Conclusion Chapter

Jonas Stulz, Peter Güttinger, Patrik Toth & Isa Roesch

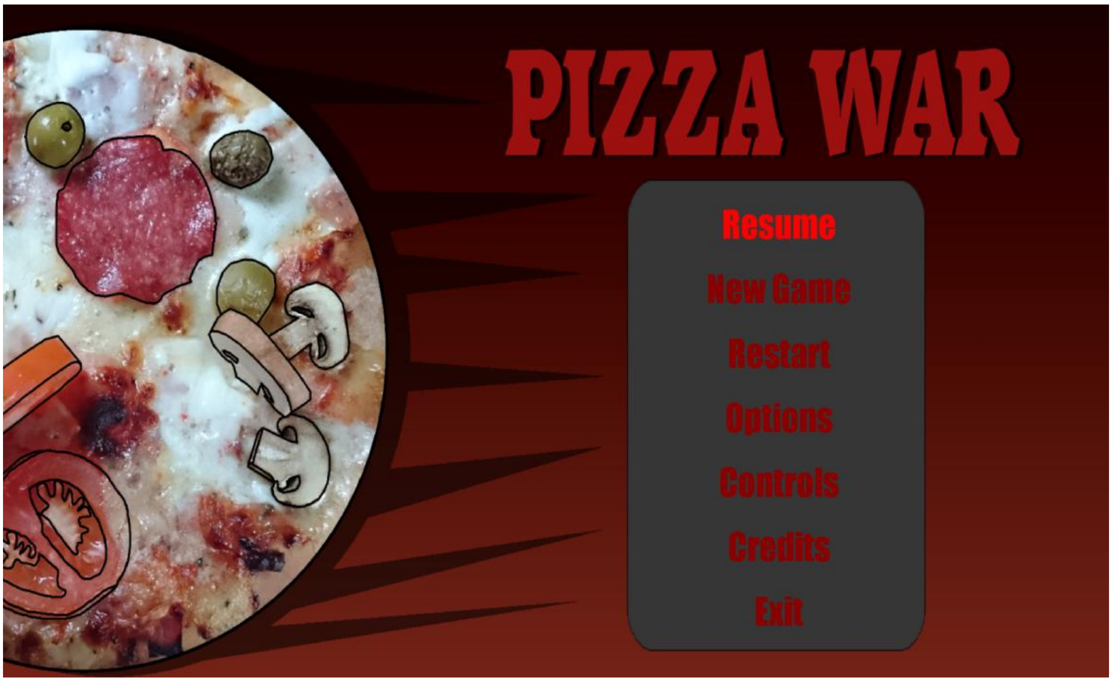
Contents

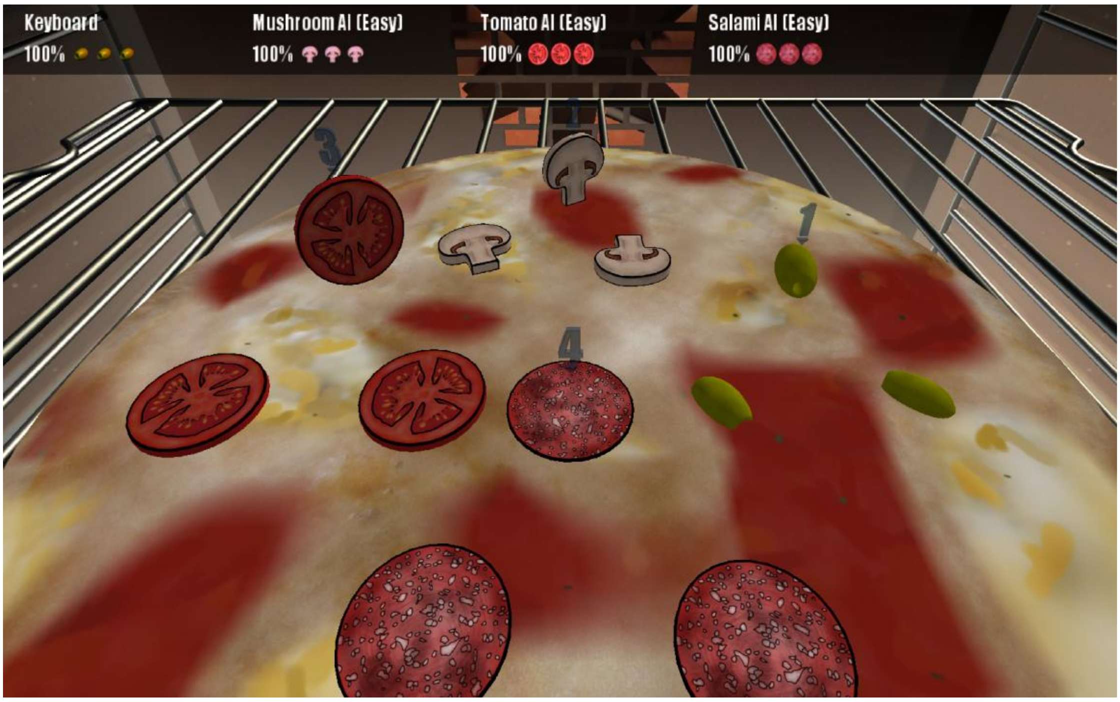
1 Final Result

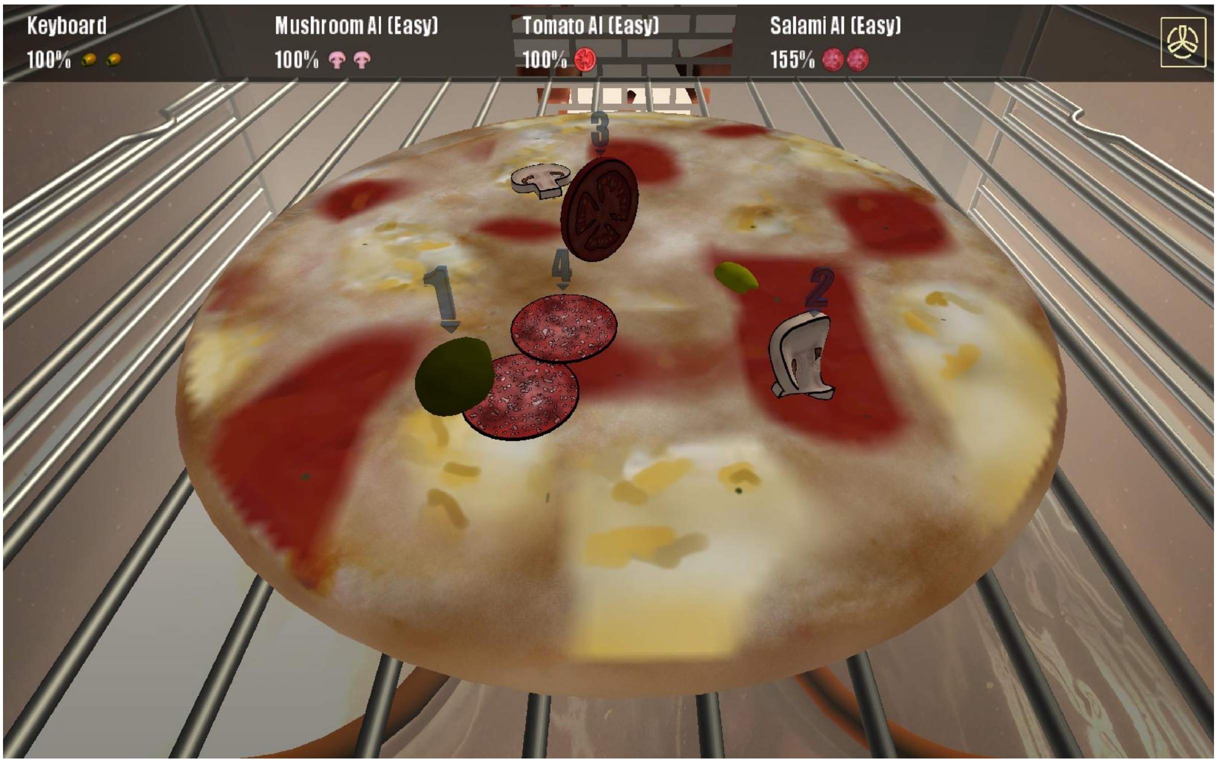
1.1 The Final Game

For the final release, we got a fully working game that is fun to play as our playtesting session confirmed. We could not implement all of the features we planned, but most of them. The highlight of our game is not the physics, as was initially planned, but the beautiful and professional models and animations.

A few impressions from the final game:





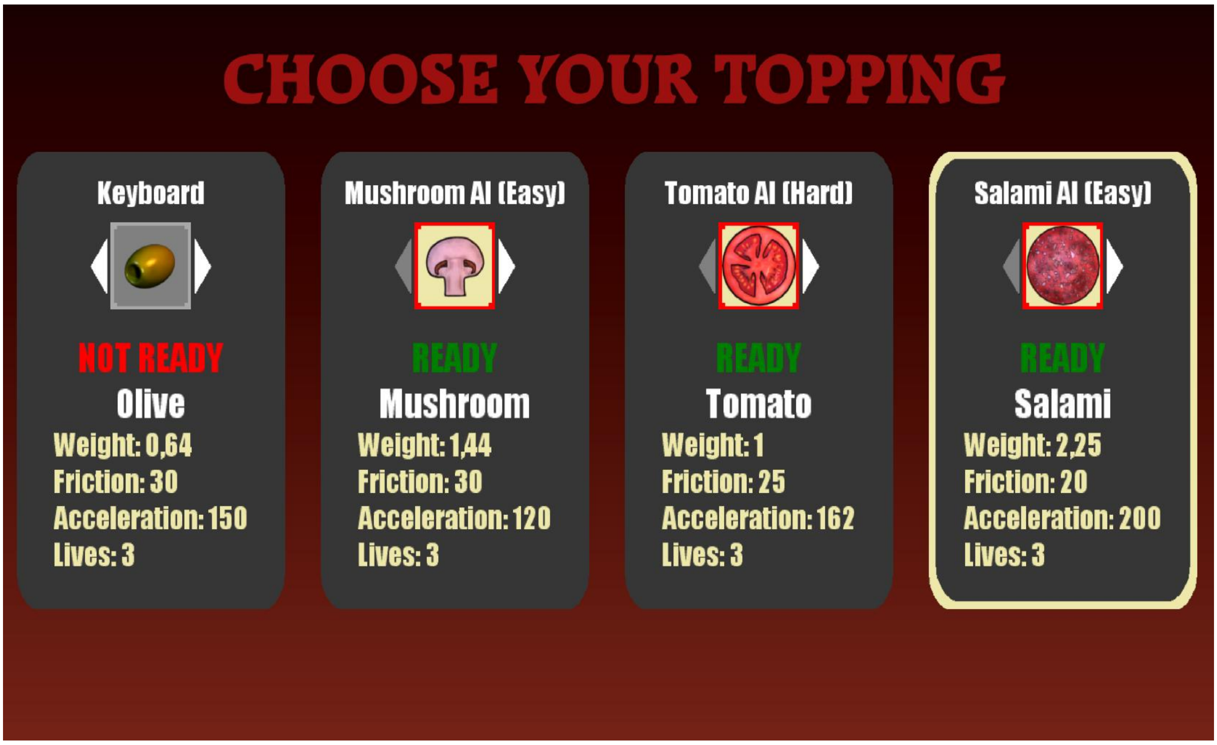


Keyboard 100%

Mushroom AI (Easy) 100%

Tomato AI (Easy) 100%

Salami AI (Easy) 155%



CHOOSE YOUR TOPPING

Keyboard
NOT READY
Olive
Weight: 0,64
Friction: 30
Acceleration: 150
Lives: 3

Mushroom AI (Easy)
READY
Mushroom
Weight: 1,44
Friction: 30
Acceleration: 120
Lives: 3

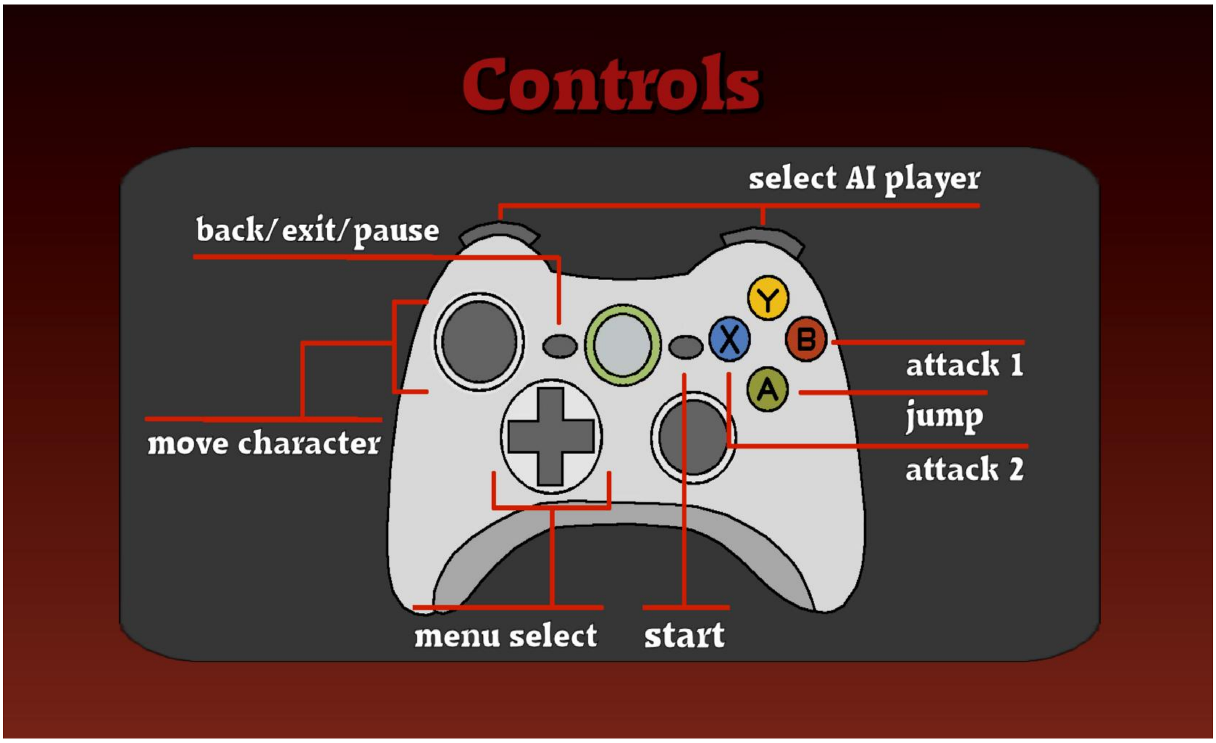
Tomato AI (Hard)
READY
Tomato
Weight: 1
Friction: 25
Acceleration: 162
Lives: 3

Salami AI (Easy)
READY
Salami
Weight: 2,25
Friction: 20
Acceleration: 200
Lives: 3

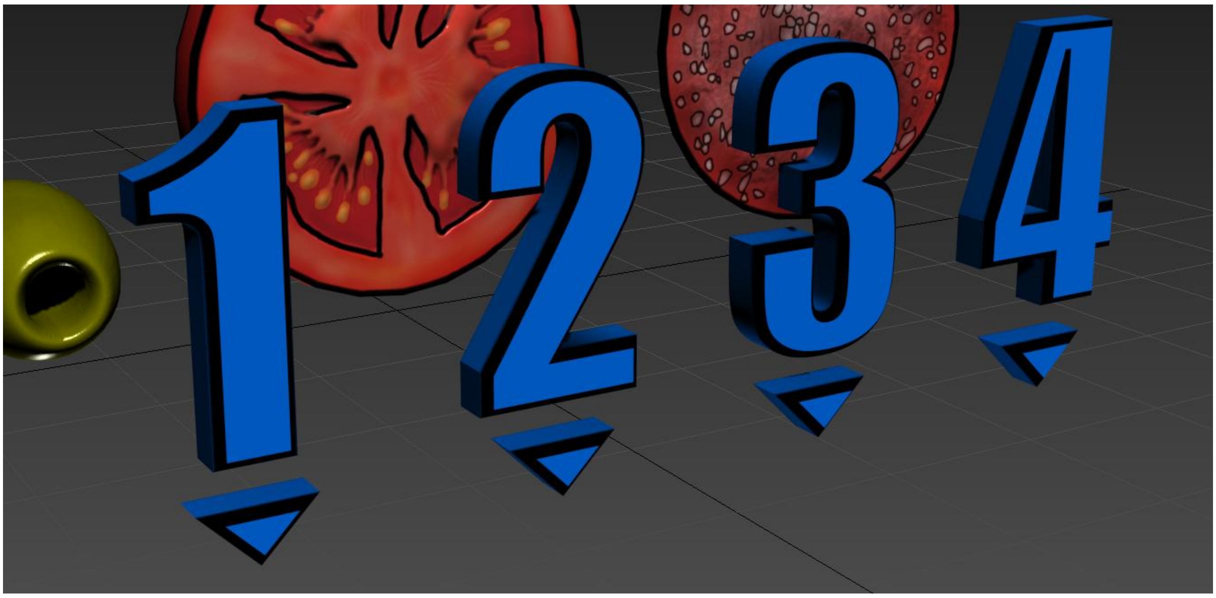
1.2 Changes since the Alpha Release

We made a bunch of changes since the Alpha release to finalize the game, some of them before the playtesting session and a lot of them afterwards.

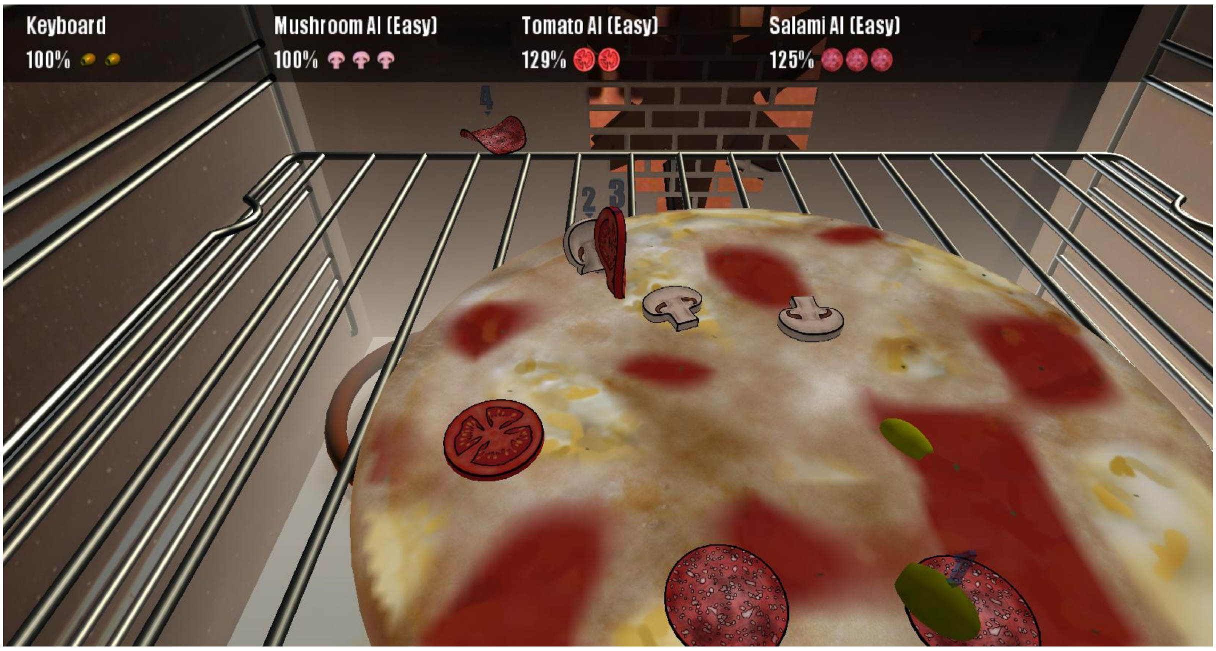
- **Friction and balancing:** Parameters of the toppings, such as speed, attack strength, friction on the pizza, etc. were overhauled several times to make the game more fun and the rounds longer.
- **Debug mode and keyboard player:** Everything debug-related, like manual camera control, debug AIs and debug bounding boxes was moved to the debug mode. The keyboard player only appears if no gamepad is connected. The debug mode can be toggled using the config.ini file.
- **Super Smash Bros mechanics (Attack multiplier):** To make the game more interesting, we implemented an attack multiplier mechanic, making characters that got hit several times more prone to being pushed down. This has also an auto-balancing effect. The attack multiplier is also shown in the game interface.
- **Attacks:** The mushroom and the olive got two attacks in total. The salami got an own unique attack, which allows him to shoot down like a hawk when flying and push other toppings down.
- **Respawn Timeout:** When a topping respawns, it is a little bit transparent for one second and invincible to attacks. The character can walk, but not attack while in this state.
- **Control Screen:** To ensure that players can play the game without trying out which key is which, we included a control screen in the game.



- **Player markers:** As the playtesting showed that our players also enjoyed mirror matches, we decided that the game should not forbid that, as it was initially planned. But we still needed something to distinguish the players, if they were playing as four olives, for example. That's why we implemented number indicators for the players.



- **Improved animations and oven:** The oven is no longer only a skybox, it is fully modelled including the ventilator. The toppings animations were also improved and the pizza got a new texture.



- **Improved air circulation:** The feedback of the air circulation was improved. The sound effect was recorded anew and its volume adjusted. The ventilator in the oven turns with a speed that is proportional to the air circulation speed. The GUI shows also the air circulation symbol when it is on. In addition, the fog level in the oven is also proportional to the air circulation speed. We decided to use fog instead of particles to make it visually more appealing.

2 Schedule and Progress

2.1 Overview

Overall, we could stick to the schedule pretty well. But because one of our team members got very sick and couldn't contribute for most of the time, we had to rearrange things a bit. The remaining team members had to implement his tasks (AI, attacks and physics) which went pretty well. But we couldn't focus as much on the physics as we wanted to. The playtesting session helped us to understand that realistic physics is not that important for a game like this, so we decided to leave most of it and only do some minor improvements and fixes.

When looking back on what we planned and what we reached, we could say it was a success. If we would have had all our team members, the schedule would have fit pretty well and we wouldn't have had any problems to reach everything we wanted to implement.

2.2 Layered Development Progress

Below we present the layered development schedule until and highlight the features which are implemented (green) and the features we could only partially implement or had to change (blue).

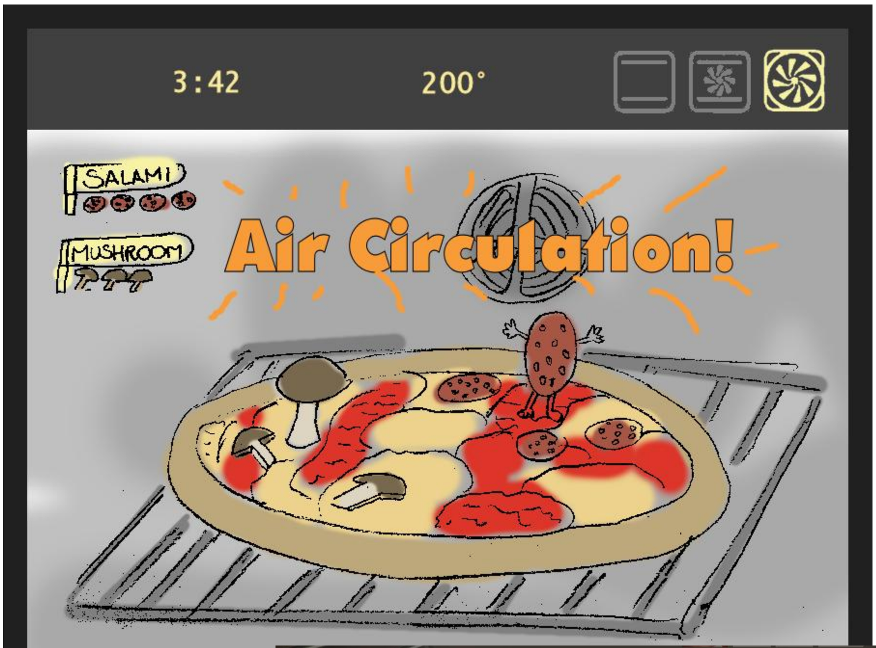
- **Functional minimum**
  - A pizza (plane) with objects (spheres) that can bump into each other and die if they move too far away from the center
  - Very simple physics
  - Minimal GUI
  - The game allows to determine and show a winner.
- **Low target**
  - 2 toppings
  - Jumping
  - A simple default attack (kick)
  - "Good enough" physics, that make the game fun to play
  - Minimal graphics to show that it is a pizza
  - Splash-, player select and winner screen
  - Simple AI
- **Desirable target**:
  - 4 toppings
  - One individual attack per topping
  - Air circulation
  - Dynamic camera
  - Sound effects
  - Good physics -> Good enough physics
  - Nice models and textures, animations (rigging) (olive missing)
  - Base toppings (cheese and tomato sauce)
  - Better AI
  - Music

- **High target**:
  - 4-6 toppings
  - Two unique attacks per topping -> two toppings have two attacks, two toppings have one
  - Particle systems (using an engine; for air circulation and tomato sauce splash) -> fog
  - Really smooth movement and good controls
  - Shaders/Textures/Lighting to give the game a unique, “toon-ish” look
  - Game mechanics to make the game harder over time
  - Nice AI, that’s actually a bit of a challenge to play against

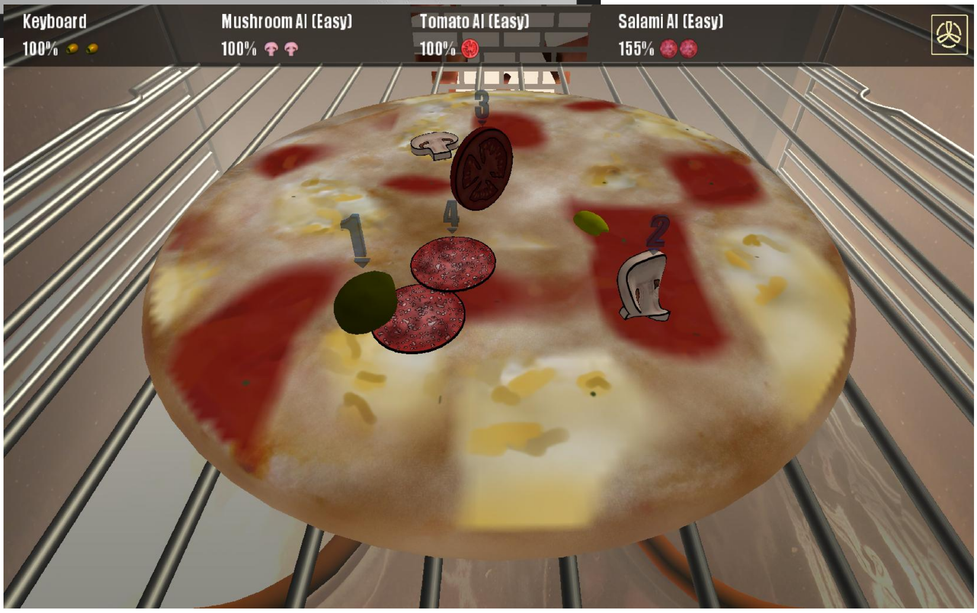
3 Conclusion

3.1 From the Idea to the Final Game

It is interesting how well we could turn our ideas into a game. When comparing the first concept art to the final screenshot, there are many similarities. From the beginning, we had a very clear idea of how our game should be and that helped a lot with the development process.



← First Concept Art



Final Screenshot →

3.2 Feedback on the course

We really liked the course, it was well structured and tackled interesting topics, so that all of us could learn something they didn't know.

The schedule was not too tight for us, because we decided to listen to all the people telling us that they had to work 40h/week because they wanted to create something really big in this short time and they couldn't even finish it. So we went with the strategy to set a relatively low target and create a "simple" game that we could finish.

The **"Food"** theme of the course certainly helped us finding a game idea, and with **Pizza War**, where you play on food and are food, we fully included the theme.

3.3 Successes and Difficulties

Overall, we're pretty happy with the results. Sure, the physics could be better, and more toppings and more attacks could be implemented. First, we were not sure if our game would be fun, but the playtesting session really helped us getting more confident with what we created.

Our greatest success is also our greatest difficulty: Getting animations into the game.

3.3.1 Regarding Monogame

The framework itself is fine and helped us to render something on the screen pretty fast. We also liked that it doesn't limit you the way Unity does. But we had also several problems with monogame.

The first one is that the community is small and there are only a few tutorials out there. Googling problems with monogame was a pain in the neck – if you Google an Unity problem, you have a lot of people with the same problem and a huge amount of answers. For Monogame, the questions weren't even answered.

Also, we had huge problems getting our animations in the game. Monogame has its very own way to deal with certains transforms in model files and we spent a lot of time to find the source of the problem. But eventually, we did it.