grainmaker_pd_external

Generated by Doxygen 1.8.17

1 Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

2 File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

3 Data Structure Documentation

3.1 _grainmaker_tilde Struct Reference

Collaboration diagram for _grainmaker_tilde:

Data Fields

- t_object **x_obj**
- t_word ∗ **x_sample**
- int **x_sample_length**
- int **offset**
- int **num_grains**
- int **grain_length**
- t_symbol ∗ **x_arrayname**
- t_float **f**
- grain_scheduler ∗ **x_scheduler**
- t_inlet ∗ **in_offset**
- t_inlet ∗ **in_num_grains**
- t_inlet ∗ **in_grain_length**
- t_outlet ∗ **out**

The documentation for this struct was generated from the following file:

- grainmaker∼.c

3.2 grain Struct Reference

The struct of a grain

```
#include <grain.h>
```

Data Fields

- int **start_sample**
- int **end_sample**
- int **current_sample**
- int **grain_size**

Related Functions

(Note that these are not member functions.)

- grain construct_grain (int sample_pos, int src_sample_length, int offset, int grain_length)

  Creates a grain to be used by grain_scheduler

3.2.1 Detailed Description

The struct of a grain

3.2.2 Friends And Related Function Documentation

3.2.2.1 construct_grain()

```
grain construct_grain (
        int sample_pos,
        int src_sample_length,
        int offset,
        int grain_length )  [related]
```

Creates a grain to be used by grain_scheduler

Parameters

sample_pos	The position of the playhead as defined from outside.
src_sample_length	The length of the source sample @param offset The offset around sample_pos in which grains can be constructed
grain_length	The length of the individual grains The construct_grain function creates grains in an area of the source sample that is defined by sample_pos and offset. It sets all grain variables.

Returns

A grain object

The documentation for this struct was generated from the following file:

- grain.h

3.3 grain_scheduler Struct Reference

A structure for a gain_scheduler object

`#include <grain_scheduler.h>`

Collaboration diagram for grain_scheduler:

Data Fields

- t_word * **src_sample**
- grain * **grains**
- int * **grain_pauses**
- int **src_sample_length**
- int **offset**
- int **num_grains**
- int **current_num_grains**
- int **grain_length**
- int **grain_spread**

Related Functions

(Note that these are not member functions.)

- grain_scheduler * grain_scheduler_new (t_word *src_sample, int src_sample_length)

  Creates a new grain_scheduler object
  This function sets the source sample and its length for the
  grain_scheduler class.

- void grain_scheduler_free (grain_scheduler *x)

  Frees a grain_scheduler object

- void grain_scheduler_set_props (grain_scheduler *x, int offset, int num_grains, int grain_length)

  Sets the properties of the grain_scheduler object as defined by outside input.

- void grain_scheduler_perform (grain_scheduler *x, int sample_pos, t_sample *out)

  Performs the grain creation and playback in realtime

3.3.1 Detailed Description

A structure for a gain_scheduler object

3.3.2 Friends And Related Function Documentation

3.3.2.1 grain_scheduler_free() `void grain_scheduler_free ( grain_scheduler * x ) [related]`

Frees a grain_scheduler object

Parameters

x	My grain_scheduler object The function frees the allocated memory of a grain_scheduler object.

3.3.2.2 grain_scheduler_new()

```
grain_scheduler * grain_scheduler_new (
        t_word * src_sample,
        int src_sample_length )  [related]
```

Creates a new grain_scheduler object
This function sets the source sample and its length for the
grain_scheduler class.

Returns

a pointer to the newly created grain_scheduler object

3.3.2.3 grain_scheduler_perform()

```
void grain_scheduler_perform (
        grain_scheduler * x,
        int sample_pos,
        t_sample * out )  [related]
```

Performs the grain creation and playback in realtime

Parameters

x	My grain_scheduler
sample_pos	The current position of the source sample around which grains are to be created and played back
out	The output vector The function grain_scheduler_perform creates and outputs grains according to user defined properties

3.3.2.4 grain_scheduler_set_props()

```
void grain_scheduler_set_props (
        grain_scheduler * x,
        int offset,
        int num_grains,
        int grain_length )  [related]
```

Sets the properties of the grain_scheduler object as defined by outside input.

Parameters

x	My grain_scheduler object
offset	The offset in which grains can be created
num_grains	The number of grains which are to be created @param grain_length The length of the grains to be created The function grain_scheduler_set_props sets the parameters for grain creation at the beginning of the perform routine. It creates the array in which grains are stored and constructs grains if it is empty or grain size has been increased since the last time the perform routine was executed.

The documentation for this struct was generated from the following file:

- grain_scheduler.h

4 File Documentation

4.1 grain.h File Reference

Object to handle the individual grains of the grainmaker~ object.

Grain includes the methods to initialize and construct individual grains.

This graph shows which files directly or indirectly include this file:

4.2 grain_scheduler.h File Reference

Object to handle the creation, deletion and scheduling of grains

Grain_scheduler manages all grains for the grainmaker~.c pd external. It constructs grains, sets the time between grains, and computes the output.

```
#include <stdio.h>
#include "m_pd.h"
#include "grain.h"
```

Include dependency graph for grain_scheduler.h:

Data Structures

- struct grain_scheduler

  A structure for a gain_scheduler object

Typedefs

- typedef struct grain_scheduler **grain_scheduler**

4.2.1 Detailed Description

Object to handle the creation, deletion and scheduling of grains

Grain_scheduler manages all grains for the grainmaker~.c pd external. It constructs grains, sets the time between grains, and computes the output.

Author

Peter Gorzo, Jonas Koerwer, Claudio Albrecht, Roman Schweikert
Audiocommunication Group, Technical University Berlin
Real-time audio programming in C, SoSe2020
A simple Grain sampler

Index