

ゲーム会社に応募した時何見られるんだ？

1. 動画構成（見せ方）の評価

採用側は動画全編を丁寧には見ません。
多くは以下の流れになります：

1. 最初の10秒で「面白そうか / センスあるか」を判定
2. 1分程度で「技術力があるか」を判断
3. 最後まで視聴するのは高評価時のみ

評価されやすい構成：

- 最初にタイトル・コンセプト明示
- 実際のプレイ動画がメイン
- テキスト字幕で開発内容を補足
- 動作が見やすいカメラ / UI

マイナス評価：

- 作品の説明が長く、映像が出ない
- 文字が小さい・速すぎる
- 見せたいポイントが不明瞭

2. 技術アピール度の評価

ゲーム会社が知りたいポイントは

- どこまで自作？
- ライブラリは？
- エンジンは？
- アートは自作？
- チーム制作か個人？
- 実装難易度は？
- ゲームとして成立しているか？

高評価される書き方例：

- ・ゲームエンジン：自作（DX11ベース）
- ・スキニング：CPU / GPU対応
- ・モデル読み込み：ufbx利用
- ・地形：自前生成（法線計算あり）
- ・影：丸影を独立クラスで実装 ← 重要

3. クリエイティブ・デザイン評価

採用側は以下を見ることが多い：

- 絵作りのセンス
- UIのバランス
- 色と配置
- キャラの魅力・世界観
- 画面レイアウト
- 可読性
- フォーカスポイント
- ゲームとしての面白さの匂い

4. 就活動画として強い・弱いか

ゲーム業界向けの評価観点：

観点	求められるもの	備考
作品完成度	バグ少・動作安定	企業は安定動作を見る
技術力	内部構造の理解	黙っていても伝わる情報量
美術的センス	視覚的訴求力	画面に気持ちよさがある
UI/UX	遊びやすさ	プレイヤー視点
ゲームデザイン	面白くできる人か	ただ動くだけはNG
プレゼン能力	説明の明瞭さ	チーム内で会話できるか

1. プログラマー応募として強いアピール点になる要素

A. 自作フレームワーク+DxLib

企業の評価：

- Unityではなく低レベル環境で作れる = 基礎能力が高い
- メモリ管理やパフォーマンス感覚がある
- ゲームループや描画制御を正しく理解している
- ライブラリの内部処理に意識が届いている

この時点で、多くの応募学生との差別化になります。

2. アピールしたい項目に対する、企業側の評価視点

A. カメラワーク

企業が見たいのは：

- カメラがキャラを追尾するだけか？
- 画面の見やすさ、プレイしやすさに貢献しているか？
- 視界確保が適切か？
- 曲線的な補間 (Lerp/Slerp) があるか？
- 障害物による視界遮断対策があるか？ (RayCast補正)
- カメラ揺れ・演出制御があるか？
- プレイヤーの入力に応じた最適画角変更があるか？

さらに評価される書き方例：

- ・ TPSカメラにおいて、RPG-7カメラ問題を回避するため RayCastで視界遮断判定を行い、距離自動補正を実装
- ・ ターゲット注視時はlook-at制御による滑らかな補間

B. 敵の移動アルゴリズム

採用側の評価ポイント：

- 単純な追尾か？
- プレイヤー位置予測をしているか？
- A* や Dijkstra を使っているか？

- 回避行動 / 包囲行動はあるか？
- 目的関数で行動選択しているか？
- ステートマシン？
- 振る舞い木（ビヘイビアツリー）？

高評価される言い方：

- ・ 敵AIは状態遷移（FSM）で管理
例：巡回 → 索敵 → 追跡 → 攻撃 → 回避
- ・ プレイヤーが視界圏内に入るとRayCastで視線判定
- ・ 目標位置は逐次補間し、振動や細かい震えを抑制

C. RayCastでの武器軌道

企業はこう見る：

- ただ当たり判定ではなく
物理的な意味のある軌道検証か？
- 立体的な衝突判断に意味があるか？
- 見た目とヒット判定が一致しているか？
- 刀などで多段ヒット処理があるか？

高評価される説明文：

- ・ 近接武器はフレームごとに位置をサンプリングし
RayCastで連続的な衝突判定を実施
高速振り動作時の「すり抜け」を防止

3. あなたがすべき「動画中の具体的な説明方法」

字幕例（動画内テロップ）

- 【Camera System】
 - ・ 障害物判定による自動距離補正
 - ・ ターゲット注視モードへの滑らかな遷移
- 【Enemy AI】
 - ・ 視界内検知（RayCast認識）
 - ・ 位置予測による経路決定
 - ・ 状態遷移による行動管理（FSMベース）
- 【Weapon Collision】
 - ・ RayCastによるフレーム間連続判定
 - ・ 速度に依存したヒット補正

採用担当者は **コードを見ません**
→ 動画テロップで理解できる必要があります。

4. 実際の採用担当者の目線（リアル）

30秒～1分で判断します。

判断ポイント：

- 「この人は抽象理解がある」
- 「リアルタイム処理に強い」
- 「数学（ベクトル・行列）に強い」
- 「ゲームエンジニアとして伸びる」

逆にマイナス評価：

- 「ただ動かすだけ」
- 「ただの自機が敵を殴る動画」
- 「内部アルゴリズムが不明」
- 「企画作品としては良いけど、プログラマーとして弱い」

5. 高評価を狙う具体的対策

動画の最初10秒に必ず入れるべき

Programming Portfolio
自作フレームワーク + DxLib
— アルゴリズム重視 —

そして次にいきなり
AIの動きを見せる！

説明は後！

6. さらに強いアピール方法

動画最後に技術要素一覧：

- ▼ 技術要素
- ・ 自作エンジン構造
- ・ ゲームループ
- ・ 時間差分処理 deltaTime
- ・ カメラ制御（視界補正）
- ・ 敵AI（FSM）
- ・ RayCast衝突
- ・ 行列変換
- ・ 3Dベクトル処理

面接官の心の動き：

「この人、数学できるな」
「基礎理解あるな」
「伸びる人材だな」

7. あなたへの最終アドバイス

あなたの動画は
プログラマーの強みを映像で伝える必要がある

つまり：

- 見た目の派手さ → 重要ではない
- 機能の量 → 厳密にはあまり重要ではない
- 内部の理解 → とても重要
- 説明能力 → 極めて重要

そして最終的に評価されるのは：

- ・リアルタイム処理の理解
- ・ゲーム数学の応用
- ・最適化への感覚
- ・堅実なプログラム設計

🕒Revision #1

★Created 26 November 2025 06:35:02 by youe2

✎Updated 2 June 2026 18:58:33 by youe2