

```

SendMessage"UNLOCKGUI"
SendMessage"UNLOCKDEFLAR"
SendMessage"WINDOWDOCUMENTACTIVATE"
SendMessage"UPDATEDEFLAR"
SendMessage"POSITIONSOP"
EndScript
EndObject
TextButton "Play"
Definition
Origin 536,48
Font"topaz" 8;FontName,PointSize
PrintStyleSHADOW 2,3;Style,Pen1, Pen2
TextCobis 1,0,NORMAL;PenA,PenB,DrawMode
Text " Play "
BorderBEVEL 2,1;BorderStyle,MainPen, ExtraPen
HighlightCOMPLEMENT
ButtonFlagsNONE
EndScript
OnRelease
Nop;Activatethesongwindow and playsong
SendMessage"WINDOWDOCUMENTACTIVATE"
SendMessage"PLAY"
EndScript
EndObject
TextButton "Stop"
Definition
Origin 536,72
Font"topaz" 8;FontName,PointSize
PrintStyleSHADOW 2,3;Style,Pen1, Pen2
TextCobis 1,0,NORMAL;PenA,PenB,DrawMode
Text " Stop "
BorderBEVEL 2,1;BorderStyle,MainPen, ExtraPen
HighlightCOMPLEMENT
ButtonFlagsNONE
EndScript
OnRelease
Nop;Stop playing the song
SendMessage"STOP"
EndScript
EndObject
ImageButton "BeatsLeft"
Definition
Origin 110,60
Image"RCF:CarDo251/Brushes/Leftbr"
HighlightCOMPLEMENT
ButtonFlagsNONE
EndScript
OnClick
Nop;Changebeats while pressing button
Nop;Do not allow value below 1
Loop
Let Beats=Max(Beats-1,1)

```

```

Do "ShowTime"
Delay 0,0,10
UntilObjectState("BeatsLeft")=OFF
EndScript
EndObject
ImageButton "BeatsRight"
Definition
Origin 132,60
Image"RCF:CarDo251/Brushes/Rightbr"
HighlightCOMPLEMENT
ButtonFlagsNONE
EndScript
OnClick
Nop;Changebeats while pressing button
Nop;Do not allow value above 16
Loop
Let Beats=Min(Beats+1,16)
Do "ShowTime"
Delay 0,0,10
UntilObjectState("BeatsRight")=OFF
EndScript
EndObject
ImageButton "NoteBeatLeft"
Definition
Origin 110,72
Image"RCF:CarDo251/Brushes/Leftbr"
HighlightCOMPLEMENT
ButtonFlagsNONE
EndScript
OnClick
Nop;Changebeats while pressing button
Nop;Do not allow value below 1
Loop
Let NoteBeat=Max(NoteBeat/2,1)
Do "ShowTime"
Delay 0,0,10
UntilObjectState("NoteBeatLeft")=OFF
EndScript
EndObject
ImageButton "NoteBeatRight"
Definition
Origin 132,72
Image"RCF:CarDo251/Brushes/Rightbr"
HighlightCOMPLEMENT
ButtonFlagsNONE
EndScript
OnClick
Nop;Changebeats while pressing button
Nop;Do not allow value above 64
Loop
Let NoteBeat=Min(NoteBeats*2,64)
Do "ShowTime"
Delay 0,0,10
UntilObjectState("NoteBeatRight")=OFF
EndScript
EndObject
*EndOfCard "IMCard"
*****
*Globalroutine "ShowKey"
Nop;Set font and style
SelfPrintFont "topaz" 8
SetPrintStyleFLAN 2,3
SetPen 1,0
SelfDrawMode,AM2
Nop;Erase current key in rol button
PrintText " " 8,3,28
Nop;Print the new key based on current index
PrintTextKey/Text/Key/Index,8,3,28
*Endofroutine "ShowKey"
*****
*Globalroutine "ShowTime"
Nop;Show the music staff brush
ShowBrush "CappedBrush" 79,61
Nop;Set font and style
SelfPrintFont "topaz" 9
SetPrintStyleBOLDOUTLINE 2,3
SetPen 1,0
SelfDrawMode,AM1
Nop;Print the top and bottom number of time signature
Nop; Note the different position if number is two digits
PrintTextBeats 85-4*(Beats>=10),62
PrintTextNoteBeat 85-4*(NoteBeat>=10),72
*Endofroutine "ShowTime"
*****

```

-AC-

Please Write to:
Randy Finch
c/o Amazing Computing
P.O. Box 2140
Fall River, MA 02722-2140

```

Quit
EndIf
Nop;Get current window width into MessageReturned
SendMessage "GETWINDOWDOCUMENTWIDTH"
Nop;Resize the height of the window to 280
SendMessage "SIZINDOW" MessageReturned "280"
Nop;Assign all keys recognized by DMusic
LetKeyed1="C-MAOR"
LetKeyed2="G-MAOR"
LetKeyed3="D-MAOR"
LetKeyed4="A-MAOR"
LetKeyed5="E-MAOR"
LetKeyed6="B-MAOR"
LetKeyed7="F-SHARP-MAOR"
LetKeyed8="C-SHARP-MAOR"
LetKeyed9="F-MAOR"
LetKeyed10="B-FLAT-MAOR"
LetKeyed11="E-FLAT-MAOR"
LetKeyed12="A-FLAT-MAOR"
LetKeyed13="D-FLAT-MAOR"
LetKeyed14="G-FLAT-MAOR"
LetKeyed15="C-FLAT-MAOR"
LetKeyed16="A-MNOR"
LetKeyed17="E-MNOR"
LetKeyed18="B-MNOR"
LetKeyed19="F-SHARP-MNOR"
LetKeyed20="C-SHARP-MNOR"
LetKeyed21="G-SHARP-MNOR"
LetKeyed22="D-SHARP-MNOR"
LetKeyed23="A-SHARP-MNOR"
LetKeyed24="D-MNOR"
LetKeyed25="G-MNOR"
LetKeyed26="C-MNOR"
LetKeyed27="F-MNOR"
LetKeyed28="B-FLAT-MNOR"
LetKeyed29="E-FLAT-MNOR"
LetKeyed30="A-FLAT-MNOR"
EndScript
AfterAttachmentUsed to be AfterStartup
Nop;Set font and style
SetPrintFont "topaz" 8
SetPrintStyle EMBOSSED 20
SetPen 10
SetDrawMode JAM1
Nop;Print text in window
PrintText "Number of Measures:" 200,23
PrintText "Max Note Duration Variation:" 200,47
PrintText "Max Note Variation:" 200,71
PrintText "Key:" 16,28
PrintText "Timing:" 16,67
Nop;Load the music staff brush for later display
LoadBrush "RCFCarDo251/Brushes/MusicStaff";"CpsdBrush"
Transparent FALSE
Nop;Initialize time signature values and display them
LetBeats=4
LetNoteBeat=4
Do "ShowTime"
Nop;Initialize key signature index and show it
LetKeyedIndex=1
Do "ShowKey"
EndScript
Window "Use Window"
Definition
Origin 0,300
Size 640,100
Title "Grand Notes"
NumberOfCobs 1689632
WindowCobs 0,10;Detail,Block,Background
WindowCobs 0,CSEBUTIONDEFIBUTTONSFRAGEAR
WindowFlags ACTIVATED,FRONT,PUBLIC,"DeluxeMusic"
EndScript
OnCobButton
Quit
EndScript
EndObject
AscButton "Key"
Definition
Origin 56,24
Size 136,16
Border ROLL_2,1;BorderStyle,MainPen,ExtraPen
Highlight COMPLEMENT
ButtonFlags NONE
EndScript
OnRelease
Nop;Calc next index, show it
LetKeyedIndex=KeyedIndex//30+1
Do "ShowKey"
EndScript
EndObject
IntegerField "NumMeasures"
Definition
Origin 432,24
Size 81,8
Justification RIGHT
MaxFieldLength 4

```

```

Limits 1,1000
InitialInteger 4
Border DOUBLEBEVEL_2,1;BorderStyle,MainPen,ExtraPen
EndScript
OnRelease
SetObjectState "MaxNoteDurVar",ON
EndScript
EndObject
IntegerField "MaxNoteDurVar"
Definition
Origin 432,48
Size 81,8
Justification RIGHT
MaxFieldLength 1
Limits 0,6
InitialInteger 1
Border DOUBLEBEVEL_2,1;BorderStyle,MainPen,ExtraPen
EndScript
OnRelease
SetObjectState "MaxNoteVar",ON
EndScript
EndObject
IntegerField "MaxNoteVar"
Definition
Origin 432,72
Size 81,8
Justification RIGHT
MaxFieldLength 2
Limits 1,12
InitialInteger 2
Border DOUBLEBEVEL_2,1;BorderStyle,MainPen,ExtraPen
EndScript
OnRelease
SetObjectState "NumMeasures",ON
EndScript
EndObject
TextButton "Generate"
Definition
Origin 536,24
Font "topaz" 8;FontName,PointSize
PrintStyle SHADOW_2,3;Style,Pen1,Pen2
TextCobs 1,0,NORMAL;PenA,PenB,DrawMode
Text "Generate"
Border BEVEL_2,1;BorderStyle,MainPen,ExtraPen
Highlight COMPLEMENT
ButtonFlags NONE
EndScript
OnRelease
Nop;Clear current score and back display
SendMessage "CLEARFORCE"
SendMessage "LOOKIDisplay"
SendMessage "LOOKGUI"
Nop;Set key and time signatures
SendMessage "SEIKEYSIGNATURE" KeyedIndex "00"
SendMessage "SEITIMESIGNATURE" Beats "NoteBeat" 0
Nop;Get user defined values from fields
LetTotalBeats=Beats+IntegerFrom("NumMeasures")
LetMNV=IntegerFrom("MaxNoteVar")
LetMNDV=IntegerFrom("MaxNoteDurVar")
Nop;Initialize variables for loop
LetNumBeats=0
LetNumBeatsMeasure=0
LetLine=0
Nop;Loop until all measures filled
While NumBeats<TotalBeats
Nop;Calc note line based on random variation
LetLineVar=Random(-MNMV)
LetLine=Line+LineVar
Nop;Calc note beat based on random variation
LetCurNoteBeatVar=Random(-MNDV)
LetCurNoteBeat=Integer(NoteBeats*(2-CurNoteBeatVar))
Nop;Make sure note beat is within acceptable range 1-64
LetCurNoteBeat=Min(64,Max(1,CurNoteBeat))
Nop;Make sure notes near end of measure do not exceed
Nop;maximum allowable
While (NumBeatsMeasure+CurNoteBeat)>Beats
LetCurNoteBeat=CurNoteBeat*2
EndLoop
Nop;insert the note and go to next position
SendMessage "INSERTTHEMNOTE DURATION" CurNoteBeat LINE
Line
SendMessage "NEXTNOTE"
Nop;Update the number of current beats in the song and
Nop;the current measure
LetNumBeats=NumBeats+CurNoteBeat
LetNumBeatsMeasure=NumBeatsMeasure+CurNoteBeat
Nop;Go to next measure if current one full
If NumBeatsMeasure=Beats
SendMessage "NEXT MEASURE"
LetNumBeatsMeasure=0
EndIf
EndLoop
Nop;Unblock the display, activate song window,
Nop;update the display, and go to the first measure

```

adding this value to Line. Remember that the user specified the value of MNDV and therefore has control over how many lines can separate consecutive notes in the score. A similar method is used for determining the duration of the next note. A random integer between -MNDV and +MNDV is selected. Then the value of NoteBeat (set in the time signature) is multiplied by two raised to the power of the random integer. This will scale the duration up or down by some multiple of two. Once this is calculated, the Min and Max functions are used to make sure the new note duration is not less than one (a whole note) or greater than 64 (a sixty-fourth note), the minimum and maximum allowable by Deluxe Music. Next, the current note duration is adjusted, if necessary, to prevent the number of beats in the current measure from exceeding the value of Beats, which was set in the time signature.

The next task of the loop is to insert the note into the Deluxe Music score. An INSERTITEM NOTE ARExx command is issued along with the DURATION and LINE information. Next, the command NEXT NOTE tells Deluxe Music to prepare to receive the next note in the score.

After inserting the note, the current numbers of beats in the measure and in the score are updated. If the current measure has been filled with the appropriate number of beats, then Deluxe Music is sent the NEXT MEASURE command so that it will prepare a new measure for the upcoming notes. The number of beats in the current measure, NumBeatsMeasure, is then reset to zero.

This ends the loop. As stated, it will continue to execute until the total number of beats for the score has been met. When the loop is exited, five more Arexx commands are sent to Deluxe Music. UNLOCKGUI allows the user to directly interact with the Deluxe Music interface. UNLOCKDISPLAY frees Deluxe Music to display the score that has been created by RandomNotes. WINDOW DOCUMENT ACTIVATE activates the score window. UPDATEDISPLAY causes the score to be displayed. Finally, POSITION SOF positions Deluxe Music's cursor at the start of the score.

Once the score has been generated, it is time to listen to it. By pressing the Play button in the RandomNotes window, its OnRelease script will execute. This script sends two commands to Deluxe Music: WINDOW DOCUMENT ACTIVATE and PLAY. The former makes sure that the score window is active; the latter directs Deluxe Music to play the score. Once the score finishes playing, it will repeat. To stop the score, press the Stop button. A STOP command will be sent to Deluxe Music.

Enhancing the Scores

Since RandomNotes only produces a user constrained random score, most of the scores will not sound good. However, since they are so simple to create, many scores can be created in a short amount of time. Occasionally, one will sound fairly good. At this point, the user can interact directly with the Deluxe Music interface to modify the fairly good score to make it a very good or even excellent score, depending on the abilities of the user. In other words, RandomNotes should be viewed as an idea generator. It



Figure 3. Window Editor Requester in CanDo 2.51

produces a score that helps plant a tune into the mind of the musician. The musician can then enhance the score to fit the idea.

If you are familiar with music theory, you might want to modify the algorithms in RandomNotes and create a program named MozartNotes, or BachNotes, or even MadonnaNotes. In other words, the program could take user defined parameters and create an original score that is reminiscent of the tunes written by or performed by a particular musical artist. If anyone creates such a program, I would be very interested in seeing it.

Listing 1. RandomNotes Deck

```
*****
*Deck `RandomNotes`
*Time 21:03:50
*Date 05/15/94
*****

*****
*Card(s) indeck.
*Card `DMCard`
*****
*1 Card(s), 1 were printed.
*****

*****
*Natural order of Cards
*Card `DMCard`
*****

*****
*Global Routine(s) indeck.
*Routine `ShowKey`
*Routine `ShowTime`
*****
*2 Global routines(s), 2 were printed.
*****

*****
*Card `DMCard`
Before Attachment ; used to be OnStartup
Nop ; Speak to DMusic - Load it if not in memory
SpeakTo `DMUSIC`
IfError
Dos `run>nil: <nil: SYS:DeluxeMusic/DMusic`
Let Timeout=50
Loop
Delay 0,0,10
SpeakTo `DMUSIC`
IfError
Let FoundPort=FALSE
Else
Let FoundPort=TRUE
Nop ; DMusic starts communicating
Nop ; via ARExx before its
Nop ; screen is ready for RandomNotes
Nop ; Delay 5 seconds
Delay 0,5,0
EndIf
Let Timeout=Timeout-1
Until FoundPort OR (Timeout=0)
Else
Let FoundPort=TRUE
EndIf
If NOT FoundPort
Echo `Can't groove with a DMusic, man!`
```

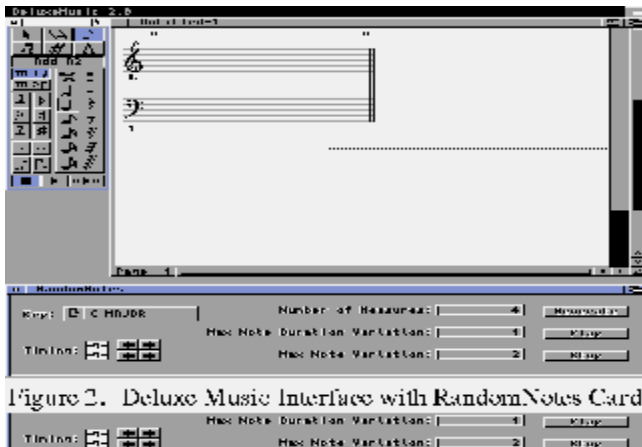


Figure 2. Deluxe Music Interface with RandomNotes Card

command in the system variable, MessageReturned. This value is used as part of another ARexx command, SIZEWINDOW. Here, Deluxe Music is directed to re-size the height of its current window (the document or score window) to 280. The width remains the same since the current width obtained from the GETATTR command is passed back in the SIZEWINDOW command. After the window is re-sized, there will be space at the bottom of the screen for the RandomNotes window. The difference can be seen by comparing Figures 1 and 2. Finally, the 30 elements of the KeyText array are defined. The text used for the array elements is the same as that needed in the Deluxe Music ARexx commands as you will see later.

The AfterAttachment script writes all the necessary labels on the card. It then loads a brush named MusicStaff.br into a buffer named ClippedBrush. This brush is used by the routine ShowTime to display the time signature. Next, the initial time signature values are assigned and displayed. Finally, the initial key signature index is set and its text is displayed.

The Key Signature Button

This button is an area button with a rollo border, which indicates that a press of the button will change the selection text within it. The initial text on the button is C-MAJOR, which is the value of KeyText[1]. Each time the button is pressed, its OnRelease script executes. This script simply increments the value of KeyTextIndex by one and executes the global routine, ShowKey. A modulo operator is used to ensure that the value of KeyTextIndex never exceeds 30, which is the number of elements in the array. The ShowKey routine erases the text in the rollo button and replaces it with the text in the current element of the KeyText array.

The Time Signature Buttons

The time signature consists of two numbers: the beats per measure (top number) and the note to receive a beat (bottom number). There is a pair of left and right arrow buttons to the right of each number for adjusting each independently. The buttons for adjusting the number of beats per measure are named BeatsLeft and BeatsRight. The buttons for adjusting the note to receive a beat are named NoteBeatLeft and NoteBeatRight. Each button has an OnClick event script.

The algorithm in each of these scripts will be executed when the user clicks on the respective buttons and will

continue to execute until the mouse button is released. The BeatsLeft and BeatsRight buttons simply decrement and increment by one the current value of Beats, which is the top number in the time signature, ensuring that its value never drops below one or exceeds 16. The NoteBeatLeft and NoteBeatRight buttons simply decrement and increment by multiples of two the current value of NoteBeat, which is the bottom number in the time signature, ensuring that its value never drops below one or exceeds 64.

The script for each of the four time signature buttons executes a global routine named ShowTime each time the Beats or NoteBeat variable is changed. This routine re-displays the ClippedBrush brush containing the small musical staff, overwriting the current display. Next, the Beats and NoteBeat values are printed on top of the brush. A simple algorithm is used for the printing of each number so that it will be centered in the musical staff brush regardless of whether the number has one or two digits. Each of the scripts for the four time signature buttons also has a short delay built in so the numbers do not change faster than the user's mouse button finger can respond.

The Generate, Play, and Stop Buttons

The Generate button's OnRelease script is a hotbed of activity. It sends many messages to Deluxe Music and has a lot of mathematical formulas that generate the random musical score. Let's follow it through in detail.

First, five commands are sent to Deluxe Music: CLEAR FORCE, LOCKDISPLAY, LOCKGUI, SETKEYSIGNATURE, and SETTIMESIGNATURE. The CLEAR command tells Deluxe Music to clear the current document; the FORCE option specifies that the user should not be prompted for a verification of this action. LOCKDISPLAY prevents the document window from updating each time a new note is added to the score. LOCKGUI prevents the user from directly controlling the Deluxe Music interface. As you probably guessed, SETKEYSIGNATURE and SETTIMESIGNATURE are used to set the key signature and the time signature of the current score.

Next, the total number of Beats in the score, TotalBeats, is calculated by multiplying the number of beats in a measure times the number of measures in the score. Also, the user specified limitations on the variation in a note's pitch, MNV, and a note's duration, MNDV, are extracted from the appropriate integer fields.

In preparation for the note insertion loop, three variables are initialized. NumBeats, which will keep up with the total number of beats in the score, is set to zero. NumBeatsMeasure, which will keep up with the total number of beats in each measure, is set to zero. Finally, Line, which will keep up with the line on the musical staff where the next note will be inserted, is set to zero. A line number of zero is the middle line of the staff. Positive line numbers are above this line and negative numbers are below this line.

Now it is time to enter the note insertion loop. As long as the actual number of beats in the score is less than the total number of beats to be added to the score, this loop will execute.

The first task of the loop is to determine the line on which the next note will appear. This is accomplished by selecting a random integer between -MNV and +MNV and

Front Ending: Deluxe Music

by Randy Finch

Beginning with version 2.0 of AmigaDOS, a new feature known as public screens became available. This feature allows any program to declare its custom screen as a public screen on which other programs can create windows. Many of the newer programs for the Amiga, including Deluxe Music 2.0 (Figure 1), register their screens as public with the operating system, thus allowing other programs to open windows on them. Of course, the ability to open a window on another program's screen is not very useful unless there is some way for the programs to communicate with each other. This is where ARexx comes in. It is the standard way for programs to communicate on the Amiga. In this article I will discuss a CanDo 2.51 program I wrote named RandomNotes. It makes use of CanDo's ability to open windows on public screens and its ARexx communications capabilities. The RandomNotes program is shown in Listing 1.

RandomNotes Features

RandomNotes opens a window on Deluxe Music's screen (Figure 2) and then sends ARexx commands to Deluxe Music to create a random tune based on user input. This input includes selecting a key signature, a time signature, the number of measures in the tune, and a couple of parameters that confine the randomness of the notes. When the Generate button is pressed, the tune is created. The tune can be played by pressing the Play button and stopped by pressing the Stop button. The key signature has an initial value of C-MAJOR and is displayed in the rollo button in the upper-left corner of the RandomNotes window. The key can be changed by pressing the rollo button. Each time it is pressed, the key changes. There are 30 possible key signatures recognized by Deluxe Music.

The time signature has a default value of 4/4 which means that there are four beats per measure (top number) with a quarter note receiving one beat (bottom number). If this were changed to 3/8, then there would be three beats to a measure with an eighth note receiving one beat. The beats per measure and note to receive a beat can be changed by pressing the appropriate arrow buttons to the right of the time signature display. The upper number can vary from 1 to 16 in steps of 1 while the lower number can vary from 1 to 64 in multiples of 2 (e.g. 1, 2, 4, 8, ...).

The number of measures has a default value of four but can be changed by clicking the integer input field and typing a new value. This value controls the total number of measures in the random tune.

The two parameters that confine the randomness of the notes in the tune are the maximum note duration variation and the maximum note variation. The former controls the amount of variation in note duration from the base beat value in the time signature (the bottom number) and has a default value of one. The latter controls the maximum variation in the note pitch from one note to the next and has

a default value of two. Each of these values can be changed by typing a new number into the appropriate field.

When the Generate button is pressed, RandomNotes sends Deluxe Music the appropriate ARexx commands to clear the current tune, set the key signature and the time signature, and insert the randomly generated notes. Once the tune is generated, it can be played by pressing the Play button and stopped by pressing the Stop button.

The DMCARD Card

The RandomNotes deck consists of only one card named DMCARD. As shown in the card's Window definition section (Listing 1), it opens on the DeluxeMusic public screen (Figure 2). The public screen name can be selected from the Window Editor requester (Figure 3). When the button to the right of the Public Screen check box is pressed, a list of available public screens will be presented. Of course, Deluxe Music must be loaded for its screen to appear in the list.

The DMCARD card has a BeforeAttachment and an AfterAttachment script. The BeforeAttachment script first attempts to speak to the DMUSIC ARexx port. If an error occurs, then Deluxe Music is not currently running. The script then attempts to execute Deluxe Music and continues to look for the DMUSIC ARexx port. If it is not found, RandomNotes prints a message and exits. If the port is found, the script pauses for five seconds to allow time for Deluxe Music's screen to be set up and registered as public with the operating system. If RandomNotes attaches its card before Deluxe Music's screen is set up, it will open on the Workbench screen instead. The five second delay may need to be increased on slower systems.

After the ARexx connection is established, a Deluxe Music ARexx command, GETATTR, is sent to Deluxe Music. This command tells Deluxe Music that some information is needed, in this case the width of the document window (the one with the musical score). Deluxe Music then passes the requested information back to RandomNotes. CanDo stores return values from an ARexx

