Robert Hendler
Paul Lehrman
Electronic Musical Instrument Design
May 10, 2018

PRIDI: The Party Ready Interactive Digital Instrument

The PRIDI is the spiritual successor to the SHIDI. Much like its original incarnation it is a table on which you roll balls to create noise. The lateral position of a ball (that is, to the user's left and right) dictates its pitch, whereas the vertical position (that is, forward and backwards along the table) dictates the volume of a note. The table is still slanted so that balls will still roll towards the quiet side of the table and sounds will tend to naturally die out.

Now, however, we've added controls for each ball, "Ball Pallets" to further increase playability, and obstacles to guide balls along their paths. Balls will also make noises when they collide with each other or with a wall.

The balls' positions are still monitored via a camera mounted above the table, and all of the processing grunt work is done by a python script. A Max patch takes in information from the python script, as well as information from an arduino inside the control box, and generates the sounds.

The Table

The table of the PRIDI is the same as the table of the SHIDI with a few improvements. Firstly, it has been painted black. It increased the contrast between the table and the balls and now the camera has an easier time picking them out from the background.

Secondly neoprene sheeting was added to the walls of the table. In the SHIDI, when a ball would hit a wall it would make a very loud wood on wood clack. We liked the idea that a collision with a wall would make a sound, but we didn't like that it wasn't produced electronically. The neoprene sheets dampen all collisions with walls so we can generate the sounds ourselves.

The next big change is the QR code blocks. Much like the SHIDI the python script uses 4 QR codes to find the playing area of the table. On the SHIDI these were printed paper, taped to the corners of the table. Those taped little pieces of paper got bent, lost, fell off the table, and all other manner of bad things happened to them. The PRIDI has replaced those SHIDI papers with carved wooden QR blocks. The blocks not only stay on the table better but they're larger as well, meaning the camera has an easier time seeing them.

The camera mount has been replaced by an adjustable mic stand and 3D printed camera/light mount. Decoupling the camera mount from the table allows us to adjust the height and orientation of the camera at will if we feel like something is off. It also, along with the removeable QR blocks, allows any table to be used as the PRIDI table, not just our custom one. Arrange the QR blocks on the four corners of, say, a pool table, and you can make some fun noises by playing pool (after you've adjusted the camera parameters we'll talk about later).

A neoprene wall was also added to the base of the table to prevent balls from rolling off. It's not backed by wood, so it can be moved out of the way in case you need the extra room to roll a ball the way you want to.

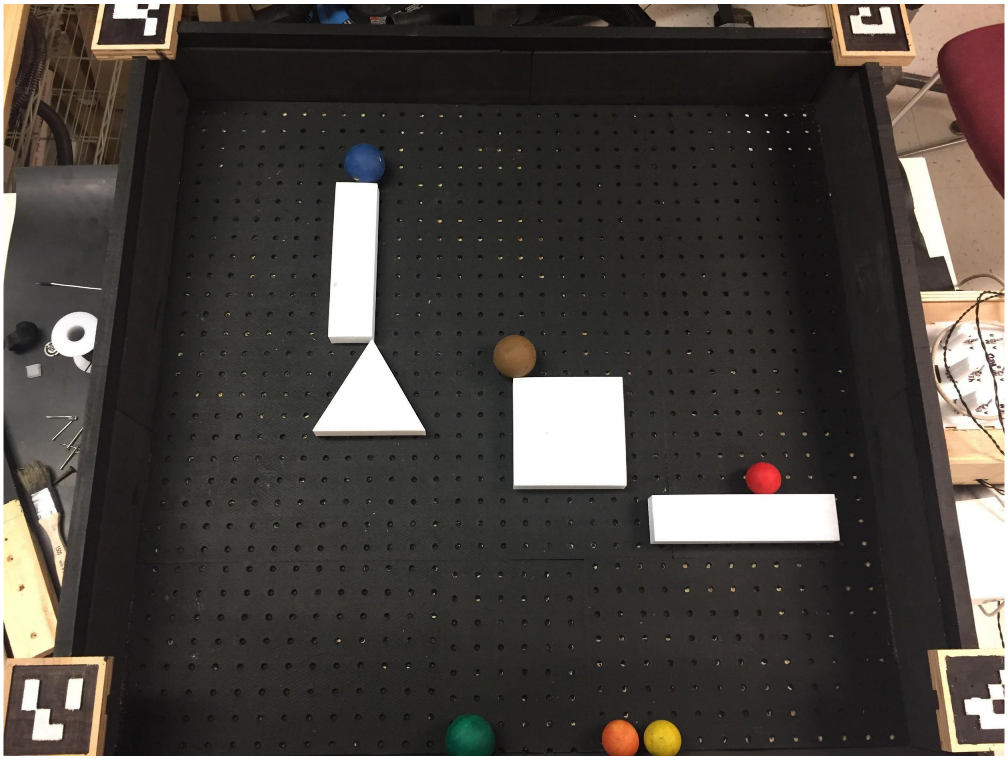

Fig 1 - The PRIDI table featuring new black paint job, QR blocks, neoprene siding

Fig 2 - The PRIDI camera and light mount in the jaws of a mic stand

The Accessories

During the creation of the SHIDI we chose to use a peg board as the face of our table. We wanted to be able to make up some obstacles to fit into the board to guide the balls along certain paths of to block out areas of the board to make it easier to play music. These obstacles didn’t come to fruition with the SHIDI, but we proudly present 3 variations with the PRIDI! A triangle, square, and bar can be seen in figure 1 above. Below you can see a CAD model of the bottom of the triangle. They are hollow, to reduce weight and waste, but mostly printing time, and they only attach at two points along the peg board. Any other shape could also be made for this purpose, provided it could fit on a 3D printer.

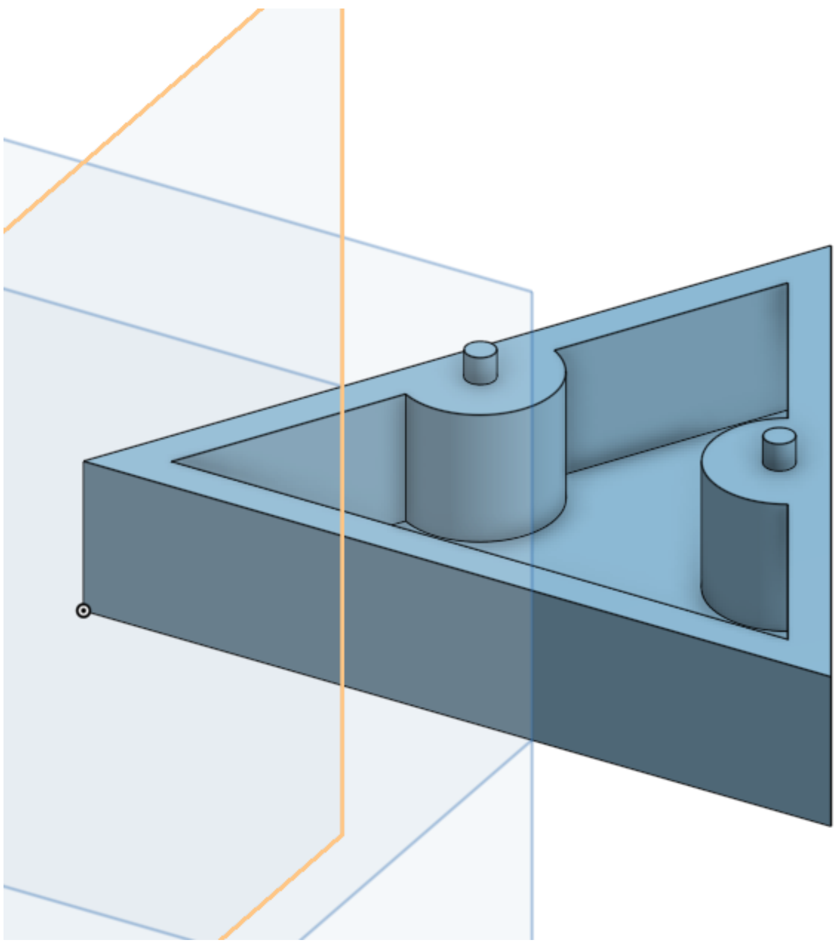

Fig 3 - The underside of a triangle obstacle

One other accessory was created to facilitate the playability of this instrument. A control box, seen below in figure 4. The control box consists of nine on-on switches, three 6-position rotary encoders, 1 power switch, a USB hub, and an arduino. The power switch is wired in line with the light next to the camera, which is a direct improvement from having to reach up and turn on the flashlight for the SHIDI.

The switches and knobs are separated into 3 groups we call pallets, each containing one knob and 3 switches. Each pallet controls the sounds for 3 similarly colored balls, and each ball on the pallet will sound similar but not the same. The switches in a pallet control whether an individual ball will play with discrete pitches or continuous pitch as it rolls along the x-axis of the table. The knob controls what patch each ball's notes are routed through. Each knob has 6 positions, the first being off (no sound for that pallet) and the other 5 being reserved for different patch groups.

Both the camera and the arduino are plugged into the USB hub inside the control box, and the outlet of the usb hub is routed to a computer. That way we only have one cord going from the box to the computer, giving the PRIDI a cleaner appearance

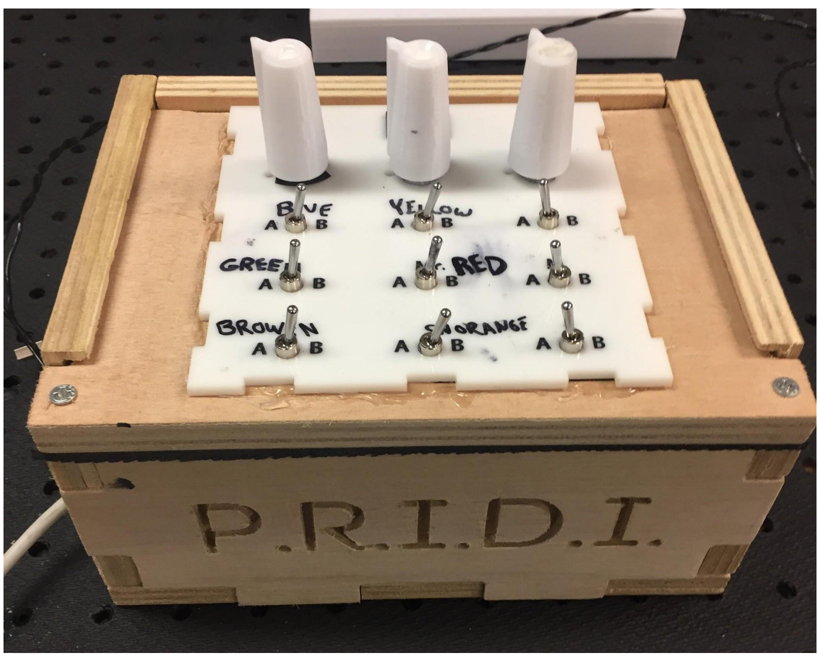

Fig 4 - The PRIDI control box. Power switch not depicted, it is hidden by the middle knob

The Code

A few quality of life additions were made to the code, first of which being an adjustment slider for the camera's exposure. When we initially painted the table black the camera assumed it was looking at a very dark room and automatically increased its exposure. In reality it was not looking at a dark room and increasing the exposure ruined the object detection. Adding the exposure slider allowed us to fix this issue.

We also pivoted from the SHIDI's size based ball differentiation to a more color based one. As mentioned above, each pallet contains 3 balls of similar colors. We decided to go that path because the camera often had trouble differentiating between ball sizes. Usually it would say that a ball was much larger than it was and it would swap its ID with another ball that was actually much larger. Now the code uses the hue and saturation of a ball to determine its ID. Unfortunately by the time we made that swap we only had enough time to set up 6 colors of ball, and so only 2 pallets were present at the final presentation.

The most major addition to the python script was to add collision detection. After applying an image mask to the frame to determine ball locations, the code will look for two types of collisions. The first is a collision between a ball and a wall. To determine if that took place, the code checks if the coordinates of the ball match up with the extremes of X and Y axes. The second collision type is between two balls. To check for that the code checks if any of the masks' coordinates overlap. Those collisions got added to the packet that gets sent from the python code to the max patch.

The max patch was majorly updated to change and add some functionality. Originally in the SHIDI notes were played more like a marimba rolling on a bar than like on sustained note. This has been fixed, and now as long as a ball is on the table it will play a continuous sustained note. It also handles the control box inputs. For instance, the figure below shows how the max patch handles a pallet being turned on or off. The figure under that shows how the patch handles changing reason patches using the control knob. It will flush all three balls' channels, and then send the python script a command saying "Recheck these three balls, and send me new note on commands if they're on the table."

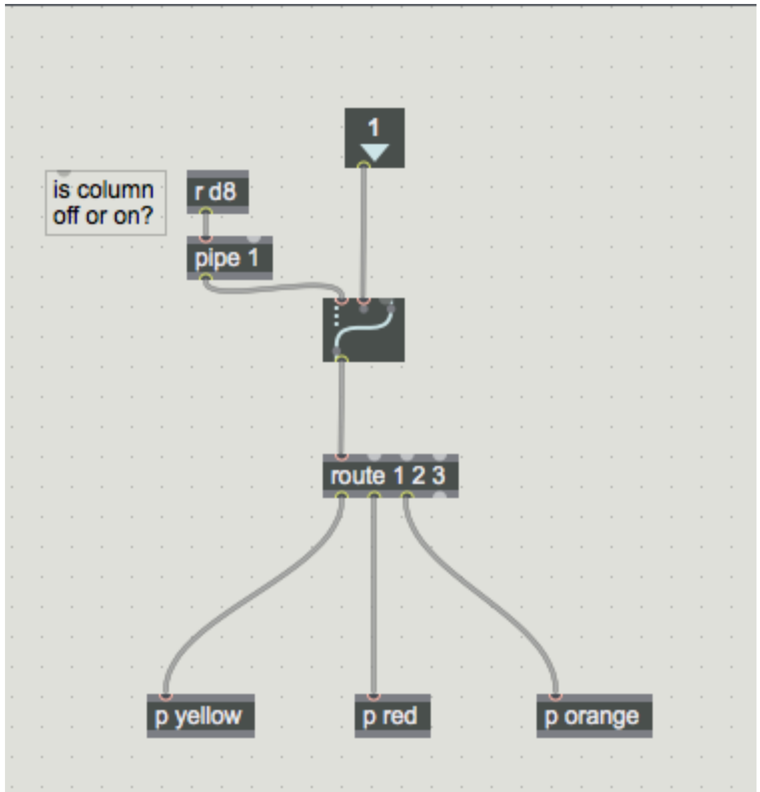

Fig 5 - Turning on or off a pallet

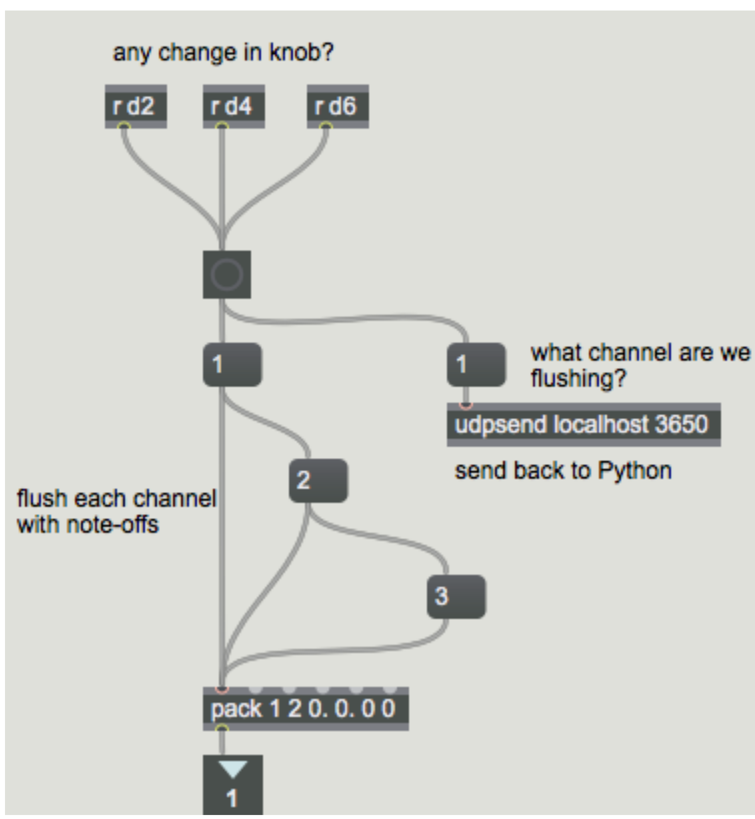

Fig 6 - Updating sounds when a knob is turned

Once the max patch figures out what sounds need to be produced it sends MIDI commands to a reason patch. Some workarounds had to be implemented though to limit the number of channels we used. Firstly, when a ball sends its note-on command to reason it will always be sent for the same note. For instance, blue is always sent as a C3. This lets us put all three balls into one instrument. Pitch bend is then applied to the notes to correct it to the actual pitch we intend to play. Without this fix we'd have to have 45 different channels for the entire instrument. With this fix we need significantly fewer. Below is an example of what was just described.

Fig 7 - One pallet's reason patch

Moving Forward

Future iterations of the PRIDI could be improved. Firstly we only added functionality for 6 colors of ball. Ideally we would have 9 colors that we could group into 3, similarly hued, groups. The PRIDI could also have an adjustable table angle. During part of our design process we had toyed with adding legs, but since the tables in the EMID lab are much thinner than the PRIDI we dropped the idea. Adjustable legs would allow a user to change how quickly a ball's sound dissipated. More obstacles could also be created, and they could be created for specific sound profiles. A certain curve could be used to play a repeatable melody for instance, allowing the user to compose music for the PRIDI.