

Pie Machine 314159

by Helen P. Read

Pie Machine 314159 Source Code

```
SetDirectory [ToFileName [{"FileName"/.NotebookInformation [EvaluationNotebook []] }[[1]]];

startup [] := Module [{text},
  flavors = Flatten [Import ["recipes.xls"]];
  data = Flatten [Import ["data.xls"]];
  audioclip1 = Import ["OldTurkeyBuzzard1.wav"];
  audioclip2 = Import ["OldTurkeyBuzzard2.wav"];
  state = 1;
  SeedRandom [Method -> "MersenneTwister"];
  ding = SoundNote ["F", 1, "Tinklebell"];
  EmitSound [Sound [{Table [SoundNote [m, 1, "Goblins"], {m, RealDigits [N[ $\pi$ , 10], 10] }[[1]]}],
  SoundNote [RealDigits [N[ $\pi$ , 15], 15] }[[1]], 10, "Sweep"],
  ding}]];
  text = Import ["startup.txt", "List"];
  For [i = 1, i <= Length [text], i++, Print [text[[i]]; Pause [10 - 3 (i - 1)]];
];

bake [] := Module [{n, virtualpie},
  n = RandomInteger [{1, Length [flavors] + state * (234 - Length [flavors]) * 30}];
  If [n > Length [flavors], misfire [],
  virtualpie = Framed [Style [flavors[[n]], Bold], FrameStyle -> Red, Background -> LightBlue,
  FrameMargins -> 10];
  EmitSound [Sound [{SoundNote [{"F1", "A2", "C2", "F#3"}, 3, "Metallic"], ding}]];
  flavors = Drop [flavors, {n}];
  Pause [3];
  Return [virtualpie]
];

misfire [] := Module [{n, output},
  n = RandomInteger [106];
  EmitSound [Sound [SoundNote [Range [-12, 12], 1, "Telephone"]]];
  EmitSound [RandomChoice [{2, 1} -> {audioclip1, audioclip2}]];
  Print [Row [{"Error Code ", n, "\nContact System Administrator"}]];
  output = Framed [Style [RandomChoice [data], Bold], FrameStyle -> Red, Background -> Yellow,
  FrameMargins -> 10];
  Print [output];
  Print [Row [Table ["Error ", {i, 5}]]];
];
```

```

seekAndDestroy [proc_String]:=Module[{text},
text=Import[proc<>".txt","List"];
EmitSound[{Sound[{Table[SoundNote[k-20,.1,"Bird"],{k,60}],
Table[SoundNote[40-k,.1,"Bird"],{k,0,60}],Table[SoundNote[27,.1,"Gunshot"],{k,60}],
SoundNote[Range[-12,12],6,"Helicopter"],Table[SoundNote[2k-20,.1,"Bird"],{k,30}],
Table[SoundNote[40-2k,.1,"Bird"],{k,30}],ding}}]}
For[i=1,i<=Length[text],i++,Print[text[[i]];Pause[6]];
state=0;
];

utter[]:=Module[{n,pistring,lastthree,styled,stripped,numberofUtterances},
If[state==1,misfire[],
EmitSound[Sound[{Table[SoundNote[RandomInteger[{-50,50}],3,"Sweep"],{i,5}],
ding}]];
n=RandomInteger[{105,314159}];
pistring=ToString[N[ $\pi$ ,n+2]];

(* grab the last three digits and style them *)
(* then reassemble *)

lastthree=StringTake[pistring,-3];
styled=Style[lastthree,Red,Bold];
stripped=StringDrop[pistring,-3];
numberofUtterances=ToExpression[lastthree];
Print[Row[{"Random integer selected...",n}]];
Pause[5];
Print[Row[{"Computing digits ",n," through ",n+2," of  $\pi$ "}]];
Pause[5];
Print[Row[{stripped,styled}]];
Print["Uttering..."];
Pause[5];
Print[Framed[Row[Table["Pie! ",{i,Range[numberofUtterances]}]],
Background→LightBlue,FrameStyle→Red]];
Print[Row[{"Uttered ",numberofUtterances," times"}]];
];
];

notify[winner_String]:=Module[{ding,string,styled},
ding=Sound[SoundNote["F",.2,"Tinklebell"]];
string=StringJoin[Table[" $\pi$  ",{k,30}]];
styled=Style[string,FontSize→14,FontColor→Purple,FontWeight→Bold];
EmitSound[Table[ding,{k,10}]];
Pause[.5];
EmitSound[audioclip1];
Print[Framed[Row[{styled,
Style["\n\nCongratulations ",FontSize→36,FontColor→Red],
Style[winner<>"!!!",FontSize→36,FontColor→Red],
Style["\n\nAFI Dave's Mom's Virtual Thanksgiving Pie
Virtual-Thanksgiving-Pie-Guessing Grand Champion 2007\n\n",
FontSize→14,FontColor→Purple,FontWeight→Bold],styled}],
FrameStyle→Red,Background→LightBlue,FrameMargins→30]];
];

```

Start the Pie Machine

Execute this first, and answer **Yes** at the initialization prompt. This loads all of the external files, defines the various Pie Machine modules, etc.

```
startup []
```

Begin Virtual Baking

Bake as many virtual pies as you need, until you have a winner. It may be necessary to go into *Sudden Death Pie Overtime* if you do not have a winner during regulation play. Remember that everyone is still in the running and can earn additional points until the threshold score or time limit has been reached.

```
bake []
```

```
bake []
```

```
bake []
```

Seek and Destroy

When you start getting sick of the misfires, execute the seekAndDestroy script to seek out and destroy any Donz processes that have infiltrated the system. There are two different misfire sound effects, so it's worth waiting until you have heard both before you seekAndDestroy.

You can seek and destroy other processes besides Donzes, provided you have an appropriate text file. The syntax is seekAndDestroy["Whatever"] assuming you have a file called Whatever.txt (use the Donz.txt file as a model, and note that the timing of sound effects is best if you have the same number of lines of text as in the Donz.txt file).

```
seekAndDestroy [ "Donz " ]
```

Continue Virtual Baking

Bake additional virtual pies as needed. If you have run the seekAndDestroy script, you won't have any more problems with misfires.

```
bake []
```

Utterances

Run the pie utterances module. Use the result for a Tie-Break if needed. If you are still getting misfires when you try to run the pie utterances, you will need to run the seekAndDestroy script (otherwise you are really doomed).

```
utter []
```

Notify Winner

(This should be self-explanatory.)

```
notify["winner's name goes here"]
```