Milestone Report 3

Interim Report for a Twitch-based Puzzle-Action-Game

1 Progress Overview

During the recent three weeks every required functional feature has been implemented to a basic level as stated in the low target from Milestone Report 1. The game now includes a board of conveyor-belt tiles, cooking pots to complete recipes in, a bar with randomly refilling tiles, a UI for current food orders, a poll system which already supports Twitch interconnection, a basic score system, ingredient spawning and travelling across the board and a console for debugging commands.

2 Implemented Features

2.1 Debugging Console

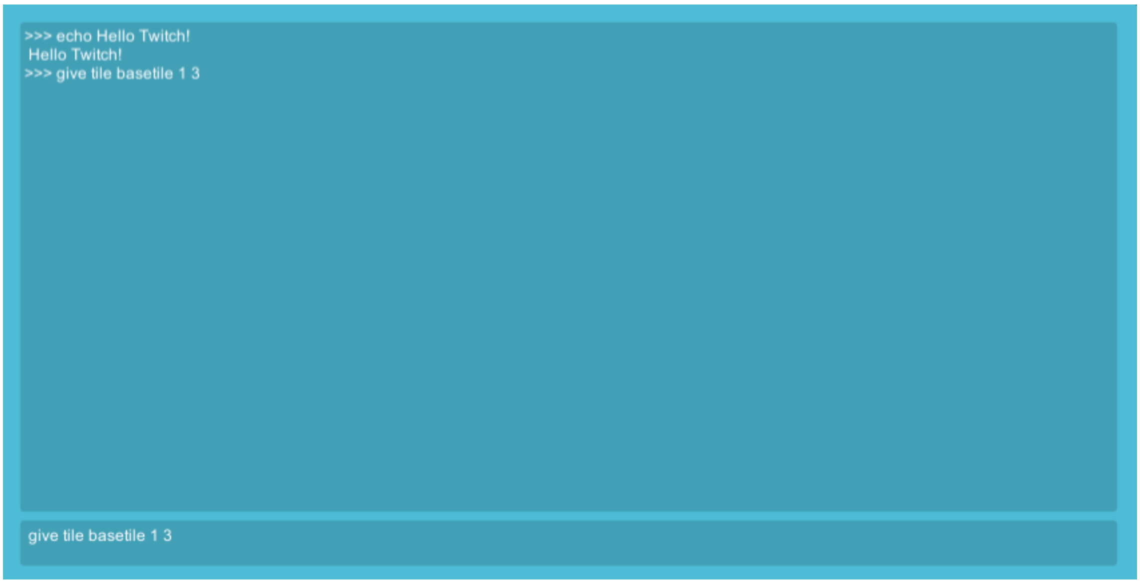
During the last milestone meeting and in the mutual critiques by our fellow peers a major problem many pointed towards was the testing and presentation of the gameplay without a large group representing the Twitch-Chat in the background. While we plan to randomize most of the processes people are allowed to vote for like the generation of tiles for testing purposes in the case of too few people being present during gameplay a basic console was additionally implemented which, amongst other things, allows developers to order conveyor-belt-tiles in order to make testing and presenting a lot easier. This console can later be used to set general hyper-parameters involving recipes, spawning of ingredients etc.

2.2 Game Board with Conveyor-Belt Tiles

The game's board which oversees the placed conveyor-belt-tiles is the core of the gameplay. A very easily extendable tile-hierarchy makes it possible to also add special tiles to the board which render the game far more exciting. Currently, only basic directional tiles are included. During the upcoming period for milestone 4 it is planned to add at least three special tiles to the game.

2.3 UI for Food Orders, Score, etc.

Most of the UI during gameplay is already functional: A very basic score system is in the game combined with a set of possible recipes each consisting of two or three ingredients. The game randomly places orders and the player can complete them to earn points which are then shown below 'Orders' (see figure 3). On the bottom-right, the required ingredients to finish a placed order are shown so the player always knows which combinations he or she needs to go for. Recipes are represented by icons showing the needed ingredients as a formula from top to bottom added up by plus-signs.



```
>>> echo Hello Twitch!
 Hello Twitch!
>>> give tile basetile 1 3
```

```
give tile basetile 1 3
```

Figure 1: A debugging console was added to support development. It can be used to add tiles and later ingredients and food orders so testing various gameplay situations is simplified.

2.4 Cooking Pots

Cooking Pots are currently used to represent the process of cooking and combining ingredients. The player can see whether a pot already contains food or is still empty. This is additionally represented by small icons below the pot showing the ingredients which are currently inside (see figure 2 bottom). If at least one thing is inside a pot, a graphical effect is displayed which resembles a boiling soup or stew.

2.5 Polling System

The polling system and its connection to Twitch-chat is one of the core technical challenges in our project. Twitch chat input is parsed and users can vote on generated ingredients as well as tiles by writing emotes into the twitch chat. The polls use timers and in case of no votes or ties a random option will be selected. In future iterations we plan on adding a more sophisticated algorithm for the possible ingredients to prevent deadlock situation if the twitch chat is trying to sabotage the player too much.

2.6 Ingredients Movement

The second core part of the game, ingredients interaction with the environment, has already its basic elements. Ingredients are spawning on predefined spawning points, in accordance to results of the polls. They move along conveyor belts, are destroyed after falling over the edge of the board or crushed with new tile and update contents of cooking pots after reaching them.

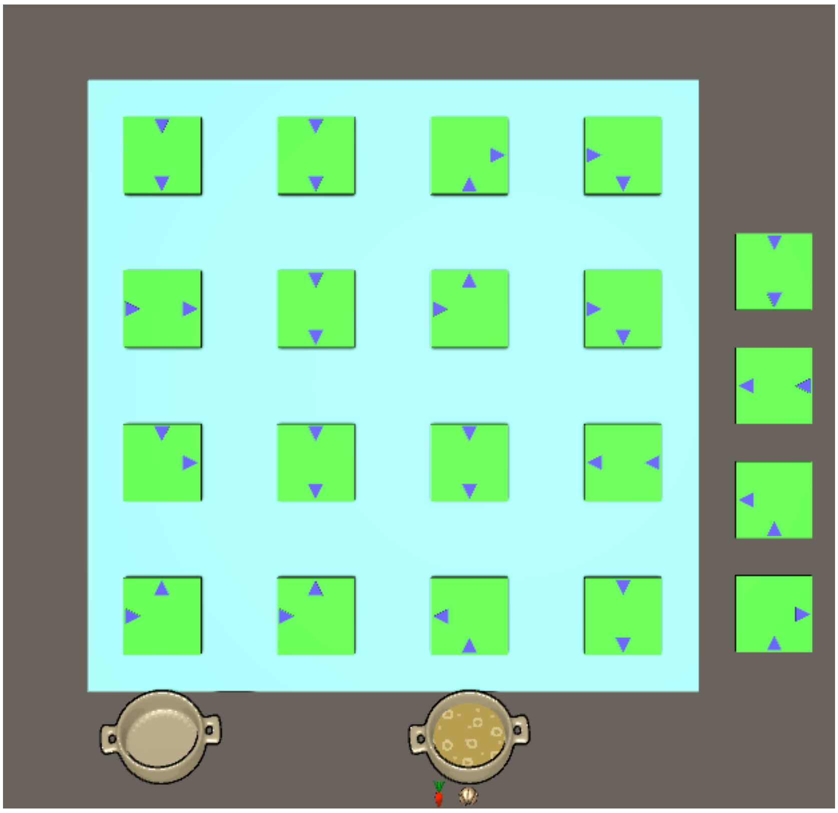


Figure 2: While not yet featuring models, the gameboard already contains required functionality and allows the addition of more complex tiles than the already present directional ones.

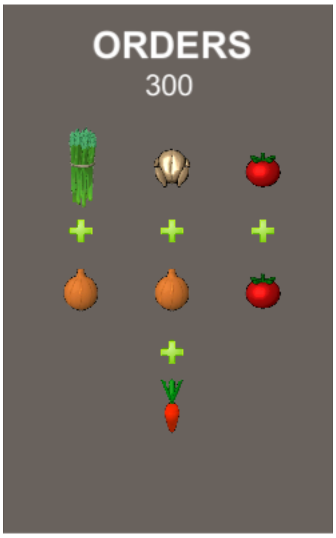


Figure 3: Food orders and current player score (currently 300 points)

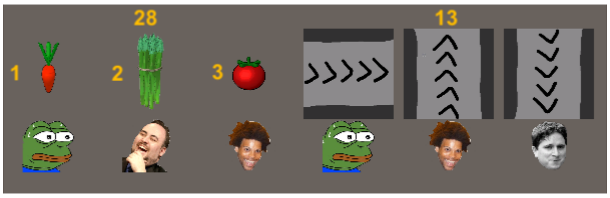


Figure 4: emote-based ingredient poll and tile poll

2.7 Assets

In order to keep the visuals consistent and simple, we've decided to create own set of assets. Chosen style should be smooth, round and simple so that all, important models can be created in the given time and in order to amplify advantages of cell shading. Few textures are limited to ephasize some core features of certain ingredients, like for example stripes on an onion. At this point we have five, different files representing tomato, onion, carrot, chicken and asparagus.

2.8 Cel-Shader

To simplify the graphics and still appear consistent, a Cel-Shader was implemented (see figure 4). The game was decided to use OpenGL 3 as graphics API and the effect was therefore written in GLSL. It computes the Phong Lighting Model and then clamps color values into intervals giving objects the appearance of being drawn. On top of that, in a second render pass an outline is generated. Many popular games and even quite old ones use this technique to improve graphical appearance. While the shader now works well it was quite a hassle to implement because Unity mainly supports DirectX. The change of graphics API also caused issues and dark rendering viewports for some team members which unfortunately costed time to resolve.

3 Design Revisions

We ran into the problem, that replacing tiles below moving ingredients leads to immersion breaking teleporation, e.g. if an ingredient is moving from top to bottom and is currently near the bottom of the tile and now the tile gets replaced by a left to right tile we would need to teleport it onto the right side instead. Therefore, we decided that replacing tiles below ingredients now instead destroys the moving ingredient. This can be visualized by crushing the ingredient under the new tile and cannot lead to game-breaking scenarios unlike other considered solutions.

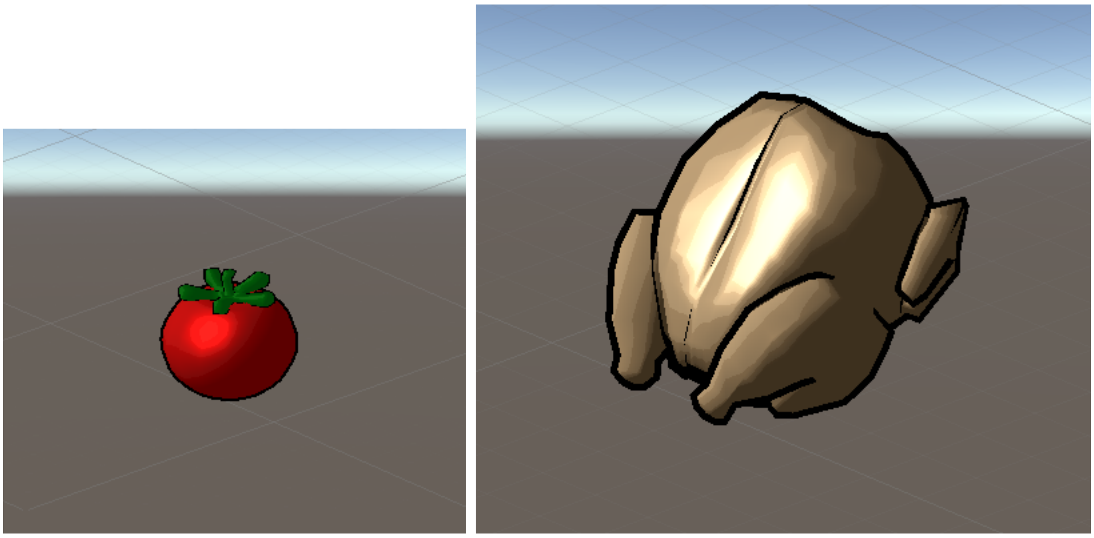


Figure 5: Cel shader as material of ingredient meshes. Assets appear to have some form of detail without the usage of bump-maps and albedo textures.