Psycle Developer Guide

C-Version, Feb 2021 (unfinished)

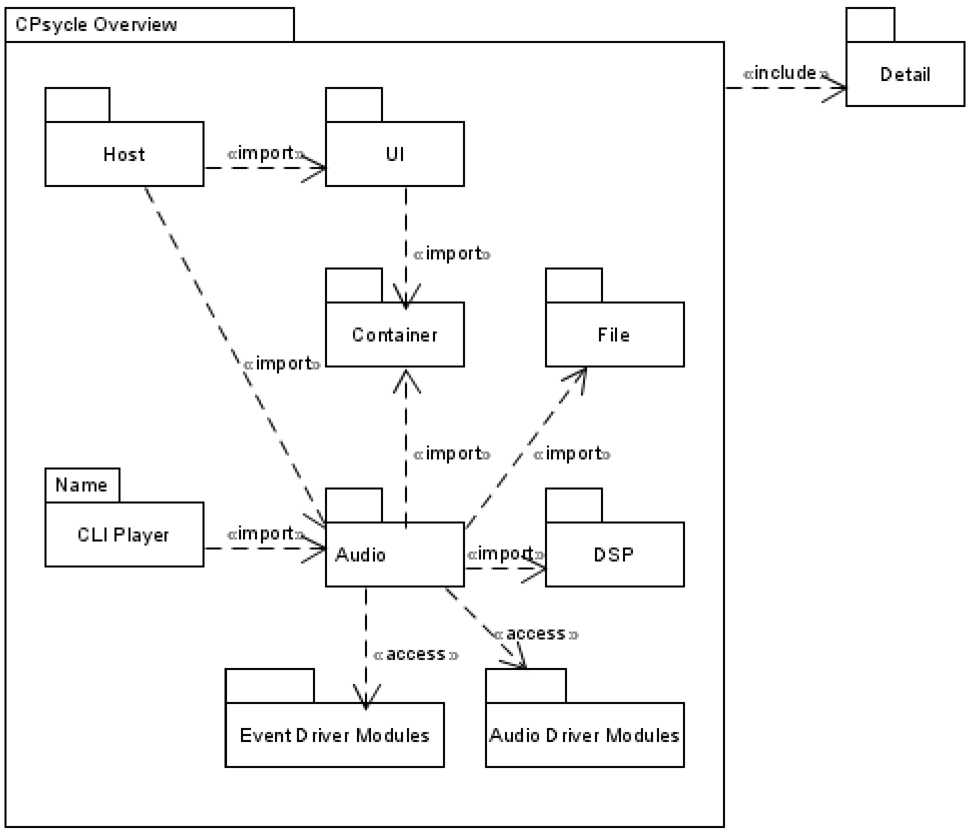
Structure of Psycle

Psycle is divided in two subsystems,

- the audio engine and
- the ui

The helpers dsp, container and file assist these subsystems.

The audio and event drivers will be loaded by the audio engine at runtime.



The Host

The host ist the application, the user starts. The program begins in psycle.c, first, initializing the ui, creating the mainframe and starting the ui event loop and finally leaves the program, if a close event occurs.

Depending on the platform the main entry point differs. Windows uses WinMain as entry point, other toolkits will use int main(int argc, char** argv)

DIVERSALIS, a collection of platform specific defines, is used to detect the platform and needed main entry. To leave the platform dependend startup, psycle_run is called as platform independed main entry point.

psycle_run:

1. The app is added to the enviremont path, that the scilexer module can be found.

Scilexer is an open source editor component, psycle uses since version 1.12.

2. psy_ui_app_init(&app, psy_ui_DARKTHEME, instance);

The ui is initialized with a darktheme. (psy_ui_LIGHTHEME starts a light theme).

3. mainframe = (MainFrame*)malloc(sizeof(MainFrame));

Mainframe is found in mainframe.c and is the composite of all views psycle will present to the user. Workspace holds the project song and configurations.

4. psy_ui_component_showstate(&mainframe->component, SW_MAXIMIZE);

Psycle is displayed.

5. Now the app main loop is started:

err = psy_ui_app_run(&app);

The main loop waits for an event (mouse, keyevent..) and triggers callbacks to the host. In case of a close event the main loop will be terminated.

6. The heap storage of the mainframe is cleaned up and the env path is restored

7. The app returns the status to the operation system and the process is terminated.

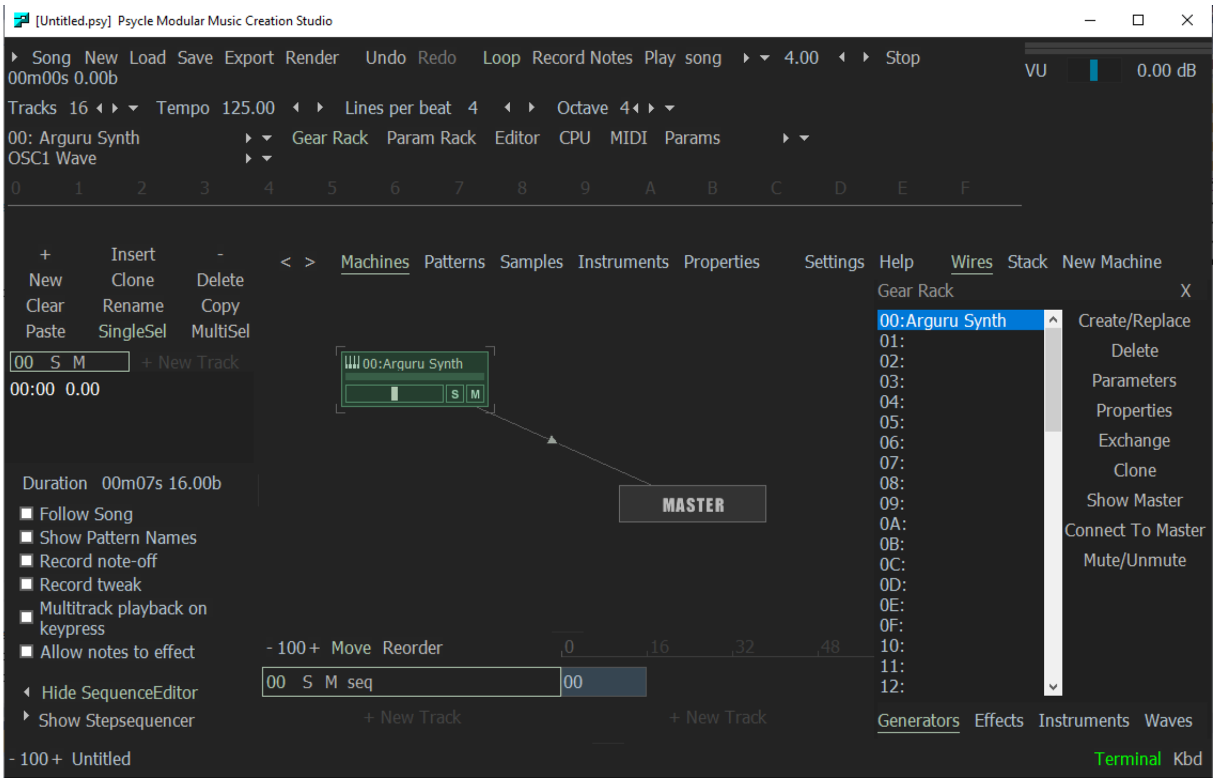
The Mainframe

The mainframe consists oft two basic systems:

The ui views and the workspace.

1. Machines, Patterns, Samples, Instruments, Help, SequencerView, and more …
2. The workspace contains the currenct song, owns the audio players and the configuration files.

Psycle is mainview orientated decorated with sidebars. The mainview structure is realized with a psy_ui_Notebook, that allows to switch components. Tabbar controls the notebook and offers the user tabs to switch it.



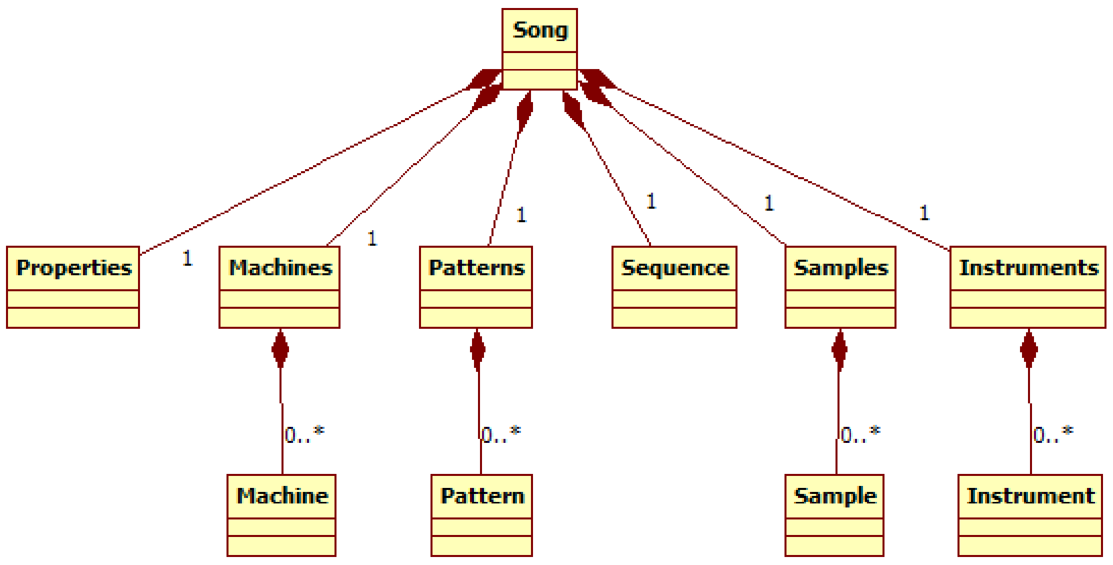
Workspace

The workspace connects the player with the psycle host ui and configures both

```
psy_audio_MachineCallback
      ^
    Workspace
        <>---- PsycleConfig;              host
        <>---- ViewHistory
        <>---- psy_audio_Player;          audio imports
        <>---- psy_audio_MachineFactory
        <>---- psy_audio_PluginCatcher
        <>---- psy_audio_Song
```

The Song

psy_audio_Song hold everything comprising a "tracker module", this include patterns, pattern sequence, machines and their initial parameters and coordinates, wavetables, ...



Song hold everything comprising a "tracker module", this include
patterns, sequence, machines, samples(wavetables), and instruments

The Player

The player controls the audio drivers. It loads a driver and has a work callback. The audio driver will call them if the soundcard buffer needs to be filled. Usually the player work callback will be in the audio driver thread (different than the ui thread).

To fill the buffer the player has a reference to a song. With help oft the song, the player has a helper, psy_audio_Sequencer, that processes the patterns oft the song. The player will then work the machines with the current events, the sequencer has selected. psy_audio_Machines compute the machine path determined by psy_audio_Connections, the wires of the Machines, from their leafs to the Master.