

Red vs. Blue Performance Instrument

by Peter O'Regan

Team Members: Jonathan Evans and Nicholas Pasquariello

Electronic Musical Instrument Design Final Project

May 9, 2008

We designed and built an interactive two-musician music performance system for our final project, codenamed Red vs. Blue. In the initial development stage of the project, we decided that we wanted to create a large instrument which required multiple player's gestures and allowed interaction between the musicians. We wanted something that could be prominently displayed, yet portable enough to adjust to different performance venues. Our final design took the form of two boxes identical in design and differing in color that were tethered together, each measuring roughly 11"x17"x6" and controlling the majority of musical expressions through knobs, switches, and faders.

The instrument plays synth lead sounds, drum kits, and bass pads. The idea behind the use of identical consoles is to give each player the opportunity to control all portions of the instrument, though at different times. While one player handles the lead, the other is able to play the pads and drum kit. When the physical connection between the players (taking the form of a string) is disturbed, the players access the controls of their partner's portion of the instrument. The type of change varies from subtle trades of controllers to completely exchanging the functions of the consoles. The musical performance is formally initiated by each player flipping a master toggle switch on his console to the "on" position.

Each console was built from an aluminum Hammond box. Ordering them prefabricated ensured that they were uniform and cut down on production time. The majority of cuts to the box were machined with a computerized cutting system. This made them very clean and precise, as well as giving extremely snug fits to the hardware. Mounted on the top panel of each box, starting from the left is a column of four function buttons, a column of four rotary encoders, a master toggle switch, an array of five pushbuttons, and an IR sensor seated above two faders. Two 1/4-inch jacks were attached on the rear panel. Inside, simple circuits were wired to connect the switch-type controls and the resistors. Components were grounded to the box housing and isolated from the housing with electrical tape, where necessary. Most of the hardware was screwed in, though some elements such as the sliders were secured with glue. Two holes in the rear panel and an interior-mounted lever switch tensioned with a spring control the tautness of the string connection (each box has one lever switch and one string terminus). The electrical connections all enter and exit the consoles through a wide piece of ribbon cable.

The functions of the buttons vary depending on whether the box is set to play the synth lead or the drum and pad set. In synth mode, the four function buttons are used to switch between sounds of varying timbre, the rotary encoders adjust the amplitude ADSR envelope of the current synth, and one of the sliders controls the volume of the synth. The second slider is used to select the pitch played, and its sounding and duration are triggered by two of the buttons. In this configuration, polyphony and sustains are possible. A pedal switch connected to a rear panel jack allows for all notes to be ended in unison.

In pad mode, the first two function buttons select between and enable the five pushbuttons to play different drum kits. The second two function buttons allow the same with pads. In all four

states, the rotary encoders select the ADSR envelope, and the faders control velocity and volume. As with the synth configuration, an attached stomp pedal to the rear panel jack allows for quick termination of all currently playing notes.

While at first it may seem that each player is constrained to his own portion of the instrument, flicking one of the two strings connecting the consoles allow the musicians to interact further than just blending sounds. One string sends light-hearted taunts each time it is struck, encouraging some competition and thus excitement between the players. The second switch enacts randomized changes to the console configurations and provides audio commentary and instructions to the performers and to the audience. Some of the temporary changes it allows are for the ADSR controls of each player's box to become routed to the other's portion of the instrument, the ability to switch the partner's synth, drum or pad timbre, and a scale change for the synth. It also can exchange the role of the two consoles, and this function is not time-limited.

Electrical commands received from the boxes are received by the attached Doepfer box and converted into MIDI signals, which then travel via an Edirol FA-101 interface into a computer running Max/MSP and Reason. Max processes the raw MIDI data into a form that the Reason patch has been programmed to play in a musical fashion.

Once note or controller data is received in Max, it runs through a splitter to separate the signals coming from the red console from those coming from the blue console. The splitter also identifies data from the string switches and toggle buttons sends it down the proper channel. Red notes and data are sent to a small routing subpatch that marks the data according to its button origin. Blue has a similar subpatch. The results of both of these identifier subpatches are then moved using the send command to the main routing module, which reads the state of the system (a code to mark red as synth or blue as synth) and opens the proper sets of gates and switches to flow the data from the consoles splitting modules to the synth and drum pad instrument modules. The button inputs here are standardized. For example the FUN1R button is the red function 1 button while FUN1B is from the blue console. These will flow into an area either FUN1S or FUN1P, depending on if the data is going to a synth or pad for that particular console.

The dedicated state-oriented button controls are received at the instrument module, where they are decoded and their functionality is specialized. In the synth module, one of the sliders determines pitch by going through a divisor and using the `zl list` command in conjunction with a scale to send out different elements in the scale based on the slider's position. This continually updates on motion with the slider, though the buttons need to be used to bang the output (starting and stopping notes with that pitch) The drum and pad module uses `zl list` commands to easily set the destination notes of the five pushbuttons to Reason, coupled with one of the slider inputs to determine its outgoing velocity when not a note off. Both modules also have the one-stomp all-notes off switch (which runs using a flush object) that is triggered from the stomp pedal and also automatically during console state changes to prevent stuck notes. Also present on each is a complete ADSR envelope control which uses accumulator objects to respond to forward and reverse motions on the rotary encoders. This method of control preserves the envelope on the individual synths when the role of the consoles switches. All note and controller data is reprogrammed for Reason in these patches and then sent out Max's output port.

The taunt and switching string events are routed to a separate subsystem within Max. The taunt trigger proceeds to a random number generator, which picks a sample number from the pool

of taunts and sends the appropriate information to Reason. The switch trigger also runs through a randomizer to determine what type of switch it will be. Using two state variables to define the consoles, the role of each console (the ConsoleState variable) and whether there is a switch event in progress can be determined (the SystemState variable). This also prevents the system from retriggering and reversing roles during a temporary switch event; instead it just elongates the time allotted and announces a new temporary switch type (when necessary). The time allotted is determined remotely and runs through the SwitchTimer subpatch, which uses a counter to determine the time allotment and send audio feedback about the change time. The entire system uses easily adjustable global variables and is highly modular for additional features and switching types.

The information about the type of switch occurring is disseminated through the various routing modules. At the base of the main routing subpatch, receivers and additional gating mechanisms allow for seamless and robust routing of specific pieces of hardware during ADSR and other types of switching.

The output result of the synth, pad drum, and audio feedback Max modules are sent to function-dedicated modules within Reason. The first four channels are used for the synth patches (Subtractor modules), the next two for drums (NN-XT modules), the next two for pads (Thor modules), and the last one is set to play all audio commands resulting from taunts or switching commentary (NN-XT module).

The performance system works extremely well, with no detectible latency from the switching and all hardware working exactly as it should. The build quality makes the unit durable to withstand heavy pounding. There were many challenges to get it to this point, however. Aside from the expected amount of debugging and tweaking within the Max patch, several hardware problems were encountered. Some, such as needing to insulate the faders from the ground, could be countered with electrical tape. Others, such as isolating the IR circuits from the other controllers, proved to be too involved for the time constraints and workarounds had to be invented. The original plan for the IR sensors, for example, was to control the pitch index; this was shifted to a controller after the IR sensors were determined to be beyond unreliable and limited in their reporting (this may have been specific to the installed ones). Given more time and the modular nature of the routing and switching patches, many additional switching features could be implemented very easily. Additional playback modes for the synth, such as glissando, could also be programmed. Further time would enable the functionality of the master toggle switches to move beyond their specifically scripted sample triggering to a more adaptable starting system (as it had been originally conceived), the taunt sample pool could be expanded, and a timer system to control the duration of the entire performance could be added.

Unimplemented bells and whistles notwithstanding, Red vs. Blue provides a fun, interactive way for two musicians to jam. Consisting of familiar knob and switch controls, it is not hard to learn how to play. Consisting of two consoles joined by strings of adjustable length, it can fill any size performance space and is relatively portable. Containing rhythmic, harmonic, and melodic aspects, it is capable of performing entire arrangements. Containing dynamic performance elements, continuously adjustable sound parameters, and a degree of randomness, it is capable of obtaining and keeping an audience's attention. Though it is unlikely we will see an influx of demand for the it any time soon, Red vs. Blue's qualities as a complete performance instrument make it a desirable and expressive musical device to have.

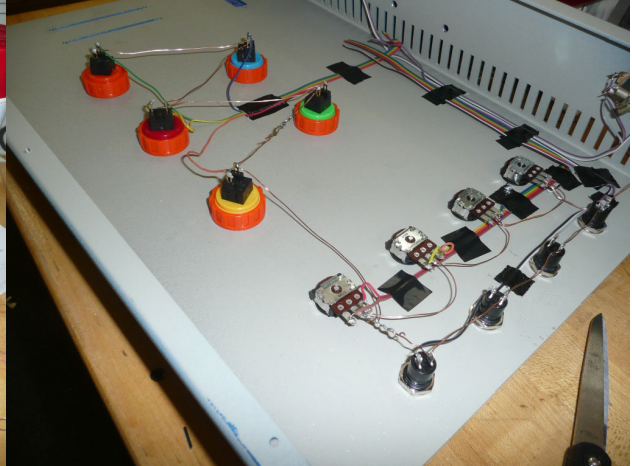
Production Photos and Diagrams

Figure A



Red console, hardware attached

Figure B



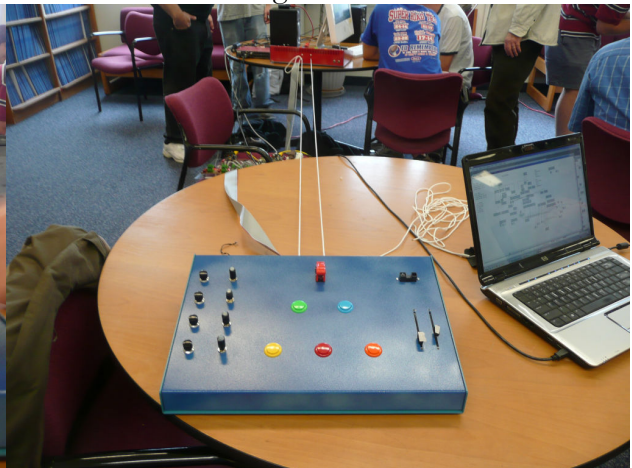
Console wiring

Figure C



Blue console in performance

Figure D



Blue console and laptop running Max

Figure G - Drum Subpatch

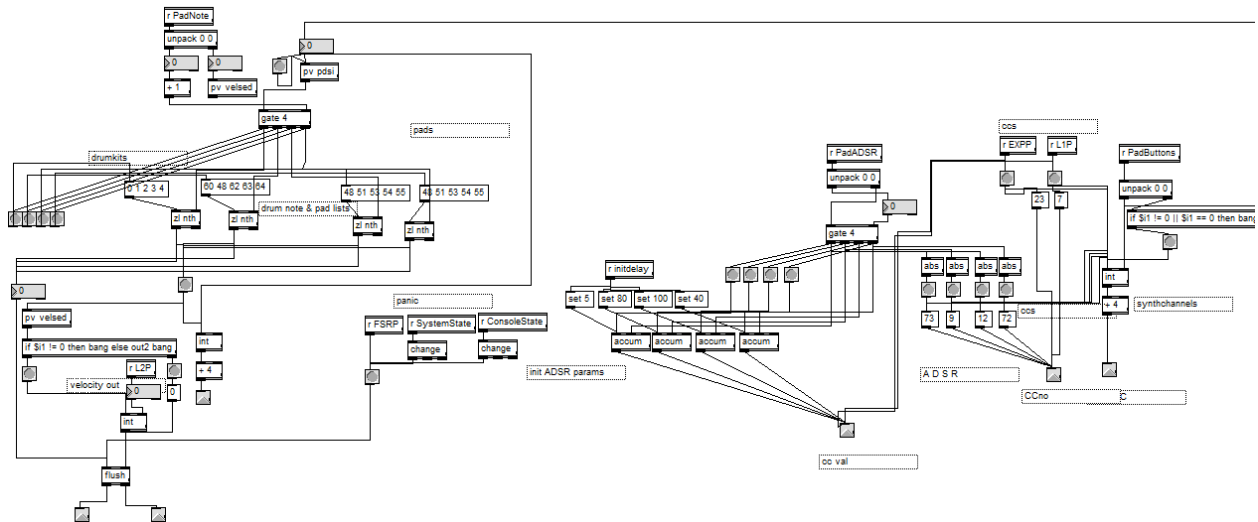


Figure H - Portion of the Routing Subpatch

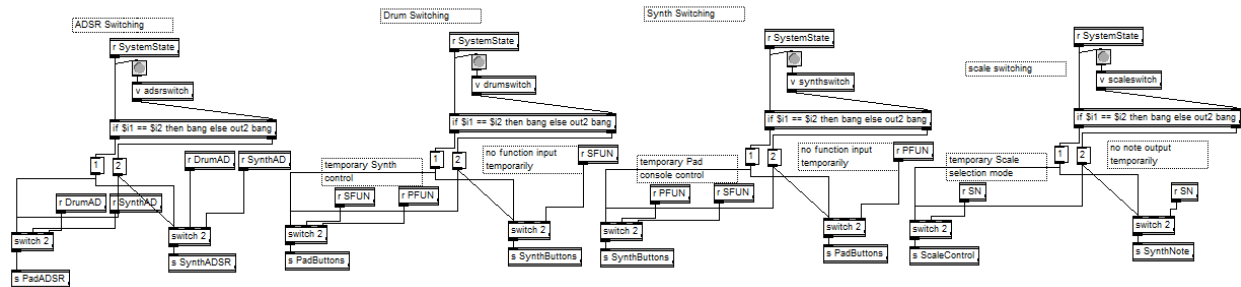


Figure I – Red Console's Preliminary Routing Subpatch

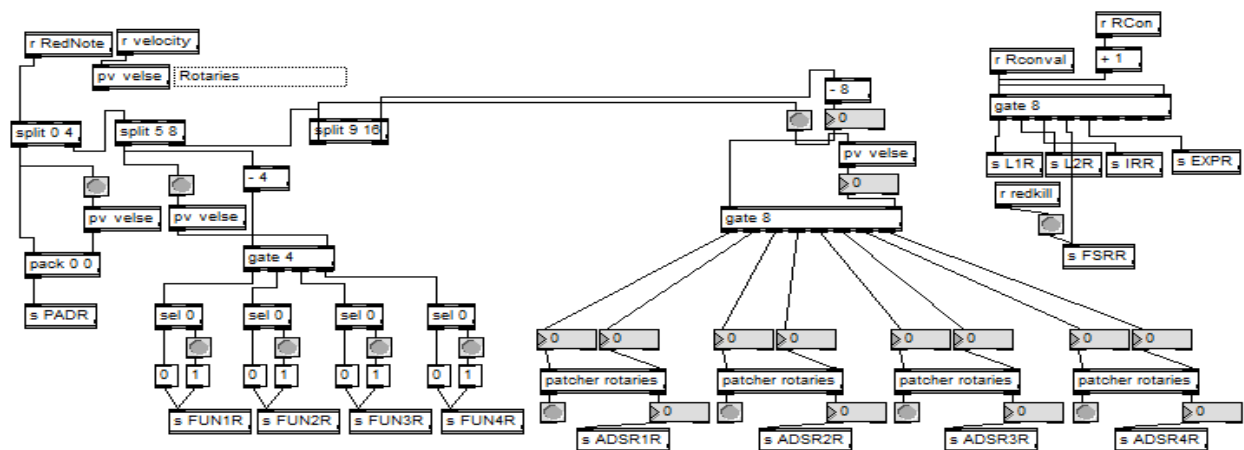


Figure J – Max Module Diagram

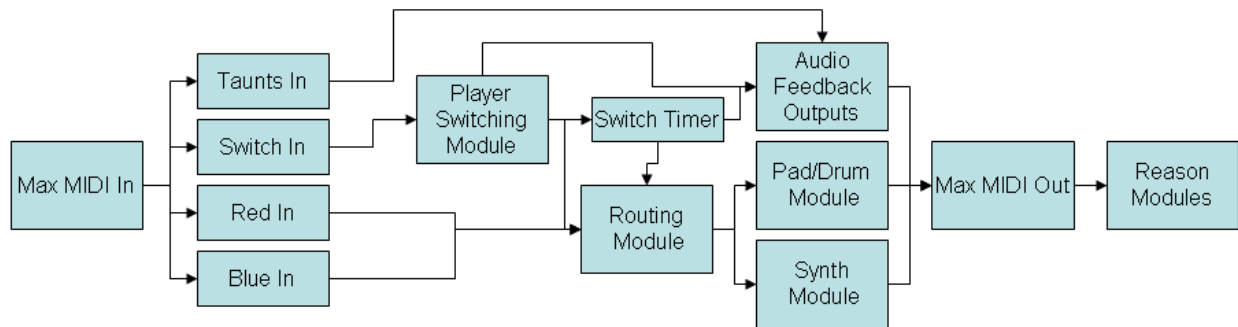


Figure K - Reason Patch

