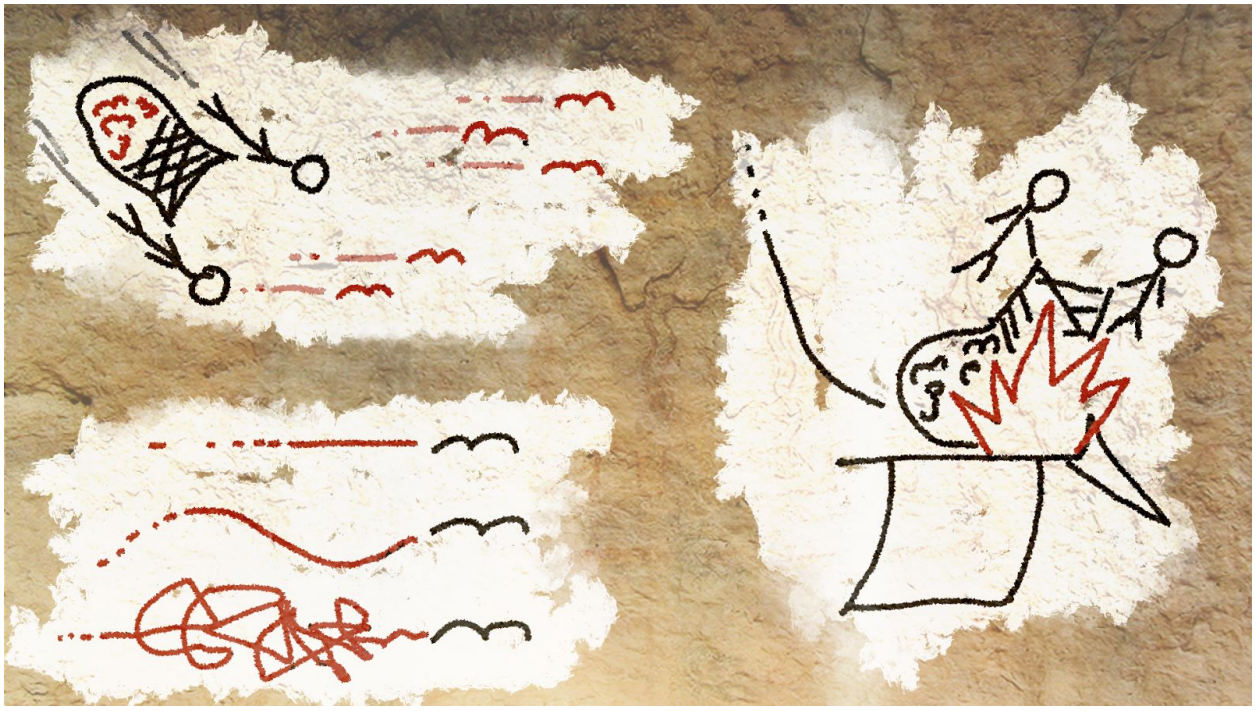
The gliding phase takes place in the same level every time but with differently spawned birds and wind zones.
Gliding can be split into three core mechanics, the player has to balance:



2.1.1. Continue gliding

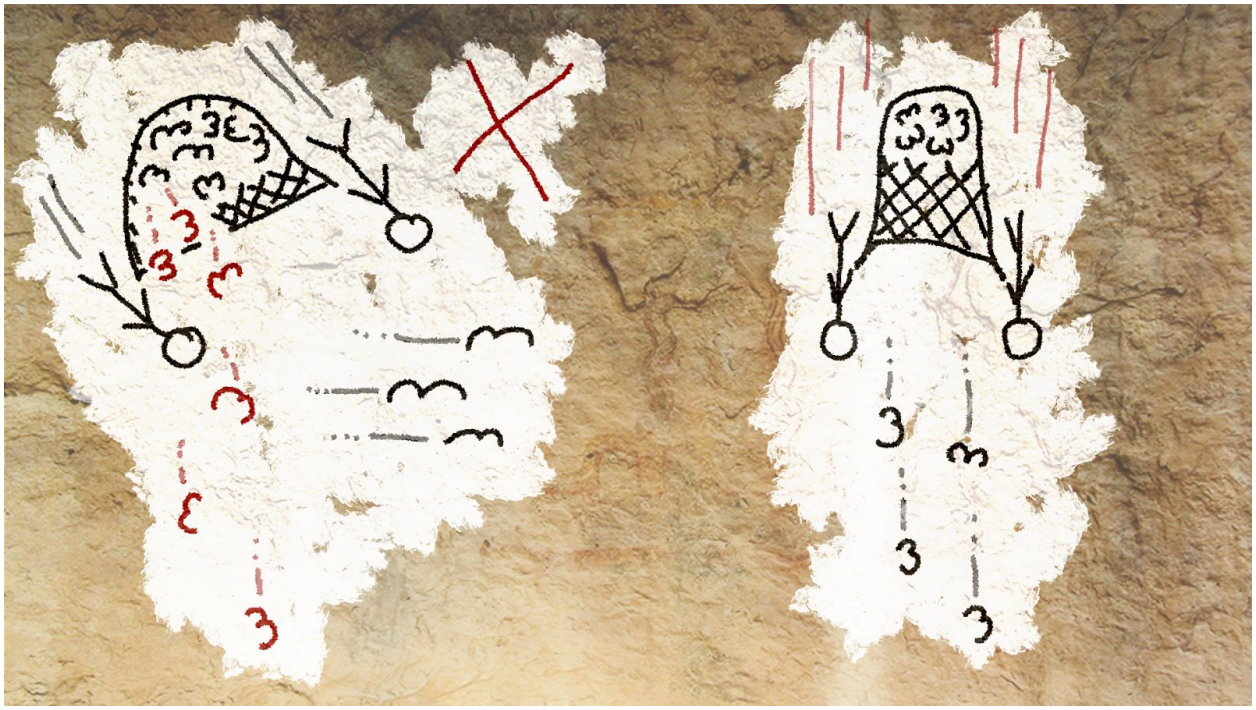
As with real physics, the cavemen drop over time. But by using troll physics, they can pull each other up again, by pulling the net. Another method is to fly into wind zones that pull the cavemen back into the air. Also they can nosedive downwards to generate speed, which lets them fly even higher up, due to troll physics. Another way to generate speed is by enhancing their aerodynamics by flying close together. To not bring a fast stop to the travel, they have to avoid crashing into big dinosaur birds.
Staying in the air influences the other two goals significantly: If the cavemen fly near together, it is harder to catch birds with the net between them. If they have to change their path to reach a wind zone or avoid big birds, it is harder to catch birds. A nosedive can easily lead to bird droping out of the net. If the player ignores gliding and drops too fast, the round is over.

2.1.2. Catch birds



The main goal is to catch as many birds as possible. Birds move in different ways and groups, which makes it harder to catch them. Also there are big birds which have to be knocked out with a net full of small birds, so they do not fidget in your net.
To catch birds there is always the balance between opening the net far enough to get them, and keeping it closed enough the prey does not drop out. Since flying is harder as the net gets filled, the game naturally draws to an end.

2.1.3. Keep birds



Your net is open, so birds can always drop out. To keep them inside, the player has to balance the size and angle of the opening of the net in relation to his current velocity.
If birds drop out of the net, catching them before they hit the ground becomes the main priority, making it a lot harder to keep yourself in the air or catch new birds.

2.2 Upgrade phase

During the upgrade phase, the player can buy upgrades which are paid for with caught birds. The upgrades give the player an overarching goal, and also change the gameplay every round. To balance the system, the player can only activate certain upgrades and every advantage comes with a disadvantage, leaving the player free to adapt the game to her taste, but not becoming overpowered.
Upgrades mostly change the flight physics (e.g. attached wings), the net (e.g. bigger net, sticky net) and the spawning (e.g. changing wind zones).

3. Technical Achievement

The game will be implemented using the Unity game engine. Due to the previous experience of the team members with the engine, the focus can completely be set on the gameplay features.

3.1 Cave painting shader

The main technical achievement is a cave painting shader, that renders the whole game as if it was drawn on cave walls, while still remaining readable for the player. The whole scope of the shader has to be tested, but there are a lot of directions to go to: E.g. generating and endless, non-repeating cave wall by combining texture chunks or using motion blur to generate the impression as if every previous frame was wiped away, as seen in chalkboard animations.

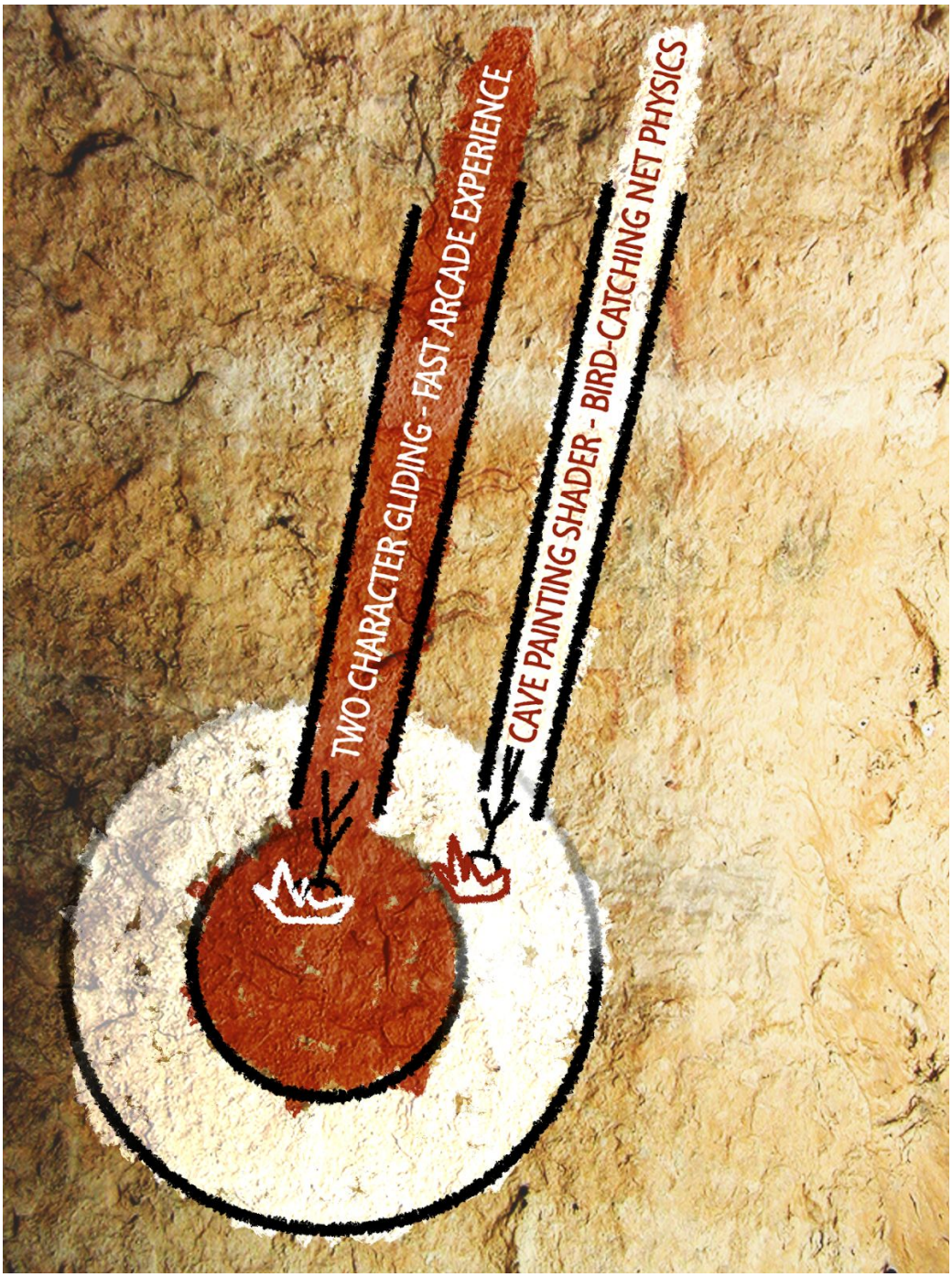
3.2 Flying and net physics

As our main mechanic, the physics have to fulfill a lot requirements: The path of flight has to be easily controllable and support maneuvers like nosediving followed by pulling straight up to gain height and speed. The net physics has to keep the birds inside the net, as well as leaving the chance for birds to drop out. Also the net has to be controllable enough to knock out big birds. Aside from the game design aspects, the physics also have to look reasonable to an extend and feel good. Achieving this requires a lot of technical tweaks and systems.

3.3 Extras

If there is enough time, we would like to try a wind grid instead of wind zones. Therefore we would simulate the wind movement in Mantaflow and import it to our Unity level.
Another interesting feature would be to implement 3D cloud shading, and projecting it then onto our cave wall.

4. Big Idea Bullseye



5. Development Schedule

Below is a sorted layered feature list. Since the core gameplay is very compressed and interdependent, most higher layered goals are mostly about effects, graphics and adding gameplay tweaks.

5.1 Layer plan

Functional minimum:
- Flying physics
- Net physics
- Birds
- Score screen

Low target:
- Cave painting shader 1 (drawn-like style)
- Upgrades 1 (longer gliding, bigger net, ...)
- Tutorial levels
- Basic sound effects (flying, hitting birds, ...)
- Simple menu
- Feedback effects (camera look-ahead, screenshake, ...)
- Static wind zones

Desirable target:
- Cave painting shader 2 (motion blur)
- Upgrades 2 (sticky net, slaves that create wind fields, ...)
- Different bird types (different sizes, different movement, ...)
- Menu in cave style
- Generated levels
- Gameplay effects (birds loosing feathers if hit, ...)
- Assets for Upgrades
- Grid based wind zones

High target:
- Fluid sim to generate wind fields
- Cloud shader (obviously drawn clouds)
- Story screens
- Cavemen voice acting
- Environment art (trees, ...)

Extras:
- Worldmap
- More bird types
- Different maps (winter, summer, other topology, ...)
- Getting bought by Tencent

5.2 Timeline

See attachments.

Colors:
Orange: Markus
Blue: Domenik
Purple: Tim
Green: Daniel

6. Assessments

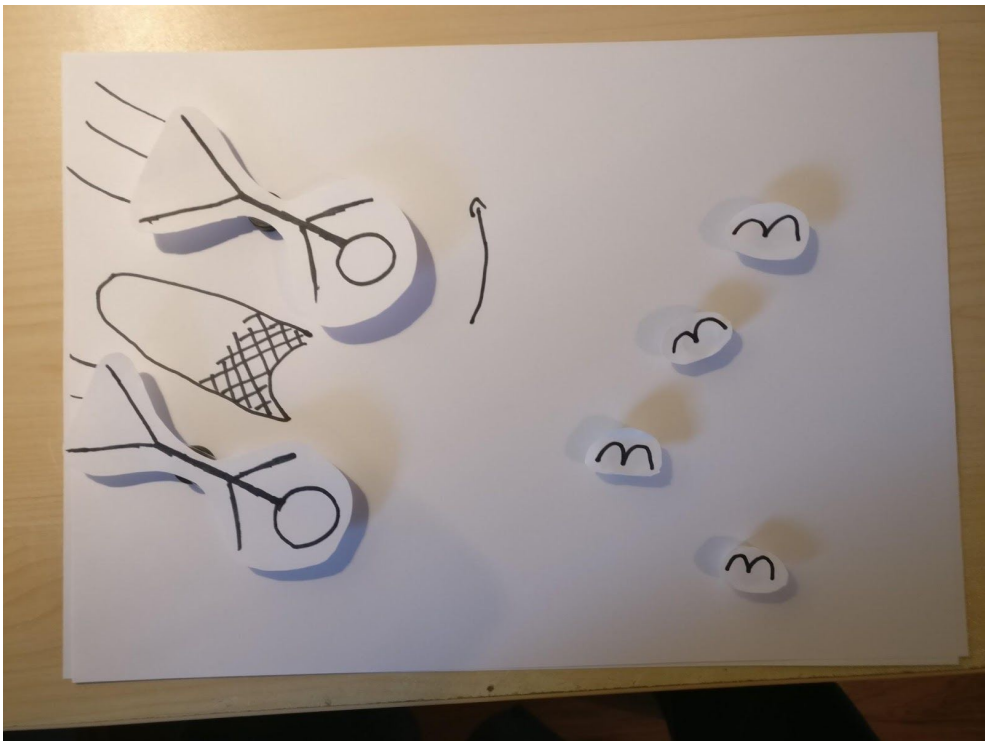
"Caviators" is intended to give the player a fast-paced arcade-like flying experience. While the basic experience should just be a cool flying controller that provides fun short rounds of gameplay, there is also another goal: Fluidly balancing all three core mechanics under the strict time pressure of the approaching ground. Therefore it is crucial that the player is able to control both characters without being confused and gain experience to master her flying skills. Via upgrades and different level spawning there should be enough variety to always start another round, as well as the possibility to shape the gameplay in favor of the player's taste. Also the player should be able to set her own goals, like catching specific birds to work towards a certain upgrade or beating a highscore.
In the end, it should also be a humorous experience. Let's be real, it's a game about two falling cavemen.

Paper Prototype

1. Design

When starting out with designing our paper prototype none of us could formalize how to capture the speedy and fun flight experience we imagined as core to our game idea.
So we went about creating a paper version of our game by putting a game master in charge of modelling physics. This game master (in the following “persona computa” or “PC”) would periodically ask for our player’s input, and then model the reaction of our cavemen-gliders and their environment, resulting in a turn based interaction with the following elements:

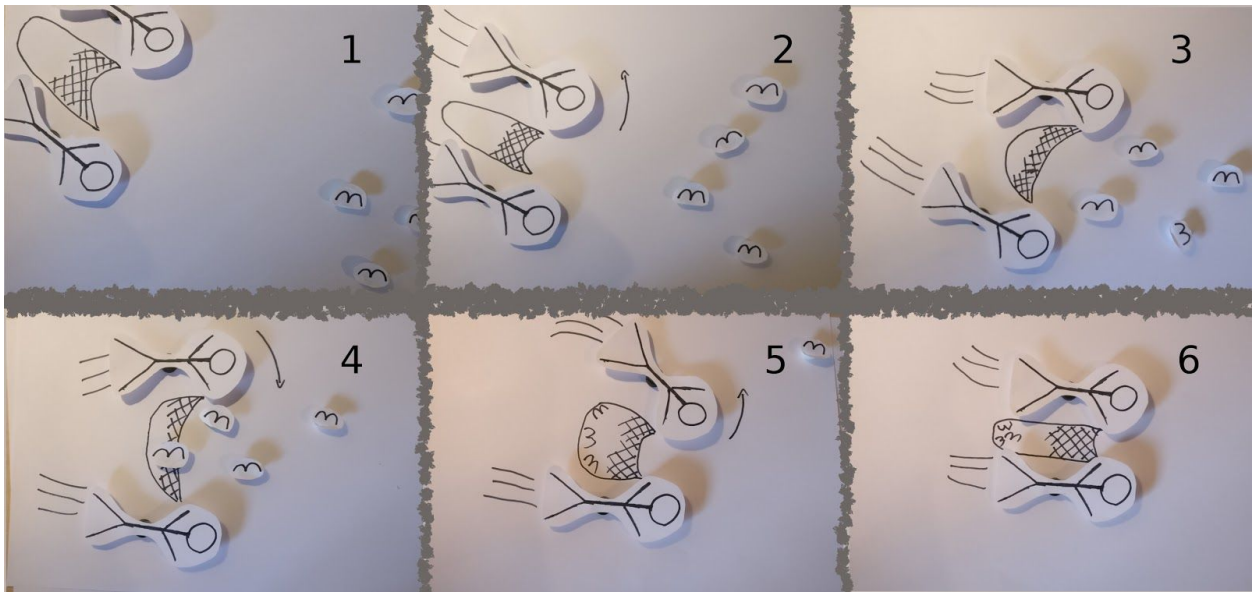
- 2 caveman-gliders
- 1 net (different versions depending on how far it’s opened)
- Either several small birds, or a single big one.



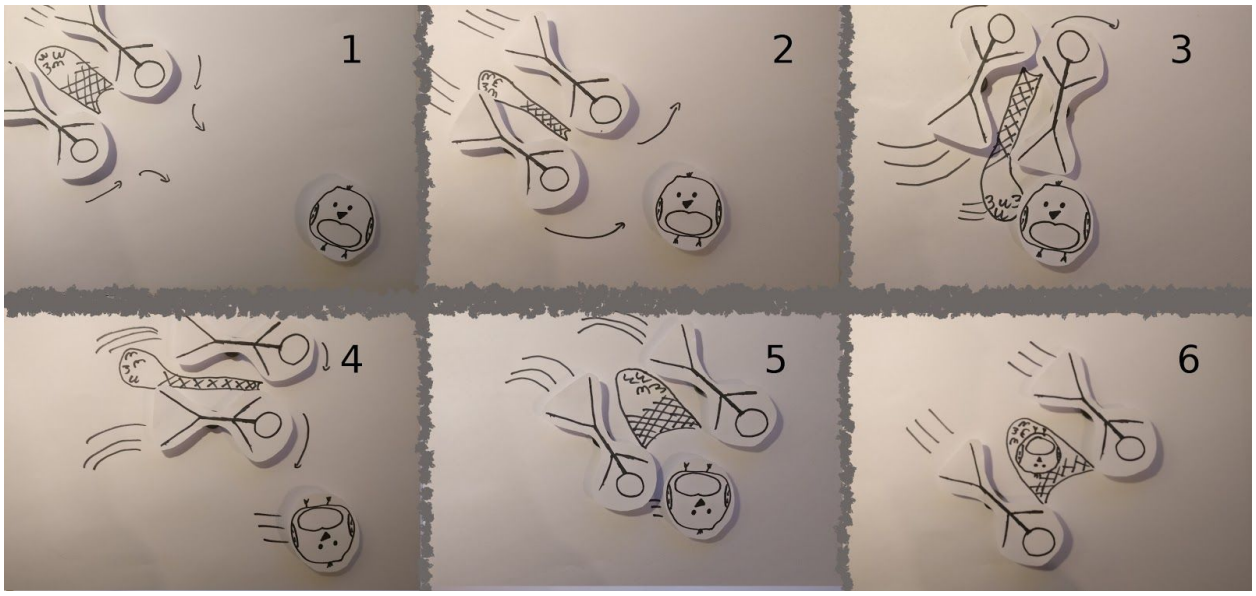
In practice that meant the player controlled whether each caveman would turn their nose upwards, (to gain some height in exchange for speed,) downwards, (gaining speed but losing height,) or continue their flight as in the previous turn.
In turn, the PC would then move the cavemen depending on the player’s input and their own understanding of flight physics, set the net between them depending on the cavemen’s positions, and finally move all available birds - either into the net, or so they would continue flying.

2. Experience

To get an idea of how the prototype works, here are visual transcripts of two rounds we played:



This first one shows the standard game situation in which the player catches several small birds. Between each pair of the six shown key turns, there were one to three game turns in which the player could give input.



In this second situation the player already has caught a couple small birds, and now chases after a fat bird in an attempt to knock it unconscious and then catch it.

2.1. Problems

During playtesting the prototype, the split of player and PC led to some quarrels about the flight behavior. The PC could not always provide an experience that was consistent with our player's expectations: the reaction rate of gliders would differ as well as the timestep size and the swing speed of the net.
Another problem was that we already had put a lot of thought into our design, meaning the PC would move the player characters in ways so that our balance triangle would always work out.
Also the net proved to be hard to represent with a physical prototype. Using a string seemed to be the most obvious solution at first, but the strings we had were either too stiff or too thin. Bringing them into positions resembling those of a net at high speeds (and keeping them there) was annoying.
Overall. the paper prototype did not convey the sense of speed and urgency that's core to an arcade-like game, so a lot of the experience could not be tested with the paper prototype.

2.2. Conclusions

Although this playtesting was not an acceptable recreation of the intended gameplay experience, it led to interesting discussions about other aspects of the game:
The disagreements between players and PCs about things like glider and net behavior showed us some actions that players wanted or expected to be able to do, even though they were physically incorrect behavior. For example players tried to pull up and slow down with one glider, in an effort to swiftly swing the net forward. The intended result could not happen unless the net's center of mass was unrealistically close to the gliders (or unless that action was accounted for in our game's physics computations.)
Another problem we already had thought about but did not grasp the scope of so far, was the problem with loops in the net. If players were to cross paths, then a two dimensional net would form a loop around our caught birds, and would prevent us from catching new birds.
Here the paper prototype helped us visualise the problem, as well as our ideas for solutions, so after a lot of discussion and explanations involving some sort of puppet theater, we agreed to use the convex hull of a simulated net to ensure that no loops would form.

3. Results

In the end, we did arrive at some points that we could incorporate into our game design plans:
The probably most important one was the decision to put more focus on fun interactions than on realistic physics. So when implementing the net, we'll aim to not only simulate it as if it was dragged along by the gliders, but that we'll try to anticipate how and when the players want their net to open, close, or swing around, and that we'll try to enable that behavior.

For the gliders themselves this means that we'll explore a wider variety of control schemes and behaviors, and that we won't discard approaches or ideas just because they sound unrealistic. Instead we'll see which combination of controls and flight behavior best supports the interaction patterns we expect, and those we have seen emerge in our prototype.

Implementation specific points we discussed while playing the paper prototype for example were disabling player-player, and player-net collisions. While our game will be 2D, we agreed to go about implementing it as if one glider always was behind the other on the third axis.
If communicated visually, this would result in game situations in which one glider passes through the other in 2D not appearing confusing to the player, while also giving the player more freedom in how they control the gliders.
Additionally, this would allow us to place the net between the two gliders, resulting in a visual representation in which it makes sense that the net never wraps around itself.

Lastly, in one session with the paper prototype we had the problem that an unconscious big bird was falling downwards directly below the gliders, which meant that the player could not catch it. It would have taken too long to first slow down to get behind the bird, before going into a nose dive to chase after it.
After we started giving big birds a boost when they were hit with a filled net this problem was resolved, so we'll incorporate such a boost when implement batting big birds out of the sky.