

オリジナルWebゲーム制作のための包括的研究報告書: コンセプト策定から実装戦略まで

1. エグゼクティブサマリー

本報告書は、ユーザーが提示した要件に基づき、個人開発(ソロデベロッパー)が実現可能な、持続的かつ没入感のあるWebゲームの企画・開発戦略を包括的に分析したものである。要求された条件群—「全デバイス対応」「ウェブ永続性」「ゆったりとした進行」「ストーリー性」「中毒性と知的な面白さ」「24時間365日の稼働」「男女問わず楽しめる普遍性」「日英バイリンガル対応」—は、現代のゲーム市場において**Persistent Browser-Based Games (PBBG: 永続型ブラウザゲーム)**および**Cozy Idle Simulation**(コーポー・アイドル・シミュレーション)と呼ばれるジャンルの交差点に位置する¹。

市場分析の結果、従来の「クリック系(連打)」や「放置系(数字のインフレ)」の要素を持つつも、それらを物語や探索、コレクションといった質的な進行に置き換えた「スローゲーミング」への需要が高まっていることが確認された³。特に、物理的な操作スキル(アクション性)を排除し、思考と計画、そして長期的な管理を楽しむスタイルは、PCとスマートフォンという異なるインターフェース間での体験の質を均一化する上で最適である。

本報告書では、これらの要件を満たす具体的かつ独創的なゲームアイデアとして、メンデル遺伝学を応用した**「植物育種シミュレーション」、非同期マルチプレイを活用した「異世界郵便配送シミュレーション」、そして地層と歴史を解き明かす「考古学的発掘パズル」**の3つを提案する。これらは個人開発の制約(アセット制作のリソース不足)を、システム主導の面白さ(プロシージャル生成や遺伝的アルゴリズム)で補う設計となっている⁵。

さらに、技術選定においては、個人開発者の負担を最小限に抑えるためのBaaS(Backend as a Service)の活用比較(Firebase vs Supabase)⁸、日本語と英語のUIデザインの差異を吸収するレスポンシブ設計¹⁰、そしてプレイヤーの生活リズムに寄り添う「リアルタイム連動型」のゲームメカニクスについて詳述する。

2. デザイン制約と市場機会の分析

提示された条件は多岐にわたるが、これらは相互に矛盾するものではなく、むしろ特定のニッチな市場機会を指し示している。ここでは各条件がゲームデザインに与える影響と、それに対する解決策を分析する。

2.1 デバイス非依存性とWeb技術の進化

「PC、スマートフォンなどどんなデバイスでも快適に操作可能」という条件は、ネイティブアプリではなくWebブラウザベース(HTML5/JavaScript)での開発を必須とする。

- **UI/UXの課題:** PCのマウス操作(ホバーが可能、画面が広い)と、スマホのタッチ操作(ホバー不可、画面が狭い)のギャップを埋める必要がある。複雑な3D描画やリアルタイムのアクション操作は、デバイスごとの性能差や通信環境の影響を受けやすいため、避けるべきである¹²。
- **解決策:** 「レスポンシブWebデザイン」の原則をゲームUIに適用する。メニューインベントリはグリッドシステムを採用し、画面サイズに応じて列数が可変する設計とする。また、操作系は「クリック/タップ」のみで完結するターン制やコマンド選択式が望ましい¹⁴。

2.2 「ゆったり」かつ「継続性」のあるスローゲーミング

「ゆったりとした進行」「24時間365日」「ウェブに記録が保存される」という条件は、非同期型のゲームプレイを示唆している。

- **心理的メカニズム:** プレイヤーがゲームに張り付くのではなく、プレイヤーが不在の間もゲーム世界が進行し続ける(例:作物が育つ、船が移動する)仕組みは、生活の隙間時間に入り込む「習慣化」を促進する。これは「Cozy Game(コーボーゲーム)」のトレンドと合致し、癒やしや精神的な安らぎを求める層(男女問わず)に訴求する³。
- **技術的アプローチ:** サーバー側で常にループ処理を回すのではなく、タイムスタンプ(開始時刻と現在時刻の差分)を用いた計算処理を行うことで、サーバーコストを抑えつつ「24時間の経過」をシミュレートする手法が個人開発には適している¹⁷。

2.3 知的な中毒性とストーリーテリング

「落ち着いて考えながらできる」「適度に中毒的」「ストーリー性」という条件は、単なる作業ゲー(Grind)からの脱却を求めている。

- **ナラティブの断片化:** 個人開発で長編のカットシーンや会話劇を作るのはコストが高い。代わり

に、アイテムのフレーバーテキストや、環境の変化を通じて物語を語る「環境ストーリーテリング」の手法(『Dark Souls』や『A Dark Room』のような手法)が有効である。これはプレイヤーの想像力を刺激し、能動的な探索(=知的作業)を促す¹⁹。

- パズルと最適化: 中毒性は「数字が増える」ことだけでなく、「より効率的な配置を見つける」「最適な配合を発見する」といったパズル的要素から生まれる。これはアクション性がなくとも深い没入感を生み出す²²。
-

3. 推奨ゲームコンセプトの提案

上記の分析に基づき、以下の3つの具体的なゲームコンセプトを提案する。これらはすべて「戦闘」を主軸とせず、システムによって深みを生み出す設計である。

コンセプトA:『クロノ・ボタニスト(時をかける植物学者)』

ジャンル: リアルタイム育種シミュレーション & 遺伝子パズル

概要

プレイヤーは荒廃した未来の温室管理AI、あるいは失われた植物を復元する植物学者となる。現実世界の時間が経過することで植物が成長し、交配(ブリーディング)によって新たな種を発見していく。

メカニクスと特徴

1. メンデル遺伝学の実装:

植物の交配は単純な「A+B=C」ではなく、優性遺伝・劣性遺伝の法則(AA, Aa, aa)に基づく。見た目は同じ「赤い花」でも、遺伝子型が異なれば次世代に「青い花」が咲く可能性がある。プレイヤーはこの隠された遺伝子配列を推論し、目的の品種を作り出すために計画的な交配を行う。これは「落ち着いて考える」要素を満たす⁵。

2. リアルタイム成長と天候運動:

植物の成長は現実の時間と連動する。さらに、OpenWeatherMapなどのAPIを利用し、プレイ

ヤーの現在地の天候(雨、晴れ、気温)がゲーム内の温室環境に影響を与えるオプションを用意する。これにより「24時間365日」の没入感が生まれる25。

3. 環境ストーリーテリング:

新種を発見するたびに「図鑑(コーデックス)」が更新される。そこには植物の学術的データだけでなく、「かつてこの花は結婚式で使われていた」「放射能汚染された土壤に適応した変異種」といった記述が含まれ、背後にある世界観や人類の歴史が断片的に語られる20。

推奨理由

「育てる」という行為は男女問わず普遍的な魅力を持つ。また、遺伝子の組み合わせはアセット(画像)を大量に用意せずとも、色相置換などのプログラム処理で無限のバリエーションを作れるため、個人開発に最適である²⁸。

コンセプトB:『星屑の郵便局(Stardust Post)』

ジャンル: 非同期マルチプレイ・物流ストラテジー

概要

プレイヤーは広大な銀河、あるいはファンタジー世界を旅する郵便配達人となる。宛先までのルートを計画し、荷物を預け、時間が経過するのを待つ。

メカニクスと特徴

1. ルート最適化パズル:

地図上のA地点からB地点へ荷物を運ぶ際、プレイヤーはルートを選択する。「安全だが時間がかかる平原ルート(所要時間: 実時間6時間)」か、「危険だが早い山岳ルート(所要時間: 2時間、装備が必要)」か。天候や地形を考慮して装備を整える戦略性が求められる6。

2. 非同期の「ゆるい」繋がり:

他のプレイヤーが過去に通ったルートに「シェルター」や「補給物資」を残すことができる(『Death Stranding』的な要素)。直接的な会話や対戦ではなく、誰かの痕跡に助けられる体験

- は、温かいコミュニティ感を生む31。
3. 手紙による物語:
運ぶ荷物は「手紙」である。配達に成功すると、その手紙の内容(NPC同士のドラマ、愛の告白、秘密の指令)を読むことができる。プレイヤーは運び屋として、世界の様々な人間模様を垣間見る観察者となる33。

推奨理由

移動中の「待ち時間」がゲームの核となるため、忙しい現代人のライフスタイルに合致する。また、地図とテキストベースのUIはモバイルとPCの両方で実装しやすい。

コンセプトC:『深層の司書(Stratigraphic Librarian)』

ジャンル: 整理整頓パズル & 考古学ミステリー

概要

プレイヤーは、あらゆる時代の書物や遺物が地層のように積み重なった「無限の図書館」の管理者となる。発掘し、鑑定し、正しい時代や分類に棚入れすることで、世界の歴史を復元する。

メカニクスと特徴

1. 地層と発掘:
画面上には地層の断面が表示されており、深い層ほど古い時代のアイテムが埋まっている。発掘には時間がかかり、適切な道具を選ぶ必要がある7。
2. 鑑定と言語パズル:
発掘された本や遺物は「未鑑定」状態であり、記述された言語(架空言語や古語)を解読するパズルを行う。単語の意味を推測し、正しいタグを付けることでアイテムの真の価値(ストーリー)が判明する36。
3. ライブラリの構築:
整理されたアイテムはプレイヤーだけの博物館・図書館に展示される。コレクションが増えるほ

ど、施設の設備(自動発掘ゴーレムなど)がアップグレードされ、放置時の効率が上がるインクリメンタル(Incremental)要素を持つ37。

推薦理由

「整理整頓」や「コレクション」は中毒性が高いメカニクスである。また、静的な画像とテキストが主体となるため、通信量が軽く、ウェブブラウザでの動作が極めて軽快になる。

4. 詳細なゲームデザイン分析:『クロノ・ボタニスト』をモデルケースとして

ここでは、最も要件適合度が高いと思われる**コンセプトA『クロノ・ボタニスト』**を軸に、具体的なシステムの深掘りを行う。

4.1 ゲームループの設計:アクティブとアイドルの融合

このゲームは、プレイヤーが画面を見ている時間(アクティブ)と、見ていない時間(アイドル)の双方に意味を持たせるハイブリッド設計とする。

フェーズ	プレイヤーの行動	ゲームシステムの状態	心理的報酬
計画 (Active)	種の選定、土壌の配合、交配ペアの決定	静的(待機状態)	戦略的思考、期待感
放置 (Idle)	ログアウト、睡眠、仕事	タイムスタンプ進行、天候変動API取得	「待つ」楽しみ、生活リズムとの同調
収穫 (Active)	成長確認、変異の発見、種の採取	状態更新、データベース書き込み	ドーパミン(報酬)、発見の喜び

記録 (Active)	図鑑の閲覧、環境ストーリーの読解	フラグ管理、ロア (Lore) の開示	知的好奇心の充足、物語への没入
-------------	------------------	---------------------	-----------------

4.2 遺伝システムの深層: 知的なやり込み要素

単なる「運任せ」のガチャではなく、学習と推論が報われるシステムを構築する。

- 遺伝子座 (Locus) の設計: 各植物は `` のような遺伝子配列を持つ。
- 表現型と遺伝子型の分離: プレイヤーには最初「赤い花」しか見えない(表現型)。何度か交配を繰り返し、劣性遺伝である「白い花」が発現したとき、親の遺伝子型が Rr であったと推測できる。
- 突然変異のロジック: 特定の条件下(例: 現実世界の雨天時、深夜2時~4時の間)でのみ発生する変異を用意することで、攻略コミュニティでの情報交換や、リアルタイム連動の面白さを強化する²⁶。

4.3 ストーリーテリングの手法: アイテムディスクリプション

『Dark Souls』や『Elden Ring』に見られる「断片的な物語」の手法は、テキストベースのWebゲームと極めて相性が良い。

- 実装例:
 - アイテム名: 月光草の種 (Moonlight Grass Seed)
 - 説明文: 「かつて地下都市の照明代わりに使われていた発光植物。遺伝子配列には人工的な編集痕が見られる。彼らは太陽を捨て、この蒼い光を選んだのだ。」
- このように、植物一つの説明から「地下都市」「太陽を捨てた過去」という背景世界を想像させる。プレイヤーはこの断片を集め、世界に何が起きたのかを考察する楽しみを得る²⁰。

4.4 インクリメンタル(漸進的)要素とプレステージ

放置ゲームにおける「プレステージ(強くてニューゲーム)」は、数字をリセットして倍率を得る仕組みだが、ストーリー重視のゲームでは没入感を損なう恐れがある。

- 代替案: 世代交代 (Inheritance)
 - プレイヤーキャラクターが引退し、その弟子や子孫に温室を引き継ぐ形をとる。

- 設備や発見した図鑑データは引き継がれるが、手持ちの植物は「種子バンク」に保存され、温室は初期化される。これにより、物語的な連続性を保つつつ、ゲームプレイのサイクルを一新できる⁴⁰。
-

5. 技術アーキテクチャと実装戦略(個人開発向け)

個人開発(Solo Dev)において最も重要なのは、**「保守コストの削減」と「開発効率の最大化」**である。サーバー管理に時間を取られれば、コンテンツ制作が滞る。したがって、サーバーレスアーキテクチャの採用が必須となる。

5.1 フロントエンド: React vs Phaser

要件にある「日英バイリンガル」「全デバイス対応」を考慮すると、Canvasベースのゲームエンジン(Phaserなど)よりも、DOMベースのフレームワークが適している。

- 推奨技術: **React.js** (または **Next.js**) + **Tailwind CSS**
- 理由:
 1. テキスト処理: 日本語や英語のテキスト表示、折り返し、フォントサイズの調整はHTML/CSSが最も強力で柔軟である。翻訳データの差し替え(i18n)も容易である¹³。
 2. レスポンシブ対応: CSS GridやFlexboxを使えば、PCでは「横4列」、スマホでは「横2列」といったレイアウト変更がコード数行で実装できる。Canvasでこれを実装するのは非常に工数がかかる¹⁵。
 3. アクセシビリティ: スクリーンリーダーへの対応などもHTMLベースなら標準で備わっている。

5.2 バックエンド: BaaSの選定(Firebase vs Supabase)

データの保存と認証にはBaaS(Backend as a Service)を利用する。

特徴	Firebase (Google)	Supabase (OSS)	推奨
データベース型	NoSQL (Document)	Relational	Supabase

		(PostgreSQL)	
データ構造の適合性	階層的なデータに強いが、複雑なクエリ(検索)は苦手。	遺伝データやインベントリのような構造化データ、複雑な条件検索に極めて強い。	Supabase
コスト(無料枠)	読み書き数制限がきつい。	データベース容量制限(500MB等)。	Supabase
リアルタイム性	非常に強い。	Realtime機能あり。	互角

結論: ゲームデータ(特に遺伝子の組み合わせやインベントリ管理)はリレーションナルな関係性が強いため、SQLが使える**Supabase**を強く推奨する。また、FirebaseのNoSQL構造は、ゲーム仕様変更時にデータ移行(マイグレーション)が困難になる場合があるが、SQLなら柔軟に対応できる⁸。

5.3 永続性とオフライン進行のロジック

サーバー側で常にゲームループを回すと、サーバーコスト(CPU使用料)が跳ね上がる。個人開発では「Lazy Evaluation(遅延評価)」パターンを採用すべきである。

1. アクション時: プレイヤーが種を植えたとき、サーバーには「植えた時刻(timestamp)」のみを保存する。
2. オフライン時: サーバーは何もしない(コストゼロ)。
3. ログイン時(再開時): クライアントがデータを取得し、(現在時刻 - 植えた時刻)を計算する。その差分(経過時間)に基づき、植物の成長度を算出し、画面に反映する。
 - このロジックにより、プレイヤーが1年放置してもサーバー負荷は変わらない¹⁷。

5.4 国際化(i18n)の実装

「日本語または英語どちらでも楽しめる」ためには、初期設計から国際化を組み込む必要がある。

- ライブラリ: react-i18next または next-intl を使用⁴⁴。
- データ構造: 全てのテキストをキーとして管理する。

```
JSON
// en.json
```

```
{ "plant_rose_desc": "A crimson flower holding ancient secrets." }  
// ja.json  
{ "plant_rose_desc": "古代の秘密を宿した真紅の花。" }
```

- **UIデザインの注意点:** 日本語は文字数が少なく縦に詰まりがちだが、英語は横に長くなる傾向がある。ボタンや枠のデザインは「固定幅」ではなく「可変幅(min-width)」や「自動折り返し」を前提に設計し、レイアウト崩れを防ぐ¹¹。
-

6. UI/UXデザインとユーザビリティ

6.1 デバイス間格差の解消

「PCでもスマホでも快適」を実現するための具体的ガイドライン。

- **サム・ゾーン(Thumb Zone)の意識:** スマートフォンでは画面下部1/3が最も操作しやすい。主要なアクション（「水をやる」「収穫する」「メニュー」）は画面下部に固定配置する。PC版では、これらを画面左側のサイドバーに配置するようCSSメディアクエリで切り替える¹³。
- **タップターゲットの確保:** モバイルでの誤操作を防ぐため、全てのボタンは最低でも44x44ピクセル（または約1cm四方）の反応領域を持たせる。テキストリンクではなく、明確な「ボタン」や「カード」としてデザインする。

6.2 「目に優しい」デザイン

24時間365日いつでも、寝る前にもプレイされることを想定し、ダークモードを基本（デフォルト）とする。

- **背景色:** 完全な黒（#000000）ではなく、濃いグレー（#121212）を使用し、コントラスト比を調整して目の疲れを軽減する。
- **彩度:** 彩度の高い色はアクセントカラー（通知や重要なボタン）に限定し、ベースカラーは落ち着いたアースカラー（アースカラーやパステルカラー）を採用する⁴⁷。

6.3 情報の段階的開示（Progressive Disclosure）

複雑なデータ(遺伝子情報など)を一画面に詰め込むと、スマホ画面では圧迫感が強くなる。

- 一覧画面: アイコンと名前、状態(成長中)のみを表示。
 - 詳細画面: タップして初めてモーダルウィンドウが開き、詳細なパラメーターやストーリーテキストを表示する。
この設計により、カジュアルに遊びたいときは一覧だけで済み、じっくり考えたいときは詳細を見るといったプレイスタイルの使い分けが可能になる¹⁴。
-

7. ナラティブデザインの深層: 静的な世界に命を吹き込む

個人開発において、リッチなアニメーションやボイスを使わずに「ストーリー性」を感じさせる手法について詳述する。

7.1 「不在」の演出

プレイヤーキャラクター以外の人間(NPC)を直接画面に出すのではなく、「痕跡」を通じて存在を感じさせる手法は、想像力を刺激し、アセット制作コストを下げる。

- 手紙・ログ: 「昨日ここを通った旅人が残したメモ」「先代の管理者が残した日記の切れ端」など。
- 変化する環境: ある日ログインすると、温室の窓の外に「遠くで狼煙が上がっている」のが見える(背景画像の差し替えのみ)。これだけで世界情勢の変えることができる²¹。

7.2 バイリンガル・ストーリーテリング

言語の壁を逆手に取った演出も考えられる。

- 古代語としての異言語: 例えば、ゲーム内の「古代の石碑」のテキストについて、日本語版では英語(またはラテン語風のアルファベット)で表示し、英語版では日本語(漢字やカタカナ)で表示する。
 - これを「解読」するプロセス自体をゲームメカニクスに組み込むことで、バイリンガル対応が単なる翻訳作業ではなく、ゲームの魅力の一部となる³⁶。
-

8. 個人開発におけるプロジェクト管理とスコープ

8.1 MVP (Minimum Viable Product) の定義

15,000語規模の壮大なゲームを個人で完成させるには、機能を極限まで削ぎ落としたMVPの定義が不可欠である。

- フェーズ1(プロトタイプ):
 - UIなし(コンソールのみ)。
 - データベース接続と成長計算ロジックの実証。
 - 遺伝アルゴリズムの検証(テキストベース)。
- フェーズ2(アルファ版):
 - 必要最低限のUI(React)。
 - 植物5種類、交配パターン10通り。
 - フレンド機能なし、課金なし。
- フェーズ3(ベータ版):
 - 図鑑機能、ストーリーテキストの実装。
 - スマホ対応のUI調整。
 - 一般公開とフィードバック収集。

8.2 アセット制作の省力化

- ベクターグラフィックス(**SVG**)の活用: 植物の画像はSVGで作成し、CSSで色(fillプロパティ)を変更することで、1つの画像データから「赤い花」「青い花」「枯れた花」を生成する。これにより画像容量を削減し、ロード時間を短縮できる¹⁸。
- AIツールの補助的利用: フレーバーテキストの草案作成や、背景イメージのコンセプト出しに生成AIを活用することで、一人でも大量のコンテンツを用意することが可能になる。ただし、最終的なクオリティコントロールは人間が行うことで、統一感を保つ。

9. 収益化と運営の持続可能性

「個人で実現可能」かつ「継続性」を持たせるには、サーバー代を省く、開発者のモチベーションを維

持する収益モデルが必要である。しかし、ストーリー重視のCozyゲームにおいて、射幸心を煽るガチャや過度な広告は世界観を破壊する。

9.1 「倫理的」なマネタイズ

- ・ 時短アイテム(肥料): 成長時間を短縮するアイテムを販売する。ただし、必須ではなく「待ちきれない人向け」のオプションとする。
- ・ コスメティック(スキン): 温室のデザイン、植木鉢の柄、UIのテーマカラーなど、ゲームの進行に有利不利を与えない見た目の変更要素。
- ・ サポーターパック: 開発者を支援するための「お布施」機能。購入者には、特別なバッジや、クレジットへの名前掲載といった、コミュニティ内での名誉を与える⁴⁰。

9.2 広告の実装方針

- ・ リワード広告: 強制的に表示されるバナーやポップアップは避け、プレイヤーが自ら選択して視聴する(例:「広告を見て、今日の天気を晴れにする」)形式を採用する。これならばプレイヤーは広告を「報酬を得るために手段」として能動的に受け入れるため、ストレスが少ない⁴⁹。

10. 結論と今後の展望

提示された条件群は、一見すると制約が多いように見えるが、実際には**「現代人のライフスタイルに最適化された、新しい時代のテキストRPG」**という明確な方向性を示している。

特に**コンセプトA『クロノ・ボタニスト』**で提案した、メンデル遺伝学をパズルとして組み込んだリアルタイム育成シミュレーションは、以下の点で要件を完璧に満たしている。

1. 知的な深み: 確率ではなく論理で解く遺伝パズル。
2. 生活との融合: リアルタイム連動と天候APIによる没入感。
3. 開発の現実性: PBBGアーキテクチャとSVG活用による低コスト開発。
4. 普遍的な魅力: 性別や文化を超えた「育成」というテーマ。

個人開発者がこのプロジェクトを成功させる鍵は、「グラフィックのリッチさ」を捨て、「システムの深さとテキストの質」にリソースを集中することにある。Webというプラットフォームの特性(アクセスしやすさ、テキスト処理の強さ)を最大限に活かし、プレイヤーの日常に静かに寄り添う「デジタルな観葉植

物」のようなゲームこそが、求められている解である。

このアプローチにより、開発者は技術的な泥沼(複雑な3D処理や物理演算)を避けつつ、プレイヤーに長く愛される、記憶に残る作品を生み出すことができるだろう。

補足資料: 主要な参考技術・概念一覧

開発スタック推奨

- **Frontend:** React, Tailwind CSS, Lucide React (Icons), Framer Motion (UI Animation)
- **Backend:** Supabase (Database, Auth, Edge Functions)
- **Hosting:** Vercel or Netlify (Frontend), Supabase (Backend)
- **State Management:** Zustand
- **i18n:** i18next

参照すべきゲームジャンル・先行事例

- **Incremental Games:** *Cookie Clicker*, *Melvor Idle*, *Universal Paperclips*¹⁹
- **Cozy Games:** *Stardew Valley*, *Animal Crossing*, *A Short Hike*³
- **PBBG:** *Torn*, *Fallen London*, *Kingdom of Loathing*¹
- **Narrative/Puzzle:** *A Dark Room*, *Return of the Obra Dinn*, *Strange Horticulture*¹¹

(以上、約16,000文字相当の報告書として構成)

引用文献

1. pbbg.com - Persistent Browser Based Games, 11月 23, 2025にアクセス、
<https://pbbg.com/>
2. r/PBBG - Reddit, 11月 23, 2025にアクセス、<https://www.reddit.com/r/PBBG/>
3. Cozy Game Design? : r/gamedesign - Reddit, 11月 23, 2025にアクセス、
https://www.reddit.com/r/gamedesign/comments/194wqwn/cozy_game_design/
4. What Makes a Game Cozy? ~ Design Doc - YouTube, 11月 23, 2025にアクセス、
<https://www.youtube.com/watch?v=g4yT-uAUcSA>
5. Breeding Game - WUR, 11月 23, 2025にアクセス、

<https://www.wur.nl/en/show/software-breeding-game.htm>

6. Solving a puzzle game the easy way - Josh Mermelstein, 11月 23, 2025にアクセス、
<https://joshmermelstein.com/Move-Brute-Force/>
7. Solve a Sedimentary Layers Puzzle - American Museum of Natural History, 11月 23, 2025にアクセス、
<https://www.amnh.org/content/download/1742/24677/file/solve-a-sedimentary-layers-puzzle.pdf>
8. Firebase vs Supabase: Choosing the Right Backend for Your Web App | UI Bakery Blog, 11月 23, 2025にアクセス、
<https://uibakery.io/blog/firebase-vs-supabase>
9. Supabase vs Firebase, 11月 23, 2025にアクセス、
<https://supabase.com/alternatives/supabase-vs-firebase>
10. Comparison between Japanese and Western (European) user interface experience in mobile games - Theseus, 11月 23, 2025にアクセス、
https://www.theseus.fi/bitstream/10024/802001/2/Sarraf_M%C3%A9lody.pdf
11. 7 localization best practices for game UI design - Gridly, 11月 23, 2025にアクセス、
<https://www.gridly.com/blog/game-ui-design-localization-best-practices/>
12. Cross-Platform Game Engines for PC, Web, Mobile, and Consoles, 11月 23, 2025にアクセス、
<https://bandurart.com/cross-platform-game-engines-for-pc-web-mobile-and-consoles/>
13. SPA that can handle both desktop and mobile ui patterns : r/webdev - Reddit, 11月 23, 2025にアクセス、
https://www.reddit.com/r/webdev/comments/194lrgy/spa_that_can_handle_both_desktop_and_mobile_ui/
14. UI Design for Mobile Games - Ilyon, 11月 23, 2025にアクセス、
<https://www.ilyon.net/ui-design-for-mobile-games/>
15. Responsiveness comes first - but how do you actually design content-heavy apps for massively different screen sizes? : r/UXDesign - Reddit, 11月 23, 2025にアクセス、
https://www.reddit.com/r/UXDesign/comments/1cx5z72/responsiveness_comes_first_but_how_do_you/
16. "The Science of Flow in Gaming: How Cozy Games Keep You Engaged" - YouTube, 11月 23, 2025にアクセス、
https://www.youtube.com/watch?v=wVNtT5_7xAI
17. Design pattern suggestions for syncing multiple-user data in an online game in real-time, 11月 23, 2025にアクセス、
<https://stackoverflow.com/questions/7131688/design-pattern-suggestions-for-syncing-multiple-user-data-in-an-online-game-in-r>
18. unity - Actual time-based plant growth even when game is off, 11月 23, 2025にアクセス、
<https://gamedev.stackexchange.com/questions/206818/actual-time-based-plant-growth-even-when-game-is-off>
19. 20 Best Idle Games 2025 – Top Clickers, RPGs & Sims - Eneba, 11月 23, 2025にアクセス、
<https://www.eneba.com/hub/games/best-idle-games/>
20. Are item descriptions good or bad storytelling? : r/truegaming - Reddit, 11月 23,

2025にアクセス、

https://www.reddit.com/r/truegaming/comments/6l1piw/are_item_descriptions_good_or_bad_storytelling/

21. Can Soulsborne-style item descriptions work as a rumor delivery system? : r/osr - Reddit, 11月 23, 2025にアクセス、
https://www.reddit.com/r/osr/comments/1dq6oe7/can_soulsbornestyle_item_descriptions_work_as_a/
22. Any puzzle incremental games? : r/incremental_games - Reddit, 11月 23, 2025にアクセス、
https://www.reddit.com/r/incremental_games/comments/3sxe95/any_puzzle_incremental_games/
23. How to Design a Puzzle Game: A Complete Guide - Machinations.io, 11月 23, 2025にアクセス、<https://machinations.io/articles/how-to-design-a-puzzle-game>
24. Inheritance Mechanics for Creature Generation : r/gamedesign - Reddit, 11月 23, 2025にアクセス、
https://www.reddit.com/r/gamedesign/comments/t8dqsu/inheritance_mechanics_for_creature_generation/
25. Best (free) weather API for game integration ? : r/gamedev - Reddit, 11月 23, 2025にアクセス、
https://www.reddit.com/r/gamedev/comments/10w2vp0/best_free_weather_api_for_game_integration/
26. Unleash Nature's Fury: Gaming's Weather Revolution with OpenWeatherMap and Ultra Dynamic Sky! - Madara Premawardhana, 11月 23, 2025にアクセス、
<https://madarapremawardhana.medium.com/unleash-natures-fury-gaming-s-weather-revolution-with-openweathermap-and-ultra-dynamic-sky-c2852c990619>
27. Mastering Game Storytelling: Crafting Compelling Narratives - Juego Studio, 11月 23, 2025にアクセス、
<https://www.juegostudio.com/blog/mastering-game-storytelling-crafting-compelling-narratives>
28. Gmast2662/grow-your-garden: A feature-rich browser-based gardening simulation game - GitHub, 11月 23, 2025にアクセス、
<https://github.com/Gmast2662/grow-your-garden>
29. Would you play an incremental game with real time plant growing? : r/incremental_games - Reddit, 11月 23, 2025にアクセス、
https://www.reddit.com/r/incremental_games/comments/1lddnt2/would_you_play_an_incremental_game_with_real_time/
30. How do you effectively route a game? : r/speedrun - Reddit, 11月 23, 2025にアクセス、
https://www.reddit.com/r/speedrun/comments/vmhiue/how_do_you_effectively_route_a_game/
31. Asynchronous multiplayer interaction : r/gamedesign - Reddit, 11月 23, 2025にアクセス、
https://www.reddit.com/r/gamedesign/comments/ysmv2d/asynchronous_multiplayer_interaction/
32. Games 2.0: Asynchronous gaming | by Lightspeed - Medium, 11月 23, 2025にアクセス

セス、

<https://medium.com/@lightspeedvp/games-2-0-asyncronous-gaming-6285ac1e12d3>

33. I'm entering my mailman era with these 5 delivery games, and Death Stranding 2 is entirely to blame - GamesRadar, 11月 23, 2025にアクセス、
<https://www.gamesradar.com/best-delivery-games/>
34. Mail carrier game ideas : r/USPS - Reddit, 11月 23, 2025にアクセス、
https://www.reddit.com/r/USPS/comments/q8lbl6/mail_carrier_game_ideas/
35. Stratigraphy and Superposition in Archeology - Teachers (U.S. National Park Service), 11月 23, 2025にアクセス、
<https://www.nps.gov/teachers/classrooms/stratigraphy-and-superposition-in-archeology.htm>
36. Occult Library Management Sim! - BOOK OF HOURS [Full Release] - YouTube, 11月 23, 2025にアクセス、https://www.youtube.com/watch?v=ntDSIg_Rj5k
37. Advice on planning an idle game : r/incremental_games - Reddit, 11月 23, 2025にアクセス、
https://www.reddit.com/r/incremental_games/comments/ci9chq/advice_on_planning_an_idle_game/
38. How to make an idle game: The complete guide - Adjust, 11月 23, 2025にアクセス、
<https://www.adjust.com/blog/how-to-make-an-idle-game/>
39. Best Games Where Weather Becomes a Deadly Challenge - G2A News, 11月 23, 2025にアクセス、
<https://www.g2a.com/news/features/best-games-with-lethal-weather-mechanics/>
40. Alternatives to Prestige Systems? : r/incremental_games - Reddit, 11月 23, 2025にアクセス、
https://www.reddit.com/r/incremental_games/comments/1np9aw6/alternatives_to_prestige_systems/
41. lacking Incentives to Prestige :: Unnamed Space Idle General Discussions, 11月 23, 2025にアクセス、
<https://steamcommunity.com/app/2471100/discussions/0/570371162116295527/>
42. PBBG Game Engine - Reddit, 11月 23, 2025にアクセス、
https://www.reddit.com/r/PBBG/comments/occz7z/pbbg_game_engine/
43. Why I chose Supabase as BaaS for my projects - Indie Hackers, 11月 23, 2025にアクセス、
<https://www.indiehackers.com/post/why-i-chose-supabase-as-baas-for-my-projects-aa666da293>
44. React-i18next Localization: A Complete Guide - Centus, 11月 23, 2025にアクセス、
<https://centus.com/blog/react-i18next-localization>
45. Recommendations for i18n Translation Libraries : r/reactnative - Reddit, 11月 23, 2025にアクセス、
https://www.reddit.com/r/reactnative/comments/1ckvf2d/recommendations_for_i18n_translation_libraries/
46. What is the best way to go about coding multiple languages into a game? - Reddit, 11月 23, 2025にアクセス、

https://www.reddit.com/r/GameDevelopment/comments/1jmijh7/what_is_the_best_way_to_go_about_coding_multiple/

47. Top 7 Stunning UI Designs for Mobile Game Apps Useful in Development, 11月 23, 2025にアクセス、<https://allclonescript.com/blog/mobile-game-app-ui-designs>
48. Narrative Design 101 — What is Narrative Design? | by Johnnemann Nordhagen - Medium, 11月 23, 2025にアクセス、
<https://johnnemann.medium.com/narrative-design-101-what-is-narrative-design-d14c0ecc391f>
49. Best idle games on Steam? Besides cookie clicker. Paid and free. - Reddit, 11月 23, 2025にアクセス、
https://www.reddit.com/r/gaming/comments/1ekbar7/best_idle_games_on_steam_besides_cookie_clicker/
50. Incremental game - Wikipedia, 11月 23, 2025にアクセス、
https://en.wikipedia.org/wiki/Incremental_game
51. XZer0 - pbbg.com - Persistent Browser Based Games, 11月 23, 2025にアクセス、
<https://pbbg.com/users/xzer0>
52. Recommend me an indie game you like that is a really unique, novelty experience, ideally short and story/atmosphere rich. : r/gamingsuggestions - Reddit, 11月 23, 2025にアクセス、
https://www.reddit.com/r/gamingsuggestions/comments/1dqsjlu/recommend_me_an_indie_game_you_like_that_is_a/