

第19回 若年者ものづくり競技大会 模倣課題

ウェブデザイン

- Web Technologies -

フロントエンド

作業時間： 3時間

はじめに

このモジュールでは、「Escape Runner」を作成する。

プロジェクトの説明

プレイヤーを操作し、草原を下りながら敵から逃げつつ障害物を避け、アイテムを収集しながらスコアを上げていくゲームを作成する。

障害物に当たると敵が1ブロック近づき、敵と重なるとゲームオーバーとなる。

ゲームの速度など、ゲーム仕様に明示されていない数値については、選手自身が数値を判断し、ゲームの仕様を満たすように設定すること。

テンプレートについて

デザインが適用されたHTMLおよびCSSのテンプレートが提供されるため、そのテンプレートを各自が組み込みながら機能実装を進めていく。

JavaScriptの実装が主な評価対象となり、HTMLおよびCSSの内容については基本的には評価対象ではないが、コンポーネント化、機能分割などの設計実装は評価対象である。

当課題について

本課題は、「[JAVADA 中央職業能力開発協会 第19回 ウェブデザイン 当日公表課題 モジュール1](#)」をベースに作成されています。オリジナルの問題や概念を尊重しつつ、一部改変や拡張を行っております。この過程で、問題の本質を維持しながら、新たな視点や応用力を養うための要素を加えています。参考元の著作者に深く感謝申し上げますとともに、本課題が学習者の理解深化と能力向上に貢献することを願っております。

APIについて

ゲームにおいて必要なデータを取得、格納できるAPIが提供される。更新系APIについては、実際にはデータベースの更新は行われないため、取得系APIのデータは全て固定である。更新系APIが正しく動作しているかどうかは、HTTPステータスコードを以て判断することとする。

- フィールド
 - フィールド取得API
 - 2次元配列のフィールドデータを取得できる
 - 例: [[0,0,0], [1,2,3], ...]
 - 0: 草
 - 1: 障害物
 - 2: スコア加算アイテム
 - 3: 無敵アイテム
 - プレイヤーは上から3ブロック目に固定のため、フィールドデータには含まれない
 - 敵は上から1ブロック目に固定のため、フィールドデータには含まれない
- スコア
 - スコア一覧取得API
 - ニックネーム、スコアの一覧を取得できる
 - スコア投稿API
 - ニックネーム、スコアを投稿できる
 - スコア更新API
 - 投稿したスコアのニックネームを更新できる
 - スコア削除API
 - 投稿したスコアを削除できる

ゲーム基本仕様

1. ゲーム画面

フィールド

- フィールドは1ブロック64px x 64pxで構成されている
- 「フィールド取得API」から取得したフィールド情報をもとに、障害物とアイテムを配置する
 - 配列の最後まで到達した後、配列を再配置する必要はない
- 「草」は縦12ブロック、横3ブロックで構成され、障害物およびアイテムが、1ブロックごとに「下から上」に動く

スコア

- 累計スコアが画面上部に表示される
 - スコアは「8桁」でゼロパディングする
- 1ブロック動くごとに、「10ポイント」加算される

プレイヤー

- プレイヤーは固定の高さに配置され、プレイヤーを左右に移動させることができる
 - 「←キー」で左に移動、「→キー」で右に移動
- スペースキーの押下によりゲームが開始する
 - スペースキーを再度押下した場合は、ゲームが一時停止する
- プレイヤーは上下移動や草から出ることはできない

敵

- 敵は固定の高さに配置され、プレイヤーの動きを追従する
- 敵は障害物やアイテムを無視する
- 障害物やアイテムに接触した時はその画像を削除する

障害物

- 「切り株」に接触すると敵がプレイヤーに1ブロック近づきます。
- プレイヤーと敵が接触するとゲームオーバーとなり、ゲームオーバー画面に遷移する
 - ライブラリの遷移メソッドによりページ遷移すること

2. ゲームオーバー画面

スコア投稿

- スコア結果を表示する
 - ローカルストレージを用いてゲーム画面の最終スコアを表示する
 - スコアは「8桁」でゼロパディングする
- 「スコア投稿」ボタンの押下後、window.promptを用いてニックネームの入力画面を表示し、「スコア投稿API」を用いてスコア結果とニックネームを投稿する
 - スコア投稿後、アラートダイアログで「スコアを投稿しました」を表示する
 - APIエラーが発生した場合には、APIから返却されたエラーメッセージを表示する

スコアランキング

- スコア投稿後、「スコア一覧取得API」からスコア一覧を取得し、スコアの高い順に上位3位のみランキングを表示する
 - スコアは「8桁」でゼロパディングする
 - 今回のスコア結果をランキングに組み込む必要はない

リプレイボタン

- リプレイボタンでゲーム画面に遷移する
 - ライブラリの遷移メソッドによりページ遷移すること

ゲーム追加仕様

ニックネーム更新

- スコア投稿後、「スコア投稿」ボタンを非表示にし、「ニックネーム更新」ボタンを表示する
- 「ニックネーム更新」ボタン押下後、window.promptを用いてニックネームの入力画面を表示し、「スコア更新API」を用いてニックネームを更新する
- APIエラーが発生した場合には、APIから返却されたエラーメッセージを表示する

投稿スコア削除

- スコア投稿後、「スコア投稿」ボタンを非表示にし、「スコア削除」ボタンを表示する
- 「スコア削除」ボタン押下後、window.confirmを用いて確認ダイアログを表示し、「スコア削除API」を用いてスコアを削除する
- APIエラーが発生した場合には、APIから返却されたエラーメッセージを表示する

スコア加算アイテム

- 「花」に接触することで、「100ポイント」スコアが加算される
 - アイテムに接触したあとは、アイテムが消滅する

無敵アイテム

- 「キノコ」に接触することで、1度だけ障害物接触を無効化できる
- アイテム取得後は、プレイヤーの画像が無敵状態だと分かるように色などを変化させる
 - 無敵状態の時に追加でアイテムを取得しても効果は重複しない
 - アイテムに接触したあとは、アイテムが消滅する
- 「切り株」に接触した後、プレイヤーの画像は通常表示に戻り、無敵状態ではない時に「切り株」に接触すると敵がプレイヤーに1ブロック近づく。

フィールド速度上昇

- 5ブロック動くごとに、フィールドの速度が上昇していく

ソースコードの評価

ソースコードは以下の観点で評価する。

- 共通的に利用する機能などは切り分けられ、再利用可能な状態になっていること
- ソースコード管理の観点から、ページだけでなく各パーツがコンポーネントとしてファイルごとに切り分けられ、再利用性や可読性に貢献していること
- JavaScriptにおいてのクラス名やメソッド名、関数名、変数名などが、保守性を考慮し正しく命名されていること
- ソースコードの中で複雑な処理を記述している箇所については、コメントで正しく内容が説明されていること。または規則的なコメントが記述されていること

選手への指示

1. 提供される「src/assets/m1」フォルダ内のファイルを使うことができる
2. パッケージマネージャー（npm）が提供されているが、パッケージのインストールを行うことはできない
3. ライブラリを利用する際は、提供されているライブラリを適切なフォルダに配置し読み込みを行うこと
4. 競技サーバ内の「www/m1」フォルダに「ビルド後」のファイルをアップロードすること完成したウェブサイトは「http://localhost:8081」で表示確認をすること
5. 競技サーバ内の「www/m1」フォルダ内に「_src」フォルダを作成し、ソースコードのデータをアップロードすること
 - ソースコードの採点は「_src」内のファイルを元に実施するため、フォルダ・ファイルが存在しない場合は採点不可となる
 - 「node_modules」など容量の大きい不要なファイルがアップロードされている場合は減点対象となる
6. **競技サーバにアップロードされたデータのみが採点対象となるので注意すること**