

# 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 + ^(^)` 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、「新しいターミナル」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

2024-09-07 Update

- クローンするリポジトリを `holorhysm/core` から `holorhysm/charts` に変更しました

- 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。
- 「リポジトリの複製」をクリックします。
- 「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。
- `holorhysm/charts` をクリックします。
- 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。
- 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。
- 「このフォルダ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。
- 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)

2024-09-07 Update

- リポジトリ分離に伴い、ブランチ名のルールを `chart_MUSIC-LABEL` から `new_N` に変更し、またsongブランチを廃止しました
- リポジトリ分離に伴い、各譜面に対応するIssueの制作手順を追加しました

- [このリンク](#) から、GitHub上で譜面Issueを作成します。
  - Title : 『楽曲名』[難易度ラベル] (例 : 『Shiny Smily Story (2022 ver.)』[Normal])
  - 追加区分 : イベント楽曲はEvent、関係なければSingle、それ以外の場合はChart-Leader(暫定でProducerが兼任)から指示あり
  - 納期 : Chart-Leader(暫定でProducerが兼任)から指示あり
  - Labels : 譜面ラベルに合わせて"Easy", "Normal", "Hard", "Extra"のうち1つを選択
  - Assignees : "assign yourself"を押す
  - 上記5項目を埋めたら、[Submit New Issue]をクリック
  - Issue番号を確認します (タイトルの右に #1 のように書いてある)
- (以降はVSCodeで操作) 左下の「main」をクリックします。
- 「新しいブランチの作成…」をクリックします。
- 「new\_(Issue番号)」を入力してEnterを押します。
  - 例 : Issue番号が#6 → new\_6
- 左下の表示が「new\_(Issue番号)」になったことを確認します。
- Visual Studio Codeで「ソース管理」画面を開きます。
- 「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。
- 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。
- `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。
  - なかったらそのフォルダを作ってください
- 「新しいファイル…」をクリックします。
- 「(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。
- 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。
- 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。
- Visual Studio Codeで「ソース管理」画面を開きます。
- 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)

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

## 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

2024-09-07 Update

- バグ修正サポート終了に伴い、1世代前のエディター(feature\_24)の記述を消しました
- 譜面編集便利ツール(JS\_magic-circle)の更新が完了したので非推奨の記述を削除
- 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。
  - その右にあるアイコン群は譜面編集の便利ツールへのリンクです。

## 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。

## テストプレイ

2024-09-07 Update

- Cloudflare Pagesで開発環境を開発チーム以外限定公開にする方法を確立したため、テストプレイの方法がすべて変わりました
  - これまでは手元のPCでLive Serverでサーバーを立てる必要がありましたが、それが不要になっています

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！！

## 譜面コンテナファイルを作成する

- テストプレイしたい譜面の譜面ファイル(.jsonc)を用意します
- その譜面に対応する音源ファイル(.mp3)を用意します
- <https://holorhysm-chart-zipper.pages.dev/>を開きます
  - Cloudflare Access (Cloudflare Zero Trust)のログイン画面が出たらGitHubを選択してログイン
    - GitHubの500エラーが出たらもう一回ログインし直したら通るかも
    - "Authorize holorhysm developer preview"が出たら[authorize holorhysm]をクリック
- 上の方の「Chart (.jsonc)」のファイル選択ダイアログを開き、譜面ファイル(.jsonc)を選択します
- その右にある「Audio (.mp3)」のファイル選択ダイアログを開き、音源ファイル(.mp3)を選択します
- その右にある[Zip]ボタンを押すと、「Chart Container (.hryc)」欄に[DL]ボタンが出るのでそれを押します
  - ボタンを押すと譜面コンテナファイル(.hryc)がダウンロードされます

## テストする

- テストに使う端末で<https://main.holorhysm.pages.dev/>を開きます
  - Chart Zipperと同様にCloudflare Access (Cloudflare Zero Trust)のログイン画面が出ますが、同様に……
- ホーム画面まで進んだらダイアル画面を開きます
- その画面から、**7357**に電話をかけます
  - 発信音のあとに、「Test mode turned ON. Hello, Charter.」と喋って通話が切れます
    - 「Test mode turned **OFF**. Goodbye, Charter.」だったらもう一回かけ直し
- この状態でホーム画面に戻って[🎵セッション]を押すと、テストプレイ選択画面に入ります
- ファイル選択ダイアログを開いて、譜面コンテナファイル(.hryc)を選択します
- [Start]ボタンを押すと読み込んだ譜面のテストプレイができます
  - リザルト画面からのリトライも動きます