



ft_transcendence

サプライズだ。

概要

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。

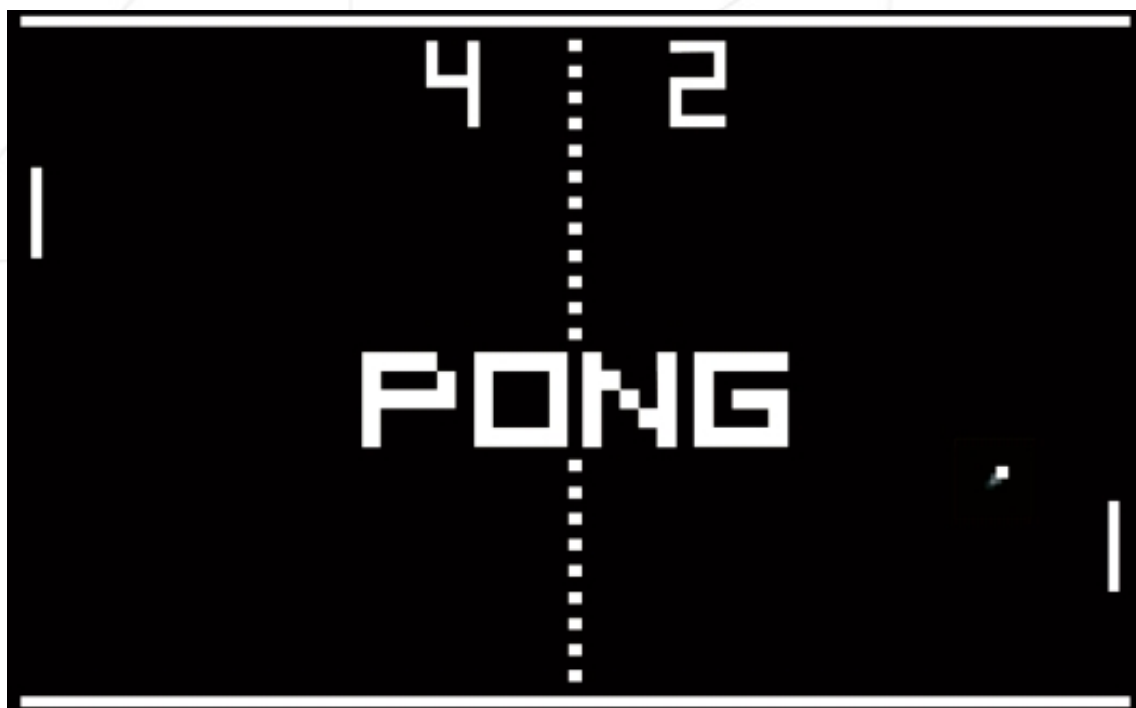
今こそ輝く時だ！

バージョン: 15.1

内容

I	前文	2
II	エッセンシャル・ポイント	3
III	必須パート	5
III.1	概要	5
III.2	最小限の技術要件	6
III.3	Game	7
III.4	セキュリティへの懸念	8
IV	モジュール	9
IV.1	概要.....	10
IV.2	ウェブ.....	12
IV.3	ユーザー管理	13
IV.4	ゲームプレイとユーザーエクスペリエンス	15
IV.5	AI-アルゴ	17
IV.6	サイバーセキュリティ	19
IV.7	デボップス.....	21
IV.8	ゲーミング.....	23
IV.9	グラフィック	24
IV.10	アクセシビリティ	25
IV.11	サーバーサイドのポン.....	27
V	ボーナス・パート	29
VI	提出と相互評価	30

章 前文



第二章 重要なポ

イント

このプロジェクトは複雑な事業であり、指定された制約の中で意思決定を行う必要があります。特定のモジュールの実装にはある程度の柔軟性があり、**テーマの範囲内**であなたの裁量に任されています（つまり、テーマの制約を迂回することはできません）。評価の際に、あなたの選択の正当性を問われることがあります。

ウェブサイトを立ち上げるために`nginx`を使う必要があると考えるなら問題はありませんが、まず自問してみてください。それがなくてもやっていけるのか？ 同様に、あなたをサポートしてくれるライブラリに直面したとき、それがあなたのタスクを満たしてくれるかどうかを理解することは非常に重要です。面白くないサブレイヤーを作り直すのではなく、提案された機能を機能させることが期待されているのです。

ある機能の実装について疑問が生じるような決断に遭遇することを理解することが極めて重要である。最初に、プロジェクトの要件を徹底的に理解することを**強く推奨する**。達成すべきことを把握したら、プロジェクトの枠内にとどまることが必要である。私たちがある技術について言及するとき、それは、要求されたフレームワーク／言語に公式に関連するものはすべて許可されるということを明確に意味します。

しかし、あるモジュールを実装したい場合、すべての制限がそのモジュールに適用されることを強調しておきます。例えば、お題で指定されたバックエンドモジュールでプロジェクトを実現したい場合、デフォルトの言語は使えません。それでもデフォルトの言語を使ってバックエンドを作成したい場合、それも可能ですが、要求

された言語/フレームワークを使っていないので、このモジュールは有効とはみなされません。

結論の前に、いくつかのモジュールは意図的に他のモジュールに強く依存していることに注意することが重要だ。

このテーマには明確な制約（しなければならないことならないこと）がある一方で、いくつかの選択の余地が残されている分野もあります。これらの選択肢は、評価の際に正当性を示す必要があります。注意してください。

コードに入る前に、時間をかけてアプリケーションのデザインを熟考することが重要です。

ft_transcendence

サプライズ

お楽しみください! :)

第III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！



- グローバルな機能やモジュールに対して、即座に完全なソリューションを提供するライブラリやツールの使用は禁止されています。
- サードパーティのライブラリやツールの使用法（can、must、can't）について、直接的な指示があれば、それに従う必要がある。
- グローバルな機能やモジュールのサブコンポーネントに相当する、シンプルでユニークなタスクを解決する小さなライブラリやツールの使用は認められる。
- 評価中、チームは、対象者が明示的に承認していないライブラリやツールの使用法であっても、対象者の制約に反しないものであれば、それを正当化する。
- 評価の間、評価者は自分の責任において、特定のライブラリやツールの使用が正当なものか（許可されているか）、あるいは機能やモジュール全体をほとんど解決していないか（禁止されているか）を定義する。

III.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と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なければなりません！

- プロジェクトは、最低限以下のガイドラインを遵守する必要があり、最終成績に貢献するのはごく一部です。

- このテーマの第2部では、以下のルールを置き換えたり、完成させたりすることができる追加モジュールを提供する。

この『課題』では、特定の言葉が緑色で強調されている。これらは、時間の経過とともに進化する技術の選択を表しています。主題のバージョンに細心の注意を払ってください。

III.2 最小限の技術要件

プロジェクトは以下のルールに従わなければならない:



繰り返しになるが、これらの制約のいくつかは、特定のモジュールの覆すことができる。

- バックエンドの有無にかかわらず、自由にサイトを開発することができます。
 - バックエンドを含める場合は、純粋なRubyで記述する必要があります。しかし、この要件はFrameworkモジュールによって上書きすることができます。
 - バックエンドやフレームワークがデータベースを使用する場合、Databaseモジュールの制約に従う必要があります。
- フロントエンドは、純粋なバニラJavascriptを使用して開発する必要があります。しかし、この要件はFrontEndモジュールによって変更することができます。
- 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。
グーグル・クローム
- 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が提供する自律型コンテナを実行するには、コマンドライン1つですべてを起動する必要があります。例: `docker-compose up --build`

コンテナ・ソリューションがDockerの場合:

クラスタ内のコンピュータがLinuxで動作している場合、セキュリティ理由からDockerをルートレスモードで使用するようになる。これには2つの側面がある:



- Docker ランタイムファイルは、/goinfre または /スゴインフレ
- コンテナ内で非root UIDが使用されている場合、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」を使用することはできない。

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、rootをユニークUIDとする独自のDockerイメージを作成する。

III.3 ゲーム

このサイトの主な目的は、他のプレイヤーとポンで対戦することです。

- そのため、ユーザーはウェブサイト上で他のプレイヤーと直接対戦するライブのポンゲームに参加できなければならない。両プレイヤーは同じキーボードを使用します。**リモートプレイヤーモジュール**は、リモートプレイヤーを使ってこの機能を強化することができます。
- プレイヤーは他のプレイヤーと対戦できなければならないが、**トーナメント**を提案することも可能でなければならない。このトーナメントは、複数のプレイヤーが順番にすることで構成されます。トーナメントをどのように実施するかは自由ですが、誰が誰と対戦するのか、またその順番は明確に表示されなければなりません。
- トーナメント開始時に、各プレイヤーはエイリアス名を入力する必要があります。エイリアスは、新しいトーナメントが始まるときにリセットされます。ただし、この要件は**Standard User Managementモジュール**を使って変更することができます。
- トーナメントシステムが参加者のマッチメイクを行い、次の試合を発表する。
- すべてのプレイヤーは同じルールを守らなければならない。この条件はAIを使用する場合にも適用され、AIは通常のプレイヤーと同じスピードを示さなければならない。
- ゲーム自体は、デフォルトのフロントエンドの制約（上記のとおり）に従って開発されなければならないが、**FrontEndモジュール**を利用することもできますし、**Graphicsモジュール**で上書きすることもできます。見た目の美しさは様々ですが、**オリジナルの Pong (1972)**のエッセンスを取り入れる必要があります。



- グローバルな機能やモジュールに対して、即座に完全なソリューションを提供するライブラリやツールの使用は禁止されています。
- サードパーティのライブラリやツールの使用法（can、must、can't）について、直接的な指示があれば、それに従う必要がある。
- グローバルな機能やモジュールのサブコンポーネントに相当する、シンプルでユニークなタスクを解決する小さなライブラリやツールの使用は認められる。
- 評価中、チームは、対象者が明確に承認していないライブラリやツールの使用法であっても、対象者の制約に反しないものであれば、それを正当化する。
- 評価の間、評価者は自分の責任において、特定のライブラリやツールの使用が正当なものか（許可されているか）、あるいは機能やモジュール全体をほとんど解決していないか（禁止されているか）を定義する。

III.4 セキュリティへの懸念

基本的な機能を備えたウェブサイトを作成するために、取り組まなければならないセキュリティ上の問題がいくつかある：

- データベースに保存されているパスワードが該当する場合は、**ハッシュ化する**必要があります。
- ウェブサイトはSQLインジェクション/XSSから保護されていなければなりません。
- バックエンドやその他の機能がある場合は、すべての面でHTTPS接続を有効にすることが必須です（wsの代わりにwssを利用する...）。
- バックエンドを使用しない場合はベースページ内で、バックエンドを使用する場合はサーバーサイドで、フォームやユーザー入力に対して何らかのバリデーションを実装しなければなりません。
- 2FA付きのJWTセキュリティ・モジュールを実装するかどうかにかかわらず、ウェブサイトのセキュリティを優先することが極めて重要です。例えば、APIを作成する場合は、ルートが保護されていることを確認してください。JWTトークンを使用しないと決めたとしても、サイトのセキュリティ確保が不可欠であることに変わりはないことを忘れないでください。



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。

第IV章 モジュ

ール

これでプロジェクトの25%を達成したなる！

機能的な基本ウェブサイトが出来上がったので、次のステップは、さらに改良するためのモジュールを選ぶことである。

プロジェクトを100%完成させるには、**最低7つの主要モジュールが必要です**。各モジュールを注意深く確認することは非常に重要です。そのため、このテーマ全体を熟読することを強くお勧めします。



- グローバルな機能やモジュールに対して、即座に完全なソリューションを提供するライブラリやツールの使用は禁止されています。
- サードパーティのライブラリやツールの使用法（can、must、can't）について、直接的な指示があれば、それに従う必要がある。
- グローバルな機能やモジュールのサブコンポーネントに相当する、シンプルでユニークなタスクを解決する小さなライブラリやツールの使用は認められる。
- 評価中、チームは、対象者が明示的に承認していないライブラリやツールの使用法であっても、対象者の制約に反しないものであれば、それを正当化する。
- 評価の間、評価者は自分の責任において、特定のライブラリやツールの使用が正当なものか（許可されているか）、あるいは機能やモジュール全体をほとんど解決していないか（禁止されているか）を定義する。



2つのマイナー・モジュールは1つのメジャー・モジュールに相当する。

IV.1 概要

• ウェブ

- **主要モジュール**: フレームワークを使ってバックエンドを構築する。
- **マイナーモジュール**: フレームワークやツールキットを使ってフロントエンドを構築する。
- **マイナーモジュール**: バックエンドにデータベースを使用する。
- **主要モジュール**: ブロックチェーンにトーナメントのスコアを保存する。

• ユーザー管理

- **主要モジュール**: 標準的なユーザー管理、認証、トーナメント間のユーザー。
- **主要モジュール**: リモート認証の実装

ゲームプレイとユーザー体験

- **主要モジュール**: リモートプレーヤー
- **主要モジュール**: マルチプレイヤー（同じ2人以上）。
- **主要モジュール**: ユーザー履歴とマッチメイキングで別のゲームを追加します。
- **マイナーモジュール**: ゲームカスタマイズオプション。
- **主要モジュール**: ライブチャット

• AI-アルゴ

- **主要モジュール**: AIの対戦相手を紹介する。
- **マイナーモジュール**: ユーザーと試合の統計ダッシュボード

• サイバーセキュリティ

- **主要モジュール**: HardenedコンフィギュレーションによるWAF/ModSecurity

の導入と、秘密管理のためのHashiCorp Vault。

- **マイナーモジュール**：ユーザー匿名化、ローカルデータ管理、アカウント削除によるGDPRコンプライアンスオプション。
- **主要モジュール**：二要素認証（2FA）とJWTの実装。

- **デボックス**

- **主要モジュール**：ログためのインフラセットアップ。
- **マイナーモジュール**：監視システム。
- **主要モジュール**：マイクロサービスとしてのバックエンドの設計。

• グラフィック

- **主要モジュール**: 高度な3D技術の使用。

• アクセシビリティ

- **マイナーモジュール**: すべてのデバイスでサポート。
- **マイナーモジュール**: ブラウザ互換性の拡張。
- **マイナーモジュール**: 複数言語サポート。
- **マイナーモジュール**: 視覚障害ためのアクセシビリティを追加しました。
- **マイナーモジュール**: サーバーサイドレンダリング (SSR) の統合。

• サーバーサイドのポン

- **主要モジュール**: 基本的なポンをサーバーサイドのポンに置き換え、APIを実装する。
- **主要モジュール**: API統合により、Webユーザーに対してCLI経由でPongゲームプレイを可能にする。

IV.2 ウェブ

これらのモジュールは、あなたのPongゲームに高度なWeb機能を統合することができます。

- **主要モジュール**: フレームワークを使ってバックエンドを構築する。

この主要なモジュールでは、バックエンド開発に特定のウェブフレームワークを利用することが求められます。



デフォルトの言語/フレームワーク(上記の必須参照)を使うことで、このモジュールの制約を使わずにバックエンドを作成することができます。しかし、このモジュールはその要件に従った場合のみ有効です。

- **マイナーモジュール**: フレームワークやツールキットを使ってフロントエンドを構築する。

フロントエンドの開発では、バニラJavascriptに加えてBootstrapツールキットを使用しなければならない。



デフォルトのフロントエンドディレクティブを使用することで、このモジュールの制約を使用せずにフロントエンドを作成することができます(上記の必須部分を参照)。しかし、このモジュールはその要件に従った場合にのみ有効です。

- **マイナーモジュール**: バックエンドにデータベースを使用する。

プロジェクト内のすべてのDBインスタンスに指定されたデータベースはPostgreSQLです。この選択は、すべてのプロジェクト構成要素におけるデータの一貫性と互換性を保証し、バックエンドのフレームワークモジュールなどの他のモジュールの前提条件となる可能性があります。

- **主要モジュール**: ブロックチェーンにトーナメントのスコアを保存する。

このメジャーモジュールでは、ブロックチェーン上にトーナメントのスコアを安全に保存する機能をPongウェブサイトを実装することに焦点を当てています。開発とテストの、テスト用のブロックチェーン環境を利用することを明確にしておくことが重要です。この実装に使用するブロックチェーンは

Ethereumで、スマートコントラクトの開発に使用するプログラミング言語はSolidityです。

- 。ブロックチェーンの統合：このモジュールの主な目標は、ブロックチェーン技術、特にイーサリアムをPongウェブサイトにしームレスに統合することです。この統合により、トーナメントスコアの安全かつ不変な保存が保証され、プレイヤーにゲーム成果の透明で改ざん不可能な記録を提供します。
- 。ソリディティ スマート・コントラクトブロックチェーンとやりとりするために、私たちはスマート・コントラクトを開発する。これらの契約は記録を担当する、ソリディティ トーナメントスコアの管理、検索。

- テスト用ブロックチェーン：前述したように、テスト用ブロックチェーンは開発とテストの目的で導入される。これにより、ブロックチェーンに関連するすべての機能が、本番のブロックチェーンに伴うリスクなしに徹底的に検証される。
- 相互運用性：このモジュールは他のモジュール、特にバックエンド・フレームワーク・モジュールに依存する可能性がある。ブロックチェーンの機能を統合するには、ブロックチェーンとの相互作用に対応するためにバックエンドの調整が必要になるかもしれない。

このモジュールを実装することで、ブロックチェーンベースのスコア保存システムを導入し、Pongウェブサイトを強化することを目指しています。ユーザーは、セキュリティと透明性のレイヤーが追加され、ゲームスコアの完全性が保証されるという恩恵を受けることができます。このモジュールでは、ブロックチェーン開発に伴うリスクを最小限に抑えるため、テスト用のブロックチェーン環境を使用することを重視しています。

IV.3 ユーザー管理

このモジュールは**ユーザー管理**の領域を掘り下げ、Pong プラットフォームにおけるユーザーとのやり取りとアクセス制御の重要な側面を扱います。このモジュールには2つの主要なコンポーネントがあり、それぞれユーザー管理と認証の重要な要素に焦点を当てています。

- **主要モジュール**：標準的なユーザー管理、認証、またがるユーザー。
 - ユーザーは安全なウェブサイトを購読できる。
 - 登録ユーザーは安全なログインできます。
 - ユーザーはトーナメントをプレーするためにユニークな表示名を選択することができます。
 - ユーザーは自分の情報を更新できる。
 - ユーザーはアバターをアップロードすることができます。

- ユーザーは他のユーザーを友達として追加し、オンラインステータスを見ることができる。
- ユーザープロフィールには、勝敗などの統計情報が表示される。
- 各ユーザーは、1対1の試合、日付、関連する詳細を含む**試合履歴を持つ**ており、ログインしたユーザーがアクセスできます。



重複したユーザー名やメールの管理は、あなたの裁量に任されています。納得のいく解決策を提供してください。

- **主要モジュール：リモート認証の実装**

この主要モジュールでは、以下の認証実装することを目標としている：

42のOAuth 2.0認証。主な機能と目的は以下の通りです：



重複したユーザー名やメールの管理は、あなたの裁量に任されています。納得のいく解決策を提供してください。

- 認証システムを統合し、ユーザーが安全にサインインできるようにする。
- 安全なログインを可能にするために、必要な認証情報と権限を権限者から取得する。
- ベストプラクティスとセキュリティ標準に準拠した、ユーザーフレンドリーなログインと承認のフローを導入する。
- ウェブアプリケーションと認証プロバイダ間で、認証トークンとユーザー安全に交換できるようにする。

この主要モジュールは、リモート・ユーザー認証を取得し、ユーザーに安全で便利なウェブ・アプリケーションへのアクセス方法を提供ことを目的としています。

IV.4 ゲームプレイとユーザー体験

これらのモジュールは、プロジェクトの一般的なゲームプレイを強化するために設計されている。

- **主要モジュール：リモートプレーヤー**

離れた場所に2人のプレイヤーを置くことも可能です。各プレイヤーは別々のコンピュータに配置され、同じウェブサイトアクセスし、同じPongゲームをプレイします。



予期せぬ切断やラグなど、ネットワークのみましよう。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。

- **主要モジュール：マルチプレーヤー**

2人以上のプレイヤーを持つことも可能です。各プレイヤーはライブコントロールが必要です（ですから、以前の "Distant players "モジュールが強く推奨されます）。ゲームを3人、4人、5人、6人...のプレイヤーでどのようにプレイするかはあなた次第です。通常の2人用ゲームと同様に、この多人数用モジュールでは、2人プレイヤー数を選択することができます。例：正方形のボードで4人でプレイし、各プレイヤーは正方形の一辺を所有する。

- **主要モジュール：ユーザー履歴とマッチメイキングで別のゲームを追加します。**

この主要モジュールでは、Pongとは異なる新しいゲームを導入し、ユーザーの履歴追跡やマッチメイキングなどの機能を組み込むことを目的としている。主な特徴と目標は以下の通りです：

- 新しい魅力的なゲームを開発し、プラットフォームの提供を多様化し、ユーザーを獲得する。
- ユーザー履歴トラッキングを実装し、個々のユーザーのゲームプレイ統計を記録・表示する。
- ユーザーが対戦相手を見つけ、公平でバランスの取れた試合に参加できるよう、マッチメイキングシステムを構築する。

- ユーザーのゲーム履歴とマッチメイキングデータが安全に保存され、最新の状態に保たれるようにします。
- 新しいゲームのパフォーマンスとレスポンスを最適化し、楽しいユーザー体験を提供する。バグを修正し、新機能を追加し、ゲームプレイを向上させるために、定期的にゲームをアップデートし、保守すること。

この主要モジュールは、新しいゲームの導入、ゲームプレイ履歴によるユーザーのエンゲージメントの強化、楽しいゲーム体験のためのマッチメイキングの促進によって、プラットフォームを拡大することを目的としています。

- **マイナーモジュール：**ゲームカスタマイズオプション。

このマイナーモジュールでは、プラットフォーム上で利用可能なすべてのゲームにカスタマイズオプションを提供することを目的としています。主な機能と目的は以下の通りです：

- パワーアップ、攻撃、別マップなど、ゲームプレイ体験を向上させるカスタマイズ機能を提供する。
- よりシンプルな体験を好む場合は、基本的な機能を備えたゲームのデフォルトバージョンを選択できるようにする。
- カスタマイズオプションが利用可能で、プラットフォーム上で提供されるすべてのゲームに適用できるようにする。
- ゲームのパラメータを調整するためのユーザーフレンドリーな設定メニューやインターフェースを実装する。
- 統一されたユーザーエクスペリエンスを提供するため、すべてのゲームにおいてカスタマイズ機能の一貫性を維持する。

このモジュールは、さまざまなカスタマイズ・オプションを提供することで、利用可能なすべてのゲームでゲーム体験をカスタマイズできる柔軟性をユーザーに提供することを目的としている。

• **主要モジュールライブチャット。**

このモジュールでユーザーのためのチャット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。そうすれば、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なりません。
- 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べきである。
- トーナメントのシステムは、次の試合が予想されるユーザーに警告を発することができるはずです。
- ユーザーは、チャット・イン・フェイ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。

IV.5 AI-アルゴ

これらのモジュールは、データ駆動型の要素をプロジェクトに導入する役割を果たし、主要モジュールはゲームプレイを向上させるAI対戦相手を導入し、マイナーモジュールはユーザーとゲームの統計ダッシュボードに焦点を当て、ユーザーに最小限の、しかし洞察に満ちたゲーム体験を垣間見せる。

- **主要モジュール：** AIの対戦相手を紹介する。

この主要モジュールでは、AIプレーヤーをゲームに組み込むことが目的である。特に、この課題では**A*アルゴリズム**の使用は許可されていない。主な特徴と目標は以下の通り：

- ユーザーに挑戦的で魅力的なゲーム体験を提供するAI対戦相手を開発する。
- つまり、AIの実装では、キーボード入力をシミュレートしなければならない。ここでの制約は、AIはゲームのビューを1秒間に1回しか更新できないため、バウンドやその他のアクションを予測する必要があることです。



を選択した場合、AIはパワーアップを利用しなければならない。
ゲームカスタマイズオプションモジュール。

- AIプレーヤーが知的で戦略的な動きをできるように、AIのロジックと意思決定プロセスを実装する。
- A*に頼ることなく、効果的なAIプレーヤーを作るための代替アルゴリズムやテクニックを探る。
- AIがさまざまなゲームプレイシナリオやユーザーとのインタラクションに適応できるようにする。



注意評価の際には、あなたのAIがどのようにいるかを詳しく説明する必要があります。何もしないAIを作ることは厳禁です。たまたに勝つ能力を持たせなければなりません。

この主要モジュールは、A*アルゴリズムに依存することなく、興奮と競争力を付加するAI対戦相手を導入することで、ゲームを強化することを目的としている。

- **マイナーモジュール：**ユーザーとゲームの統計ダッシュボード。

このマイナーモジュールでは、個々のユーザーとゲームセッションの統計を表示するダッシュボードを導入することを目的としています。主な機能と目的は以下の通りです：

- ユーザーフレンドリーなダッシュボードを作成し、ユーザー自身のゲーム統計に関する洞察を提供します。

- 試合セッション用の独立したダッシュボードを開発し、各試合の詳細な統計、結果、履歴データを表示する。
- ダッシュボードが、データを追跡・分析するための直感的で有益なユーザー・インターフェイスを提供することを確認する。
- 図表やグラフなどのデータ視覚化テクニックを導入し、統計情報をわかりやすく視覚的にアピールする。
- ユーザーが自分のゲーム履歴やパフォーマンス指標にアクセスし、便利に探索できるようにします。
- 有用と思われる指標を自由に追加してください。

このマイナーモジュールは、ユーザーフレンドリーなダッシュボードを通じてゲーム統計やゲームセッションの詳細をモニターし、ゲーム体験の包括的なビューを提供する能力をユーザーに与えることを目的としています。

IV.6 サイバーセキュリティ

これらのサイバーセキュリティモジュールは、プロジェクトのセキュリティ態勢を強化するために設計されており、主要モジュールは、Webアプリケーションファイアウォール（WAF）とModSecurityの設定、および安全な秘密管理のためのHashiCorp Vaultによる堅牢な保護に重点を置いています。マイナーモジュールは、GDPR準拠、ユーザーデータの匿名化、アカウント削除、二要素認証（2FA）、JSON Webトークン（JWT）のオプションを追加することで、この取り組みを補完し、データ保護、プライバシー、認証セキュリティに対するプロジェクトのコミットメントを総体的に保証します。

- **主要モジュール**HardenedコンフィギュレーションによるWAF/ModSecurityの導入と、秘密管理のためのHashiCorp Vault。

この主要モジュールでは、いくつかの重要なコンポーネントを実装することにより、プロジェクトのセキュリティ・インフラを強化することを目的とする。主な特徴と目標は以下の通り：

- Webアプリケーションファイアウォール(WAF)とModSecurityを厳密で安全な構成で設定、導入し、Webベースの攻撃から保護する。
- HashiCorp Vaultを統合することで、APIキー、認証情報、環境変数などの機密安全に管理・保管し、これらの機密情報を適切に暗号化・隔離することができます。

この主要モジュールは、Webアプリケーション保護のためのWAF/ModSecurityや、安全でセキュアな環境を確保するための秘密管理のためのHashiCorp Vaultなど、堅牢なセキュリティ対策を導入することで、プロジェクトのセキュリティインフラを強化することを目的としている。

- **マイナーモジュール**：ユーザー匿名化、ローカルデータ管理、アカウント削除によるGDPRコンプライアンスオプション。

このマイナーモジュールでは、ユーザーがデータプライバシーの権利を行使できるようにするGDPRコンプライアンスオプションを紹介することを目的としています。主な特徴と目的は以下の通りです：

- GDPRに準拠した機能を実装し、ユーザーが個人データの匿名化を要求できるようにする。
- システム内に保存されている個人情報を読覧、編集、削除する機能など、ユーザーが自分のローカルデータを管理するためのツールを提供する。
- すべての関連データを含むアカウントの永久削除をユーザーが要求するための合理化されたプロセスを提供し、データ保護規制の遵守を確保する。
- データ・プライバシーの権利に関するユーザーとの明確で透明性のあるコミュニケーションを維持し、これらの権利を行使するためのアクセスしやすいオプションを用意する。

このマイナーモジュールは、ユーザーがシステム内で個人情報を管理し、データプライバシーの権利を行使できるようにするGDPRコンプライアンスオプションを提供することで、ユーザーのプライバシーとデータ保護を強化することを目的としています。

一般データ保護規則(GDPR)をご存じでない方は、次のようにお読みください。

は、特にユーザーデータの管理とプライバシーに関して、その原則と意味を理解することが不可欠です。GDPRは、欧州連合（EU）および欧州経済領域（EEA）内の個人データとプライバシーを保護することを目的とした規制です。組織が個人データをどのように扱い、処理すべきかについて、厳格な規則とガイドラインを定めています。

GDPRとその要件について理解を深めるには、データ保護に関する欧州委員会の公式ウェブサイトをご覧になることを強くお勧めします¹。このウェブサイトでは、GDPRの原則、目的、ユーザーの権利など、GDPRに関する包括的な情報を提供しています。また、このトピックをより深く掘り下げ、規制へのコンプライアンスを確保するための追加的な情報源も提供しています。

GDPRをよくご存じない方は、このプロジェクトを進める前に、提供されたリンクにアクセスし、規制をよく理解してください。

• **主要モジュール：二要素認証（2FA）とJWTの実装。**

この主要モジュールでは、二要素認証（2FA）を導入し、JSONウェブトークン（JWT）を活用することで、セキュリティとユーザー認証を強化することを目的としています。主な特徴と目的は以下の通りです：

- 二要素認証（2FA）をユーザーアカウントの追加セキュリティ層として導入し、パスワードに加えてワンタイムコードなどの二次認証方法をユーザーに提供することを義務付ける。
- 認証と認可のためのセキュアな方法としてJSON Webトークン（JWT）を活用し、ユーザー・セッションとリソースへのアクセスが安全に管理されるようにします。
- SMSコード、認証アプリ、または電子メールベースの認証のオプションを使用して、2FAを有効にするためのユーザーフレンドリーな設定プロセスを提供する。
- JWT トークンが安全に発行・検証され、ユーザーアカウントや機密データへの不正アクセスが防止されるようにする。

この主要モジュールは、二要素認証（2FA）を提供し、JSONウェブトークン

(JWT) を使用して認証と認可を強化することにより、ユーザーアカウントのセキュリティを強化することを目的としている。

¹ https://commission.europa.eu/law/law-topic/data-protection/data-protection-eu_en

IV.7 デボックス

主なモジュールは、ELK（Elasticsearch、Logstash、Kibana）を使用した効率的なログのためのインフラセットアップ、柔軟性とスケーラビリティのためのマイクロサービスとしてのバックエンドの設計、包括的なシステム監視のためのPrometheus/Grafanaの実装です。

- **主要モジュール:** ELK（Elasticsearch、Logstash、Kibana）によるログ管理のためのインフラセットアップ。

この主要モジュールでは、ELKスタック（Elasticsearch、Logstash、Kibana）を使用したログ管理と分析のための堅牢なインフラストラクチャを確立することを目的としています。主な特徴と目標は以下の通りです:

- Elasticsearchを導入することで、ログデータを効率的に保存・インデックス化し、簡単に検索・アクセスできるようになります。
- 様々なソースからログデータを収集、処理、変換し、Elasticsearchに送信するようにLogstashを設定する。
- ログデータの可視化、ダッシュボードの作成、ログイベントからの洞察の生成のためにKibanaを設定する。
- ログデータの保存を効果的に管理するために、データ保持とアーカイブのポリシーを定義する。
- ログデータとELKスタックコンポーネントへのアクセスを保護するためのセキュリティ対策を実施する。

この主要モジュールは、ELKスタックを使用した強力なログ管理・分析システムを構築し、効果的なトラブルシューティング、モニタリング、システムの運用とパフォーマンスへの洞察を可能にすることを目的としています。

- **マイナーモジュール:** 監視システム。

このマイナーモジュールでは、PrometheusとGrafanaを使用して包括的な監視システムをセットアップすることを目的としています。主な機能と目標は以下の通りです:

- Prometheusを監視およびアラートツールキットとして導入し、メトリクスを収集し、さまざまなシステムコンポーネントの健全性とパフォーマンスを監視します。
- さまざまなサービス、データベース、インフラストラクチャ・コンポーネントからメトリクスを取得するために、データ・エクスポーターとインテグレーションを構成します。
- Grafanaを使用してカスタムダッシュボードとビジュアライゼーションを作成し、システムメトリクスとパフォーマンスに関するリアルタイムの洞察を提供します。
- Prometheusでアラートルールを設定し、重要な問題や異常をプロアクティブに検出して対応します。
- 過去の測定基準データについて、適切なデータ保持と保存戦略を確保する。
- セキュアな認証とアクセス制御メカニズムをGrafanaに実装し、機密性の高いモニタリングデータを保護する。

このマイナー・モジュールは、以下を使用して堅牢なモニタリング・インフラを確立することを目的としている。

PrometheusとGrafanaは、システムのパフォーマンスと信頼性を向上させるために、システム・メトリクスをリアルタイムで可視化し、問題をプロアクティブに検出します。

- **主要モジュール：** マイクロサービスとしてのバックエンドの設計。

この主要モジュールでは、マイクロサービス・アプローチを使ってシステムのバックエンドをアーキテクトすることが目標である。主な特徴と目的は以下の通り：

- バックエンドをより小さく、疎結合のマイクロサービスに分割し、それぞれが特定の機能や特徴を担当する。
- マイクロサービス間の明確な境界とインターフェイスを定義し、独立した開発、デプロイメント、スケーリングを可能にする。
- RESTフルAPIやメッセージキューなど、マイクロサービス間の通信メカニズムを実装し、データ交換や連携を容易にする。
- 各マイクロサービスが単一の、明確に定義されたタスクまたはビジネス能力を担当するようにし、保守性とスケーラビリティを促進する。

この主要モジュールは、ミ・クロスサービス設計アプローチを採用することでシステムのアーキテクチャを強化し、バックエンドコンポーネントの柔軟性、拡張性、保守性を高めることを目的としている。

IV.8 ゲーミング

これらのモジュールは、プロジェクトのゲーミフィケーションの側面を強化するように設計されており、メジャーモジュールは新しいゲーム、ユーザー履歴の追跡、マッチメイキングを紹介し、マイナーモジュールは利用可能なすべてのゲームにわたってカスタマイズオプションを提供することに重点を置いている。

- **主要モジュール:** ユーザー履歴とマッチメイキングで別のゲームを追加します。

この主要モジュールでは、Pongとは異なる新しいゲームを導入し、ユーザーの履歴追跡やマッチメイキングなどの機能を組み込むことを目的としている。主な特徴と目標は以下の通りです:

- 新しい魅力的なゲームを開発し、プラットフォームの提供を多様化し、ユーザーを獲得する。
- ユーザー履歴トラッキングを実装し、個々のユーザーのゲームプレイ統計を記録・表示する。
- ユーザーが対戦相手を見つけ、公平でバランスの取れた試合に参加できるよう、マッチメイキングシステムを構築する。
- ユーザーのゲーム履歴とマッチメイキングデータが安全に保存され、最新の状態に保たれるようにします。
- 新しいゲームのパフォーマンスとレスポンスを最適化し、楽しいユーザー体験を提供する。バグを修正し、新機能を追加し、ゲームプレイを向上させるために、定期的にゲームをアップデートし、保守すること。

この主要モジュールは、新しいゲームの導入、ゲームプレイ履歴によるユーザーのエンゲージメントの強化、楽しいゲーム体験のためのマッチメイキングの促進によって、プラットフォームを拡大することを目的としています。

- **マイナーモジュール:** ゲームカスタマイズオプション。

このマイナーモジュールでは、プラットフォーム上で利用可能なすべてのゲームにカスタマイズオプションを提供することを目的としています。主な機能と目的は以下の通りです:

- パワーアップ、攻撃、別マップなど、ゲームプレイ体験を向上させるカスタマイズ機能を提供する。
- よりシンプルな体験を好む場合は、基本的な機能を備えたゲームのデフォルトバージョンを選択できるようにする。
- カスタマイズオプションが利用可能で、プラットフォーム上で提供されるすべてのゲームに適用できるようにする。
- ゲームのパラメータを調整するための、ユーザーフレンドリーな設定メニューやインターフェースを実装する。
- 統一されたユーザーエクスペリエンスを提供するため、すべてのゲームにおいてカスタマイズ機能の一貫性を維持する。

このモジュールは、さまざまなカスタマイズ・オプションを提供することで、利用可能なすべてのゲームでゲーム体験をカスタマイズできる柔軟性をユーザーに提供することを目的としている。

IV.9 グラフィック

- 専攻モジュール高度な3Dテクニックの実装

グラフィックス知られるこの主要モジュールは、Pongゲームの視覚的側面の強化に焦点を当てています。より没入感のあるゲーム実現するための高度な3D技術の活用を紹介します。具体的には、ThreeJS/WebGLを使用してPongゲームを開発し、望ましい視覚効果を実現します。

- 高度な3Dグラフィックス：このモジュールの主な目標は、高度な3Dグラフィックス技術を実装し、Pongゲームのビジュアルクオリティを向上させることです。ThreeJS/WebGLを利用することで、プレイヤーをゲーム環境に没入させるような素晴らしいビジュアルエフェクトを作成することを目指しています。
- 没入感のあるゲームプレイ：高度な3D技術を取り入れることで、ユーザーに視覚的に魅力的で魅惑的なPongゲームを提供し、ゲームプレイ体験全体を向上させます。
- 技術の統合：このモジュールのために選ばれたテクノロジーは、ThreeJS/WebGLです。これらのツールは、互換性と最適なパフォーマンスを保証する3Dグラフィックスを作成するために使用されます。

この主要モジュールは、高度な3D技術を導入することで、Pongゲームのビジュアル要素に革命を起こすことを目的としています。ThreeJS/WebGLを利用すること、プレイヤーに没入感のある視覚的なゲーム体験を提供することを目指します。

IV.10 アクセシビリティ

これらのモジュールは、ウェブアプリケーションのアクセシビリティを強化するために設計されており、あらゆるデバイスにおける互換性の確保、ブラウザサポートの拡大、多言語機能の提供、視覚障害ユーザー向けのアクセシビリティ機能の提供、パフォーマンスとユーザーエクスペリエンスの向上のためのサーバーサイドレンダリング（SSR）の統合などに重点を置いています。

- **マイナーモジュール：**すべてのデバイスでサポート。

このモジュールでは、あなたのウェブサイトがあらゆる種類のデバイスでシームレスに動作するようにすることに主眼を置きます。主な特徴と目的は以下の通りです：

- デスクトップ、ノートパソコン、タブレット、スマートフォンで一貫したユーザーエクスペリエンスが得られるよう、さまざまな画面サイズや表示方式に対応するレスポンシブなウェブサイトであることを確認してください。
- ユーザーが使用しているデバイスに応じて、タッチスクリーン、キーボード、マウスなど、さまざまな入力ウェブサイトを簡単にナビゲートし、操作できるようにします。

このモジュールは、すべての一貫性のあるユーザーフレンドリーな体験を提供し、アクセシビリティとユーザー満足度を最大化することを目的としています。

- **マイナーモジュール：**ブラウザ互換性の拡張。

このマイナーモジュールでは、追加のウェブブラウザのサポートを追加することによって、ウェブアプリケーションの互換性を強化することを目的としています。主な機能と目的は以下の通りです：

- ブラウザのサポートを拡張し、追加のウェブブラウザを含めることで、ユーザーがシームレスにアプリケーションにアクセスし、使用できるようにします。

- ウェブアプリケーションが正しく機能し、新しくサポートされたブラウザで正しく表示されるよう、徹底的なテストと最適化を行う。
- 追加されたウェブブラウザで発生する可能性のある互換性の問題やレンダリングの不一致に対処する。
- ユーザビリティと機能性を維持しながら、サポートされているすべてのブラウザで一貫したユーザーエクスペリエンスを確保する。

このマイナーモジュールは、追加のウェブブラウザをサポートすることで、ウェブアプリケーションのアクセシビリティを広げ、ユーザーにブラウジング体験の選択肢を提供することを目的としている。

- **マイナーモジュール：複数言語サポート。**

このマイナーモジュールでは、多様なユーザーベースに対応するために、ウェブサイトが確実に多言語をサポートすることを目的としています。主な機能と目標は以下の通りです：

- 幅広い読者を獲得するため、ウェブサイトですべての国語に対応する。

- ユーザーが好みに応じてウェブサイトの言語を簡単に変更できるよう、言語スイッチャーまたはセレクターを提供する。
- ナビゲーションメニュー、見出し、主要情報など、ウェブサイトの重要なコンテンツを対応言語に翻訳します。
- 選択した言語に関係なく、ユーザーがウェブサイトをシームレスにナビゲートし、やり取りできるようにします。
- 翻訳プロセスを簡素化し、異なる言語間での一貫性を維持するために、言語パックまたはローカリゼーション・ライブラリの使用を検討してください。
- ユーザーが希望する言語を、次回以降のウェブサイト訪問時のデフォルト選択として設定できるようにする。

このマイナーモジュールは、多言語でコンテンツを提供することにより、ウェブサイトのアクセシビリティとインクルーシビリティを高め、多様な国際的視聴者にとってより使いやすいものにするを目的としています。

・ **マイナーモジュール：** 視覚障害ためのアクセシビリティを追加しました。

このマイナーなモジュールでは、あなたのウェブサイトを視覚障害者にとってよりアクセシブルなものにするを目的としています。主な機能は以下の通りです：

- スクリーンリーダーや支援技術のサポート。
- 画像には明確で説明的なaltテキストを付ける。
- 読みやすさを重視したハイコントラストな配色。
- キーボードナビゲーションとフォーカス管理。
- テキストサイズを調整するオプション。
- アクセシビリティ基準を満たすための定期的な更新。

このモジュールは、視覚障害者にとってのウェブサイトの使いやすさを向上させ、アクセシビリティ基準に確実に準拠することを目的としています。

・ **マイナーモジュール：** サーバーサイドレンダリング（SSR）の統合。

このマイナーモジュールでは、サーバーサイドレンダリング (SSR) を統合して、ウェブサイトのパフォーマンスとユーザーエクスペリエンスを向上させることに焦点を当てます。主な目的は以下の通りです：

- SSRを導入し、ウェブサイトの読み込み速度と全体的なパフォーマンスを改善する。
- コンテンツがサーバー上でプリレンダリングされ、ユーザーのブラウザに配信されるようにすることで、最初のページロードを高速化。
- 検索エンジンにプリレンダリングされたHTMLコンテンツを提供することで、SEOを最適化します。
- SSRの利点を享受しながら、一貫したユーザーエクスペリエンスを維持する。

このモジュールは、ページロードの高速化とユーザーエクスペリエンスの向上のためにサーバーサイド・レンダリングを統合することで、ウェブサイトのパフォーマンスとSEOを向上させることを目的としています。

IV.11 サーバーサイドのポン

- **主要モジュール：** 基本的なPongをサーバーサイドPongに置き換え、APIを実装する。

この主要モジュールでは、基本的なPongゲームをサーバーサイドのPongゲームに置き換え、APIを実装することを目標とする。主な特徴と目標は以下の通りです：

- ゲームプレイ、ボールの移動、得点、プレイヤーとのインタラクションを処理するPongゲームのサーバー側ロジックを開発する。
- Pongゲームと対話するために必要なリソースとエンドポイントを公開するAPIを作成し、コマンドラインインターフェイス（CLI）とWebインターフェイスを介してゲームを部分的に使用できるようにする。
- ゲームの初期化、プレイヤーのコントロール、ゲーム状態の更新をサポートするAPIエンドポイントを設計、実装する。
- サーバーサイドのPongゲームがレスポンスよく動作し、魅力的で楽しいゲーム体験を提供できるようにします。
- サーバーサイドのPongゲームをウェブアプリケーションと統合し、ユーザーがウェブサイト上で直接ゲームをプレイできるようにする。

この主要モジュールは、Pongゲームをサーバーサイドに移行することによって、ゲームリソースや機能に簡単にアクセスするためのAPIを提供しながら、ウェブインターフェイスとCLIの両方を通してインタラクションを可能にすることによって、Pongゲームを向上させることを目的としている。

- **主要モジュール：** API統合により、Webユーザーに対してCLI経由でPongゲームプレイを可能にする。

この主要なモジュールでは、ユーザーがウェブ版のゲームを使っているプレイヤーとPongを対戦できるようにするコマンドラインインターフェイス（CLI）を開発することが目標である。CLIはウェブアプリケーションにシームレスに接続し、CLIユーザーがウェブプレイヤーに参加し、対話できるようにする

必要があります。主な機能と目的は以下の通りです：

- ウェブサイト上で利用可能なPongのゲームプレイ体験を再現する堅牢なCLIアプリケーションを作成し、CLIユーザーにPongの試合を開始したり参加したりする機能を提供する。
- APIを使用してCLIとWebアプリケーション間の通信を確立し、CLIユーザーがサイトに接続してWebプレーヤーとやり取りできるようにします。
- CLI内のユーザー認証メカニズムを開発し、CLIユーザーが安全にWebアプリケーションにログインできるようにする。
- CLIとウェブユーザー間のリアルタイム同期を実装し、ゲームプレイのインタラクションをシームレスかつ一貫性のあるものにします。
- CLIユーザーがウェブプレーヤーとPongマッチに参加したり、作成したりできるようにし、クロスプラットフォームのゲームプレイを容易にします。

- ウェブユーザとのポン対戦のためにCLIを効果的に使用方法について、包括的な文書とガイダンスを提供する。

この主要モジュールは、API統合を通じてCLIユーザーとウェブプレーヤーをシームレスに接続するCLIを作成し、統一されたインタラクティブなゲームプレイ環境を提供することで、Pongのゲーム体験を向上させることを目的としています。



このモジュールの希望する場合は、前のモジュールの受講を強く勧める。

第V章 ボーナス

・ パート

このプロジェクトでは、ボーナス・セクションはわかりやすくデザインされている。より多くのモジュールを含めることが要求される。

- ・ 各マイナー・モジュールには5点が与えられる。
- ・ 各主要モジュールに10点が与えられる。



ボーナス・パートは、必須パートがパーフェクトである場合にのみ査定される。パーフェクトとは、必須パートが統合的に行われ、不具合なく動作することを意味します。必須すべてクリアしていない場合、ボーナス・パートは評価されません。

第六章

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業のみが、ディフェンス中に評価されます。ファイル名が正しいか再確認することをお勧めします。



- グローバルな機能やモジュールに対して、即座に完全なソリューションを提供するライブラリやツールの使用は禁止されています。
- サードパーティのライブラリやツールの使用法（can、must、can't）についての直接的な指示は、それに従う必要があります。
- グローバルな機能やモジュールのサブコンポーネントに相当する、シンプルでユニークなタスクを解決する小さなライブラリやツールの使用は認められる。
- 評価中、チームは、対象者が明確に承認していないライブラリやツールの使用法であっても、対象者の制約に反しないものであれば、それを正当化する。
- 評価の間、評価者は自分の責任において、特定のライブラリやツールの使用が正当なものか（許可されているか）、あるいは機能やモジュール全体をほとんど解決していないか（禁止されているか）を定義する。

10001

16997B3EAAA8F965C2C99AD2D1E7C1AD78B53E5AF3CCB21228CB747F6
F9717A4824FFC9EF55AE6E3DC3045320031A4899E0B7A5368BDC2A4B9246
F40E923432A3437CB5703BA745EB9



53616C7465645F5F549209D53E8A1D597DFEF0A034A52CF06BF4132EA61FE
0C907AB090B88EF6E0DF3250E47AB70367C522E351BCD6FE2683B44409CFC
0A0A81D31546431B43D4CE504A554660E0806598302B8F776A1B72407A813
C22D077BADE4F866A7FCF426864823BC25833264770351BC0EF3CD8C47930
FDC58CBA86B49C5445ECD9C840D1C737D17B8D4A5CA935994847DED79443B
3B79475AB2C9506F2C94E9FECC933D1689714AE02FE3C55F278E502E1B806
D7B917906C25612B25700502525825CDC5D23F3856BD9B95F2EC6CA89CBC6
ED26011D60D7CB6EAD304924717BCC06DC297BE7E9B29E7185B23614AB59C
66E2B254C24D5DD6FE54133D4C676F88ED318516F7795B15404A4C014CC12
E5BEAF0A3CBE66F9E1A778AD1F06DA87230D1127DB08E11086F16B1B09905
C31B41262725E8CDD5B18CFB03C6A5CDBF72674C4E76F6FB1851AB7466C5A 4