

# Welcome New Charter!

holorhysm developer teamにようこそ！

このファイルは、新しく譜面制作者としてholorhysm developer teamに加入した人向けの、制作準備の手順を示したドキュメントです。

まずはこのドキュメントの手順に従って譜面制作環境を整えてください。

## 注意

- わからないことがあったときは、Discordサーバーの「#雑90\_婆さんや飯はまだか」まで。なるべく回答します

## Visual Studio Codeのインストールと設定

Visual Studio Codeは、プログラムを編集したりするための「コードエディター」です。  
譜面制作においても、譜面の保存やテストプレイなどのために使用します。

### Windows (VSCode設定)

1. VSCode本体と"Japanese Language Pack for Visual Studio Code"をインストールします。
  - [龍谷大学 理工学研究科の授業資料](#)を参考にしてください。
2. 以下の拡張機能をインストールします。
  - "Japanese Language Pack for Visual Studio Code" (Microsoft)
    - `ms-ceintl.vscode-language-pack-ja` で検索
    - 上記参考資料を最後まで進めるとインストールされているはずです。
  - "EditorConfig for VS Code" (EditorConfig.org)
    - `EditorConfig.EditorConfig` で検索
  - "Live Server" (Ritwick Dey)
    - `ritwickdey.LiveServer` で検索

### macOS (VSCode設定)

1. VSCode本体と"Japanese Language Pack for Visual Studio Code"をインストールします。
  - [龍谷大学 理工学研究科の授業資料](#)を参考にしてください。
2. 以下の拡張機能をインストールします。
  - "Japanese Language Pack for Visual Studio Code" (Microsoft)
    - `ms-ceintl.vscode-language-pack-ja` で検索
    - 上記参考資料を最後まで進めるとインストールされているはずです。
  - "EditorConfig for VS Code" (EditorConfig.org)
    - `EditorConfig.EditorConfig` で検索
  - "Live Server" (Ritwick Dey)
    - `ritwickdey.LiveServer` で検索

## GitHubアカウントの作成と設定

1. 普段使っているブラウザで、<https://github.com/>にアクセスします。
2. ページ右上の「サインアップ」をクリックします。

3. なんか始まるので画面の指示通り入力して進めていきます。
  - 英語が読めない？そんなあなたに[Google翻訳](#)
    - enter : (動)入力する
    - receive : (動)受け取る
    - verify : (動)検証する
  - ユーザーネームは覚えておいてください。
4. ユーザーネームを綾坂ことに伝えます。
  - Discord鯖でメンションしてください。( @綾坂こと ユーザーネーム〇〇です、GitHub招待お願いします など)
5. 綾坂側の作業を待ちます。
6. 登録に使ったメールアドレスに届いているメールを確認します。
  - "invited you to"みたいな文言が入ってると思います
  - そのメールの"join holorhysm"的なボタンを押してください。
7. <https://github.com/holorhysm>が開けることを確認してください。
  - 404 Not Found もしくは 403 Forbidden といったエラーが出たら「#雑90\_婆さんや飯はまだか」まで。

## Gitをインストール

### Windows (Gitインストール)

1. <https://git-scm.com/downloads>にアクセスします。
2. 「**64-bit Git for Windows Setup**」をクリックします。
3. ダウンロードされたファイルを開いて、画面の指示に従ってインストールを進めます。
  - "Choosing the default editor used by Git"はVisual Studio Codeを探して選択してください。
  - "Adjusting the name of the initial branch in new repositories"は"Let Git Decide"でいいです。
4. Gitのインストールが終了したら、Visual Studio Codeを再起動します。

### macOS (Gitインストール)

1. homebrewをインストールします。
  - <https://brew.sh/ja/>を開きます。
  - "インストール"の下にあるコマンドをコピーして、ターミナル.appで実行します。
  - "y/n"みたいなことを聞かれると思いますが、"y"を入力しないと進まないと思います。
2. ターミナル.appで `brew install git` を実行します。
3. Gitのインストールが終了したら、Visual Studio Codeを再起動します。

## VSCodeでGitHubにログイン

1. Visual Studio Code左下の歯車マークの直上にある人マークをクリックします。
2. 「サインインして設定を同期する」をクリックします。
  - 「バックアップと同期の設定」かも。その場合は全部チェックして「サインイン」をクリック。
  - GitHubとMicrosoftのどちらでサインインするかを聞かれたら「GitHubでサインイン」をクリック。
3. たぶんブラウザでGitHubのログイン画面が開くので、ログインします。
  - リダイレクトを許可するか聞かれたら許可します。
4. (1)でクリックした人マークをクリックし、「[GitHubのユーザー名] (GitHub)」という選択肢があることを確認します。
5. Visual Studio Codeで「ターミナル」を開きます。
  - Windowsなら `Ctrl + Shift + @`、macOSなら `control + ⌘ shift + ^ (へ)` で開けます。
  - ウィンドウ上部の「ターミナル(T)」をクリックして、「新しいタイミなる」をクリックすると開けます。
6. 開いたターミナルで、`git config --global user.email "メールアドレス"` を実行します。
  - メールアドレスの部分には、GitHubアカウントの登録で使ったアドレスを入力します。

- 例 : GitHubの登録に `test@example.com` を使ったなら、`git config --global user.email "test@example.com"` を実行
7. 同様に、`git config --global user.name "ユーザー名"` を実行します。
    - ユーザー名の部分には、GitHubアカウントのユーザー名を入力します。
    - 例 : ユーザー名が `octocat` なら、`git config --global user.name "octocat"` を実行
    - ここまで終わったらターミナル画面は閉じてOKです。
      - ターミナル画面右上の「×」をクリックすると閉じます。

## リポジトリのクローン

1. Visual Studio Codeで「ソース管理」画面を開きます。
  - 画面左に並んでいるいくつかのアイコンのうち、小さい丸3つを線が繋いでるアイコンをクリックすると開けます。
  - Windowsなら `Ctrl + Shift + G`、macOSなら `control + ⌘shift + G` のショートカットでも開けます。
2. 「リポジトリの複製」をクリックします。
3. 「GitHubから複製」をクリックします。
  - "拡張機能 'GitHub' が GitHub を使用してサインインしようとしています。"が出たら「許可」をクリック。
4. 「holorhysm/core」をクリックします。
5. フォルダ選択画面が出るので、任意のフォルダを選択します。
  - 何もわからなければ、デスクトップフォルダを選択しておけば困り果てることはないと思います。
6. 少し待つと、「クローンしたリポジトリを開きますか?」と出るので、「開く」をクリックします。
7. 「このフォルダー内のファイルの作成者を信頼しますか?」と出たら、「はい、作成者を信頼します」をクリックします。
8. なんかいっぱいファイルがあること、左下の方に「main」と書かれていることを確認します。

## 譜面制作をはじめるとき(ファイルの作成)

1. 左下の「main」をクリックします。
2. 「origin/song\_(その楽曲を表す文字列)」をクリックします。
  - (その楽曲を表す文字列) : Shiny Smily Story (2022 ver.)なら `SSStory2022_IdolPJ` など。
3. 左下の表示が「song\_(その楽曲を表す文字列)」になったことを確認して、そこをもう一度クリックします。
4. 「新しいブランチの作成…」をクリックします。
5. 「chart\_(その楽曲を表す文字列)-(譜面難易度)」を入力してEnterを押します。
  - (譜面難易度) : Normal、Hardなど。
6. 左下の表示が「chart\_(その楽曲を表す文字列)-(譜面難易度)」になったことを確認します。
7. Visual Studio Codeで「ソース管理」画面を開きます。
8. 「ブランチの発行」をクリックします。
  - クリックして数秒後、Discordの「#程22\_github-notify」に通知が来るはずです。
9. Visual Studio Codeで「エクスプローラー」画面を開きます。
  - 画面左に並んでいるいくつかのアイコンのうち、2枚の書類が重なったアイコンをクリックすると開けます。
  - Windowsなら、`Ctrl + Shift + E` のショートカットでも開けます。
  - macOSなら、`⌘command + ⌘shift + E` のショートカットでも開けます。
10. chartsフォルダの中の(その楽曲を表す文字列)フォルダを右クリックします。
11. 「新しいファイル…」をクリックします。
12. 「(譜面難易度).jsonc」と入力してEnterを押します。
  - この「(譜面難易度).jsonc」が譜面ファイルになります。
13. 後述するテキスト(譜面ファイルのテンプレート)をコピーして、開いているエディタに貼り付けます。
14. ファイルを保存します。
  - Windowsなら `Ctrl + S`、macOSなら `⌘command + S` で保存できます。ショートカットキーに慣れると便利です。
15. Visual Studio Codeで「ソース管理」画面を開きます。
16. メッセージ欄に適当な文言を入力します。("とりあえずファイル作成"など)

17. 「コミット」の右側にある「v」をクリックします。
18. 「コミットして同期」をクリックします。
  - なんかできたら「Don't show again」みたいなやつを押しておくと思います。

## 譜面ファイルのテンプレート

```
{
  "bpm": "120",
  "beats": "[4, 4]",
  "soflan": "1.00",
  "offset": 0.00,
  "notes": [
    { "type": "push", "where": [-3.00, 3.00], "when": [ 1, 0, 16], "speed": 1.00, },
  ],
  "decorator": [
    {
      "color": "pseudolong",
      "start": { "where": [-3.00, 3.00], "when": [ 1, 0, 16], },
      "end": { "where": [ 0.00, 0.00], "when": [ 2, 0, 16], },
      "easing": { "left": "Linear_InOut", "right": "Linear_InOut", }, "speed": 1.00,
    },
  ],
  "function": [],
}
```

## 譜面を確認する

- holorhysmの譜面は専用のビューワーがあります。譜面の確認などに使ってください。
  - [holorhysm editor \(JSONC直接入力ver.\)](#)
  - ページ左側の入力欄に譜面ファイル(.jsonc)の内容をコピペすると譜面が確認できます。
  - [♪]をクリックして楽曲の音声ファイルを選択すると、譜面が音楽に合わせて自動スクロールします。
  - その右にある4つのアイコンは譜面編集の便利ツールへのリンクです。
    - 2024-08-04現在、リンク先のツールはバグがあるので使用非推奨。
  - 1世代前のeditorも一応残しています。こちらは各項目の入力欄があります。
    - [holorhysm editor \(各項目入力ver.\)](#)
    - こちらはバグの修正以外触らないつもりです。
    - ノーツ数が増えてくると重くなるので非推奨です。

## 譜面を編集する

holorhysmの譜面は、拡張子こそ .jsonc ですが中身はテキストファイルです。

お好きなテキストエディタで編集できます。

別紙「譜面ファイルの仕様」「譜面レギュレーション(暫定)」を参考に自由に編集してください。

- Visual Studio Code上で編集すると、タイプミスなどが見つけやすいので便利です。
- holorhysm editor上で編集すると、見た目を確認しながら編集できるので便利です。

## 譜面編集用便利ツール

譜面編集の省力化のために、よくやる操作を拙作「JS\_magic-circle」でできるようにしています。

『プログラミングは現代の魔法なので、呪文詠唱する魔法円的なものがあったらいいと思うんだ』  
綾坂こと制作のJavaScript魔法詠唱ツールです(?)  
([https://github.com/AyaExpTech/JS\\_magic-circle](https://github.com/AyaExpTech/JS_magic-circle))

使い方は「#雑90\_婆さんや飯はまだか」とかで聞いてください  
とりあえずリンクだけ置いときます

- [ノーツずらしコピー](#)
- [ノーツ左右反転](#)
- [デコレーターずらしコピー](#)
- [デコレーター左右反転](#)
- [疑似ロング生成\(始点基準\)](#)
- [疑似ロング生成\(楽曲基準\)](#)

## テストプレイ

VSCodeの拡張機能「Live Server」を使うと、PCや実機で譜面のテストプレイができます。

手順の説明が面倒なのでとりあえず流れだけ書いておきます。  
ヘルプは「#雑90\_婆さんや飯はまだか」などで受け付けます！！

## ファイルを準備する

- 本体プログラムを最新の状態に更新します
  - mainブランチからchartブランチにpullすればいいはず
- 譜面DBファイル( /charts/\_list.jsonc )に楽曲・譜面の記載があることを確認します
  - たぶん譜面制作を開始する前に綾坂が用意してるはず……
- 楽曲音声ファイルがあることを確認します
  - たぶん譜面制作を開始する前に綾坂が用意してるはず……

## PCでテストする

- VSCode右下にある「Go Live」を押します
- ちょっと待つとデフォルト設定しているブラウザでテスト画面が開きます
- 楽曲一覧の中に制作中の楽曲・譜面の選択肢が入ってるはず
- Chrome DevToolsはスマホ画面のエミュレートができるので、それを使うとより実機に近い形で確認できるはず
  - [これ](#)

## スマホでテストする

たぶんテストプレイは実機でやりたいと思うので、その流れも一応。

- (両方) PCとスマホを同じネットワークに接続します
  - Wi-Fi環境なら同じルーターに接続
  - 有線LAN環境の人は知らねえ！がんばれ！！
- (PC) PCのローカルIPアドレスを確認します
  - ほとんどの人は 192.168.xxx.xxx の形のはずです
    - 192.168.xxx.xxx の他に、10.xxx.xxx.xxx、172.(16~31).xxx.xxx も正しいローカルIPアドレスです
  - Windowsならipconfigコマンド、macOSは設定画面から確認できると思います
  - 参考：[CMAN.jpの記事](#)
  - ローカルIPアドレス固定しないと厄介かも。固定方法はルーターの機種によってまちまち。
- (PC) Live ServerのHTTPS設定を有効にします

- [@ma2shita氏の記事](#)などを参考に……
  - macOSの人は証明書はmkcertで作りましょう、[こちらの記事](#)参照
    - `brew install mkcert, mkcert -install, cd ./Desktop/, mkcert (ローカルIPアドレス)`
      - 4コマンド目は記事内では `mkcert localhost` ですが `mkcert (ローカルIPアドレス)` に読み替えてください
    - `key.pem` で終わる方のファイルは流出すると悪者が攻撃し放題になるので要注意
  - この手順いらないかもしれないけど、必要だった気がするので書くだけ書いておきます
- (PC) VSCode右下にある「Go Live」を押します
- (PC) ちょっと待つとデフォルト設定しているブラウザでテスト画面が開きます
- (スマホ) holorhysm推奨環境に指定されているブラウザで、そのIPアドレスの5500番ポートにアクセスします
  - ローカルIPアドレスが 192.168.0.128 なら、アドレスバーに `https://192.168.0.128:5500/` と入力します
  - なんか危険とか言われるかもしれませんが、「詳細を表示」→「〇〇にアクセス」とかを押してください
  - おそらくこれでテスト画面が開くので、あとはPCでのテストと同様に。