Changelog

2022-07-08

- Added functionality to adjust the opacity of background image, buttons, piano keys.

2022-07-06

- Added functionality to display note names on piano keys.

2022-06-15

- Added choose MIDI device functionality, now you can right click on `MIDI KEYBOARD` button to open the choose MIDI device window, where you can choose which MIDI device you want to use in Ideal Piano. The functionality of pressing keys to show current device is removed.
- The functionality of detect current key you are playing by music analysis algorithm is under experimental development, the algorithm will show you some most possible keys you are currently playing, the accuracy of the algorithm needs to be improved.

2022-05-30

- Switch from tkinter to PyQt5 for the browse window and settings window, since on macOS there are always plenty of bugs when pyglet and tkinter work together, and tkinter is not that powerful for a GUI toolkit compares to PyQt5. After completely rewriting the codes for the browse window and settings window for PyQt5, I find that the functional logic and framework of PyQt5 is actually a lot more easier and flexible than tkinter, and there are no longer bugs appear on macOS when pyglet and PyQt5 works together.
- For the browse window, the interval functionality is removed, since it is not that useful actually.
- Retested the MIDI keyboard mode with a MIDI keyboard and a sustain pedal, fixed all of the bugs that appeared during the testing.
- Remove the change settings standalone executable, now the change settings functionality is built into the software, you can click `Settings` button to open the change settings window.

2022-05-28

- Added functionality to show notes and chord types in sharp or flat accidentals, you can choose the accidentals by setting the parameter `show_chord_accidentals` to `'sharp'` or `'flat'` in the settings. The default value is `'sharp'`.
- Added `Settings` button to the main screen, you can click it to open the settings.
- Added functionality to show chord details on the main screen, you can open it by setting the parameter `show_chord_details` to `True` in the settings.

2022-05-01

- Added drag and drop files functionality, now you can drag and drop local image files onto the main window to change the background image, and drag and drop MIDI files onto the main window to load the MIDI files to play. (drag and drop functionality is currently not supported for macOS)

2021-12-12

- Completely refactored the whole code of Ideal Piano, mainly in code structures and readability, since the original code of Ideal Piano is pretty messy to be honest. Now classes are used as namespaces to differentiate different part of the piano engine, and other modules imported outside are used as namespaces as much as possible, to avoid naming conflicts of functions and variables between different packages, and improve readability.
- The name of the settings file changed from `config.py` to `piano_config.py`.
- The code of Ideal Piano was improved to be the same under Windows, Linux and macOS, which reduces the work to make changes to 3 different versions when adding new features to Ideal Piano.

2021-12-08

- I installed another Linux virtual machine which runs Ubuntu 20.04, and try to install fluidsynth on it using apt, unlike Ubuntu 18.04, the apt install fluidsynth 2.1.1, which is a much newer version of fluidsynth. And then I try to test the pause and unpause functionalities of this version of fluidsynth on Ubuntu 20.0.4, and it works perfectly. This means that the codes of Windows version can actually run on Ubuntu 20.04 with almost no changes with a Linux executable.

  So since this issue of pause and unpause functionalities are different on different versions of Linux distributions, I will consider adding an extra setting parameter `render_as_audio` to the Linux version, the users could switch between using the rendered audio as playback when playing MIDI files or using fluidsynth as the MIDI player by setting this parameter to True or False.

2021-12-07

- The macOS version is ready, with some tweaks I finally achieve making tkinter works with pyglet with no problem. However, the pygame's mixer cannot pause the MIDI files when playing on macOS, to be more specifically, the function `pygame.mixer.music.pause()` does not work, this issue only happens on macOS, for Windows and Linux it works. I tried the newest pygame version 2.1.0 at this time, and 2.0.0, 2.0.1, none of them works. This is not a problem that causes when pygame is working with pyglet and tkinter, because I test the pause function when playing MIDI files in another program where only pygame is used, and get the same result.

  So before pygame's developers fix this bug, for the macOS version, the default settings of MIDI file playing which uses pygame's mixer cannot pause MIDI files when playing. If you want to pause and unpause MIDI files when playing in Ideal Piano using macOS version currently, you can switch to use fluidsynth to play MIDI files in Ideal Piano by changing `use_soundfont` in the settings file to True, and this requires you to install fluidsynth on macOS. It is pretty easy, you can use homebrew to install fluidsynth, run `brew install`

`fluidsynth` and wait for fluidsynth to be installed on your computer. You can also use different SoundFont files to play MIDI files using fluidsynth.

2021-12-05

- The keyboard module is removed, now the detection of computer keyboard in this software is handled by pyglet, this update is applied to both Windows and Linux version. This improvement has many benefits.

  Firstly, pyglet is the main library for creating this software, so since it supports reading keyboard input, it is not necessary to use another python library to detect key press.

  Secondly, the python keyboard library has a problem that it detect key press globally, which means when you are entering computer keyboard playing mode, even if you are not focused on the screen of the software, for example if you are minimized the window and typing in another software, the keyboard module still detect the key press and generate sounds of piano notes like when you are using it. Using pyglet to detect key press won't have this problem, the key presses are detected only if you are focused on the screen of the software.

  Thirdly, the Linux version need to use sudo to open because the keyboard module requires root priviledge, now you can double click on the executable and open the software. The macOS version, which is under developement, requires sudo for the same reason, now at least improve at opening a little bit.

- Improve the import of musicpy and browse, which makes the code much safer in terms of namespaces of functions and variables.

2021-12-03

- After the updates of sf2_loader package, now Ideal Piano is able to implement a SoundFont player in itself, which allows playing MIDI file using SoundFont files without pre-rendering to audio, and could pause, unpause, stop at real time. This makes playing MIDI files using SoundFont files a lot faster to startup, since the MIDI data is directly sent to the player to manipulate at real time.

- **Note: This update is currently only for Windows version**, since on Linux the latest fluidsynth version installed by `apt-get` is 1.9.1 (I tested Linux version on Ubuntu 18.04), which is actually an old version from 2018, the latest version of fluidsynth at this time is 2.2.4, this old version does not have many important features that are introduced in newer versions of fluidsynth, which are used a lot in sf2_loader and also in Ideal Piano, such as `fluid_player_seek`.

  Many functions of fluidsynth player such as `fluid_player_stop` and `fluid_player_play` also behave differently from the newer version, for example, in version 2.2.4, if you stop the player, the sound will pause almost immediately, but in the latest version that is installed by `apt-get` on Ubuntu (which is 1.9.1) this will cause the current playing notes last playing forever, which is very annoying (which means that in the newer version the developers have already fixed this bug).

  Well, I can send all sounds off events to all of the channels to stop the sound, which works, but the most annoying issues comes from when you try to unpause the player, in the version 2.2.4 you can just call `fluid_player_play` again and the player will continue to play the MIDI files with almost no waiting time, but in the 1.9.1 version on Ubuntu this will cause a

very huge waiting time for the player to continue playing, and it is different for each time, which ranges from about 3 seconds to like 30 seconds, during the waiting time, the note bars are already continue moving a lot, so the sounds and the note bars will be totally out of sync. This makes the pause and unpause function of MIDI file playing in Ideal Piano using fluidsynth on Linux totally unmanipulable.

I try to use different audio driver on Linux, including alsa, pulseaudio and some other available ones out there, but they make no differences when trying to pause and unpause. I struggle to find a method to install a newer version of fluidsynth on Ubuntu like 2.1.0 or 2.2.4, but currently cannot find one (maybe I will need to build from source). For Windows version, I have the newest version 2.2.4 of fluidsynth binaries inside the sf2_loader pacakage, so it works fine.

So as a conclusion, currently the Linux version will still use rendered audio to play when you choose SoundFont files to play MIDI files, and the config parameter `use_soundfont` is still present. In other words, for Linux version you can ignore this update for now. In the future, I will try to find out a solution that fixes the pause and unpause issues for the fluidsynth version 1.9.1 on Linux (actually this is the only issue that matters, other functionalities works fine as in Windows), maybe I will try to build from source to get the latest version of fluidsynth that works on Linux, or install a newer version from somewhere.

2021-11-08

- Improved handling of piano pedals in MIDI keyboard playing mode, now supports three different types of piano pedals in MIDI CC64, CC66, CC67.
- Add the function of displaying another color when the note played is released when the piano pedal is depressed.
- Changing the way the notes are playing to the end to fade out, which improves the listening experience when playing the notes.

2021-11-07

- Added the display of border of note bar and piano keyboard, which makes the demonstration effect more elegant, the border width and border color could be adjusted in the config file.

2021-11-01

- If you are using SoundFont to play, you can change the instruments by changing preset number and bank number when playing using the combination of keys (when config key is left `Ctrl` by default):

  `Ctrl + 1` (previous preset)

  `Ctrl + 2` (next preset)

  `Ctrl + 3` (previous bank)

  `Ctrl + 4` (next bank)

2021-10-31

- Now you can load SoundFont files to play MIDI files, in the config file, please set `use_soundfont` to True, `play_as_midi` to True, and then set the file path of SoundFont files using `sf2_path`.
  If you are using SoundFont files, Ideal Piano will render the MIDI files to audio data internally using [sf2_loader](#), then use pygame to play.
  Now you can also use SoundFont files to play using computer keyboard or MIDI keyboard, in the config file, please set `play_use_soundfont_file` to True, you can customize choosing the instruments in the SoundFont files and the playing note's duration, volume using a series of parameters such as `bank`, `preset`.
- Use `Ctrl + S` to open the change settings window, use `Ctrl + R` to reload the changed settings (when config key is `Ctrl` by default). If you press the `save` button on the change settings window or press `Ctrl + S` to save the settings, the settings will automatically reload after you close the change settings window.

2021-10-28

- Improved compatibility with Linux, now works fine on Ubuntu 18.04.5 in a virtual machine. I will try to find time to adapt it to macOS in the next few months.

2020-12-27

- Added a way to view the current midi device connection of your computer in midi keyboard mode, and to see which id corresponds to the name of your instrument if your midi keyboard's midi_device_id is not connected. And it will show you the error message (the specific reason why you can't connect). After clicking the button to enter the midi keyboard mode, press the shift key on the computer keyboard to display the id of the current computer default midi device, and the names of the midi devices with ids 0 to 9 respectively, one of which should be your instrument. Then if your midi keyboard is not connected, a specific error message will also be listed at the end of the display. You can turn off the display by pressing the ctrl key.
- Now I have also improved the code logic of the software for midi piano reading a lot, now there is no need to open the midi keyboard first and then open the software, you can open the software first and then connect the midi keyboard and open the midi keyboard and then click the midi keyboard mode button to identify the midi keyboard (in the case that the midi_device_id is on your (in the case of the instrument), turn off the midi keyboard in the middle, then open it again and click the back button and then re-enter the midi keyboard mode, the software will re-identify the midi keyboard, then you can continue to play (before you need to re-open the software once again, now you do not need to)

2020-08-20

- Redesigned the piano structure, before it was a whole piano picture on the screen, now it adds the draw piano mode, set draw_piano_keys to True in the settings file to enter this mode. In draw_piano mode, the corresponding keys will light up when the midi keyboard is playing or the computer keyboard is playing, including when the midi file is played in drop note mode, the notes will also light up when they fall on the keys. The piano is drawn using black and white keys that directly follow the structure of the piano's 88 keys, according to settable parameters, and each key can change color.

Underneath the drawing of the 88 keys there is a black background image, which is mainly used to show the gaps between the piano keys (for filling). Instructions for setting the parameters of the piano keyboard drawing will be written in the manual for setting the parameters. You can turn off the note mode (note_mode is set to a value other than dots, bars, bars drop) and just turn on the piano drawing mode, the corresponding piano keys will light up when you play, and the current note will light up when you play the midi file. It is also possible to use any of the note modes and turn on Draw Piano mode.

2020-08-17

- Added a mode to play directly as a midi file audio source, set play_as_midi to True in the settings file to enter this mode. When you select merge all tracks, the notes of all tracks will be merged for demo, and the sound is played directly from the midi file inside the software. If one of the tracks is selected, the selected track is temporarily written to a midi file and then played.
- The program to modify the settings adds buttons to input Boolean values (True, False) directly by clicking on them, making it easy to modify the parameters of the Boolean values.

2020-08-13

- Added note bar (drop) mode, available only in the mode of playing midi files, to use note bar (drop) mode, just set the value of parameter note_mode to 'bars drop'.
- If the parameter sort_invisible is set to True, the sorted content of the chord will not be displayed if the chord is of sort type, only the chord name will be displayed.

2020-08-12

- Real-time chord analysis algorithm adds poly_chord_first parameter, which is a compound chord mode, so that when the chord is very complex and the time needed to judge it will cause a lag, the chord can be split into two chords and judged in the form of a compound chord (or a chord with a lowest note under it), which can greatly improve the efficiency of chord judging, especially for This mode is especially suitable for jazz music.
- The parameter get_off_drums has been added to play midi files with the option to remove the drum tracks from the midi file. Set it to True to remove the drum tracks when reading the midi file to avoid being disturbed by the drum tracks when analyzing the chords in real time.

2020-08-11

- Added the option to merge all tracks when playing a midi file.
- Fixed the bug that the file path of the background image and piano image sometimes could not be found.
- Added the parameter show_chord that allows you to choose whether to display chords in real-time analysis.

2020-08-10

- Newly made a program to change the setting parameters, with a search bar function, so you can quickly find the parameters to change.
- I wrote a software parameter manual, explaining the meaning of each parameter, so that you can modify it easily.
- The original default font, Comic Sans MS, was not very formal, so I changed the default font to Cambria, a very neat and formal English font, changed the default font size from 20 to 23, and made the button size bigger (the new parameter button_resize_num was added, which is a scaling factor for the button size) Because the original button size feels a bit small.
- Newly added the option to merge all the tracks of a midi file and play it when playing a midi file.
- P.S. I believe that Cambria is the official font that most people recognize, but if not, you can change it to the font you want in the settings file.

2020-08-09

- Note bar mode has been added. Previously, only the note points were displayed on the piano keys, but this time the note bar mode is a bit more appreciated. The notes appear and rise from the keys, both when playing on your own and when playing midi files. If you press and hold, the note bar will keep elongating, and when you release it, the note bar will float upwards. When playing a midi file, the relative length of the notes is calculated directly from the length of each note. And the strength of the notes corresponds to the transparency of the notes, (this function can be set in the corresponding parameters in the settings file on or off) and then you can also choose to appear monochrome or random arbitrary color, the effect is still good. Now the mode of the note bar (bars) exists as two modes together with the previous note dots (dots), you can set which mode you want in the settings file. The position where the notes appear, the length and width of the notes, the number of steps the notes go up each time (how many pixels they go), the color of the notes, single color display or random color display, the default transparency of the notes, the speed at which the notes elongate, etc. can all be set in config.py.
- I'm going to make all the parameters in config.py into change_settings.py, so that all these parameters can be modified.

2020-08-08

- The default chord display alignment mode has been updated to left-aligned, replacing the original centered display, which is more comfortable to look at and avoids the
It is more comfortable to look at and avoids the situation where the chord name length keeps changing in real time analysis, which makes it hard for the eyes to keep up.
- Since my software is in English, I made a Chinese patch package, which includes: the buttons of the interface, the names of the displayed chord types, (the original chords will keep the other English names, the first one is the Chinese name), the monophonic and interval, the interface for selecting midi files and each setting item. The procedure for changing the settings. The download link for the Chinese patch installer is in README.