Game Programming Lab 2016

Dance with the Cooks

Goran Saric, Christian Schmidhalter, Stefan Blumer, Pius von Däniken

Interim Report Chapter:

Design & Animation:

Ideas

Our intention was to implement an art style based on a mixture between caricature illustrations, clear form language and lean on disney characteristics. These elements we want to combine with a "disco" flavour, since our game is about dancing tv cook battles. The overall setting should be humorous and comical.

Since we have never worked with 2d animations, we decided to try out 2d keyframe animations. So we split our character / environment sprites into separate parts, add joints and bones and join them together in a tool that provides an IK (inverse kinematics) feature. With this technique we can save a lot of time animating the assets and we will have enough sprite frames for a fluent animation which is a must for this kind of setting / gameplay.
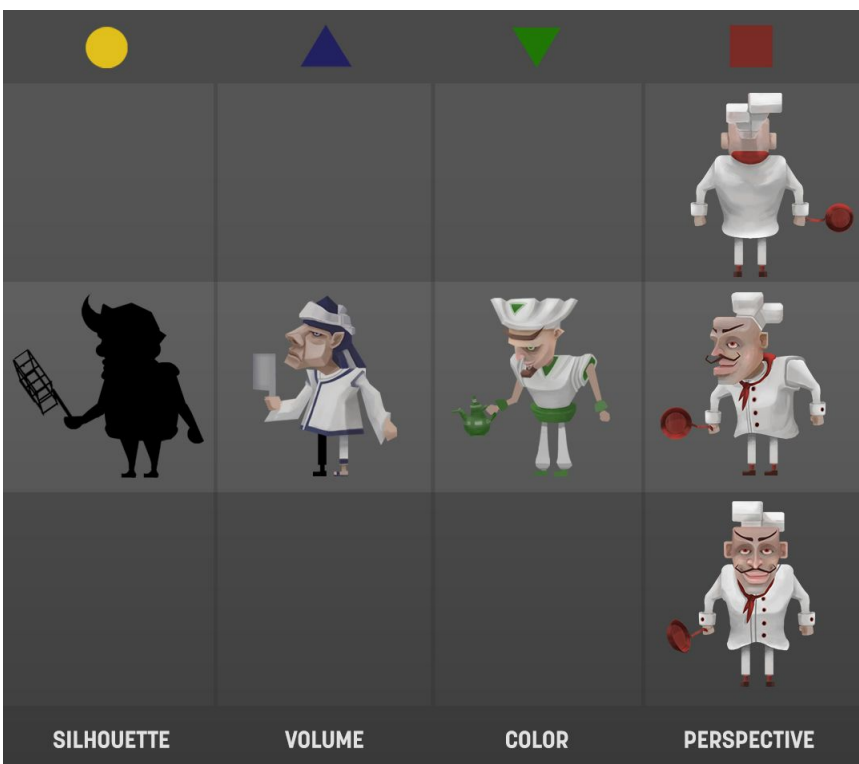
We started with these moodboards:
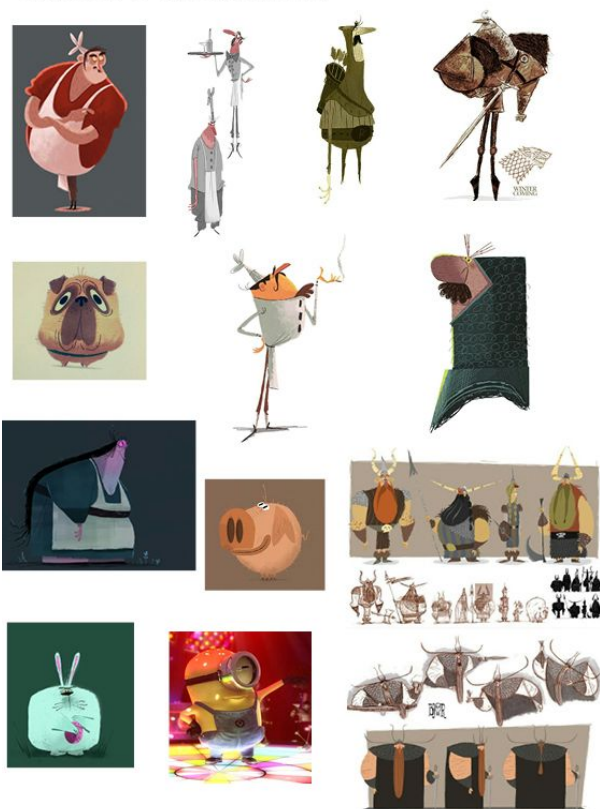
And tried to make a simple mockup just to talk about the same setting:

In the next step we tried to find a working silhouette concept for the characters:


We decided to have four different characters based on four basic geometrical forms and colors:

COOK F   COOK J   COOK U   COOK A

SILHOUETTE   VOLUME   COLOR   PERSPECTIVE

Tools

We used Photoshop CS6 to draw all the pixel graphics. For each body part we created a separate layer. In Adobe After Effects CS6 we imported the Photoshop *.psd file and gave each body part several Deform-Points with the Puppet-Tool. With the free plugin Duik (http://duduf.net/index.php/products/after-effects/duik/) we were able to create bones out of the Deform-Points and to connect them to a functional IK-Rig. After animating the animation set we could export each frame (48 frames per second) as PNG files. With the browser tool "Stitches" we generated horizontal spritesheets: https://draeton.github.io/stitches

The benefit of doing the whole 2d keyframe animation in After Effects is, that the integrated Puppet-Tool is able to bend pixel graphics. We couldn't find any other tool doing this properly. We initially tried out Spriter (https://brashmonkey.com) and Spine (http://de.esotericsoftware.com) without sufficient results.
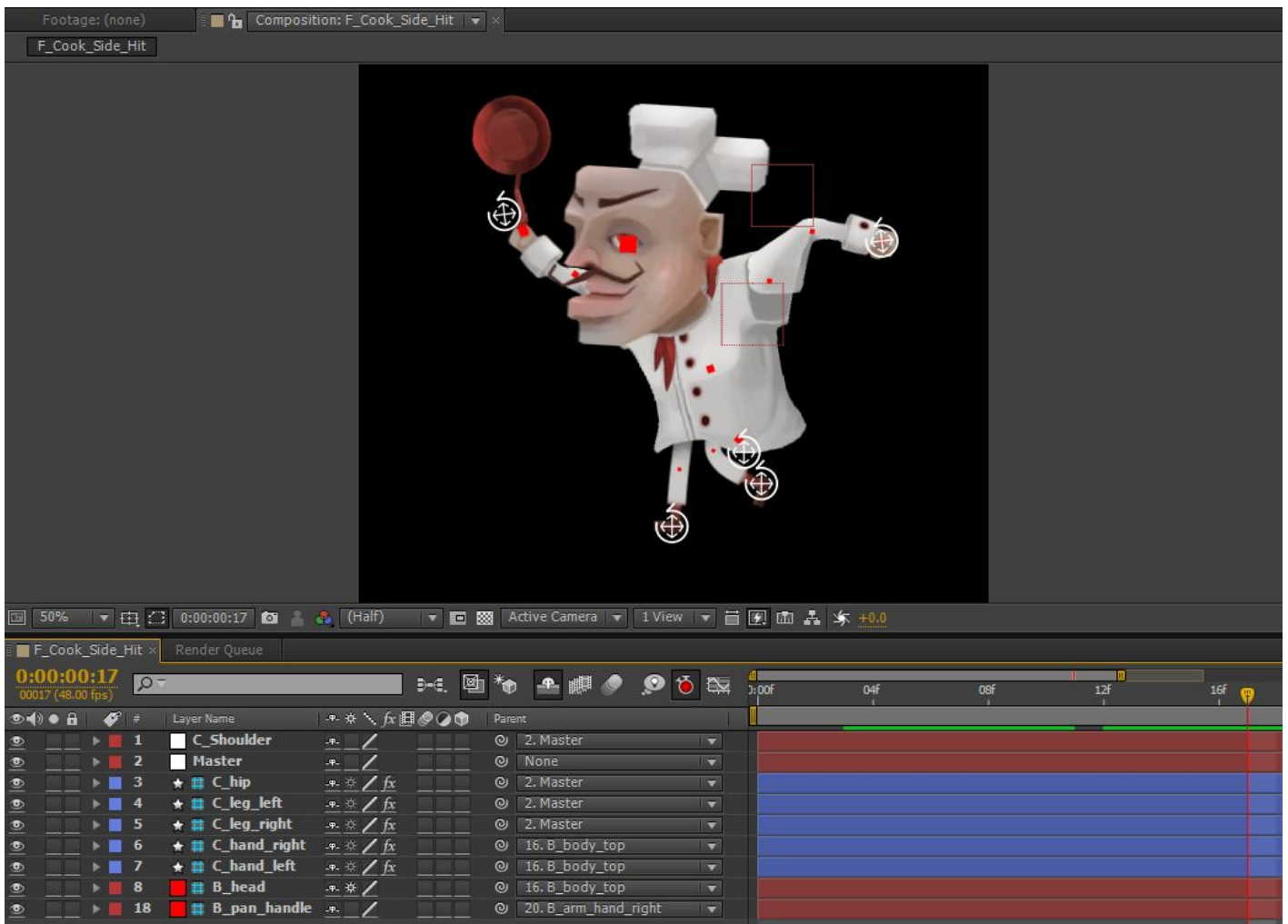
Difficulties

To have a working IK-Rig with Duik it takes a good amount of time to understands the way the different layers and bones need to be parented with each other. We had to create empty objects as parents for certain parts to have control over the shoulder for example or to simply be able to move the pan of the cook around.

Screenshots

We currently have a full animation set of the first cook:

Game Logic:

There are no major changes to the game logic. To achieve more chaos and randomness, we decided to add environment game objects. Players will be able to interact with these and some objects will act on their own. To achieve this goal, we had to do a total rewrite of the gamelogic so far.

We use an inheritance structure for the different types of game objects. We have also planned to add a system to allow a player to drop stars to the ground in certain situations.

With this new structure we have so far implemented two environment game objects:
- Chair: Is being knocked back, when hit by a player. If hitting a player, the chair is destroyed. In the final version a player who's been hit should drop a star to the ground.
- Subwoofer: Simple eye-candy to visually support the beat of the music.

With the new environment object system, we have dropped the idea to add alternative attacks, or items. We think there will be enough possibilities to interact with environment objects.

There is still some unfair behavior, where the player with a lower id can steal the one remaining star of another cook, if he's hit by two players simultaneously.
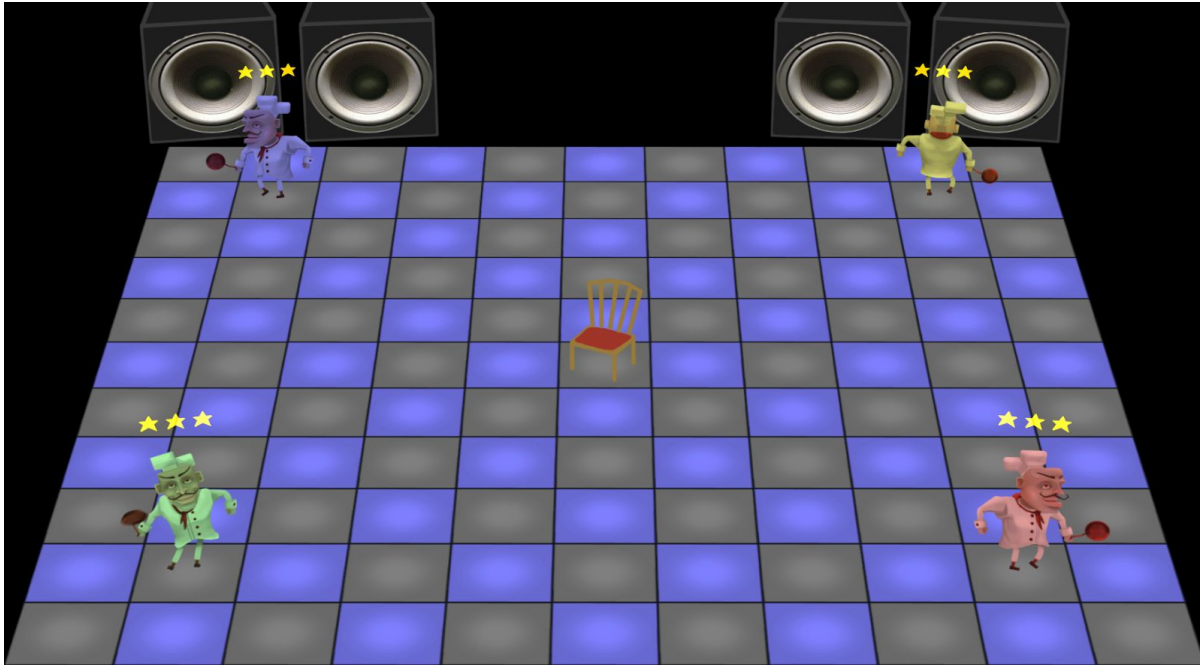
Graphics:

We keep the very simple 2d billboard and perspective playing field technique. We animate all objects with sprite-sheets.

We integrated the graphics of the first cook and have him temporarily drawn in different colors. The graphics and animations for the chair and the subwoofer are only placeholders.

For the final version we have spotlights and smoke particles planed.

Music:

We have discarded our initial idea of taking a melodic phrase as an input and generating the music around that melody, as this approach would not have allowed us to make the music very dynamic and react to game events.

Instead, we decided to use a bottom-up approach, where we first generate the chord progression. We then use various events, triggered by the game, to manipulate the music generation process. This way, ingame events like one player hitting another, will will result in an immediate acoustic feedback, like changing a pattern or playing one note of the current chord. To achieve this, the music is only generated one beat at a time and the main game loop updates the music generation pipeline once per beat.

The generation process in itself is based on simple rhythmic patterns, which repeat after one bar. Every instrument gets its own pattern to play, but in order for it to sound more pleasing we have to restrict the number of different interacting patterns and control the level of repetition within the pattern. For example for our game we want a high level of repetition for the kick drum to clearly show where the beat is, whereas we can be more free with the randomness of the hi-hat or arpeggiation patterns.

We also moved away from the previous more Jazz inspired music and are now trying to incorporate more modern electronic sounds, we think this style of music is more appropriate for our new approach.