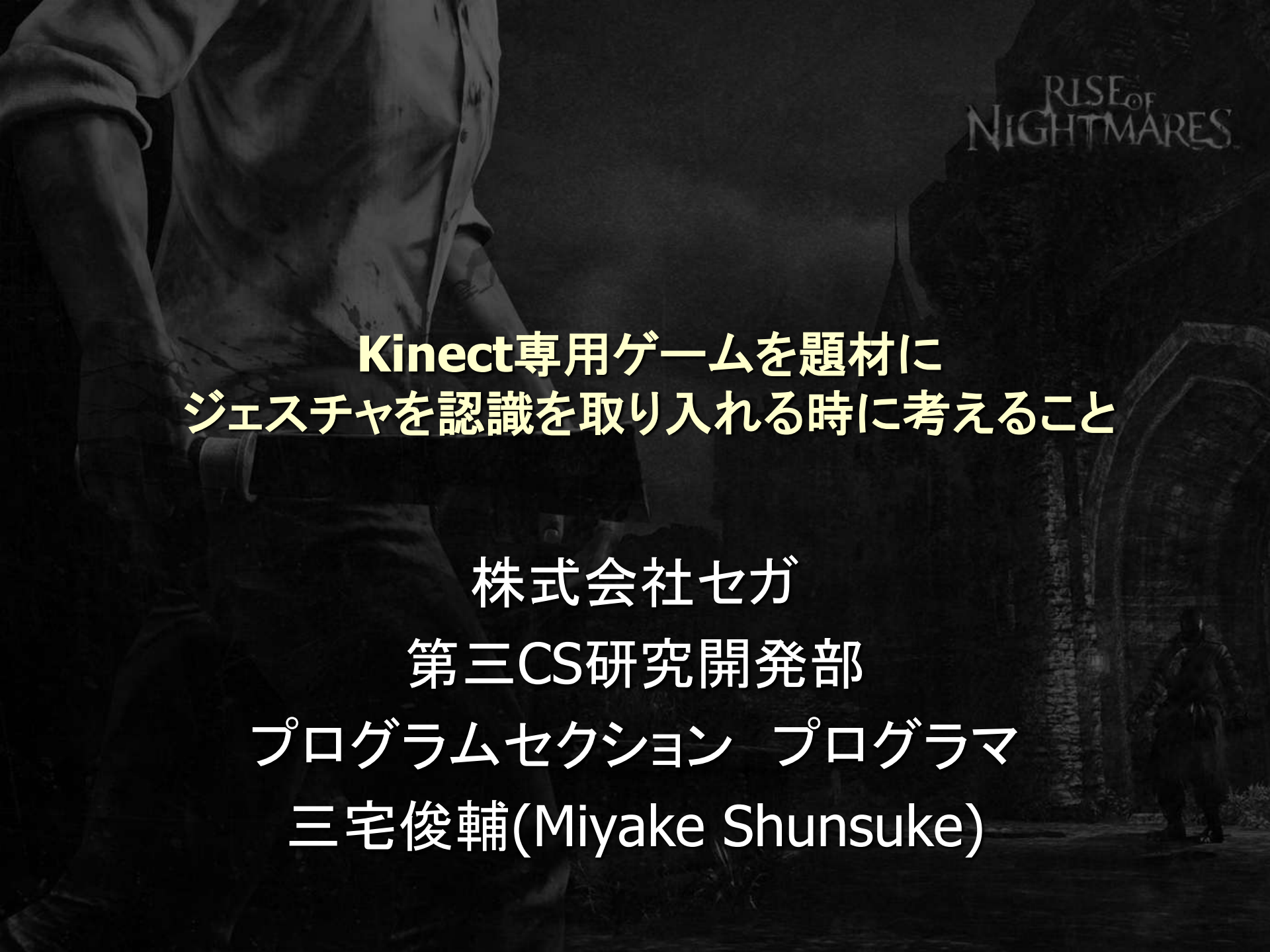
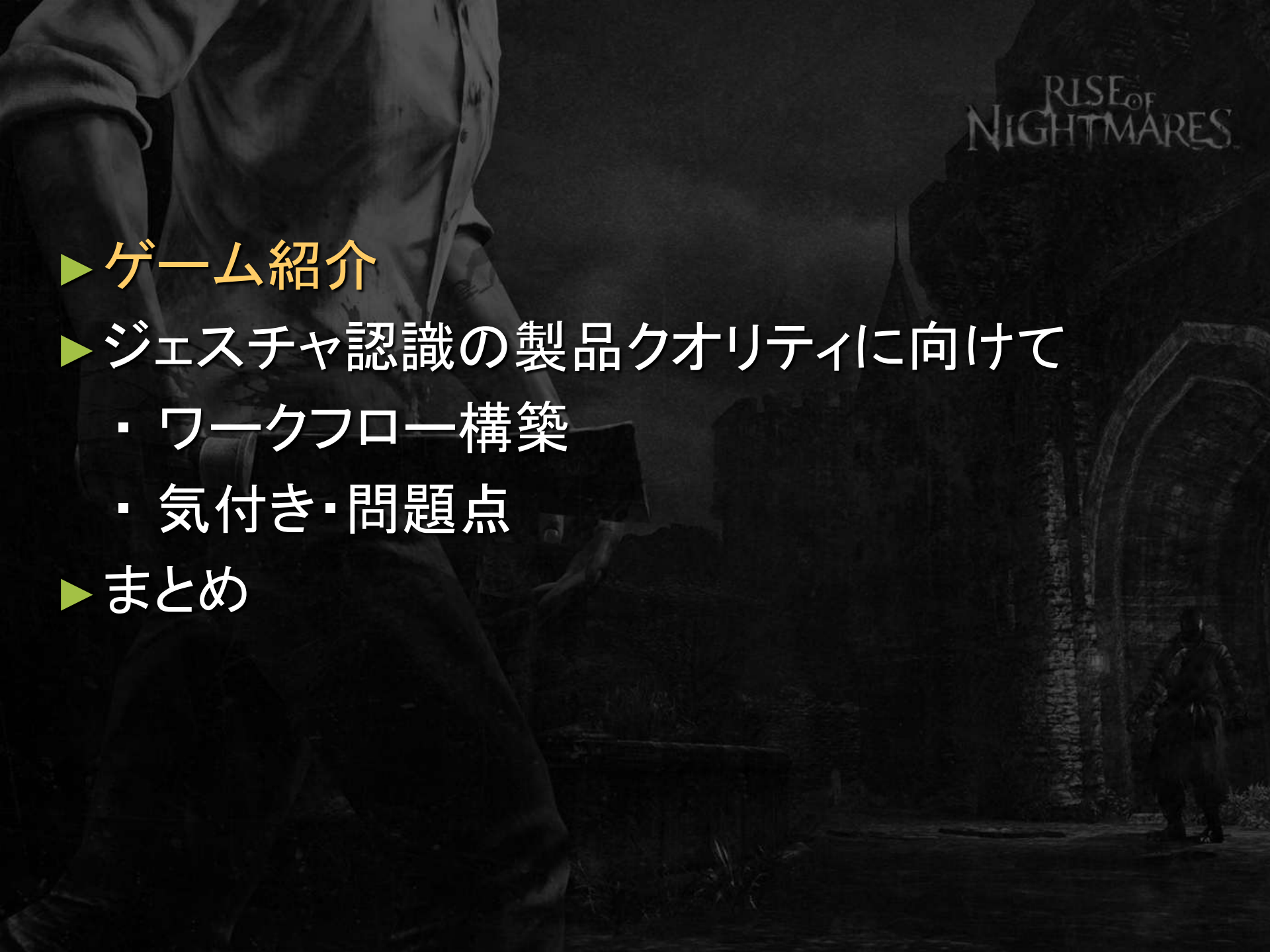


Kinect専用ゲームを題材に
ジェスチャを認識を取り入れる時に考えること

株式会社セガ
第三CS研究開発部
プログラムセクション　プログラマ
三宅俊輔(Miyake Shunsuke)

アジェンダ

- ゲーム紹介
- ジェスチャ認識の製品クオリティに向けて
  - ワークフロー構築
  - 気付き・問題点
- まとめ



- ゲーム紹介
- ジェスチャ認識の製品クオリティに向けて
  - ワークフロー構築
  - 気付き・問題点
- まとめ

RISE OF NIGHTMARES

ジャンル：体感ホラーアドベンチャー
プレイ人数：一人
対応機種：Xbox360(Kinect専用)

主な操作方法(移動)

- 前進・後退

- 旋回

主な操作方法(アクセスポインタ)

ポインターによって扉やはしご、アイテムにアクセスする事が可能

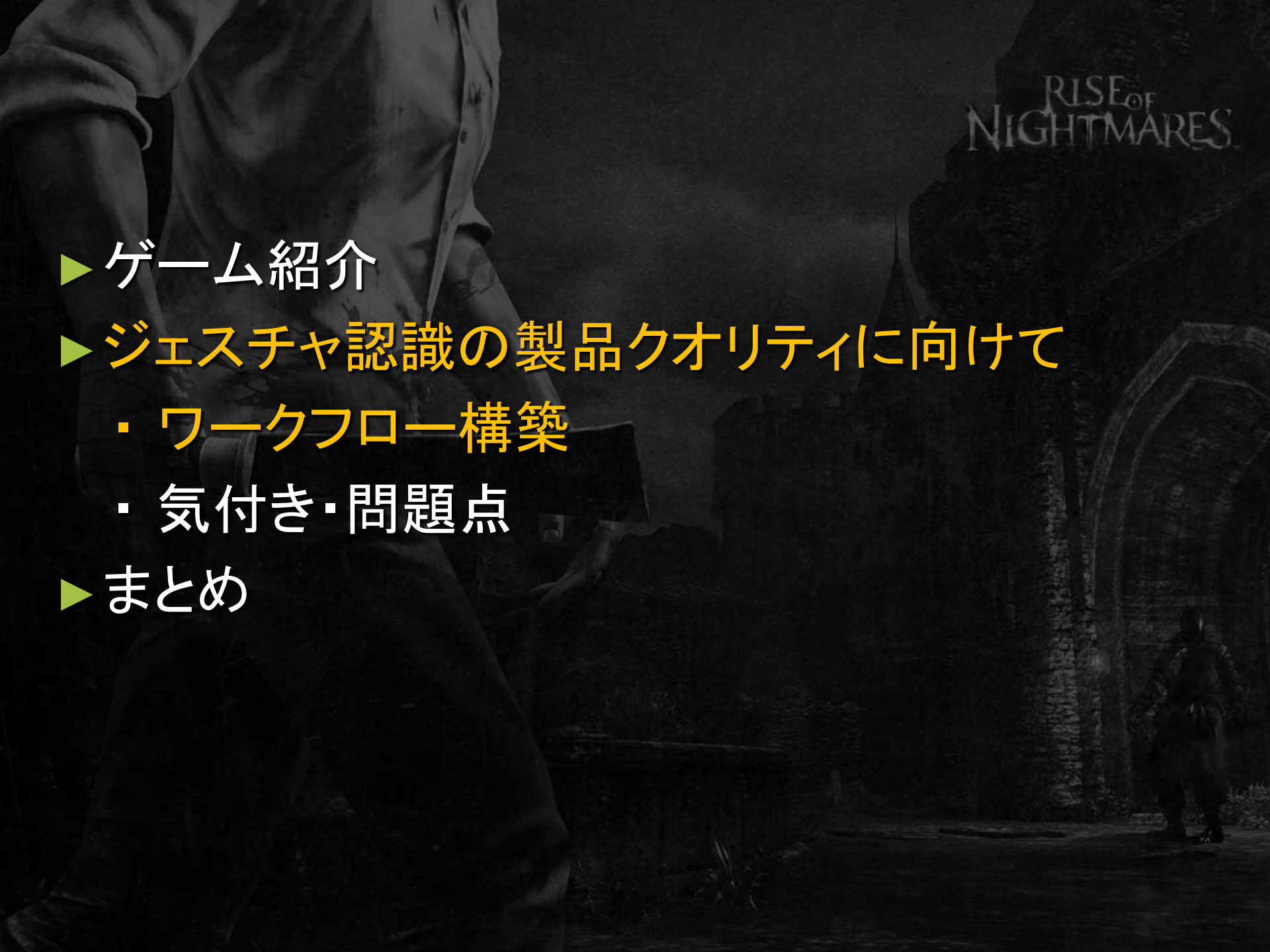
主な操作方法(ギミック操作)

扉、はしご、ハンドル、レバー、潜る穴、etc

アクセスした後、アイコンの指示に従ってジェスチャをすると反応する。

主な操作方法(戦闘)

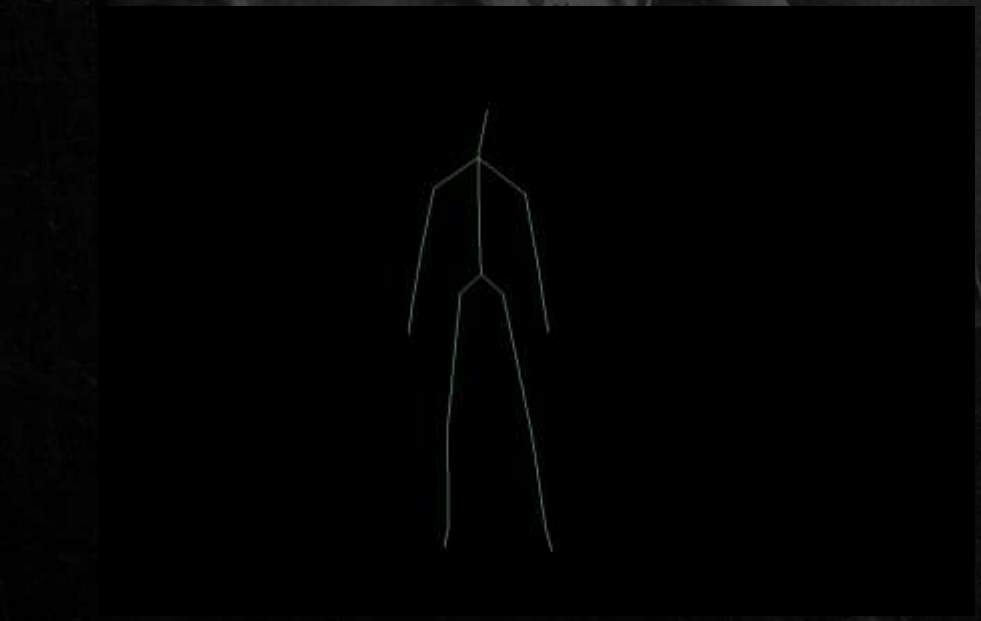
ナックル、ナイフ、投擲武器、チェーンソー、大バサミ、etc
持っている武器によってジェスチャは異なる



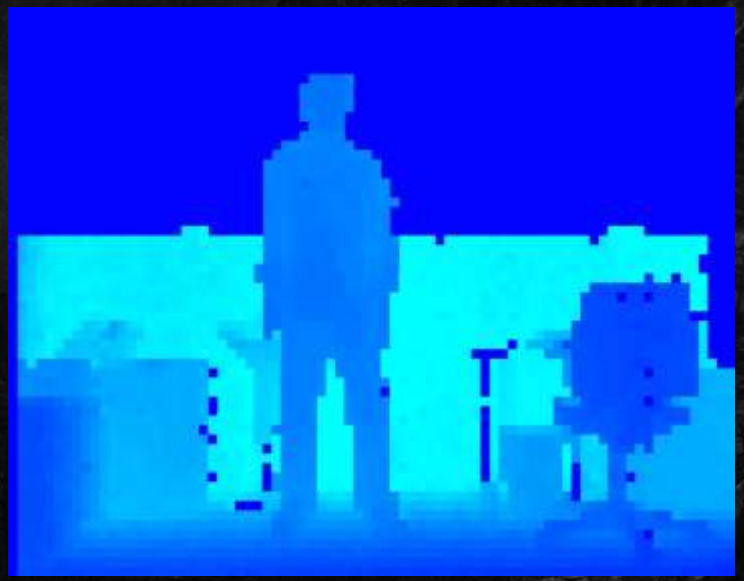
- ゲーム紹介
- ジェスチャ認識の製品クオリティに向けて
  - ワークフロー構築
  - 気付き・問題点
- まとめ

Kinectから取得可能な主な情報

- 骨格情報(体の各部位の座標)と深度情報、シルエット情報が30FPSで取得可能



[骨格情報]



[深度+シルエット情報]

今回は上記の情報を使いルールベースで実装

試行錯誤段階のワークフロー

1. Kinectの前に立ち、必要そうな情報をデバッグ表示
2. 仕様のジェスチャをしながらデータを見て推測で実装
3. 誤認識があれば、Kinectの前に立って原因追求

原因が突き止めにくい

1. 30FPSで更新されるデータを正確に目で追いづらい
2. 席から離れているので、ブレークポイントを貼りづらい

自分が認識すれば他人も認識するわけではない

- 体格差による骨格の違いや動きの癖などが、分析できていない

認識精度を高めるためには

・ジェスチャを試した人数
・正しく認識した人数/ジェスチャを試した人数

の値を体格、性別などのタイプ別に高める必要がある。

ワークフロー改善案

- 骨格・深度情報を記録して分析

→ サンプリング環境の構築

→ 定期的なジェスチャ認識テストでサンプリング

→ 収集したデータを再生して分析

サンプリング環境の構築

- 素早く簡単にデータを記録 & 再生の環境構築
  → ボタン一つで記録 & 再生機能をゲーム内に組み込み
  → ボタンを押すと自動で10秒間記録してファイル出力
  → ファイル名、種類別フォルダ分けは自動化

定期的なジェスチャ認識テスト

■単体テスト

→　指示に従って言われたジェスチャをその場でする。

→　１日1時間 (1人30分で2ライン：計4人をテスト)

■統合テスト

→　任意のステージをプレイしてもらって問題無く操作できているかのテスト。問題が出たら記録する。

→　時間制限は設けず、1人2～3ステージプレイ。

■実環境テスト

→　ソファやソフトドリンクなどを用意しての、実際の家庭でのプレイを想定してのテスト

→　時間制限は設けず、1人2～3ステージプレイ。

ジェスチャ認識テスト実績

- チーム内

期間：およそ十ヶ月。テストができるようになってから毎日
チェック内容:単体テスト、統合テスト
サンプリング人数累計：72人

- チーム外

期間：実装コストが高いジェスチャを実装してから不定期
チェック内容：単体テスト、統合テスト、実環境テスト
サンプリング人数累計：60人

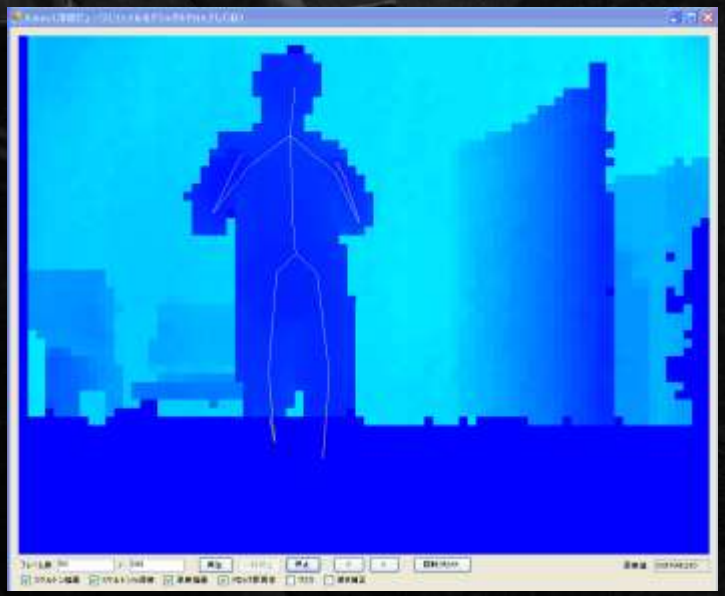
- 実際にサンプリングしたデータ

ジェスチャ種類：43種類 (オミットになったジェスチャも含む)
種類別ファイル数：最大で600ファイル程
ファイルサイズ：3MB(骨格情報、深度情報)

簡易ビューワで再生

- 記録した骨格・深度情報再生ビューワ

→　一通りの情報は確認可能



パッと見で問題がないか確認するのに利用

実機再生①

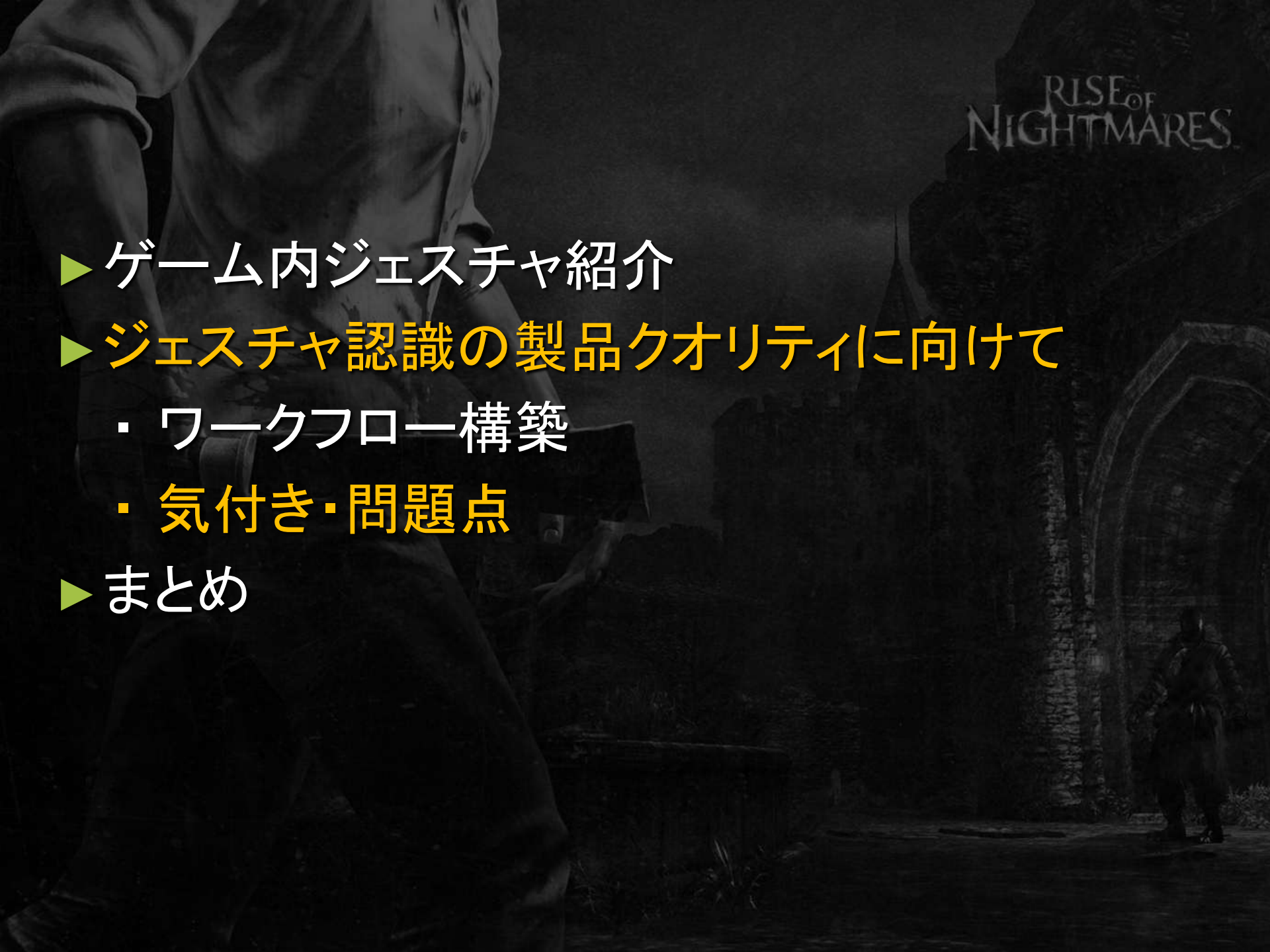
- 記録したデータは、実機再生で入念に検証

実機再生②

- 実装に手を加えたジェスチャに関しては、過去のデータを再生して実績を積むのにも利用。

最終的なワークフロー

1. 新規ジェスチャの仕様が決まる
2. チーム内の数人からデータを記録
3. データを分析しながら実装（60～80%の精度）
4. 定期的なテスト
5. 問題が発生すれば、データを再生して原因追究
6. プログラム修正 or 仕様担当者と話し合って解決
7. 過去のデータを再生してエンバグチェック
8. 4.の定期的なテストに戻る



- ゲーム内ジェスチャ紹介
- ジェスチャ認識の製品クオリティに向けて
  - ワークフロー構築
  - 気付き・問題点
- まとめ

仕様変更をもたらしかねない要素

1. 疲れることへの否定的意見
2. 想定していたジェスチャができない人がいる
3. 想定外の動きをする人がいる
4. 想定外のタイミングでジェスチャをする人がいる
5. 同時に認識する動きが衝突

1. 疲れることへの否定的意見①

- 前進ジェスチャ(足踏みと足前出し)

←[足踏み]

[足前出し]→

最終的な仕様は足前出し（反復運動は疲れる）

疲れることへの否定的意見②

- 旋回ジェスチャ(腕旋回と肩旋回)

←[腕旋回]

[肩旋回]→

最終的な仕様は肩旋回（腕を上げているのは疲れる）

■ 疲れる動きをさせる場合は、十分な検討とテストを。

2. 想定していたジェスチャができない人がいる

- しゃがむ

←[想定]

[代替手段]→

体型的に難しい人がいた

追加仕様発生：どちらでも認識するように対応

■ 想定していた動きを誰でもできるのか検討とテストを。

3. 想定外の動きをする人がいる

- 扉を開ける

←[想定]

[想定外]→

キックで扉を開けようとする人がいた

追加仕様発生：どちらでも認識するように対応

■ 動きの癖は十分な検討を。ただ初めから想定しきるのは難しいので早期のテストを。

4. 想定外のタイミングでジェスチャする人がいる

- 扉を開く

ジェスチャアイコンが出る前にジェスチャする人がいた

仕様追加発生：先行入力に対応

■ 認識処理を常に行っているわけでない場合、ユーザーがいつどのタイミングでジェスチャをしたいと思うのかの早期テストを。

5. 同時に認識する動きが衝突

- 『しゃがむ』と『足を一歩下げる』

足を一歩下げてしゃがもうとする人がいる

仕様変更:『しゃがむ』と『足を一歩下げる』を
同時で認識する状況を無くした

■ ジェスチャをする時の予備動作に注意

見積もりに影響を与える要素

1. 同時に認識するジェスチャがあるか
2. 繰り返し認識する必要があるか
3. 認識成立状態を維持する必要があるか
4. 認識結果にバリエーションがあるか
5. 認識しやすい(しにくい)ことでユーザー不利益はあるか

■ この要素が検討しておかないと、リスケジュールが発生する可能性があるので十分な検討を。テスト優先度の指標にも。

1．同時に認識するジェスチャがあるか

- パンチ(ストレートパンチ、フック)

構えとストレートパンチ、旋回とフックが誤認識しない工夫が必要

■ 同時に認識するジェスチャが増えれば増えるほど実装工数は増える。

2.繰り返し認識する必要があるか

- キック

一回のキックで複数回キックを出ない処理が必要

■ 繰り返し認識する必要があると、実装工数は増える。

3.認識成立状態を維持する必要があるか

- アクセスポインタ

アクセスポインターが出現後、勝手に解除されない処理が必要

■ 認識後もその状態を維持する必要があると実装工数は増える。

4. 認識結果にバリエーションがあるか

- 旋回

捻り角度が欲しい。滑らかに変化する必要がある

■ 欲しい認識結果にバリエーションがあると実装工数は増える。

5．認識しやすい(しにくい)ことで ユーザー不利益はあるか？

- 扉を開ける

認識しやすいことでのユーザー不利益はほぼない。

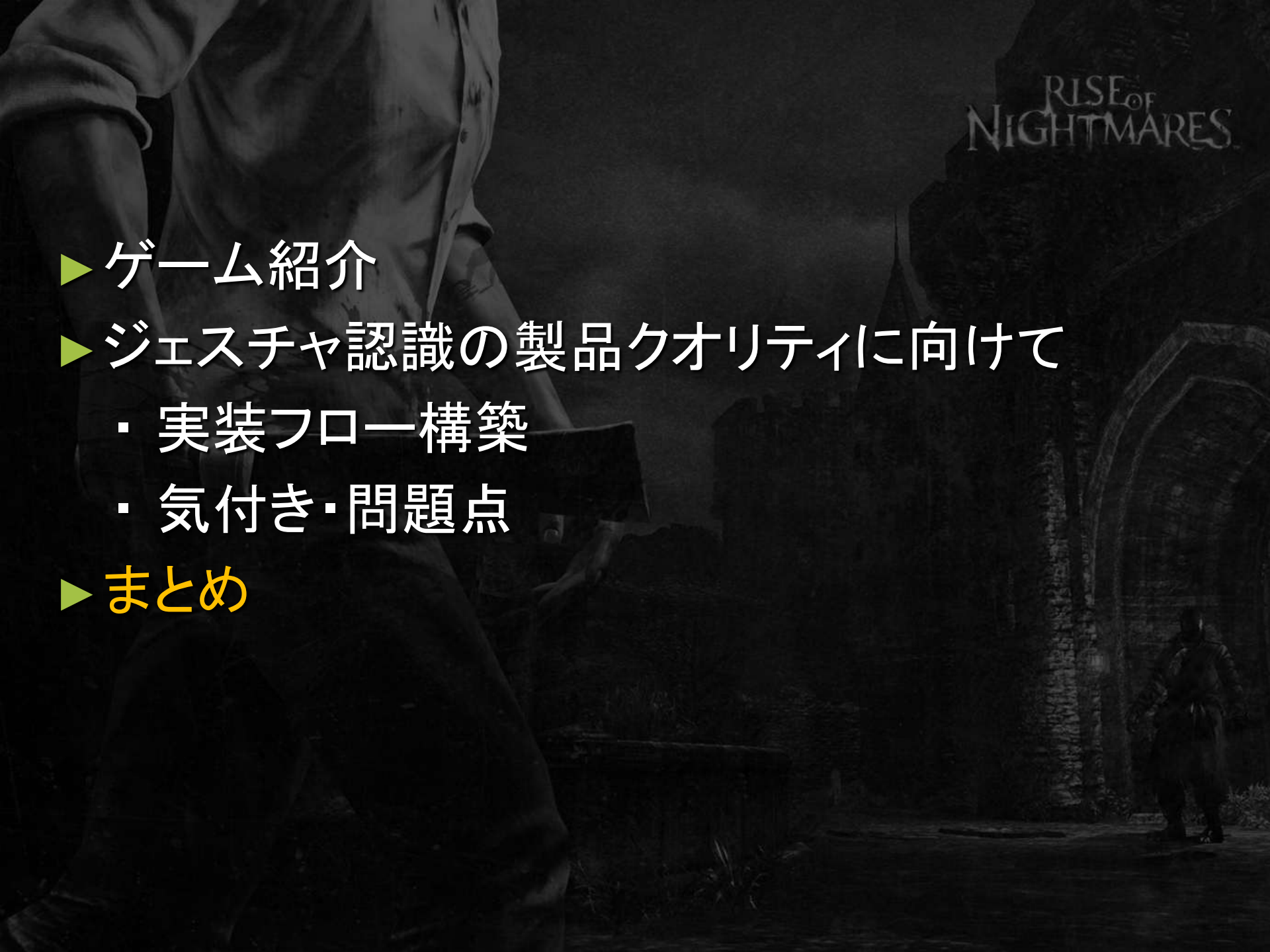
■ 認識しやすくても認識しにくくても、ユーザーにとって不利益にならないなら無理に認識精度を高めない。

TIPS:人はキャラクタの動きを真似しやすい

- 虫を払い落とす

キャラクタの動きを見てジェスチャを判断する人が多かった

■ ユーザーにさせたいジェスチャを、自然に理解してもらうためにキャラクタの動きで示すのは効果的。



- ゲーム紹介
- ジェスチャ認識の製品クオリティに向けて
  - 実装フロー構築
  - 気付き・問題点
- まとめ

ワークフロー構築時に考えて良かったこと

1. サンプリング環境を構築できているか
   → どれだけ手順や人手を減らせるか
   → ファイル名、フォルダ分けに時間がかかっていないか
2. 定期的にテストができているか
   → 体格や性別の違う人をテストできているか
3. 過去に記録したデータは蓄積して、再利用しているか

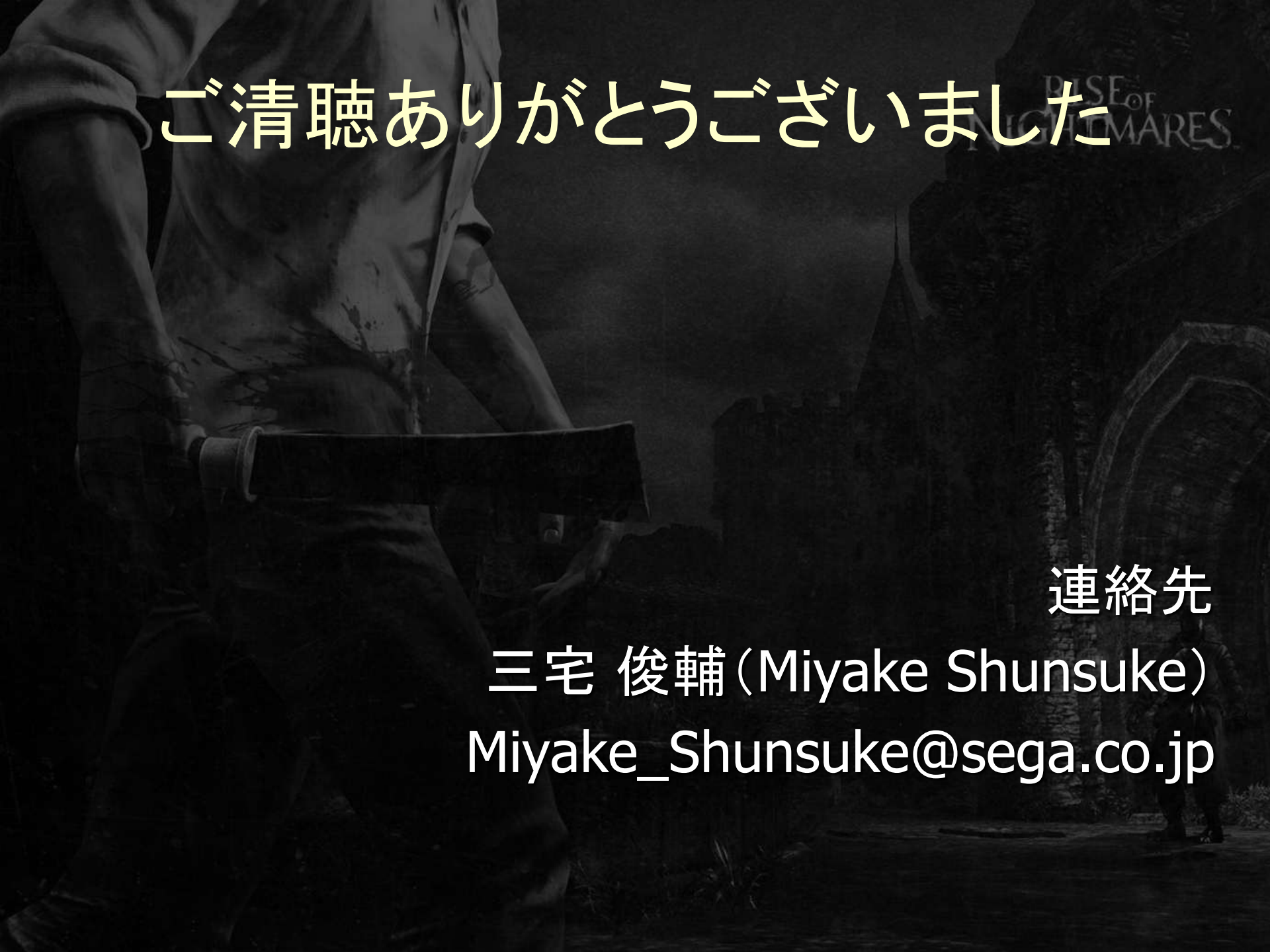
仕様検討・テスト時に意識して良かったこと

1. どれだけ疲れるか？
2. すべての人が本当にできるか？
3. 人によって、どのような癖があるか？
4. ユーザーはどういうタイミングでジェスチャをするか？
5. 同時認識するジェスチャと衝突する可能性はあるか？

見積もりの時に考えて良かったこと

1. 同時認識するジェスチャはあるか？
2. 繰り返し認識する必要はあるか？
3. 認識成立状態を維持する必要はあるか？
4. どのような認識結果が欲しいのか？
5. 認識しやすいこと、認識しにくいことでのユーザー不利益はあるか？

質疑応答



ご清聴ありがとうございました

連絡先
三宅 俊輔（Miyake Shunsuke）
Miyake_Shunsuke@sega.co.jp