

# ゲーム制作関連

ゲーム制作するにあたってなんやかんや

- [アイデア出し～プロトタイピング～製作開始](#)
- [ゲーム制作計画書 \(エンジニア志望向け\)](#)

# アイデア出し～プロトタイピング ～製作開始

## ☑ 全体フロー（5ステップ）

フェーズ	内容	出力物
① コンセプトエンジニアリング	テーマ・体験価値・独自性を設計	コンセプトシート
② アイデア出し・検証	発想を広げ、成立性・技術的可能性を検討	アイデアマップ／プロト案
③ プロトタイピング	コア要素を最小限で実装	動作するプロトタイプ
④ 作品制作	完成度・見た目・操作感を磨く	就職用プレイアブル作品
⑤ プレゼン資料作成	開発意図・技術アピールを可視化	ポートフォリオ+技術レポート

## ☑ ① コンセプトエンジニアリング

「どんな体験を作るか」を“設計”する段階です。

要素	質問例	メモ
体験価値	プレイヤーにどんな感情を与えたい？	楽しい／怖い／気持ちいい／頭を使う
コアメカニクス	何をするゲーム？	移動・撃つ・つなぐ・押す・傾ける etc.
差別化点	他の作品と何が違う？	見た目・物理感・操作感・AIなど
技術テーマ	自分がアピールしたい技術は？	AI、物理、レンダリング、地形、ネット etc.

☑ 出力：

→ A4 1枚の「コンセプトシート（体験+技術）」

例：「傾く地形に合わせて姿勢を変えるAIキャラの群れアクション」

## ☑ ② アイデア出し・検証

ブレズ的に考えますが、評価軸を明確にして選ぶのがポイント。

手法	内容	目的
マンダラート/KJ法	思いつく関連語を広げる	着想の拡散
スケッチ・モック	紙や簡易UIで操作を確認	体験の可視化
技術検証	UnityやDX11で部分的に試作	実現可能性の確認

☑ 選定の基準：

- 自分の得意技術が活かせるか？
- 2～3か月で完成できる規模か？
- 体験が一言で説明できるか？

## 🔧 ③ プロトタイピング

最小限のシステムで「面白さの核」を確かめます。

作業	内容
コアループ実装	プレイヤーの操作→結果→フィードバック
簡易アセット	立方体・テキスト・線で十分
フィードバック検証	教員・友人に触ってもらう
記録	動画キャプチャを残す（ポートフォリオで使える）

☑ ここで「技術的にどう作ったか（AI、制御、描画など）」を説明できるようにするのが就職で強いです。

## ☒ ④ 作品制作

完成度と見栄えを上げるフェーズ。

要素	内容
グラフィック	テクスチャ、シェーダー、UI整備
サウンド	BGM・効果音の統一感
難易度調整	クリア体験のテンポ調整
デバッグ	プレイヤー導線・バグ修正
動作環境	EXE出力 or WebGL or 動画作品化

# ☒ ゲーム制作計画書（エンジニア志望向け）

✓ その通りって訳じゃなく、あくまで構成のための参考だよ！

## 1☒ 表紙・概要

- プロジェクト名（ゲームタイトル）
- チーム名・担当メンバー（役割明記）
- 制作期間・使用ツール（Unity / Unreal / 自作エンジン / DirectX11 など）
- 開発ターゲット（PC / Switch / Web / モバイル）

“☒ 「技術×企画」を見せたい学生は、ここで“技術的特徴”の一言キャッチを添えると印象的  
例：「自作物理エンジンで動くローグライクRPG」

## 2☒ ゲームコンセプト(1枚)

- ゲームジャンル・プレイスタイル  
（例：3Dアクション／2Dパズル／シミュレーション）
- 一言コンセプト（30字以内）
- 狙い（他作品との違い／開発動機）

“☒ “なぜこのゲームを作るのか”を1文で。  
例：「地形傾斜の物理処理を活かしたプレイヤー体験を作りたい」

## 3☒ ゲーム概要図(1～2枚)

- スクリーンレイアウト図・操作イメージ
- コアとなる1プレイループ（図解）
- プレイヤーの目的・勝敗条件

“☒ ここでは「プレイ体験を定義」し、企画力+理解力を示す。

## 4☒ ゲームシステム仕様

- プレイヤー行動一覧（移動・攻撃・アイテム・カメラなど）
- 敵AI・ギミック仕様
- ステージ構造・進行フロー（図で示す）
- ルール・スコアシステム

“☒ エンジニア志望はここで“システム構造を技術的に説明”できると強い。  
例：「各オブジェクトはBaseObjectクラスを継承し、Update/Drawをオーバーライド」

## 5☒ 技術構成・開発アーキテクチャ(超重要☒)

“ここがエンジニア学生の“見せ場”です。

- 使用エンジン・ライブラリ (Unity / Unreal / DxLib / 自作エンジン)
- 技術的特徴・工夫
  - 自作シェーダー (影・ポストエフェクト・反射など)
  - FBXローダー、モーション再生
  - 衝突判定・物理シミュレーション
  - スクリプトエンジン・イベント駆動
  - データ構造 (マップエディタ / CSV読み)
- アーキテクチャ図 (クラス構成・モジュール図)

“☑ 企業は「どんな仕組みで動いているか」を知りたい。  
UML風でも手書きでもOK。自分のコアスキルを中心に配置するとGood。

## 6☑ 開発環境・ツールフロー

- 使用ツール一覧  
(IDE: Visual Studio / Blender / Photoshop / GitHub など)
- バージョン管理・ブランチ運用方法
- ビルド・テスト・デバッグ手順
- チーム制作の場合：役割分担と連携方法 (Gitフロー図など)

“☑ 「チームでの技術運用力」も評価対象。

## 7☑ 開発スケジュール・進捗管理

- ガントチャート形式で工程を明示  
(企画 → プロトタイプ → α版 → β版 → 完成)
- 自身の担当範囲と目標
- チェックポイント (技術課題・到達目標)

“☑ “実行力”と“見通しの良さ”を評価されるポイント。

## 8☑ 技術デモ・スクリーンショット

- 実際の動作画面・技術デモGIF・短い動画キャプチャ
- 特に「自分が実装した部分」を強調

“☑ 「見た目だけでなく、裏で動いている仕組み」も簡潔に添えると印象的。

## 9☑ 想定課題・今後の展望

- 実装で苦戦している部分
- 今後挑戦したい技術 (AI、ネットワーク、最適化など)
- 完成後の展開 (次のステップ・ポートフォリオ化)

“☑ 「成長意欲・技術課題への理解」を示す項目です。

## ☑ まとめ・自己アピール

- 「この作品で見せたい技術力」

- 「エンジニアとしての強み・興味分野」
- 「次に挑戦したい開発領域」

“ ☒ 例：「DirectX11を自作エンジンとして統合し、リアルタイムシャドウを実装。将来的にはレンダリングエンジニアを目指したい」

## ☒ 提案：スライド構成例（全体の流れ）

スライド番号	内容
1	表紙・概要
2	コンセプト
3	ゲーム概要図
4-5	システム仕様
6-8	技術構成・アーキテクチャ
9	開発環境・ツール
10	スケジュール
11	技術デモ
12	課題・今後の展望
13	自己アピール・まとめ