



ft_transcendence

やがて、自分が知らないと思っていたことを
すでに知っていることに気づくだろう。

概要

もうC言語はいらない! もうC++はいらない!

このプロジェクトは、これまでやったことのないことをやるというものだ

。

コンピュータ・サイエンスへの旅の始まりを思い出してください。

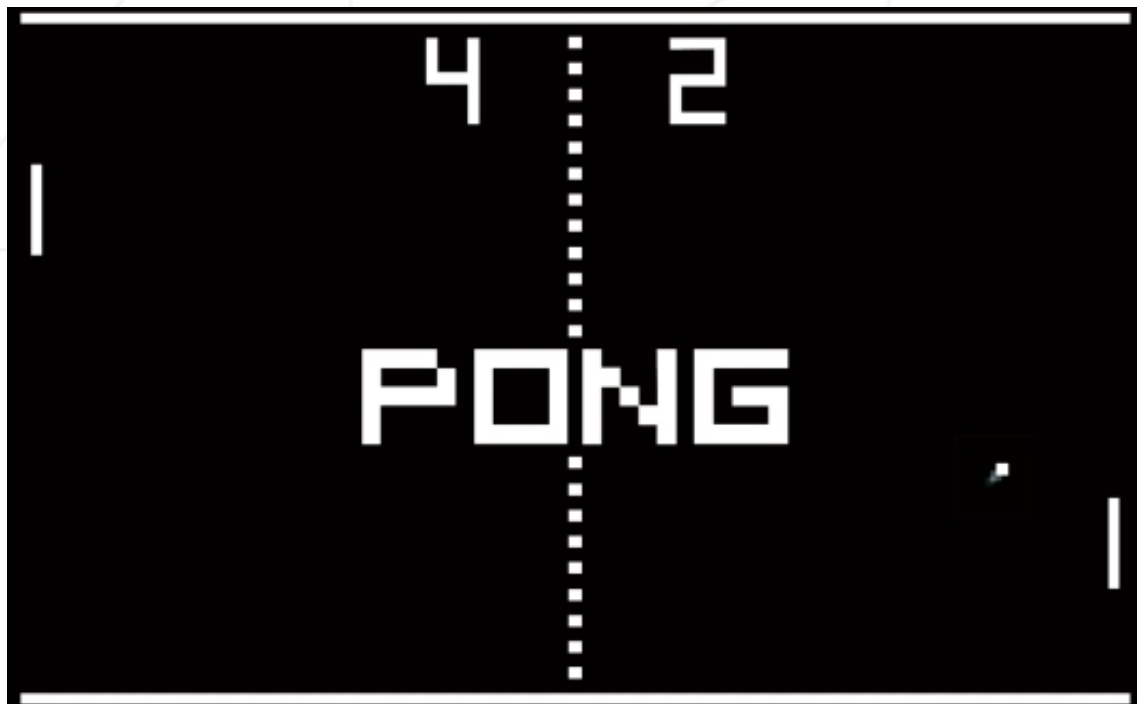
今こそ輝く時だ!

バージョン: 12.1

内容

I	前文	2
II	必須パート	3
II.1	概要	3
II.2	セキュリティ上の 懸念.....	4
II.3	ユーザーアカウント .	4
II.4	チャット .	5
II.5	Game	5
III	提出と相互評価	6

第1章 前文



第II章 必須パート

このプロジェクトは、マイティー・ポン・コンテストのウェブサイトを作成することである！

II.1 概要

あなたのウェブサイトのおかげで、ユーザーは他の人とPongをプレイすることができます。素敵なユーザーインターフェイス、チャット、リアルタイムのマルチプレイヤーオンラインゲームを提供します！

あなたの作品は以下のルールに従わなければならない：

- あなたのウェブサイトのバックエンドはNestJSで書かれていなければなりません。
- フロントエンドは好みのTypeScriptフレームワークで書かなければならない。
- この文脈では、どのライブラリを使うのも自由です。ただし、プロジェクトで使用するすべてのライブラリやフレームワークは、**最新の安定版**を使用しなければなりません。
- PostgreSQLデータベースを使わなければなりません。他のデータベースは使えません。
- ウェブサイトは[シングルページのアプリケーション](#)でなければなりません。ユーザーはブラウザの「戻る」と「進む」ボタン。
- あなたのウェブサイトは、以下の**最新の安定した最新バージョン**と互換

性がなければなりません。

*Google Chrome*および好きなウェブブラウザ1つ。

- ユーザーがウェブサイトを開覧する際に、未処理のエラーや警告が発生しないこと。
- `docker-compose up --build` を呼び出すだけで、すべてが起動する。



クラスタ内のコンピュータがLinuxで動作している場合、セキュリティ上の理由からDockerをルートルесモードで使用するようになる。これには2つの側面があります：1) Dockerのランタイムファイルは/`goinfre`または/`sgoinfre`に置かなければならない。2) ホストとコンテナ間で、いわゆる "バインドマウントボリューム" を使用できない。

コンテナプロジェクトや状況、文脈に応じて、いくつかのフォールバックが存在する：VMでDockerする、変更後にコンテナを再構築する、`root`をユニークUIDとする独自のDockerイメージを作成する。

II.2 セキュリティへの懸念

十分に機能するウェブサイトを作成するために、あなたに取り組まなければならないセキュリティ上の懸念事項をいくつか挙げよう：

- データベースに保存されているパスワードはすべてハッシュ化されていなければなりません。
- ウェブサイトはSQLインジェクションから保護されていなければなりません。
- フォームやあらゆるユーザー入力に対して、何らかのサーバーサイド・バリデーションを実装する必要があります。



強力なパスワード・ハッシュ・アルゴリズムを使用していることを確認してください。



明らかなセキュリティ上の理由から、認証情報、APIキー、環境変数などは.envファイルにローカルに保存し、gitでは無視する必要があります。公に保存された認証情報は、プロジェクトの失敗に直結します。

II.3 ユーザーアカウント

- ユーザーは、42イントラネットのOAuthシステムを使用してログインする必要があります。
- ユーザーは、ウェブサイトに表示されるユニークな名前を選ぶことができるはずだ。
- ユーザーはアバターをアップロードできなければなりません。ユーザーがアバターをアップロードしない場合は、デフォルトのアバターを設定する必要があります。

- ユーザーは、**二要素認証**を有効にすることができるはずです。 例えば

*Google Authenticator*または携帯電話にテキストメッセージを送信する。

- ユーザーは、他のユーザーをフレンドとして追加し、現在のステータス（オンライン、オフライン、ゲーム中など）を確認できるようにする必要があります。
- 統計（勝敗、ラダーレベル、実績など）はユーザープロフィールに表示されなければならない。
- 各ユーザーは、1vs1ゲーム、ラダー、その他有用なものを含む**マッチヒストリー**を持つべきである。ログインしている人なら誰でも参照できるはずです。

II.4 チャット

また、ユーザーのためにチャットを作成する必要があります：

- ユーザーは、公開、非公開、パスワードで保護された**チャンネル**（チャットルーム）を作成する必要があります。
- ユーザーは他のユーザーに**ダイレクトメッセージ**を送ることができなければならない。
- ユーザーは他のユーザーをブロックすることができるはずです。そうすることで、ブロックしたアカウントからのメッセージを見ることができなくなります。
- 新しいチャンネルを作成したユーザは、自動的に**チャンネル・オーナー**に設定されます。
彼らがそこを去るまで。
 - チャンネルのオーナーは、チャンネルにアクセスするために必要なパスワードを設定したり、変更したり、削除したりすることができます。
 - チャンネルオーナーはチャンネル**管理者**です。他のユーザを管理者に設定できます。
 - チャンネルの管理者であるユーザーは、他のユーザーをキック、禁止、ミュート（期間限定）することができますが、チャンネルのオーナーはできません。
- ユーザーは、チャットインターフェースを通じて、他のユーザーをポンゲームに招待できるようにする必要がある。
- ユーザーは、チャットインターフェースを通じて他のプレーヤーのプロフィールにアクセスできるべきである。

II.5 ゲーム

このサイトの主な目的は、他のプレーヤーとポンで対戦することです。

- したがって、ユーザーはウェブサイト上で直接、他のプレーヤーとライブのポンゲームをプレイできるはずだ。
- **マッチメイキングシステム**がなければならない。ユーザーは、自動的に誰かとマッチングされるまで、キューに参加することができる。
- キャンバスのゲームでもいいし、3Dでレンダリングされたゲームでもいいし、醜くてもいいが、いずれにせよ、**オリジナルの『Pong』**（1972年）に忠実でなければならない。
- いくつかの**カスタマイズオプション**（例えば、パワーアップや異なるマップ）を提供する必要があります。ただし、ユーザーが望むなら、追加機能のないデフォルトバージョンを選択できるようにする必要があります。
- 試合は**手応えのあるもの**でなければならない！



予期せぬ切断やラグなど、ネットワークの問題について考えてみましょう。可能な限り最高のユーザー体験を提供しなければなりません。

第三章

提出と相互評価

通常通り、Gitリポジトリに課題を提出してください。あなたのリポジトリ内の作品だけが、ディフェンス中に評価されます。ファイル名が正しいかどうか、ためらわずにダブルチェックしてください。

