Pandemic - The Arcade-Style Shuffleboard

The Dutch Shuffleboard

A Dutch shuffleboard is a long wooden board with 4 gates (holes) at the far end. The goal of the game, called "sjoelen", is to shove 30 wooden pucks into those gates. A number above the gates indicates the score you get when your puck enters that gate (1 - 4). But you also get 10 points extra when you get one puck in every gate for a total of 20 points. When all pucks have been shoved, the player takes back the pucks that didn't enter any gate for the next round. The game ends after 3 rounds.

In the Netherlands, this is a classic family game that is played by young and old in every generation (mostly in the days when there was no XBox or Playstation). Have a look at the shoveling moves of [Siem Oostenbrink](), who is 3 times world champion!

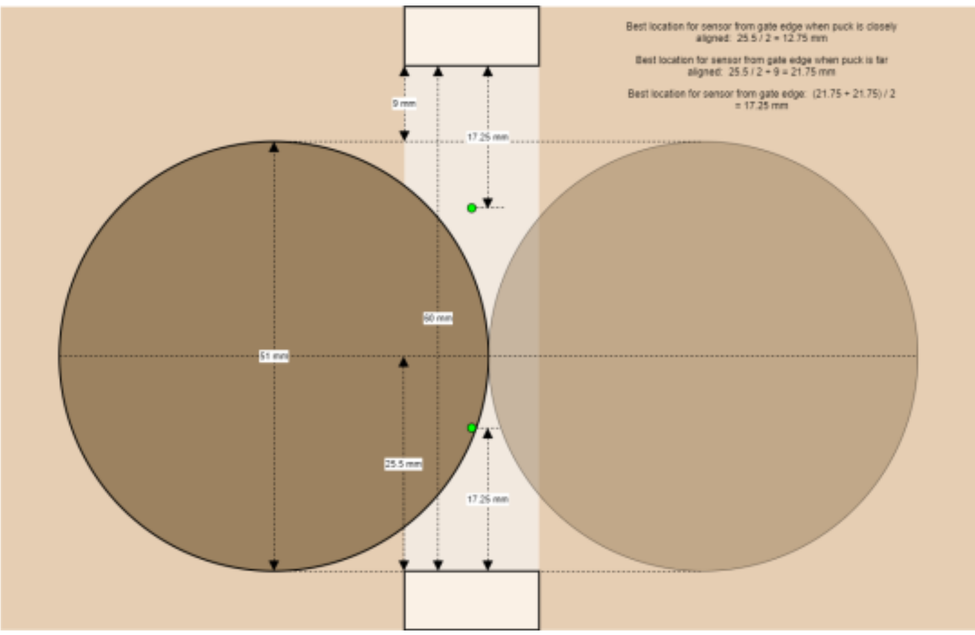
The Concept

One day I woke up with this idea to turn a classic Dutch shuffleboard into a flashy, arcade-style game, the same style that pinball machines have. This is what I had in mind:

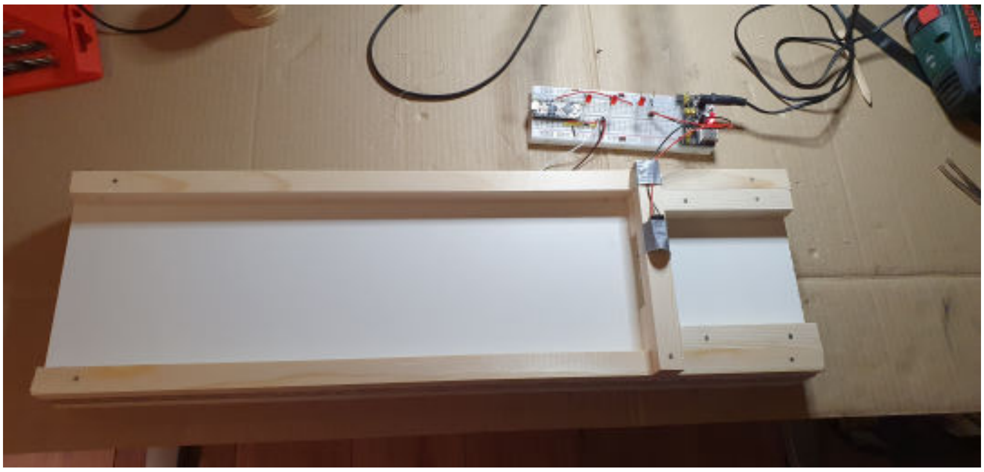
- LED Matrix display with text and flashy animations.
- Sound effects.
- Illuminated buttons.
- Score counting.

Something I also considered, was bringing pucks back to the front automatically. But that idea was short lived as it was clearly going to be very complex.

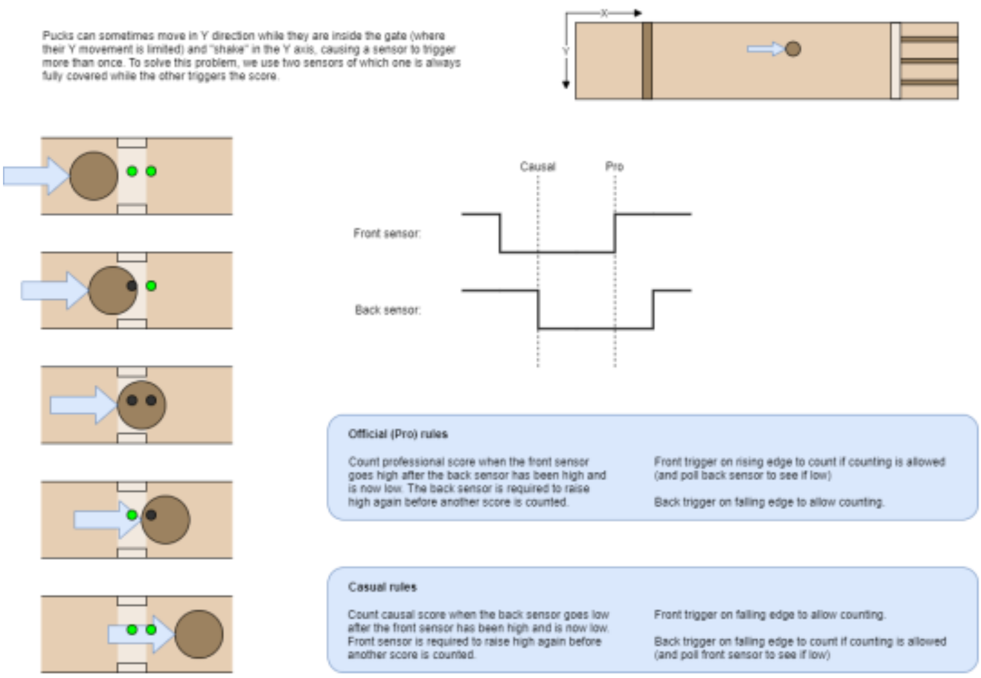
The idea for the score counting was to do this with infrared LEDs and sensors. I hadn't done this before and wasn't sure if it would work. First I calculated what the best position for the sensor would be to see pucks pass by the gates. In the middle is not a good idea because two pucks can push each other causing them to touch and the sensor can't tell the difference between the two pucks. Also, along the sides of the gate is not a good idea because there is some space between the diameter of the puck and the width of the gate. So a puck could pass unseen.

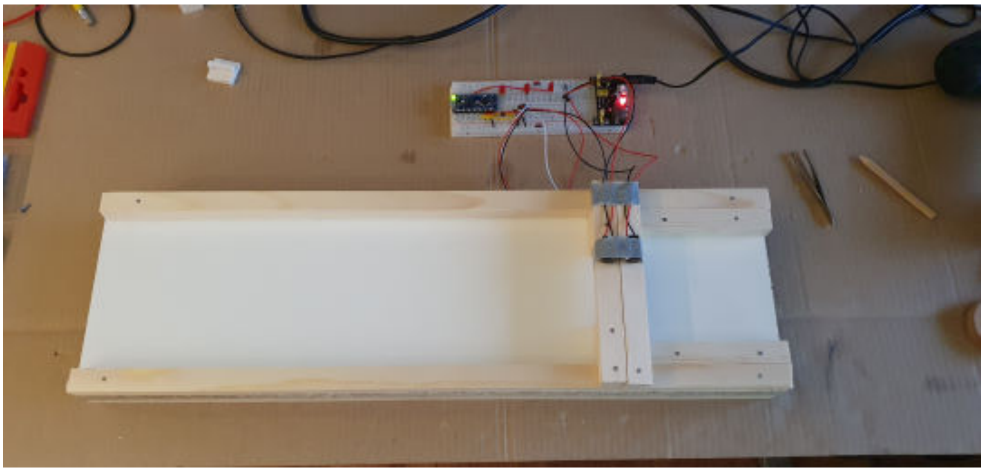


To make sure that this will work, I made a test setup. This consists of a mini shuffleboard with 1 gate that is the same width as a normal gate. An Arduino with an LED and a sensor should detect the puck.



The test showed that the pucks are sometimes counted more than once when they go through the gate. This happens because the pucks can still move along the "Y" axis halfway into the gate and break the sensor beam more than once. The solution for this was to use 2 sensors in succession.





The Design

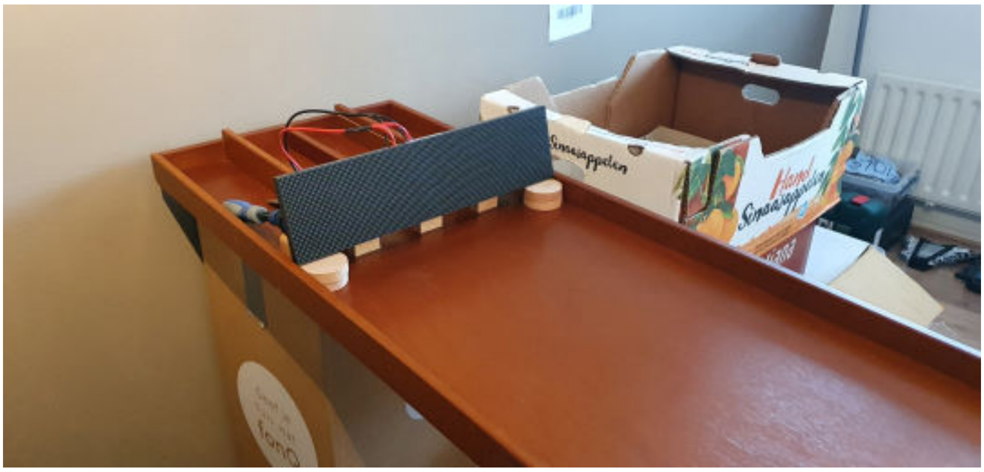
Because I only have 1 shuffleboard, I work as much as possible according to the waterfall model (first-time-right). This means I need to have a pretty good design before I start sawing and drilling.

For the design I chose to use Fusion 360. This is because it is free and my colleagues also used it (and praise it). I had never used it before, but with tips from my colleagues I came a long way. So I accurately measured the original shuffleboard and made a 3D model of it.

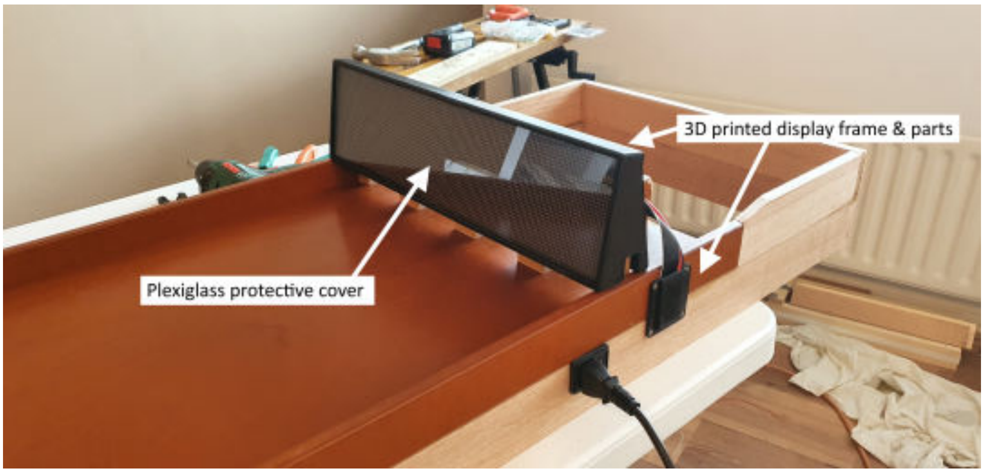
Underneath the shuffleboard, I added a wooden housing which makes the whole thing 5.5 cm higher. This gives room for the electronics, the speakers and the buttons. The 5.5 cm was chosen partly based on a minimum height that I need for the electronics and partly on the size of the wooden planks I could find.

One of the downsides of the original shuffleboard is that the slots behind the gates don't offer enough space to store pucks side by side. Someone should always be stacking pucks behind the gates to keep the slots from filling up. I solved this by removing a large part of the slots (about 15 cm behind the gates) and letting the pucks fall into a bin. Thanks to the extra 5.5 cm height, this was easily possible.

I wanted the display to sit above the ports, so that the display can also "point out" the ports where you need to score. That also happens to be a nice distance for the player to see the display well. I have chosen 2 Adafruit LED Matrix displays of 64x32 pixels each for a total size of 128x32 pixels. These just fit inside the width of the shuffleboard itself, a perfect size!



To make it nice, I will print a plastic frame in which they fit neatly. The frame is designed to sit loosely on top of the ports so that the entire display can be easily removed for transport. There is a slot in the front of the frame for a plexiglass sheet that protects the LEDs from flying pucks. The display is tilted 10 degrees backwards for better visibility from the player's position. The bottom also slopes so that it nicely overlaps over the original dots above the ports (and this gives extra stability).



I also mapped out the electronics and thought about how they work together. For the main computing I use a Raspberry Pi. For controlling the displays, I used an Adafruit Matrix Raspberry Pi HAT. This seems fitting, as the displays are from Adafruit as well. The HAT uses almost all GPIO pins (including all PWM pins) and the software library for this also uses an internal timer that is normally used for the standard audio output. This means that there are no GPIO pins left for the sensors and I have no audio output.

I therefore connect the sensors to an Arduino (Nano Every, but an Uno would do as well). The Arduinuo is dedicated to reading the IR sensors in real-time, so that every fast movement is picked up. The Arduino creates simple messages that are sent to the Raspberry Pi via the serial interface (over USB). These messages do not have to be picked up in real-time. There are 10 infrared LEDs, 2 of which are high power, to illuminate the sensors. Together, these require too much power to pull from the (USB-powered) Arduino. That's why I feed them directly from the PSU. They will be continuously lit (which reduces the lifespan), but I can always replace the LEDs if needed.

For the audio I connect via USB an audio module (PCM DAC chip) which generates an analog output, which I then pass through a small Adafruit MAX98306 amplifier for the 3W stereo speakers.