An Exploration of Peg Solitaire as a Compositional Tool

Kirsty Keatch
Department of Music
University of Edinburgh
Alison House
12 Nicolson Square
Edinburgh
EH8 9DF
kcikeatch@gmail.com

ABSTRACT

Sounds of Solitaire is a novel interface for musical expression based on an extended peg solitaire board as a generator of live musical composition. The classic puzzle game, for one person, is extended by mapping the moves of the game through a self contained system using Arduino and Raspberry Pi, triggering both analogue and digital sound. The solitaire board, as instrument, is presented as a wood and Perspex box with the hardware inside. Ball bearings function as both solitaire pegs and switches, while a purpose built solenoid controlled monochord and ball bearing run provide the analogue sound source, which is digitally manipulated in real-time, according to the sequences of game moves. The creative intention of *Sounds of Solitaire* is that the playful approach to participation in a musical experience, provided by the material for music making in real-time, demonstrates an integrated approach to concepts of composing, performing and listening.

Keywords

Interface, Peg Solitaire, Monochord, Arduino, Raspberry Pi, Pure Data.

1. INTRODUCTION

Sounds of Solitaire is an undergraduate final year music technology project presented to Edinburgh University. The aim is to develop a novel interface to allow for interactivity between the mapping of the moves of a traditional game to both analogue and digital music systems, in order to generate musical compositions.

2. CONTEXT

In the mid 20th century, John Cage's aleatoric theatre piece, *Reunion* (1968), used a “sound distributing chess board” [1], built by Lowell Cross. There, the moves of the game played by Cage and his Dadaist friend Duchamp, triggered sounds played by musicians, essentially using the chess board as a form of mixing desk.

Despite the semi indeterminate nature of the playful characteristic of game pieces of music, the rules of the game do help with "harnessing"[2] compositional formats. Such was the approach taken by John Zorn and his teams of improvisers in pieces such as *Lacrosse* (1976) and *Cobra* (1984). The application of the game format, in *Sounds of Solitaire,* is useful in structuring musical experiences where a playful approach is taken to composition, by way of interaction with the musical material provided.

2.1 Peg Solitaire

Peg Solitaire has been a popular game for a single player for centuries (see Figure 1). The object of the game is to begin with a peg in each of the 33 holes, except the centre, and then, by making a series of leapfrog style moves, the peg that is jumped over is removed until the situation is reduced to a single peg in the centre. The game does not stop at one solution: it is also possible to play the game from starting points other than the centre, returning to the chosen point.

Figure 1. Traditional peg solitaire game

3. EXPERIMENTATION

Inspiration for the experimentation stage was taken from kinetic artist, Jean Tinguely's curious sound sculpture *Meta Matic 17*, which reportedly made “40,000 unique drawings”[3] on paper strips reminiscent of the graphic scores used in pianolas and programmable music boxes. The first stage in the project was to experiment with a way of mapping the moves to a musical context using an analogue music generator. The moves of a correct solution were mapped onto a punched paper strip to be wound through a music box mechanism. An analogue sound was produced simply by turning the handle of the music box, but the result was limited to a fixed melody punched laboriously onto the paper, the focus of attention at this point was turned to real-time melodic generation and the question of action reaction to improve user experience.

3.1 Real-time Interaction

Joel Chadabe defined interactive music as: "A performance process wherein a performer shares control of the music by interacting with an instrument that itself generates new material”[4]. In the context of the project, a Raspberry Pi and Arduino were brought in to make the augmented peg solitaire board operate as a stand-alone instrument. Pure Data was used to trigger analogue sounds in real-time using solenoids, while simultaneously manipulating the sounds through the use of digital signal processing.

3.2 Autonomous self-contained system

The first prototype experimented with incorporating the hardware inside the installation. I have a slight issue that some real-time performances and installations have laptops on show even when not in active use by the performer. So one of the requirements of the prototyping stage was to use compact and relatively powerful hardware such as the Arduino and Raspberry Pi.

Raspberry Pi is a tiny computer that can run open source software and due to its small size is ideal for use inside the peg solitaire installation. The Raspberry Pi runs the CCRMA Satellite image to load a Pure Data patch on startup, allowing the box to run autonomously, waiting for user interaction, through the Arduino. The Raspberry Pi comes with only one audio out socket, so in order to feed it audio for installation, a compact Behringer Xenyx 302USB soundcard was used to improve audio output to that of the Raspberry Pi, while providing the necessary balanced input for the contact microphone, as well as channels for the monochord. A flowchart of connections in the board is outlined in Figure 2.

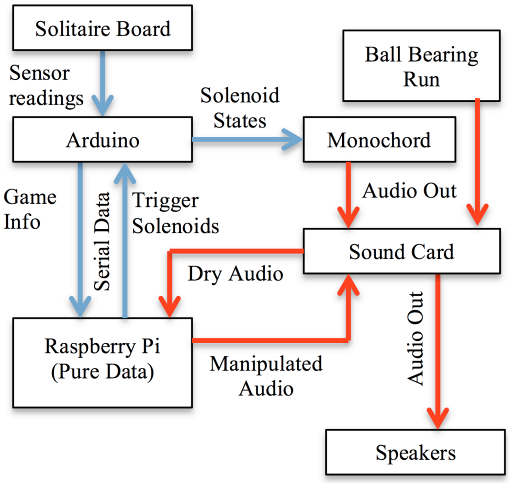


Figure 2. Flowchart of connections

Useful for its compact size, the open source Arduino Nano was chosen as a micro controller for interfacing the switches and solenoids with Pure Data, run on the Raspberry Pi. Like the majority of Arduino boards it only has 8 analogue inputs: to extend the number of inputs to match the 33 positions of the game, five CD-4051 BE multiplexing chips were used, providing 40 inputs. The rules of the game were programmed onto the Arduino, using serial communication to send game statistics such as number of available moves, peg locations and move stage detection as a continuous stream, for mapping in Pure Data.

3.3 The Board

The first working prototype was a wooden box topped with an array of split washers as peg holes, copper caps completed the circuits to trigger sound (see Figure 3). The Arduino and Raspberry Pi were wired inside the box. The copper caps were rejected as they had stability problems and were awkward to handle.

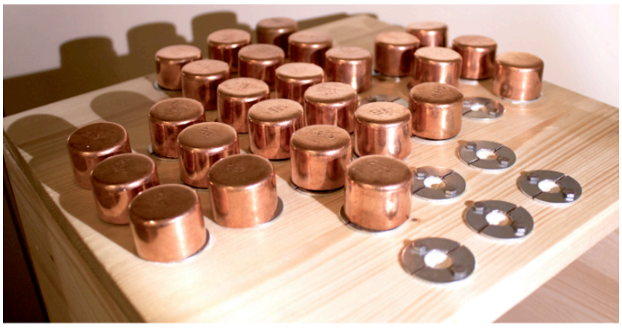


Figure 3. The prototype

In a more elegant solution, the final peg solitaire board is designed with an array of 33 x 3 contact points forming tripods to hold 20mm ball bearings. The ball bearings act as conductive pegs to serve as switches: placing the ball bearing on the tripod sets the switch to "on" completing the circuit with the tripod and so a musical event can be triggered. (see Figures 4 and 5).

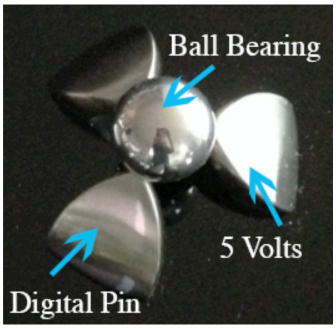


Figure 4. The tripod switch

Figure 5. Profile of the switches

4. AUDIO DEVELOPMENT

The main focus of the musical intention was the experimentation with automation of an electric guitar. Rather than preparing a ready-made guitar, an automated electric monochord was built, played by an EBow and solenoids, triggered from within Pure Data, interacting with the solitaire game.

4.1 The Monochord

The monochord is built using a single electric guitar string, resonated using an EBow [5] (see Figure 6). A 24 gauge unwound string was chosen as it requires less vibration to produce the same volume as a thinner string and also provides a richer tone. EBows set up electromagnetic fields causing the string to vibrate, this is positioned directly above a hum-bucker pickup for optimum resonance.

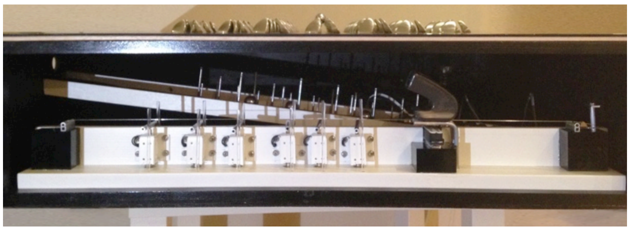


Figure 6. Front side of board showing monochord

To generate seven different notes, six 12v solenoids are positioned to act as movable frets (see Figure 7). Often solenoids are used in similar applications to press down on strings as a mechanical finger, however, the approach of pushing up on the string from below seems to be a more elegant, space-saving solution.

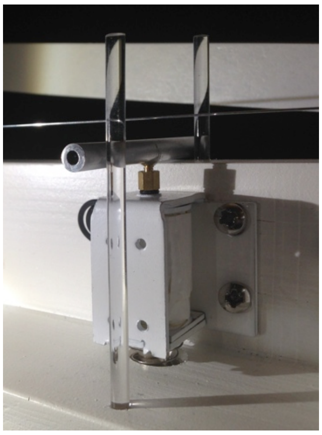


Figure 7. Solenoid moveable fret

The solenoids and EBow are controlled from serial communication from Pure Data to the Arduino. Each time a move is made, Pure Data triggers a solenoid to raise a fret bar to contact the string, depending on the coordinate of the ball bearing on the board. Two clear rods are positioned at either end of the fret to keep it orthogonal to the string. The EBow is used to control duration of the notes and is controlled in the same way as the solenoids. Each solenoid is connected as shown in the circuit diagram in Figure 8.

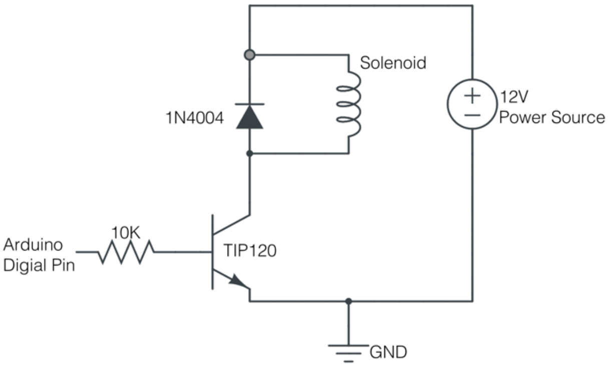


Figure 8. Circuit diagram for one solenoid

4.2 Ball Bearing Run

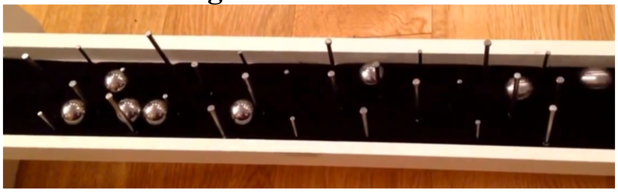


Figure 9. Ball bearing run

There was a small issue with what could be done with the ball bearing once removed from the board. For *Sounds of Solitaire*, they are placed through a hole in the side of the box to roll down a wooden run, punctuated with a pathway of nails of various sizes, adding a percussive dimension to accompany the melody of the monochord (see Figure 9). As the ball bearings zig zag down the incline colliding with the nails, the sound is picked up by a contact microphone. The floor of the run is lined with velvet to reduce the sounds of rolling, allowing the nail sounds to be emphasised, adding an industrial timbre explored in the compositional development.

4.3 Composition Generation

4.3.1 Interaction of player, game and monochord

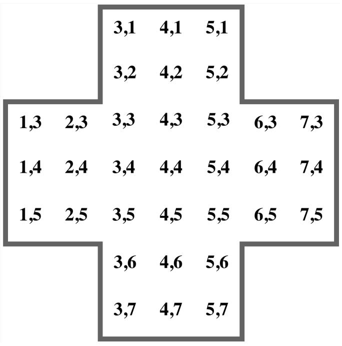
The composition generation occurs in Pure Data which is responsible for the mapping of values to melodic patterns for the monochord and audio manipulations. *Sounds of Solitaire* invites participation following the rules of the game. Every interaction with the board has the potential to create a unique composition within the constraints of the hardware, depending on the way in which the game is played. The real-time musical composition generation begins with the first move of the game. The Arduino has been programmed to report legal moves; if an illegal move is made it recognises this and waits for the move to be rectified or until a new move is made.

4.3.2 Solenoid Melody

The monochord solenoid melody is determined by the coordinates of the peg that is lifted to be repositioned (see Figure 10). Each move generates 3 or 4 different notes depending on the direction of the move. The overall effect is that positive direction moves have an ascending melodic pattern whereas negative direction moves have a descending melodic pattern.

4.3.3 Note Duration

Note duration is determined by the time taken for each move to be completed. The EBow is switched on when a peg is lifted to be repositioned and switched off after the time taken to complete the move has elapsed since the move was completed. The compositional effect of the duration feature is that if the game is played quickly then the notes will have a shorter duration and vice versa.



		3,1	4,1	5,1		
		3,2	4,2	5,2		
1,3	2,3	3,3	4,3	5,3	6,3	7,3
1,4	2,4	3,4	4,4	5,4	6,4	7,4
1,5	2,5	3,5	4,5	5,5	6,5	7,5
		3,6	4,6	5,6		
		3,7	4,7	5,7		

Figure 10. Diagram of peg co-ordinates

4.3.4 Effects Modules

Adding to the sound character of the board, four effects modules were programmed: delay, reverb, bit crushing and cross synthesis. The effects process both the marble run sounds and the monochord; the effects applied depend on the location in which the peg is repositioned. Each effect pertains to one of the four quadrants of the board: the upper quadrant triggers reverb, the right quadrant triggers bit crushing, the bottom quadrant triggers delay and the left quadrant triggers cross synthesis (see Figure 11). The middle peg has no effect, and there are 4 pegs that have two effects, these are pegs of importance when the common solution is played. The effect parameters are controlled by statistics from the game: direction, lift duration, peg count and available moves; these values are scaled appropriately for each effect. The order in which the effects modules are triggered builds structure into the pieces.

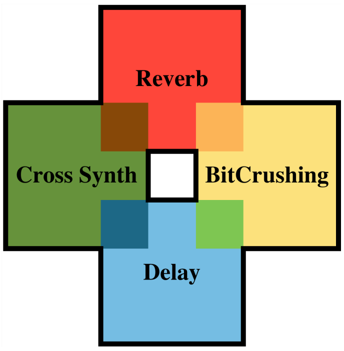


Figure 11. Locations of effect application

4.3.5 Loopers

Two four-track loopers were implemented: one for the marble run and another for the monochord, to establish a sense of coherence in the composition. Loop playback speed is independently controlled on each track, determined by the number of moves available when at rest and the direction of the preceding move, if the direction was negative then the loop is played in reverse and vice versa. The monochord looper records a new track each time a new solenoid is triggered; the loops are played back at slow speeds determined by the number of available moves. The loops are cleared one at a time to create a slowly morphing drone texture.

4.3.6 Structure

The instrumental structure is hard coded to change when specific numbers of pegs are on the board; determining different combinations of marble run, monochord and loopers. The variations in time taken by the player to complete these sections mean that each composition acquires an individual nature.

The final board design is constructed in wood and Perspex, with the electronics contained inside (see Figure 12). A demonstration of *Sounds of Solitaire* can be found at: http://youtu.be/GUkLDZmi9bM

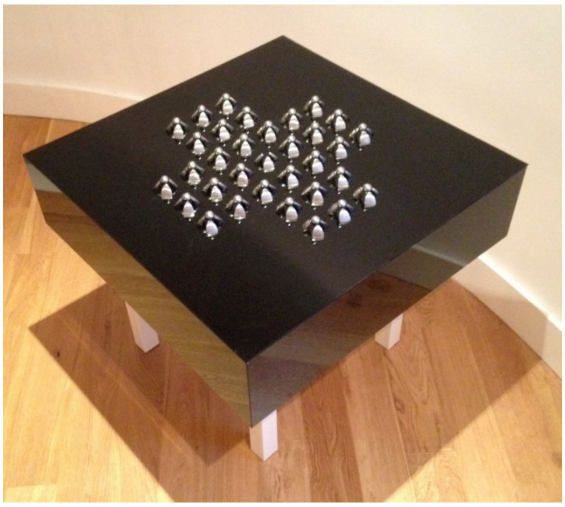


Figure 12. Final board design

Once the hardware is set up, ultimately it would be possible to change the way in which Pure Data responds to the moves. By altering the default patch via SSH connection, the solenoid mappings, effects and structure can be changed and consequently generate radically different pieces. This opens up possibilities for changing contexts.

5. CONCLUSIONS

Sounds of Solitaire is designed with a playful approach to the realisation of an algorithmic, interactive interface, capable of generating multiple runs of music. Due to the correlation between game participation and the material presented for making music in real-time, the participant generates variations in musical output. Through the restricted format of a game, *Sounds of Solitaire* presents a recognisable interface for real-time composition generation.

There may be a possibility of using the Raspberry Pi's networking potential to allow a larger audience to interact with the music generating process via smart phones; for the time being, the project is characterised by its direct physical participatory nature.

6. ACKNOWLEDGEMENTS

Michael Edwards and Peter Nelson (University of Edinburgh)

7. REFERNCES

[1] L. Cross, Reunion: John Cage, Marcel Duchamp, Electronic Music and Chess, *Leonardo Music Journal* Vol. 9, (1999) 35-42.

[2] C. Cox, and D. Warner. *Audio Culture.* Continuum, New York, 2008.

[3] M. Landy, and J. Tinguely. *Joyous Machines*. Tate Publishing, London, UK, 2009.

[4] J. Chadabe. The Performer Is Us, *Contemporary Music Review,* 18(3) (1999), 25-30.

[5] EBow Website : http://www.ebow.com/