```
*************************************************
* ASSEMBLY ALL PLAYERS
*
*
* This a musical score in form of a set of performance instructions
* It is inspired by the assembly program language or machine code instructions
* but the players in this case is not to be seen as machines but as performing artists
* Perhaps by this we will better learn the difference between human and machine
*
* Basic guide:
*
* This is a comment -lines that starts with * is only more or less helpful comments
* Lines that doesn't start with * is real instructions
* After this initial chunk of comment the code begins with the procedures
* Go further in the document and find the actual starting point
*
* The lines in the instructions is of four different categories:
* Procedure calls (find the procedure being called and perform it line by line)
* Jump calls (the instruction 'JUMP TO' and a number, jump to the given number in the instructions)
* Conditional calls (if the condition is met continue, otherwise jump to the line that starts with 'END' and the condition)
* Direct calls (Basic calls that the Player is assumed to know and react to)
*
* Direct calls:
*
* RESET (Stop any ongoing procedure and empty memory from previous instructions)
* GOTO (Move to another physical location if neccesary)
* PREPARE PLAY (Do sufficient preparations to be able to make sound from instrument)
* MEMORIZE (Memorize a given instructions)
* PLAY (Play instrument with instuction in memory, or if memory is empty Just Play!!)
* SURPRISE (Do something unpredictable)
* SPAM (Give the audience or a fellow player an unwanted message)
* ASSEMBLY (Two or more players are gathered)
*
*
* Basic conditions that could be identified by the conditional calls:
*
* ERROR (Something goes wrong and the instructions can't be followed)
* NONINTEREST (The players or the audicence finds the output not interesting)
* NONCOMPRENDE (The instructions is to complex to understand)
* (Note that these conditions applies to all players, and also the audience in the case of NONINTEREST)
*
*
* In the procedures there are some required parameters and some optional parameters
* These are also conditioned like this 'IF OPTIONAL Parameter'. Like before, if the condition isn't met jump to 'END'
*
*
* Different outputs from the procedures:
*
* None (tacet)
* Music (intonate pitches from the chromatic scale, use some kind of pulse as a rhythmical basis)
* Noise (a clear departure from musical intention, intonation and pattern)
* Sound (any interesting sound, but not Music or Noise)
*
* (Percussive sounds are either Noise or Sound depending, but not Music unless a distinct pitch can be heard)
* (Music, Noise and Sound can include short pauses, but no longer continuations of silence)
* (The player may or may not listen to his fellow players)
* (If output is optional the player can as implied choose optional output)
*
*
* The time parameter is set in minutes:seconds, but no exact timing is required
* An IMPORTANT NOTE is that the execution of the instructions continues even though the specified time has not passed.
* i. e. the players is cued by the start of the instruction and not by the finished performance of the instruction
*
*
*************************************************
```

```
**************************************************
* PROCEDURE: Initialize
* CALL: Init
* REQUIRED PARAMETERS: Player, Instrument
* OPTIONAL PARAMETERS: None
* OUTPUT: None
* DESCRIPTION: Initialize the Player with an Instrument, and gets the Player ready to play Instrument.
**************************************************

1 RESET Player
2 Player GOTO Instrument
3 Player PREPARE PLAY


**************************************************
* PROCEDURE: Improvise
* CALL: Impro
* REQUIRED PARAMETERS: Player, Time
* OPTIONAL PARAMETERS: Chords, Scale
* OUTPUT: Music
* DESCRIPTION: Player improvises on the current instrument with the given instruction, or freely.
**************************************************

1 RESET Player
2 IF OPTIONAL Chords
3 Player MEMORIZE Chords
4 END IF
5 IF OPTIONAL Scale
6 Player MEMORIZE Scale
7 END IF
8 Player PLAY Time


**************************************************
* PROCEDURE: Play notes
* CALL: PlayNotes
* REQUIRED PARAMETERS: Player, Pitches [pitch1, pitch2, pitch3...], Time
* OPTIONAL PARAMETERS: None
* OUTPUT: Music
* DESCRIPTION: Player plays the given note in optional octave on the current instrument
**************************************************

1 RESET Player
2 Player MEMORIZE Pitches
3 Player PLAY Time


**************************************************
* PROCEDURE: Perform Noise
* CALL: PerfNoise
* REQUIRED PARAMETERS: Player, Time
* OPTIONAL PARAMETERS: None
* OUTPUT: Noise
* DESCRIPTION: Player performs noise on the current instrument
**************************************************

1 RESET Player
2 Player PLAY Time


**************************************************
█
**************************************************
* PROCEDURE: Perform Sound
* CALL: PerfSound
* REQUIRED PARAMETERS: Player, Time
* OPTIONAL PARAMETERS: None
* OUTPUT: Sound
* DESCRIPTION: Player performs sound on the current instrument
**************************************************

1 RESET Player
2 Player PLAY Time


**************************************************
* PROCEDURE: Sing, move around, play piano or other objects
* CALL: SingMovePlay
* REQUIRED PARAMETERS: Player, Time
* OPTIONAL PARAMETERS: Object
* OUTPUT: Optional
* DESCRIPTION: Player sings, moves around, plays piano or other objects
**************************************************

1 RESET Player
2 IF OPTIONAL Object
3 Init Object
4 END IF
5 Player SURPRISE Time
6 Player SPAM


**************************************************
* PROCEDURE: Create continuation
* CALL: CreaCont
* REQUIRED PARAMETERS: Player, Time
* OPTIONAL PARAMETERS: None
* OUTPUT: Music
* DESCRIPTION: Player makes a musical interpretation or continuation of previous instruction on the current instrument
**************************************************

1 Player PLAY Time

**************************************************
* PROCEDURE: Freak out
* CALL: FreakOut
* REQUIRED PARAMETERS: Player, Time
* OPTIONAL PARAMETERS: None
* OUTPUT: Optional
* DESCRIPTION: Player freakes out in the players own way
**************************************************

1 RESET Player
2 Player SURPRISE Time
```

```
**************************************************
*************************************************
*************************************************
* Read the instructions and
* START HERE!!!
*************************************************
*************************************************
*************************************************


1 IF NONCOMPRENDE JUMP TO 37
2 Init Gelsey vocals
3 Init Andrew double bass
4 Init Paul drums
5 Init Erin saxophones
6 Init Dave clarinet
7 Init Jeff violin
8 Andrew PlayNotes [A] 0:10
9 Erin PlayNotes [C#] 0:10
10 Dave PlayNotes [E] 0:10
11 Jeff PlayNotes [G] 0:10
12 Gelsey PlayNotes [A] 0:10
13 Paul PerfNoise 0:20
14 Andrew PerfNoise 0:10
15 Erin PerfNoise 0:10
16 Dave PerfNoise 0:10
17 Jeff PerfNoise 0:10
18 Gelsey Impro 0:10
19 IF NONINTEREST
20 Gelsey SingMovePlay 0:20
21 END IF
22 Erin PlayNotes [A,C#,E,D#,G,F#] 0:10
23 ASSEMBLY Andrew Dave Jeff
24 Erin PerfSound 0:10
25 Paul FreakOut 0:10
26 Paul SingMovePlay 0:15 assembled objects
27 IF ERROR JUMP TO 37
28 Paul CreaCont 0:10
29 Erin CreaCont 0:10
30 Gelsey CreaCont 0:10
31 Init Andrew piano
32 Init Dave piano
33 Init Jeff piano
34 Andrew impro 0:20
35 Dave impro Am 0:20
36 Jeff impro 0:20
37 ASSEMBLY All Players
```