```
'* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
'*                                                         *
'*              WITTGENSTEIN FRAGMENTS V1.1                *
'*                       8/27/92                           *
'*                                                         *
'*  a chamber concerto for computer and synthesizers       *
'*                                                         *
'*           Copyright (C) 1992 by David Snow              *
'*                                                         *
'* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

defint a-z
library "gemdos","xbios","gemaes"
rem $include I:\witgnstn.BH
rem $option l20,y+,v+

CONST piano=0,piano_lh=1,piano_rh=2,vibes=3,guitar=4,drums=5
CONST bells=6,bass=7,flute=8,oboe=9,bassoon=10,violin_arco=11
CONST violin_pizz=12,cello_arco=13,cello_pizz=14,timpani_hard=15
CONST timpani_soft=16,xylophone=17,sax=18,tamtam=19,effects=20
CONST ride_cymbal=76,midiport=3,note_on=144,note_off=128,aftertouch=208

' THE PROGRAM
gosub initialize
gosub performance
gosub reset_midi
system

' THE SUBROUTINES
performance:
        gosub buffer_setup
        do
                gosub intro
                gosub part_1
                gosub part_2
                gosub part_3
                gosub part_4
                gosub part_5
                gosub cadenza
                gosub part_6
                gosub coda
                gosub save_midi_file
        loop while restart_flag
        return

intro:
```

```
	start_time!=timer

	large_ensemble_flag=1
	gosub pulsed_unison_staccato
	large_ensemble_flag=0

	start_part_1!=timer+12
	cadenza_flag=1
	do while timer<start_part_1!
		gosub ensemble_sustain_1
	wend
	cadenza_flag=0
	return

part_1:
	large_ensemble_flag=1
	gosub pulsed_unison_staccato
	large_ensemble_flag=0

	start_part_2!=timer+30+int(rnd*20)
	do while timer<start_part_2!
		gesture=int(rnd*5)
		select case gesture
		=0
			gosub ensemble_sustain_1
		=1
			gosub unison_staccato
		=2
			gosub unison_melody
		=3
			gosub pulsed_unison_staccato
		=4
			gosub pulsed_unison_melody
		end select
	loop
	return

part_2:
	large_ensemble_flag=1
	gosub pulsed_unison_staccato
	large_ensemble_flag=0

	start_part_3!=timer+30+int(rnd*20)
	do while timer<start_part_3!
		gesture=int(rnd*3)
		select case gesture
```

```
                =0
                        gosub ensemble_sustain_2
                =1
                        gosub piano_jazz
                =2
                        gosub free_solo_with_chords
                end select
        loop
        return

part_3:
        large_ensemble_flag=1
        gosub pulsed_unison_staccato
        large_ensemble_flag=0

        start_part_4!=timer+30+int(rnd*20)
        do while timer<start_part_4!
                gesture=int(rnd*10)
                select case gesture
                =0
                        gosub ensemble_sustain_1
                =1
                        gosub unison_staccato
                =2
                        gosub unison_melody
                =3
                        gosub solo_staccato
                =4
                        gosub pulsed_staccato
                =5
                        gosub pulsed_unison_staccato
                =6
                        gosub pulsed_unison_melody
                =7
                        gosub pulsed_staccato_repeat
                =8
                        gosub drum_solo
                =9
                        gosub free_solo_with_jazz
                end select
        loop
        return

part_4:
        large_ensemble_flag=1
        gosub pulsed_unison_staccato
```

```
        large_ensemble_flag=0

        start_part_4a!=timer+20+int(rnd*20)
        do while timer<start_part_4a!
                gesture=int(rnd*6)
                select case gesture
                =0
                        gosub free_solo_with_jazz
                =1
                        gosub blues_solo_with_jazz
                =2
                        gosub drum_solo
                =3
                        gosub fast_solo_with_jazz
                =4
                        gosub pulsed_unison_staccato
                =5
                        gosub pulsed_staccato_repeat
                end select
        loop

        start_part_5!=timer+30+int(rnd*20)
        do while timer<start_part_5!
                gosub pulsed_unison_staccato
        loop

        large_ensemble_flag=1
        gosub pulsed_staccato_repeat
        gosub pulsed_staccato_repeat
        gosub pulsed_staccato_repeat
        large_ensemble_flag=0
        return

part_5:
        large_ensemble_flag=1
        gosub pulsed_unison_staccato
        large_ensemble_flag=0

        start_cadenza!=timer+20+int(rnd*20)
        do while timer<start_cadenza!
                gesture=int(rnd*10)
                select case gesture
                =0
                        gosub ensemble_sustain_1
                =1
                        gosub unison_staccato
```

```
                =2
                        gosub unison_melody
                =3
                        gosub pulsed_unison_staccato
                =4
                        gosub pulsed_unison_melody
                =5
                        gosub pulsed_staccato_repeat
                =6
                        gosub free_solo_with_jazz
                =7
                        gosub blues_solo_with_jazz
                =8
                        gosub drum_solo
                =9
                        gosub fast_solo_with_jazz
                end select
        loop
        return

cadenza:
        large_ensemble_flag=1
        gosub pulsed_unison_staccato
        large_ensemble_flag=0

        cadenza_flag=1
        start_part_6!=timer+30+int(rnd*20)
        do while timer<start_part_6!
                gosub ensemble_sustain_2
        loop
        cadenza_flag=0
        return

part_6:
        start_coda!=timer+20+int(rnd*20)
        do while timer<start_coda!
                gesture=int(rnd*3)
                select case gesture
                =0
                        gosub ensemble_sustain_2
                =1
                        gosub piano_jazz
                =2
                        gosub free_solo_with_chords
                end select
        loop
```

```
        return

coda:
        end_coda!=timer+15+int(rnd*15)
        do while timer<end_coda!
                gosub pulsed_unison_staccato
        loop

        large_ensemble_flag=1
        gosub pulsed_staccato_repeat
        gosub pulsed_staccato_repeat
        gosub pulsed_staccato_repeat
        large_ensemble_flag=0

        total_seconds=cint(timer-start_time!)
        minutes=int(total_seconds/60)
        minutes$=str$(minutes)+" minutes "
        seconds$=str$(total_seconds-(60*minutes))+" seconds"
        alert$="[3][ Performance duration:|"+minutes$+seconds$+"][Restart|
Quit ]"
        mouse 0
        if FNform_alert(2,alert$)=1 then
                restart_flag=1
        else
                restart_flag=0
        end if
        mouse -1
        return

check_key:
        if inp(-2)<>0 then
                junk=inp(2)
                mouse 0
                alert$="[3][Abort this performance?][Yes|No]"
                if FNform_alert(1,alert$)=1 then
                        system
                end if
                mouse -1
        end if
        return

ensemble_sustain_1:
        if cadenza_flag then
                gosub piano_solo
        else
                ensemble=int(rnd*6)
```

```
                select case ensemble
                =0
                        gosub mixed_ensemble_1
                =1
                        gosub mixed_ensemble_2
                =2
                        gosub percussion_ensemble
                =3
                        gosub string_ensemble
                =4
                        gosub piano_solo
                =5
                        gosub percussion_solo
                end select
        end if

        for iteration=1 to int(rnd*32)+8
                gosub check_key
                instrument=active_instrument(int(rnd*active_voices))
                velocity=int(rnd*64)+64

                for index=0 to int(rnd*(polyphony(instrument)+1))
                        if instrument=drums then

note_buffer(instrument,index)=drum_note(int(rnd*9))
                        else

note_buffer(instrument,index)=int(rnd*range(instrument))+low_note(instrumen
t)
                        end if
                        out midiport,note_on+midi_channel(instrument)
                        out midiport,note_buffer(instrument,index)
                        out midiport,velocity
                        gosub delta_conversion
                        pokeb
smf_buffer&+smf_index&,note_on+midi_channel(instrument)
                        incr smf_index&
                        pokeb
smf_buffer&+smf_index&,note_buffer(instrument,index)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,velocity
                        incr smf_index&
                next

                timespan!=1/(int(rnd*12)+1)
                delay!=timer+timespan!
```

```
                delta_time&=delta_time&+(timespan!*steps_per_second!)
                while delay!>timer: wend


                for index=0 to polyphony(instrument)
                        if note_buffer(instrument,index)>-1
                                out
midiport,note_off+midi_channel(instrument)
                                out midiport,note_buffer(instrument,index)
                                out midiport,0
                                gosub delta_conversion
                                pokeb
smf_buffer&+smf_index&,note_off+midi_channel(instrument)
                                incr smf_index&
                                pokeb
smf_buffer&+smf_index&,note_buffer(instrument,index)
                                incr smf_index&
                                pokeb smf_buffer&+smf_index&,velocity
                                incr smf_index&
                                note_buffer(instrument,index)=-1
                        end if
                next
        next
        return

ensemble_sustain_2:
        if cadenza_flag then
                gosub piano_solo
        else
                ensemble=int(rnd*6)
                select case ensemble
                =0
                        gosub mixed_ensemble_1
                =1
                        gosub mixed_ensemble_2
                =2
                        gosub percussion_ensemble
                =3
                        gosub string_ensemble
                =4
                        gosub piano_solo
                =5
                        gosub percussion_solo
                end select
        end if
```

```
        for iteration=1 to int(rnd*16)+1
                gosub check_key
                instrument=active_instrument(int(rnd*active_voices))
                velocity=int(rnd*64)+32

                for index=0 to polyphony(instrument)
                        if note_buffer(instrument,index)>-1
                                out
midiport,note_off+midi_channel(instrument)
                                out midiport,note_buffer(instrument,index)
                                out midiport,0
                                gosub delta_conversion
                                pokeb
smf_buffer&+smf_index&,note_off+midi_channel(instrument)
                                incr smf_index&
                                pokeb
smf_buffer&+smf_index&,note_buffer(instrument,index)
                                incr smf_index&
                                pokeb smf_buffer&+smf_index&,0
                                incr smf_index&
                                note_buffer(instrument,index)=-1
                        end if
                next

                for index=0 to int(rnd*(polyphony(instrument)+1))
                        if instrument=drums then

note_buffer(instrument,index)=drum_note(int(rnd*9))
                        else

note_buffer(instrument,index)=int(rnd*range(instrument))+low_note(instrumen
t)
                        end if
                        out midiport,note_on+midi_channel(instrument)
                        out midiport,note_buffer(instrument,index)
                        out midiport,velocity
                        gosub delta_conversion
                        pokeb
smf_buffer&+smf_index&,note_on+midi_channel(instrument)
                        incr smf_index&
                        pokeb
smf_buffer&+smf_index&,note_buffer(instrument,index)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,velocity
                        incr smf_index&
                next
```

```
                timespan!=2/(int(rnd*6)+1)
                delay!=timer+timespan!
                delta_time&=delta_time&+(timespan!*steps_per_second!)
                while delay!>timer: wend
        next

        for instrument=0 to 20
                for index=0 to polyphony(instrument)
                        if note_buffer(instrument,index)>-1 then
                                out
midiport,note_off+midi_channel(instrument)
                                out midiport,note_buffer(instrument,index)
                                out midiport,0
                                gosub delta_conversion
                                pokeb
smf_buffer&+smf_index&,note_off+midi_channel(instrument)
                                incr smf_index&
                                pokeb
smf_buffer&+smf_index&,note_buffer(instrument,index)
                                incr smf_index&
                                pokeb smf_buffer&+smf_index&,0
                                incr smf_index&
                                note_buffer(instrument,index)=-1
                        end if
                next
        next
        return

solo_staccato:
        ensemble=int(rnd*6)
        select case ensemble
        =0
                gosub mixed_ensemble_1
        =1
                gosub mixed_ensemble_2
        =2
                gosub percussion_ensemble
        =3
                gosub string_ensemble
        =4
                gosub piano_solo
        =5
                gosub percussion_solo
        end select
```

```
        for iteration=1 to int(rnd*8)
                gosub check_key
                instrument=active_instrument(int(rnd*active_voices))
                velocity=int(rnd*32)+96

                for index=0 to int(rnd*(polyphony(instrument)+1))
                        if instrument=drums then

note_buffer(instrument,index)=drum_note(int(rnd*9))
                        else

note_buffer(instrument,index)=int(rnd*range(instrument))+low_note(instrumen
t)
                        end if
                        out midiport,note_on+midi_channel(instrument)
                        out midiport,note_buffer(instrument,index)
                        out midiport,velocity
                        gosub delta_conversion
                        pokeb
smf_buffer&+smf_index&,note_on+midi_channel(instrument)
                        incr smf_index&
                        pokeb
smf_buffer&+smf_index&,note_buffer(instrument,index)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,velocity
                        incr smf_index&
                next

                timespan!=.1
                delay!=timer+timespan!
                delta_time&=delta_time&+(timespan!*steps_per_second!)
                while delay!>timer: wend

                for index=0 to polyphony(instrument)
                        if note_buffer(instrument,index)>-1 then
                                out
midiport,note_off+midi_channel(instrument)
                                out midiport,note_buffer(instrument,index)
                                out midiport,0
                                gosub delta_conversion
                                pokeb
smf_buffer&+smf_index&,note_off+midi_channel(instrument)
                                incr smf_index&
                                pokeb
smf_buffer&+smf_index&,note_buffer(instrument,index)
                                incr smf_index&
```

```
                                pokeb smf_buffer&+smf_index&,0
                                incr smf_index&
                                note_buffer(instrument,index)=-1
                        end if
                next

                timespan!=1/(int(rnd*8)+1)
                delay!=timer+timespan!
                delta_time&=delta_time&+(timespan!*steps_per_second!)
                while delay!>timer: wend
        next
        return

unison_staccato:
        ensemble=int(rnd*6)
        select case ensemble
        =0
                gosub mixed_ensemble_1
        =1
                gosub mixed_ensemble_2
        =2
                gosub percussion_ensemble
        =3
                gosub string_ensemble
        =4
                gosub piano_solo
        =5
                gosub percussion_solo
        end select

        for iteration=1 to int(rnd*10)+1
                gosub check_key

                timespan!=1/(int(rnd*15)+2)
                delay!=timer+timespan!
                delta_time&=delta_time&+(timespan!*steps_per_second!)
                while delay!>timer: wend


                for index=0 to active_voices-1
                        if active_instrument(index)=drums then

note_buffer(active_instrument(index),0)=drum_note(int(rnd*9))
                        else

note_buffer(active_instrument(index),0)=int(rnd*range(active_instrument(ind
```

```
ex)))+low_note(active_instrument(index))
                        end if
                next

                for index=0 to active_voices-1
                        out
midiport,note_on+midi_channel(active_instrument(index))
                        out
midiport,note_buffer(active_instrument(index),0)
                        out midiport,127
                        gosub delta_conversion
                        pokeb
smf_buffer&+smf_index&,note_on+midi_channel(active_instrument(index))
                        incr smf_index&
                        pokeb
smf_buffer&+smf_index&,note_buffer(active_instrument(index),0)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,127
                        incr smf_index&
                next

                timespan!=.1
                delay!=timer+timespan!
                delta_time&=delta_time&+(timespan!*steps_per_second!)
                while delay!>timer: wend

                for index=0 to active_voices-1
                        out
midiport,note_off+midi_channel(active_instrument(index))
                        out
midiport,note_buffer(active_instrument(index),0)
                        out midiport,0
                        gosub delta_conversion
                        pokeb
smf_buffer&+smf_index&,note_off+midi_channel(active_instrument(index))
                        incr smf_index&
                        pokeb
smf_buffer&+smf_index&,note_buffer(active_instrument(index),0)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,0
                        incr smf_index&
                next
        next
        return

unison_melody:
```

```
        ensemble=int(rnd*6)
        select case ensemble
        =0
                gosub mixed_ensemble_1
        =1
                gosub mixed_ensemble_2
        =2
                gosub percussion_ensemble
        =3
                gosub string_ensemble
        =4
                gosub piano_solo
        =5
                gosub percussion_solo
        end select

        for iteration=1 to int(rnd*13)+1
                gosub check_key
                pitch=int(rnd*range(0))+low_note(0)
                drum_hit=drum_note(int(rnd*9))
                velocity=int(rnd*64)+64
                for index=0 to active_voices-1
                        if pitch>=low_note(active_instrument(index)) and
pitch<=high_note(active_instrument(index)) then
                                out
midiport,note_on+midi_channel(active_instrument(index))
                                if active_instrument(index)=drums then
                                        out midiport,drum_hit
                                else
                                        out midiport,pitch
                                end if
                                out midiport,velocity
                                gosub delta_conversion

                                pokeb
smf_buffer&+smf_index&,note_on+midi_channel(active_instrument(index))
                                incr smf_index&

                                if active_instrument(index)=drums then
                                        pokeb
smf_buffer&+smf_index&,drum_hit
                                else
                                        pokeb smf_buffer&+smf_index&,pitch
                                end if
                                incr smf_index&
```

```
                                pokeb smf_buffer&+smf_index&,velocity
                                incr smf_index&
                        end if
                next

                timespan!=1/(int(rnd*7)+2)
                delay!=timer+timespan!
                delta_time&=delta_time&+(timespan!*steps_per_second!)
                while delay!>timer: wend

                for index=0 to active_voices-1
                        out
midiport,note_off+midi_channel(active_instrument(index))
                        if active_instrument(index)=drums then
                                out midiport,drum_hit
                        else
                                out midiport,pitch
                        end if
                        out midiport,0
                        gosub delta_conversion

                        pokeb
smf_buffer&+smf_index&,note_off+midi_channel(active_instrument(index))
                        incr smf_index&

                        if active_instrument(index)=drums then
                                pokeb smf_buffer&+smf_index&,drum_hit
                        else
                                pokeb smf_buffer&+smf_index&,pitch
                        end if
                        incr smf_index&

                        pokeb smf_buffer&+smf_index&,0
                        incr smf_index&
                next
        next
        return

pulsed_staccato:
        ensemble=int(rnd*6)
        select case ensemble
        =0
                gosub mixed_ensemble_1
        =1
                gosub mixed_ensemble_2
        =2
```

```
                gosub percussion_ensemble
        =3
                gosub string_ensemble
        =4
                gosub piano_solo
        =5
                gosub percussion_solo
        end select

        duration!=((rnd*4)/100)+.055
        for iteration=1 to int(rnd*16)
                gosub check_key
                instrument=active_instrument(int(rnd*active_voices))
                if instrument=drums then
                        pitch=drum_note(int(rnd*9))
                else

pitch=int(rnd*range(instrument))+low_note(instrument)
                end if
                velocity=int(rnd*32)+96

                out midiport,note_on+midi_channel(instrument)
                out midiport,pitch
                out midiport,velocity
                gosub delta_conversion
                pokeb
smf_buffer&+smf_index&,note_on+midi_channel(instrument)
                incr smf_index&
                pokeb smf_buffer&+smf_index&,pitch
                incr smf_index&
                pokeb smf_buffer&+smf_index&,velocity
                incr smf_index&

                timespan!=duration!
                delay!=timer+timespan!
                delta_time&=delta_time&+(timespan!*steps_per_second!)
                while delay!>timer: wend

                out midiport,note_off+midi_channel(instrument)
                out midiport,pitch
                out midiport,0
                gosub delta_conversion
                pokeb
smf_buffer&+smf_index&,note_off+midi_channel(instrument)
                incr smf_index&
                pokeb smf_buffer&+smf_index&,pitch
```

```
                incr smf_index&
                pokeb smf_buffer&+smf_index&,0
                incr smf_index&

                timespan!=duration!
                delay!=timer+timespan!
                delta_time&=delta_time&+(timespan!*steps_per_second!)
                while delay!>timer: wend
        next
        return

pulsed_unison_staccato:
        if large_ensemble_flag then
                gosub large_ensemble
        else
                ensemble=int(rnd*6)
                select case ensemble
                =0
                        gosub mixed_ensemble_1
                =1
                        gosub mixed_ensemble_2
                =2
                        gosub percussion_ensemble
                =3
                        gosub string_ensemble
                =4
                        gosub piano_solo
                =5
                        gosub percussion_solo
                end select
        end if

        for iteration=1 to int(rnd*10)+1
                gosub check_key

                timespan!=.06
                delay!=timer+timespan!
                delta_time&=delta_time&+(timespan!*steps_per_second!)
                while delay!>timer: wend


                for index=0 to active_voices-1
                        if active_instrument(index)=drums then

note_buffer(active_instrument(index),0)=drum_note(int(rnd*9))
                        else
```

```
note_buffer(active_instrument(index),0)=int(rnd*range(active_instrument(ind
ex)))+low_note(active_instrument(index))
                        end if
                next

                for index=0 to active_voices-1
                        out
midiport,note_on+midi_channel(active_instrument(index))
                        out
midiport,note_buffer(active_instrument(index),0)
                        out midiport,127
                        gosub delta_conversion
                        pokeb
smf_buffer&+smf_index&,note_on+midi_channel(active_instrument(index))
                        incr smf_index&
                        pokeb
smf_buffer&+smf_index&,note_buffer(active_instrument(index),0)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,127
                        incr smf_index&
                next

                timespan!=.06
                delay!=timer+timespan!
                delta_time&=delta_time&+(timespan!*steps_per_second!)
                while delay!>timer: wend
                for index=0 to active_voices-1
                        out
midiport,note_off+midi_channel(active_instrument(index))
                        out
midiport,note_buffer(active_instrument(index),0)
                        out midiport,0
                        gosub delta_conversion
                        pokeb
smf_buffer&+smf_index&,note_off+midi_channel(active_instrument(index))
                        incr smf_index&
                        pokeb
smf_buffer&+smf_index&,note_buffer(active_instrument(index),0)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,0
                        incr smf_index&
                next
        next
        return
```

```
pulsed_unison_melody:
        ensemble=int(rnd*6)
        select case ensemble
        =0
                gosub mixed_ensemble_1
        =1
                gosub mixed_ensemble_2
        =2
                gosub percussion_ensemble
        =3
                gosub string_ensemble
        =4
                gosub piano_solo
        =5
                gosub percussion_solo
        end select

        for iteration=1 to int(rnd*13)+1
                gosub check_key
                pitch=int(rnd*range(0))+low_note(0)
                drum_hit=drum_note(int(rnd*9))
                velocity=int(rnd*64)+64
                for index=0 to active_voices-1
                        if pitch>=low_note(active_instrument(index)) and
pitch<=high_note(active_instrument(index)) then
                                out
midiport,note_on+midi_channel(active_instrument(index))
                                if active_instrument(index)=drums then
                                        out midiport,drum_hit
                                else
                                        out midiport,pitch
                                end if
                                out midiport,velocity
                                gosub delta_conversion
                                pokeb
smf_buffer&+smf_index&,note_on+midi_channel(active_instrument(index))
                                incr smf_index&
                                if active_instrument(index)=drums then
                                        pokeb
smf_buffer&+smf_index&,drum_hit
                                else
                                        pokeb smf_buffer&+smf_index&,pitch
                                end if
                                incr smf_index&
                                pokeb smf_buffer&+smf_index&,velocity
                                incr smf_index&
```

```
                        end if
                next

                timespan!=.12
                delay!=timer+timespan!
                delta_time&=delta_time&+(timespan!*steps_per_second!)
                while delay!>timer: wend
                for index=0 to active_voices-1
                        out
midiport,note_off+midi_channel(active_instrument(index))
                        if active_instrument(index)=drums then
                                out midiport,drum_hit
                        else
                                out midiport,pitch
                        end if
                        out midiport,0

                        gosub delta_conversion
                        pokeb
smf_buffer&+smf_index&,note_off+midi_channel(active_instrument(index))
                        incr smf_index&
                        if active_instrument(index)=drums then
                                pokeb smf_buffer&+smf_index&,drum_hit
                        else
                                pokeb smf_buffer&+smf_index&,pitch
                        end if
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,0
                        incr smf_index&
                next
        next
        return

piano_jazz:

        for index=0 TO active_voices-1

junk=FNobjc_change(playback_dialog_ptr&,instrument_button(active_instrument
(INDEX)),RX,RY,RW,RH,0,1)
        next

junk=FNobjc_change(playback_dialog_ptr&,instrument_button(piano),RX,RY,RW,R
H,1,1)
        chord_bank=int(rnd*3)
        octave=int(rnd*3)
        for iteration=1 to int(rnd*8)+1
```

```
                gosub check_key
                pitch=int(rnd*13)+60+(12*octave)
                velocity=int(rnd*16)+56
                chord_number=int(rnd*8)

                note_index=0
                out midiport,note_on+midi_channel(piano)
                out midiport,pitch
                out midiport,velocity
                gosub delta_conversion
                pokeb smf_buffer&+smf_index&,note_on+midi_channel(piano)
                incr smf_index&
                pokeb smf_buffer&+smf_index&,pitch
                incr smf_index&
                pokeb smf_buffer&+smf_index&,velocity
                incr smf_index&
                do while jazz(chord_bank,chord_number,note_index)>-1
                        out midiport,pitch-
jazz(chord_bank,chord_number,note_index)
                        out midiport,velocity
                        gosub delta_conversion
                        pokeb
smf_buffer&+smf_index&,note_on+midi_channel(piano)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,pitch-
jazz(chord_bank,chord_number,note_index)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,velocity
                        incr smf_index&
                        incr note_index
                loop

                timespan!=2/(int(rnd*6)+1)
                delay!=timer+timespan!
                delta_time&=delta_time&+(timespan!*steps_per_second!)
                while delay!>timer: wend


                note_index=0
                out midiport,note_off+midi_channel(piano)
                out midiport,pitch
                out midiport,0
                gosub delta_conversion
                pokeb smf_buffer&+smf_index&,note_off+midi_channel(piano)
                incr smf_index&
                pokeb smf_buffer&+smf_index&,pitch
```

```
                incr smf_index&
                pokeb smf_buffer&+smf_index&,0
                incr smf_index&
                do while jazz(chord_bank,chord_number,note_index)>-1
                        out midiport,pitch-
jazz(chord_bank,chord_number,note_index)
                        out midiport,0
                        gosub delta_conversion
                        pokeb
smf_buffer&+smf_index&,note_off+midi_channel(piano)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,pitch-
jazz(chord_bank,chord_number,note_index)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,0
                        incr smf_index&
                        incr note_index
                loop
        next

junk=FNobjc_change(playback_dialog_ptr&,instrument_button(piano),RX,RY,RW,R
H,0,1)
        return

pulsed_staccato_repeat:
        if large_ensemble_flag then
                gosub large_ensemble
        else
                ensemble=int(rnd*6)
                select case ensemble
                =0
                        gosub mixed_ensemble_1
                =1
                        gosub mixed_ensemble_2
                =2
                        gosub percussion_ensemble
                =3
                        gosub string_ensemble
                =4
                        gosub piano_solo
                =5
                        gosub percussion_solo
                end select
        end if

        duration!=(rnd*1)/2
```

```
        velocity=note_off-(note_off*duration!)
        for index=0 to active_voices-1
                for note=0 to
int(rnd*(polyphony(active_instrument(index))+1))
                        if active_instrument(index)=drums then

note_buffer(active_instrument(index),note)=drum_note(int(rnd*9))
                        else

note_buffer(active_instrument(index),note)=int(rnd*range(active_instrument(
index)))+low_note(active_instrument(index))
                        end if
                next
        next

        for iteration=1 to int(rnd*5)+3
                gosub check_key

                timespan!=duration!
                delay!=timer+timespan!
                delta_time&=delta_time&+(timespan!*steps_per_second!)
                while delay!>timer: wend

                for index=0 to active_voices-1
                        for note=0 to polyphony(active_instrument(index))
                                if
note_buffer(active_instrument(index),note)>-1 then
                                        out
midiport,note_on+midi_channel(active_instrument(index))
                                        out
midiport,note_buffer(active_instrument(index),note)
                                        out midiport,velocity
                                        gosub delta_conversion
                                        pokeb
smf_buffer&+smf_index&,note_on+(active_instrument(index))
                                        incr smf_index&
                                        pokeb
smf_buffer&+smf_index&,note_buffer(active_instrument(index),note)
                                        incr smf_index&
                                        pokeb
smf_buffer&+smf_index&,velocity
                                        incr smf_index&
                                end if
                        next
                next
```

```
                timespan!=.1
                delay!=timer+timespan!
                delta_time&=delta_time&+(timespan!*steps_per_second!)
                while delay!>timer: wend

                for index=0 to active_voices-1
                        for note=0 to polyphony(active_instrument(index))
                                if
note_buffer(active_instrument(index),note)>-1 then
                                        out
midiport,note_off+midi_channel(active_instrument(index))
                                        out
midiport,note_buffer(active_instrument(index),note)
                                        out midiport,0
                                        gosub delta_conversion
                                        pokeb
smf_buffer&+smf_index&,note_off+(active_instrument(index))
                                        incr smf_index&
                                        pokeb
smf_buffer&+smf_index&,note_buffer(active_instrument(index),note)
                                        incr smf_index&
                                        pokeb smf_buffer&+smf_index&,0
                                        incr smf_index&
                                end if
                        next
                next

                timespan!=duration!
                delay!=timer+timespan!
                delta_time&=delta_time&+(timespan!*steps_per_second!)
                while delay!>timer: wend
        next

        for index=0 to active_voices-1
                for note=0 to polyphony(active_instrument(index))
                        note_buffer(active_instrument(index),note)=-1
                next
        next

        return

free_solo_with_jazz:

junk=FNobjc_change(playback_dialog_ptr&,instrument_button(piano),RX,RY,RW,R
H,1,1)
        ensemble=int(rnd*7)
```

```
        select case ensemble
        =0
                gosub mixed_ensemble_1
        =1
                gosub mixed_ensemble_2
        =2
                gosub percussion_ensemble
        =3
                gosub string_ensemble
        =4
                gosub piano_solo
        =5
                gosub percussion_solo
        =6
                gosub jazz_solo
        end select

        chord_duration!=2/(int(rnd*3)+2)
        note_count=int(rnd*3)+3
        note_duration!=chord_duration!/note_count

        chord_bank=int(rnd*3)
        octave=int(rnd*3)
        for iteration=1 to int(rnd*8)+1
                gosub check_key
                pitch=int(rnd*13)+60+(12*octave)
                velocity=int(rnd*16)+56
                chord_number=int(rnd*8)

                note_index=0
                out midiport,note_on+midi_channel(piano)
                out midiport,pitch
                out midiport,velocity
                gosub delta_conversion
                pokeb smf_buffer&+smf_index&,note_on+midi_channel(piano)
                incr smf_index&
                pokeb smf_buffer&+smf_index&,pitch
                incr smf_index&
                pokeb smf_buffer&+smf_index&,velocity
                incr smf_index&

                do while jazz(chord_bank,chord_number,note_index)>-1
                        out midiport,pitch-
jazz(chord_bank,chord_number,note_index)
                        out midiport,velocity
                        gosub delta_conversion
```

```
                        pokeb
smf_buffer&+smf_index&,note_on+midi_channel(piano)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,pitch-
jazz(chord_bank,chord_number,note_index)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,velocity
                        incr smf_index&
                        incr note_index
                loop

                note_count=int(rnd*3)+3

                for count=1 to note_count

instrument=active_instrument(int(rnd*active_voices))
                        velocity=int(rnd*64)+64

                        if instrument=drums then

note_buffer(instrument,0)=drum_note(int(rnd*9))
                        else

note_buffer(instrument,0)=int(rnd*range(instrument))+low_note(instrument)
                        end if
                        out midiport,note_on+midi_channel(instrument)
                        out midiport,note_buffer(instrument,0)
                        out midiport,velocity
                        gosub delta_conversion
                        pokeb
smf_buffer&+smf_index&,note_on+midi_channel(instrument)
                        incr smf_index&
                        pokeb
smf_buffer&+smf_index&,note_buffer(instrument,0)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,velocity
                        incr smf_index&

                        timespan!=note_duration!
                        delay!=timer+timespan!
                delta_time&=delta_time&+(timespan!*steps_per_second!)
                        while delay!>timer: wend

                        if note_buffer(instrument,0)>-1
                                out
midiport,note_off+midi_channel(instrument)
```

```
                                out midiport,note_buffer(instrument,0)
                                out midiport,0
                                gosub delta_conversion
                                pokeb
smf_buffer&+smf_index&,note_off+midi_channel(instrument)
                                incr smf_index&
                                pokeb
smf_buffer&+smf_index&,note_buffer(instrument,0)
                                incr smf_index&
                                pokeb smf_buffer&+smf_index&,0
                                incr smf_index&
                                note_buffer(instrument,0)=-1
                        end if
                next

                note_index=0
                out midiport,note_off+midi_channel(piano)
                out midiport,pitch
                out midiport,0
                gosub delta_conversion
                pokeb smf_buffer&+smf_index&,note_off+midi_channel(piano)
                incr smf_index&
                pokeb smf_buffer&+smf_index&,pitch
                incr smf_index&
                pokeb smf_buffer&+smf_index&,0
                incr smf_index&
                do while jazz(chord_bank,chord_number,note_index)>-1
                        out midiport,pitch-
jazz(chord_bank,chord_number,note_index)
                        out midiport,0
                        gosub delta_conversion
                        pokeb
smf_buffer&+smf_index&,note_off+midi_channel(piano)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,pitch-
jazz(chord_bank,chord_number,note_index)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,0
                        incr smf_index&
                        incr note_index
                loop
        next

junk=FNobjc_change(playback_dialog_ptr&,instrument_button(piano),RX,RY,RW,R
H,0,1)
        return
```

```
blues_solo_with_jazz:

junk=FNobjc_change(playback_dialog_ptr&,instrument_button(piano),RX,RY,RW,R
H,1,1)
        ensemble=int(rnd*7)
        select case ensemble
        =0
                gosub mixed_ensemble_1
        =1
                gosub mixed_ensemble_2
        =2
                gosub percussion_ensemble
        =3
                gosub string_ensemble
        =4
                gosub piano_solo
        =5
                gosub percussion_solo
        =6
                gosub jazz_solo
        end select

        chord_duration!=2/(int(rnd*3)+2)
        note_count=int(rnd*3)+3
        note_duration!=chord_duration!/note_count

        transposition=int(rnd*7)-3
        blues_index=6
        chord_bank=int(rnd*3)
        octave=int(rnd*3)
        for iteration=1 to int(rnd*8)+1
                gosub check_key
                pitch=int(rnd*13)+60+(12*octave)
                velocity=int(rnd*16)+56
                chord_number=int(rnd*8)

                note_index=0
                out midiport,note_on+midi_channel(piano)
                out midiport,pitch
                out midiport,velocity
                gosub delta_conversion
                pokeb smf_buffer&+smf_index&,note_on+midi_channel(piano)
                incr smf_index&
                pokeb smf_buffer&+smf_index&,pitch
                incr smf_index&
```

```
                pokeb smf_buffer&+smf_index&,velocity
                incr smf_index&
                do while jazz(chord_bank,chord_number,note_index)>-1
                        out midiport,pitch-
jazz(chord_bank,chord_number,note_index)
                        out midiport,velocity
                        gosub delta_conversion
                        pokeb
smf_buffer&+smf_index&,note_on+midi_channel(piano)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,pitch-
jazz(chord_bank,chord_number,note_index)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,velocity
                        incr smf_index&
                        incr note_index
                loop

                note_count=int(rnd*3)+3

                direction=int(rnd*3)-1
                for count=1 to note_count

instrument=active_instrument(int(rnd*active_voices))
                        velocity=int(rnd*32)+64

                        if instrument=drums then

note_buffer(instrument,0)=drum_note(int(rnd*9))
                        else
                                blues_index=blues_index+direction
                                if blues_index<0 then
                                        blues_index=6
                                elseif blues_index>12 then
                                        blues_index=6
                                end if

note_buffer(instrument,0)=blues_scale(blues_index)+transposition
                        end if
                        out midiport,note_on+midi_channel(instrument)
                        out midiport,note_buffer(instrument,0)
                        out midiport,velocity
                        gosub delta_conversion
                        pokeb
smf_buffer&+smf_index&,note_on+midi_channel(instrument)
                        incr smf_index&
```

```
                        pokeb
smf_buffer&+smf_index&,note_buffer(instrument,0)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,velocity
                        incr smf_index&

                        timespan!=note_duration!
                delay!=timer+timespan!
                        delta_time&=delta_time&+(timespan!
*steps_per_second!)
                        while delay!>timer: wend

                        if note_buffer(instrument,0)>-1
                                out
midiport,note_off+midi_channel(instrument)
                                out midiport,note_buffer(instrument,0)
                                out midiport,0
                                gosub delta_conversion
                                pokeb
smf_buffer&+smf_index&,note_off+midi_channel(instrument)
                                incr smf_index&
                                pokeb
smf_buffer&+smf_index&,note_buffer(instrument,0)
                                incr smf_index&
                                pokeb smf_buffer&+smf_index&,0
                                incr smf_index&
                                note_buffer(instrument,0)=-1
                        end if
                next

                note_index=0
                out midiport,note_off+midi_channel(piano)
                out midiport,pitch
                out midiport,0
                gosub delta_conversion
                pokeb smf_buffer&+smf_index&,note_off+midi_channel(piano)
                incr smf_index&
                pokeb smf_buffer&+smf_index&,pitch
                incr smf_index&
                pokeb smf_buffer&+smf_index&,0
                incr smf_index&
                do while jazz(chord_bank,chord_number,note_index)>-1
                        out midiport,pitch-
jazz(chord_bank,chord_number,note_index)
                        out midiport,0
                        gosub delta_conversion
```

```
                        pokeb
smf_buffer&+smf_index&,note_off+midi_channel(piano)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,pitch-
jazz(chord_bank,chord_number,note_index)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,0
                        incr smf_index&
                        incr note_index
                loop
        next

junk=FNobjc_change(playback_dialog_ptr&,instrument_button(piano),RX,RY,RW,R
H,0,1)
        return

drum_solo:
        for index=0 TO active_voices-1

junk=FNobjc_change(playback_dialog_ptr&,instrument_button(active_instrument
(INDEX)),RX,RY,RW,RH,0,1)
        next

junk=FNobjc_change(playback_dialog_ptr&,instrument_button(drums),RX,RY,RW,R
H,1,1)

        duration!=(rnd*1)/8+.06
        for iteration=1 to int(rnd*24)+1
                gosub check_key
                pitch=drum_note(int(rnd*9))
                velocity=int(rnd*48)+48
                out midiport,note_on+midi_channel(drums)
                out midiport,pitch
                out midiport,velocity
                gosub delta_conversion
                pokeb smf_buffer&+smf_index&,note_on+midi_channel(drums)
                incr smf_index&
                pokeb smf_buffer&+smf_index&,pitch
                incr smf_index&
                pokeb smf_buffer&+smf_index&,velocity
                incr smf_index&

                timespan!=duration!
                delay!=timer+timespan!
                delta_time&=delta_time&+(timespan!*steps_per_second!)
                while delay!>timer: wend
```

```
                out midiport,note_off+midi_channel(drums)
                out midiport,pitch
                out midiport,velocity
                gosub delta_conversion
                pokeb smf_buffer&+smf_index&,note_off+midi_channel(drums)
                incr smf_index&
                pokeb smf_buffer&+smf_index&,pitch
                incr smf_index&
                pokeb smf_buffer&+smf_index&,velocity
                incr smf_index&
        next

junk=FNobjc_change(playback_dialog_ptr&,instrument_button(drums),RX,RY,RW,R
H,0,1)
        return

fast_solo_with_jazz:

junk=FNobjc_change(playback_dialog_ptr&,instrument_button(piano),RX,RY,RW,R
H,1,1)

junk=FNobjc_change(playback_dialog_ptr&,instrument_button(bass),RX,RY,RW,RH
,1,1)

junk=FNobjc_change(playback_dialog_ptr&,instrument_button(drums),RX,RY,RW,R
H,1,1)
        gosub jazz_solo

        instrument=active_instrument(int(rnd*active_voices))
        solo_note=int(rnd*range(instrument))+low_note(instrument)

        for iteration=1 to int(rnd*6)+3
                gosub check_key
                chord_pitch=int(rnd*7)+49
                velocity=int(rnd*48)+16
                out midiport,note_on+midi_channel(piano)
                out midiport,chord_pitch
                out midiport,velocity
                gosub delta_conversion
                pokeb smf_buffer&+smf_index&,note_on+midi_channel(piano)
                incr smf_index&
                pokeb smf_buffer&+smf_index&,chord_pitch
                incr smf_index&
                pokeb smf_buffer&+smf_index&,velocity
                incr smf_index&
```

```
out midiport,chord_pitch+6
out midiport,velocity
gosub delta_conversion
pokeb smf_buffer&+smf_index&,note_on+midi_channel(piano)
incr smf_index&
pokeb smf_buffer&+smf_index&,chord_pitch+6
incr smf_index&
pokeb smf_buffer&+smf_index&,velocity
incr smf_index&

out midiport,chord_pitch+11
out midiport,velocity
gosub delta_conversion
pokeb smf_buffer&+smf_index&,note_on+midi_channel(piano)
incr smf_index&
pokeb smf_buffer&+smf_index&,chord_pitch+11
incr smf_index&
pokeb smf_buffer&+smf_index&,velocity
incr smf_index&

out midiport,chord_pitch+16
out midiport,velocity
gosub delta_conversion
pokeb smf_buffer&+smf_index&,note_on+midi_channel(piano)
incr smf_index&
pokeb smf_buffer&+smf_index&,chord_pitch+16
incr smf_index&
pokeb smf_buffer&+smf_index&,velocity
incr smf_index&

out midiport,chord_pitch+21
out midiport,velocity
gosub delta_conversion
pokeb smf_buffer&+smf_index&,note_on+midi_channel(piano)
incr smf_index&
pokeb smf_buffer&+smf_index&,chord_pitch+21
incr smf_index&
pokeb smf_buffer&+smf_index&,velocity
incr smf_index&

out midiport,chord_pitch+26
out midiport,velocity
gosub delta_conversion
pokeb smf_buffer&+smf_index&,note_on+midi_channel(piano)
incr smf_index&
```

```
                pokeb smf_buffer&+smf_index&,chord_pitch+26
                incr smf_index&
                pokeb smf_buffer&+smf_index&,velocity
                incr smf_index&

                note_count=int(rnd*5)+3
                for count=0 to note_count
                        bass_flag=bass_flag xor 1
                        IF bass_flag then
                                out midiport,note_off+midi_channel(drums)
                                out midiport,ride_cymbal
                                out midiport,0
                                gosub delta_conversion
                                pokeb
smf_buffer&+smf_index&,note_off+midi_channel(drums)
                                incr smf_index&
                                pokeb smf_buffer&+smf_index&,ride_cymbal
                                incr smf_index&
                                pokeb smf_buffer&+smf_index&,0
                                incr smf_index&

                                out midiport,note_off+midi_channel(bass)
                                out midiport,bass_pitch
                                out midiport,0
                                gosub delta_conversion
                                pokeb
smf_buffer&+smf_index&,note_off+midi_channel(bass)
                                incr smf_index&
                                pokeb smf_buffer&+smf_index&,bass_pitch
                                incr smf_index&
                                pokeb smf_buffer&+smf_index&,0
                                incr smf_index&

                                cymbal_velocity=int(rnd*16)+52
                                out midiport,note_on+midi_channel(drums)
                                out midiport,ride_cymbal
                                out midiport,cymbal_velocity
                                gosub delta_conversion
                                pokeb
smf_buffer&+smf_index&,note_on+midi_channel(drums)
                                incr smf_index&
                                pokeb smf_buffer&+smf_index&,ride_cymbal
                                incr smf_index&
                                pokeb
smf_buffer&+smf_index&,cymbal_velocity
                                incr smf_index&
```

```
bass_pitch=int(rnd*range(bass))+low_note(bass)
                                out midiport,note_on+midi_channel(bass)
                                out midiport,bass_pitch
                                out midiport,64
                                gosub delta_conversion
                                pokeb
smf_buffer&+smf_index&,note_on+midi_channel(bass)
                                incr smf_index&
                                pokeb smf_buffer&+smf_index&,bass_pitch
                                incr smf_index&
                                pokeb smf_buffer&+smf_index&,64
                                incr smf_index&
                        END IF
                        direction=int(rnd*5)-2
                        if direction<>0 then
                                if instrument=drums then
                                        solo_note=drum_note(int(rnd*9))
                                else
                                        solo_note=solo_note+direction
                                        if solo_note<low_note(instrument)
then
                                                solo_note=solo_note+12
                                        elseif
solo_note>high_note(instrument) then
                                                solo_note=solo_note-12
                                        end if
                                end if

                                velocity=int(rnd*16)+64
                                out
midiport,note_on+midi_channel(instrument)
                                out midiport,solo_note
                                out midiport,velocity
                                gosub delta_conversion
                                pokeb
smf_buffer&+smf_index&,note_on+midi_channel(instrument)
                                incr smf_index&
                                pokeb smf_buffer&+smf_index&,solo_note
                                incr smf_index&
                                pokeb smf_buffer&+smf_index&,velocity
                                incr smf_index&
                        end if

                        timespan!=.1
```

```
                        delay!=timer+timespan!
                        delta_time&=delta_time&+(timespan!
*steps_per_second!)
                        while delay!>timer: wend


                        if direction<>0 then
                                out
midiport,note_off+midi_channel(instrument)
                                out midiport,solo_note
                                out midiport,0
                                gosub delta_conversion
                                pokeb
smf_buffer&+smf_index&,note_off+midi_channel(instrument)
                                incr smf_index&
                                pokeb smf_buffer&+smf_index&,solo_note
                                incr smf_index&
                                pokeb smf_buffer&+smf_index&,0
                                incr smf_index&
                        end if
                next

                out midiport,note_off+midi_channel(piano)
                out midiport,chord_pitch
                out midiport,0
                pokeb smf_buffer&+smf_index&,0
                incr smf_index&
                pokeb smf_buffer&+smf_index&,note_off+midi_channel(piano)
                incr smf_index&
                pokeb smf_buffer&+smf_index&,chord_pitch
                incr smf_index&
                pokeb smf_buffer&+smf_index&,0
                incr smf_index&

                out midiport,chord_pitch+6
                out midiport,0
                pokeb smf_buffer&+smf_index&,0
                incr smf_index&
                pokeb smf_buffer&+smf_index&,note_off+midi_channel(piano)
                incr smf_index&
                pokeb smf_buffer&+smf_index&,chord_pitch+6
                incr smf_index&
                pokeb smf_buffer&+smf_index&,0
                incr smf_index&

                out midiport,chord_pitch+11
```

```
        out midiport,0
        pokeb smf_buffer&+smf_index&,0
        incr smf_index&
        pokeb smf_buffer&+smf_index&,note_off+midi_channel(piano)
        incr smf_index&
        pokeb smf_buffer&+smf_index&,chord_pitch+11
        incr smf_index&
        pokeb smf_buffer&+smf_index&,0
        incr smf_index&

        out midiport,chord_pitch+16
        out midiport,0
        pokeb smf_buffer&+smf_index&,0
        incr smf_index&
        pokeb smf_buffer&+smf_index&,note_off+midi_channel(piano)
        incr smf_index&
        pokeb smf_buffer&+smf_index&,chord_pitch+16
        incr smf_index&
        pokeb smf_buffer&+smf_index&,0
        incr smf_index&

        out midiport,chord_pitch+21
        out midiport,0
        pokeb smf_buffer&+smf_index&,0
        incr smf_index&
        pokeb smf_buffer&+smf_index&,note_off+midi_channel(piano)
        incr smf_index&
        pokeb smf_buffer&+smf_index&,chord_pitch+21
        incr smf_index&
        pokeb smf_buffer&+smf_index&,0
        incr smf_index&

        out midiport,chord_pitch+26
        out midiport,0
        pokeb smf_buffer&+smf_index&,0
        incr smf_index&
        pokeb smf_buffer&+smf_index&,note_off+midi_channel(piano)
        incr smf_index&
        pokeb smf_buffer&+smf_index&,chord_pitch+26
        incr smf_index&
        pokeb smf_buffer&+smf_index&,0
        incr smf_index&

next
out midiport,note_off+midi_channel(drums)
out midiport,ride_cymbal
```

```
        out midiport,0
        pokeb smf_buffer&+smf_index&,0
        incr smf_index&
        pokeb smf_buffer&+smf_index&,note_off+midi_channel(drums)
        incr smf_index&
        pokeb smf_buffer&+smf_index&,ride_cymbal
        incr smf_index&
        pokeb smf_buffer&+smf_index&,0
        incr smf_index&

        out midiport,note_off+midi_channel(bass)
        out midiport,bass_pitch
        out midiport,0
        pokeb smf_buffer&+smf_index&,0
        incr smf_index&
        pokeb smf_buffer&+smf_index&,note_off+midi_channel(bass)
        incr smf_index&
        pokeb smf_buffer&+smf_index&,bass_pitch
        incr smf_index&
        pokeb smf_buffer&+smf_index&,0
        incr smf_index&


junk=FNobjc_change(playback_dialog_ptr&,instrument_button(piano),RX,RY,RW,R
H,0,1)

junk=FNobjc_change(playback_dialog_ptr&,instrument_button(bass),RX,RY,RW,RH
,0,1)

junk=FNobjc_change(playback_dialog_ptr&,instrument_button(drums),RX,RY,RW,R
H,0,1)
        return

free_solo_with_chords:

junk=FNobjc_change(playback_dialog_ptr&,instrument_button(piano),RX,RY,RW,R
H,1,1)
        ensemble=int(rnd*3)
        select case ensemble
        =0
                gosub mixed_ensemble_2
        =1
                gosub string_ensemble
        =2
                gosub piano_solo
        end select
```

```
        chord_duration!=(int(rnd*3)+1)*.82
        note_count=int(rnd*3)+2
        note_duration!=chord_duration!/note_count

        octave=int(rnd*2)
        for iteration=1 to int(rnd*8)+1
                gosub check_key
                pitch=int(rnd*12)+72+(12*octave)
                velocity=int(rnd*16)+56
                chord_number=int(rnd*8)

                note_index=0
                out midiport,note_on+midi_channel(piano)
                out midiport,pitch
                out midiport,velocity
                gosub delta_conversion
                pokeb smf_buffer&+smf_index&,note_on+midi_channel(piano)
                incr smf_index&
                pokeb smf_buffer&+smf_index&,pitch
                incr smf_index&
                pokeb smf_buffer&+smf_index&,velocity
                incr smf_index&
                do while chords(chord_number,note_index)>-1
                        out midiport,pitch-chords(chord_number,note_index)
                        out midiport,velocity
                        gosub delta_conversion
                        pokeb
smf_buffer&+smf_index&,note_on+midi_channel(piano)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,pitch-
chords(chord_number,note_index)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,velocity
                        incr smf_index&
                        incr note_index
                loop

                note_count=int(rnd*3)+2

                for count=1 to note_count

instrument=active_instrument(int(rnd*active_voices))
                        register=int(rnd*(range(instrument)-12))
                        velocity=int(rnd*32)+48
```

```
                        stop_note_flag=INT(RND*3)
                        IF stop_note_flag=0 THEN
                                if note_buffer(instrument,0)>-1
                                        out
midiport,note_off+midi_channel(instrument)
                                        out
midiport,note_buffer(instrument,0)
                                        out midiport,0
                                        gosub delta_conversion
                                        pokeb
smf_buffer&+smf_index&,note_off+midi_channel(instrument)
                                        incr smf_index&
                                        pokeb
smf_buffer&+smf_index&,note_buffer(instrument,0)
                                        incr smf_index&
                                        pokeb smf_buffer&+smf_index&,0
                                        incr smf_index&
                                        note_buffer(instrument,0)=-1
                                end if

                                if instrument=drums then

note_buffer(instrument,0)=drum_note(int(rnd*9))
                                else

note_buffer(instrument,0)=int(rnd*13)+low_note(instrument)+register
                                end if
                                out
midiport,note_on+midi_channel(instrument)
                                out midiport,note_buffer(instrument,0)
                                out midiport,velocity
                                gosub delta_conversion
                                pokeb
smf_buffer&+smf_index&,note_on+midi_channel(instrument)
                                incr smf_index&
                                pokeb
smf_buffer&+smf_index&,note_buffer(instrument,0)
                                incr smf_index&
                                pokeb smf_buffer&+smf_index&,velocity
                                incr smf_index&
                        END IF
                        timespan!=note_duration!
                        delay!=timer+timespan!
                        delta_time&=delta_time&+(timespan!
*steps_per_second!)
                        while delay!>timer: wend
```

```
                next

                note_index=0
                out midiport,note_off+midi_channel(piano)
                out midiport,pitch
                out midiport,0
                gosub delta_conversion
                pokeb smf_buffer&+smf_index&,note_off+midi_channel(piano)
                incr smf_index&
                pokeb smf_buffer&+smf_index&,pitch
                incr smf_index&
                pokeb smf_buffer&+smf_index&,0
                incr smf_index&
                do while chords(chord_number,note_index)>-1
                        out midiport,pitch-chords(chord_number,note_index)
                        out midiport,0
                        gosub delta_conversion
                        pokeb
smf_buffer&+smf_index&,note_off+midi_channel(piano)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,pitch-
chords(chord_number,note_index)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,0
                        incr smf_index&
                        incr note_index
                loop
        next

        for instrument=0 to 20
                if note_buffer(instrument,0)>-1
                        out midiport,note_off+midi_channel(instrument)
                        out midiport,note_buffer(instrument,0)
                        out midiport,0
                        gosub delta_conversion
                        pokeb
smf_buffer&+smf_index&,note_off+midi_channel(instrument)
                        incr smf_index&
                        pokeb
smf_buffer&+smf_index&,note_buffer(instrument,0)
                        incr smf_index&
                        pokeb smf_buffer&+smf_index&,0
                        incr smf_index&
                        note_buffer(instrument,0)=-1
                end if
        next
```

```
junk=FNobjc_change(playback_dialog_ptr&,instrument_button(piano),RX,RY,RW,R
H,0,1)
        return

initialize:
        if FNrsrc_load("WITGNSTN.RSC")=0 then
                alert$="[1][WITGNSTN.RSC missing.][Abort]"
                junk=FNform_alert(1,alert$)
                system
        end if

        mouse -1
        junk=FNrsrc_gaddr(0,witgnstn_dialog%,witgnstn_dialog_ptr&)
        form_center witgnstn_dialog_ptr&,rx,ry,rw,rh
        form_dial 0,0,0,0,0,0,0,0,0
        form_dial 1,320,200,1,1,rx,ry,rw,rh
        junk=FNobjc_draw(witgnstn_dialog_ptr&,0,7,rx,ry,rw,rh)

        randomize(timer)
        dim midi_channel(20)
        dim low_note(20)
        dim high_note(20)
        dim range(20)
        dim polyphony(20)
        dim note_buffer(20,5)
        dim active_instrument(20)
        dim instrument_button(20)
        dim selection(20)
        dim drum_note(8)
        dim blues_scale(12)

        active_voices=1

        restore range_data
        for instrument=0 to 20
                read midi_channel(instrument)
                read low_note(instrument)
                read high_note(instrument)
                read range(instrument)
                read polyphony(instrument)
                for note=0 to 5
                        note_buffer(instrument,note)=-1
                next
        next
```

```
restore drum_note_data
for index=0 to 8
        read drum_note(index)
next

restore jazz_chord_data
dim jazz(2,7,7)
for chord_bank=0 to 2
        for chord_number=0 to 7
                for pitch=0 to 7
                        read jazz(chord_bank,chord_number,pitch)
                next
        next
next
chord_bank=0

restore chord_data
dim chords(7,7)
for chord_number=0 to 7
        for pitch=0 to 7
                read chords(chord_number,pitch)
        next
next

restore blues_scale_data
for index=0 to 12
        read blues_scale(index)
next

restore dump_data
dim sysex(78)
for index&=0 to 78
read word$
pokew varptr(sysex(0))+index&*2,val(word$)
next

restore mode_data
dim combi_mode(8)
for index&=0 to 8
read word$
pokew varptr(combi_mode(0))+index&*2,val(word$)
next

instrument_button(0)=piano_button%
instrument_button(1)=lh_button%
instrument_button(2)=rh_button%
```

```
instrument_button(3)=vibes_button%
instrument_button(4)=guitar_button%
instrument_button(5)=drums_button%
instrument_button(6)=bells_button%
instrument_button(7)=bass_button%
instrument_button(8)=flute_button%
instrument_button(9)=oboe_button%
instrument_button(10)=bassoon_button%
instrument_button(11)=vln_arco_button%
instrument_button(12)=vln_pizz_button%
instrument_button(13)=celloarco_button%
instrument_button(14)=cello_piz_button%
instrument_button(15)=timp_hard_button%
instrument_button(16)=timp_soft_button%
instrument_button(17)=xylophone_button%
instrument_button(18)=sax_button%
instrument_button(19)=tamtam_button%
instrument_button(20)=effects_button%

for index=0 to 17
        out 3,peekb(varptr(combi_mode(0))+index)
next

delay!=.25
while delay!>timer: wend

for index=0 to 157
        out 3,peekb(varptr(sysex(0))+index)
next

out midiport,aftertouch+midi_channel(flute)
out midiport,76
out midiport,aftertouch+midi_channel(oboe)
out midiport,76
out midiport,aftertouch+midi_channel(guitar)
out midiport,48
out midiport,aftertouch+midi_channel(sax)
out midiport,127

drive_no=FNdgetdrv%
getdir$=space$(128)
getdir&=sadd(getdir$)
dgetpath getdir&,0
index=0: while peekb(getdir&+index)<>32: index=index+1: wend
path$=chr$(drive_no+65)+":"+left$(getdir$,index-1)
midifile_path$=path$+"\*.MID"
```

```
        buffer_size&=fre(0)-&h20000
        dim smf_buffer(buffer_size&\2)
        smf_buffer&=varptr(smf_buffer(0))

        'SET UP SMF HEADER CHUNK
        pokeb smf_buffer&,asc("M")
        pokeb smf_buffer&+1,asc("T")
        pokeb smf_buffer&+2,asc("h")
        pokeb smf_buffer&+3,asc("d")
        pokel smf_buffer&+4,6           'length of header
        pokew smf_buffer&+8,0           'SMF format 0
        pokew smf_buffer&+10,1          '1 track

        'SET UP TRACK CHUNK
        pokeb smf_buffer&+14,asc("M")
        pokeb smf_buffer&+15,asc("T")
        pokeb smf_buffer&+16,asc("r")
        pokeb smf_buffer&+17,asc("k")
        pokel smf_buffer&+18,0          'dummy track length

        a=inp(2)
        form_dial 2,320,200,1,1,rx,ry,rw,rh
        form_dial 3,0,0,0,0,rx,ry,rw,rh

        junk=FNrsrc_gaddr(0,playback_dialog%,playback_dialog_ptr&)
        form_center playback_dialog_ptr&,rx,ry,rw,rh
        form_dial 0,0,0,0,0,0,0,0,0
        form_dial 1,320,200,1,1,rx,ry,rw,rh
        junk=FNobjc_draw(playback_dialog_ptr&,0,7,rx,ry,rw,rh)

        return

reset_midi:
        out midiport,aftertouch+midi_channel(flute)
        out midiport,0
        out midiport,aftertouch+midi_channel(oboe)
        out midiport,0
        out midiport,aftertouch+midi_channel(guitar)
        out midiport,0
        out midiport,aftertouch+midi_channel(sax)
        out midiport,0
        mouse 0
        return

large_ensemble:
```

```
        selection(0)=piano
        selection(1)=vibes
        selection(2)=guitar
        selection(3)=drums
        selection(4)=bells
        selection(5)=bass
        selection(6)=flute
        selection(7)=oboe
        selection(8)=bassoon
        selection(9)=sax
        selection(10)=violin_arco
        selection(11)=violin_pizz
        selection(12)=cello_arco
        selection(13)=cello_pizz
        selection(14)=timp_hard
        selection(15)=timp_soft
        selection(16)=xylophone
        selection(17)=tamtam
        selection_range=18
        maximum_range=9
        minimum_size=10
        gosub create_ensemble
        return

mixed_ensemble_1:
        for index=0 to 20
                selection(index)=index
        next
        selection_range=21
        maximum_range=21
        minimum_size=1
        gosub create_ensemble
        return

mixed_ensemble_2:
        selection(0)=piano
        selection(1)=piano_rh
        selection(2)=vibes
        selection(3)=flute
        selection(4)=oboe
        selection(5)=bassoon
        selection(6)=violin_arco
        selection(7)=violin_pizz
        selection(8)=cello_arco
        selection(10)=cello_pizz
        selection(11)=bass
```

```
        selection_range=12
        maximum_range=12
        minimum_size=1
        gosub create_ensemble
        return

percussion_ensemble:
        selection(0)=piano
        selection(1)=piano_lh
        selection(2)=piano_rh
        selection(3)=vibes
        selection(4)=drums
        selection(5)=bells
        selection(6)=timpani_hard
        selection(7)=timpani_soft
        selection(8)=xylophone
        selection(9)=tamtam
        selection_range=10
        maximum_range=10
        minimum_size=1
        gosub create_ensemble
        return

string_ensemble:
        selection(0)=piano
        selection(1)=piano_lh
        selection(2)=piano_rh
        selection(3)=violin_arco
        selection(4)=violin_pizz
        selection(5)=cello_arco
        selection(6)=cello_pizz
        selection(7)=bass
        selection_range=8
        maximum_range=8
        minimum_size=1
        gosub create_ensemble
        return

piano_solo:
        selection(0)=piano
        selection(1)=piano_lh
        selection(2)=piano_rh
        selection_range=3
        maximum_range=3
        minimum_size=1
        gosub create_ensemble
```

```
        return

percussion_solo:
        selection(0)=piano
        selection(1)=piano_lh
        selection(2)=piano_rh
        selection(3)=vibes
        selection(4)=drums
        selection(5)=timpani_hard
        selection(6)=timpani_soft
        selection(7)=xylophone
        selection_range=7
        maximum_range=1
        minimum_size=1
        gosub create_ensemble
        return

jazz_solo:
        selection(0)=piano_rh
        selection(1)=vibes
        selection(2)=drums
        selection(3)=flute
        selection(4)=sax
        selection(5)=bass
        selection(6)=violin_arco
        selection(7)=cello_arco
        selection_range=8
        maximum_range=1
        minimum_size=1
        gosub create_ensemble
        return

create_ensemble:
        for index=0 TO active_voices-1

junk=FNobjc_change(playback_dialog_ptr&,instrument_button(active_instrument
(INDEX)),RX,RY,RW,RH,0,1)
        next
        active_voices=int(rnd*maximum_range)+minimum_size
        voice_index=0
        for voice_count=1 to active_voices
                selection_pointer=int(rnd*selection_range)
                active_instrument(voice_index)=selection(selection_pointer)
                incr voice_index
                decr selection_range
                if selection_pointer<selection_range then
```

```
                        for shift=selection_pointer to selection_range-1
                                selection(shift)=selection(shift+1)
                        next
                end if
        next
        for index=0 TO active_voices-1

junk=FNobjc_change(playback_dialog_ptr&,instrument_button(active_instrument
(index)),RX,RY,RW,RH,1,1)
        next
        return

BUFFER_SETUP:
        smf_buffer&=varptr(smf_buffer(0))
        smf_index&=22

        tempo=120
        delta_time=0
        steps_per_beat=96
        beats_per_second!=tempo/60
        steps_per_second!=steps_per_beat*beats_per_second!

        pokew smf_buffer&+12,steps_per_beat
        gosub set_smf_tempo

        while inp(-3)<>0
                junk=inp(3)
        wend
        RETURN

set_smf_tempo:
        smf_tempo&=(60/tempo)*1000000
        smf_tempo_ptr&=varptr(smf_tempo&)

        pokeb smf_buffer&+smf_index&,&h00       'delta time
        incr smf_index&
        pokeb smf_buffer&+smf_index&,&hff       'meta event
        incr smf_index&
        pokeb smf_buffer&+smf_index&,&h51
        incr smf_index&
        pokeb smf_buffer&+smf_index&,&h03
        incr smf_index&
        pokeb smf_buffer&+smf_index&,peekb(smf_tempo_ptr&+1)
        incr smf_index&
        pokeb smf_buffer&+smf_index&,peekb(smf_tempo_ptr&+2)
        incr smf_index&
```

```
        pokeb smf_buffer&+smf_index&,peekb(smf_tempo_ptr&+3)
        incr smf_index&
        return

SAVE_MIDI_FILE:
        'APPEND END-OF-TRACK META-EVENT

        gosub delta_conversion
        pokeb smf_buffer&+smf_index&,&hff       'meta-event
        incr smf_index&
        pokeb smf_buffer&+smf_index&,&h2f
        incr smf_index&
        pokeb smf_buffer&+smf_index&,&h00

        pokel smf_buffer&+18,smf_index&-22      'update sequence length

        save_midi:
        file_name$=""
        fsel_input midifile_path$,file_name$,ok
        if ok then
                if file_name$="" then
                        alert$="[3][Are you sure you want|to discard MIDI
file?][Yes|No]"
                        if FNform_alert(1,alert$)=2 then
                                goto save_midi
                        end if
                else
                        mouse 2
                        index=len(midifile_path$)
                        while index>0 and mid$(midifile_path$,index,1)<>"\"
                                decr index
                        wend
                        exit_file$=left$(midifile_path$,index)+file_name$
                        BSAVE exit_file$,SMF_BUFFER&,SMF_INDEX&
                        mouse 0
                end if
        else
                alert$="[3][Are you sure you want|to discard MIDI
file?][Yes|No]"
                if FNform_alert(1,alert$)=2 then
                        goto save_midi
                end if
        end if
        RETURN

'DELTA TIME VARIABLE-LENGTH QUANTITY CONVERSION
```

```
delta_conversion:
        vlq_buffer&=0
        vlq_pointer&=varptr(vlq_buffer&)
        pokeb vlq_pointer&+3,delta_time& and &h7f
        pointer_offset&=2
        delta_time&=delta_time&\&h80
        while delta_time&<>0
                pokeb vlq_pointer&+pointer_offset&,delta_time& or &h80
                decr pointer_offset&
                delta_time&=delta_time&\&h80
        wend
        incr pointer_offset&
        while pointer_offset&<4
                pokeb
smf_buffer&+smf_index&,peekb(vlq_pointer&+pointer_offset&)
                incr smf_index&
                incr pointer_offset&
        wend
        return

range_data:
'piano
DATA 00,24,096,73,05
'piano left hand
DATA 00,24,053,30,01
'piano right hand
DATA 00,65,096,32,02
'vibes
DATA 01,53,084,32,03
'guitar
DATA 02,40,076,37,00
'drums
DATA 03,36,096,61,00
'bells
DATA 04,65,089,25,00
'bass
DATA 05,40,072,33,00
'flute
DATA 06,60,101,42,00
'oboe
DATA 07,72,096,25,00
'bassoon
DATA 07,36,060,25,00
'violin arco
DATA 10,55,096,42,01
'violin pizz
```

```
DATA 11,55,096,42,01
'cello arco
DATA 12,36,079,44,01
'cello pizz
DATA 11,36,079,44,01
'timpani hard
DATA 13,52,065,14,00
'timapni soft
DATA 13,66,080,15,00
'xylophone
DATA 13,36,051,16,01
'sax
DATA 14,48,080,33,00
'tamtam
DATA 13,81,096,16,01
'effects
DATA 14,81,096,16,00

drum_note_data:
DATA 40,41,49,50,51,52,53,54,55

jazz_chord_data:
' BANK 1
DATA 05,07,12,17,19,24,-1,00
DATA 07,12,15,23,-1,00,00,00
DATA 05,07,11,16,-1,00,00,00
DATA 12,-1,00,00,00,00,00,00
DATA 07,-1,00,00,00,00,00,00
DATA 05,-1,00,00,00,00,00,00
DATA 03,-1,00,00,00,00,00,00
DATA 04,-1,00,00,00,00,00,00
' BANK 2
DATA 04,09,16,20,26,-1,00,00
DATA 05,07,12,17,19,24,-1,00
DATA 05,07,12,17,19,24,28,-1
DATA 05,10,15,20,26,-1,00,00
DATA 04,09,15,20,26,-1,00,00
DATA 04,09,15,20,26,30,-1,00
DATA 07,12,15,23,-1,00,00,00
DATA 05,07,11,16,-1,00,00,00
' BANK 3
DATA 04,09,16,20,26,42,-1,00
DATA 05,07,12,17,19,24,35,-1
DATA 05,10,15,20,26,30,-1,00
DATA 05,10,15,20,26,36,-1,00
DATA 04,09,15,20,26,-1,00,00
```

```
DATA 04,09,15,20,26,30,-1,00
DATA 07,12,15,23,-1,00,00,00
DATA 05,07,11,16,-1,00,00,00

chord_data:
DATA 09,16,23,31,38,-1,00,00
DATA 04,09,11,14,19,-1,00,00
DATA 05,08,10,15,19,-1,00,00
DATA 03,07,10,14,17,-1,00,00
DATA 07,15,22,31,38,-1,00,00
DATA 04,09,11,16,19,-1,00,00
DATA 05,08,15,19,24,-1,00,00
DATA 07,08,15,17,24,-1,00,00

blues_scale_data:
DATA 60,63,65,66,67,70
DATA 72,75,77,78,79,82,84

mode_data:
DATA &hF042,&h3F19,&h4E00,&h10F7,&hF042,&h3F19,&h4101,&h082A,&h00F7

dump_data:
DATA &hF042,&h3F19,&h4900,&h5769,&h7474,&h6765,&h6E00,&h7374,&h6E04,&h0901
DATA &h6400,&h6413,&h1300,&h001F,&h0A00,&h0030,&h0E05,&h0003,&h0400,&h1A00
DATA &h281E,&h2E00,&h0001,&h7D29,&h5000,&h0005,&h7F08,&h007F,&h017F,&h000F
DATA &h5000,&h0000,&h047F,&h007F,&h0101,&h7F01,&h0448,&h0000,&h0310,&h7F00
DATA &h7F01,&h7F02,&h0908,&h5000,&h0005,&h7F00,&h7F02,&h017F,&h0325,&h5000
DATA &h0020,&h067F,&h007F,&h017F,&h0400,&h1A50,&h0000,&h077F,&h0004,&h7F01
DATA &h7F05,&h1250,&h0040,&h0002,&h7F00,&h7F01,&h7F00,&h0620,&h5000,&h0008
DATA &h7F08,&h007F,&h017F,&h07F7,&hF042,&h3F19,&h4101,&h082A,&h00F7

'CHANNEL        PATCH                   INSTRUMENT
'       1               piano   #41             M1
'       1               LH only #41             M1
'       1               RH only #41             M1
'       2               vibes   #15             M1
'       3               guitar  #04             M1
'       4               drums   #09             M1
'       5               bells   #37             M1
'       6               a. bass #26             M1
'       7               flute   #18             M1
'       8               oboe    #32             M1
'       8               bassoon #32             M1
'       11              violin (arco)   TX16W
'       12              violin (pizz)   TX16W
'       13              cello (arco)    TX16W
```

```
'       12              cello (pizz)    TX16W
'       14              timpani                 TX16W
'       14              xylophone               TX16W
'       15              sax                             TX16W
'       14              tamtam                  TX16W
'       15              effects                 TX16W
```