BANDAI NAMCO Studios

BLUE PROTOCOLの個性豊かなキャラクターを動かす意思決定システム

株式会社バンダイナムコスタジオ
長谷 洋平


BLUE PROTOCOL™

BLUE PROTOCOL™

ブループロトコル

The world is on the brink of devastation, now is the time to unite.
March on with friends and strangers, and defeat foes beyond your might.
Travel through space and time, to change the future beyond this fight !

- オンラインアクションRPG
- 操作できる劇場アニメ
- パーティ vs パーティ のバトル
- 多数の個性的なキャラクターが敵として登場

目標

1. 多種多様なキャラクターの表現

2. どのようなキャラクターの組み合わせでも
   成立するパーティバトル

3. 運営時のキャラクターやコンテンツの追加に
   対応できる柔軟性

4. 工数削減

アジェンダ

BANDAI NAMCO Studios

- アーキテクチャ
- 個性づけ／量産
- パーティバトル
- まとめ

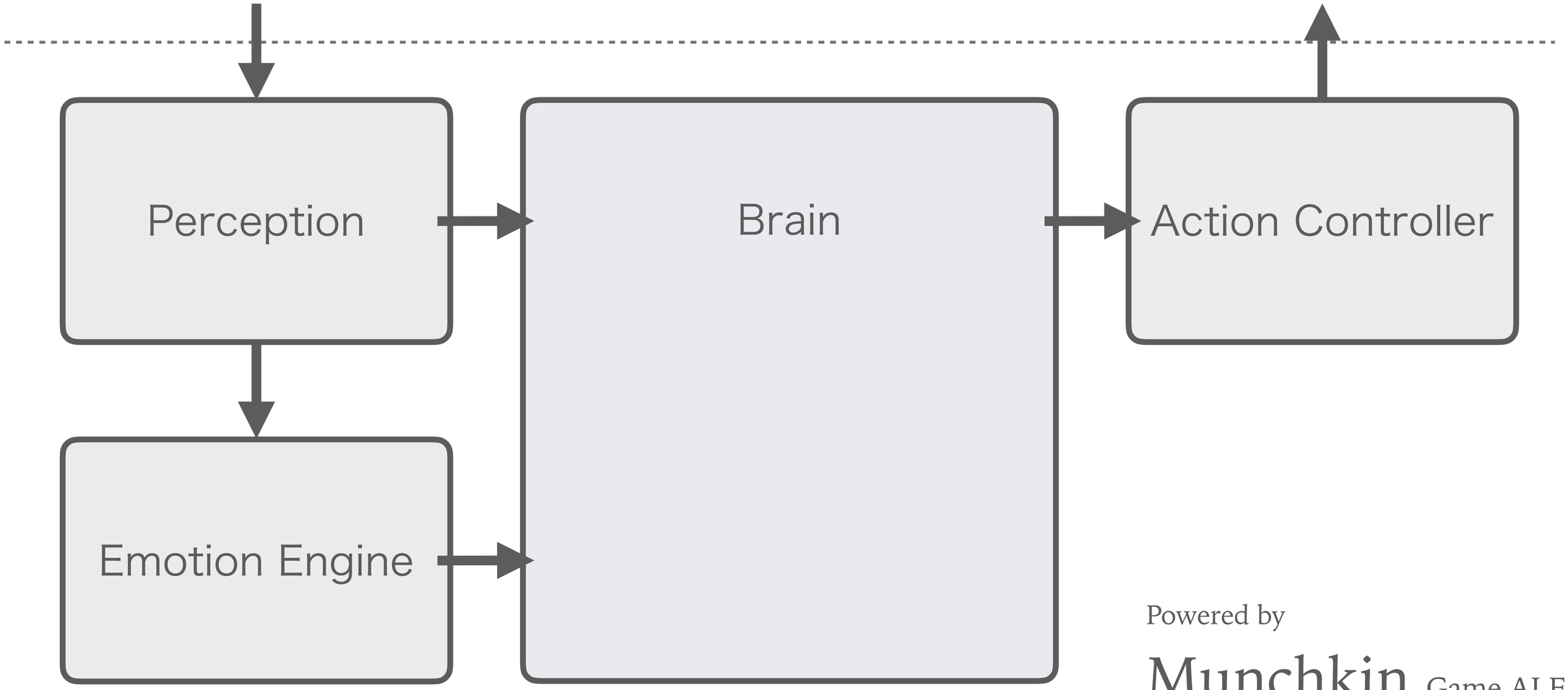
アジェンダ

BANDAI NAMCO Studios

- [x] アーキテクチャ
- [ ] 個性づけ／量産
- [ ] パーティバトル
- [ ] まとめ

エージェントアーキテクチャ

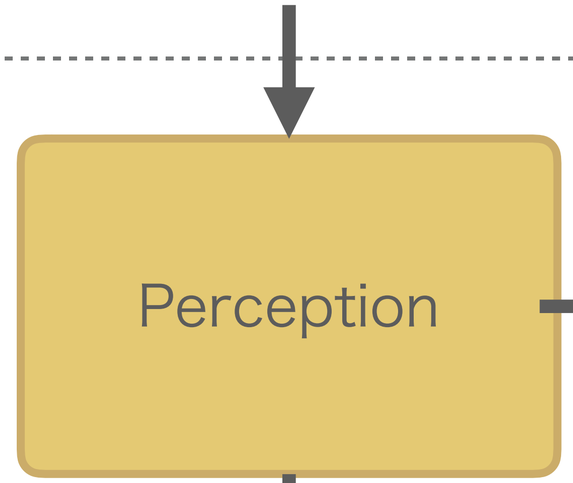
BANDAI NAMCO Studios



Powered by

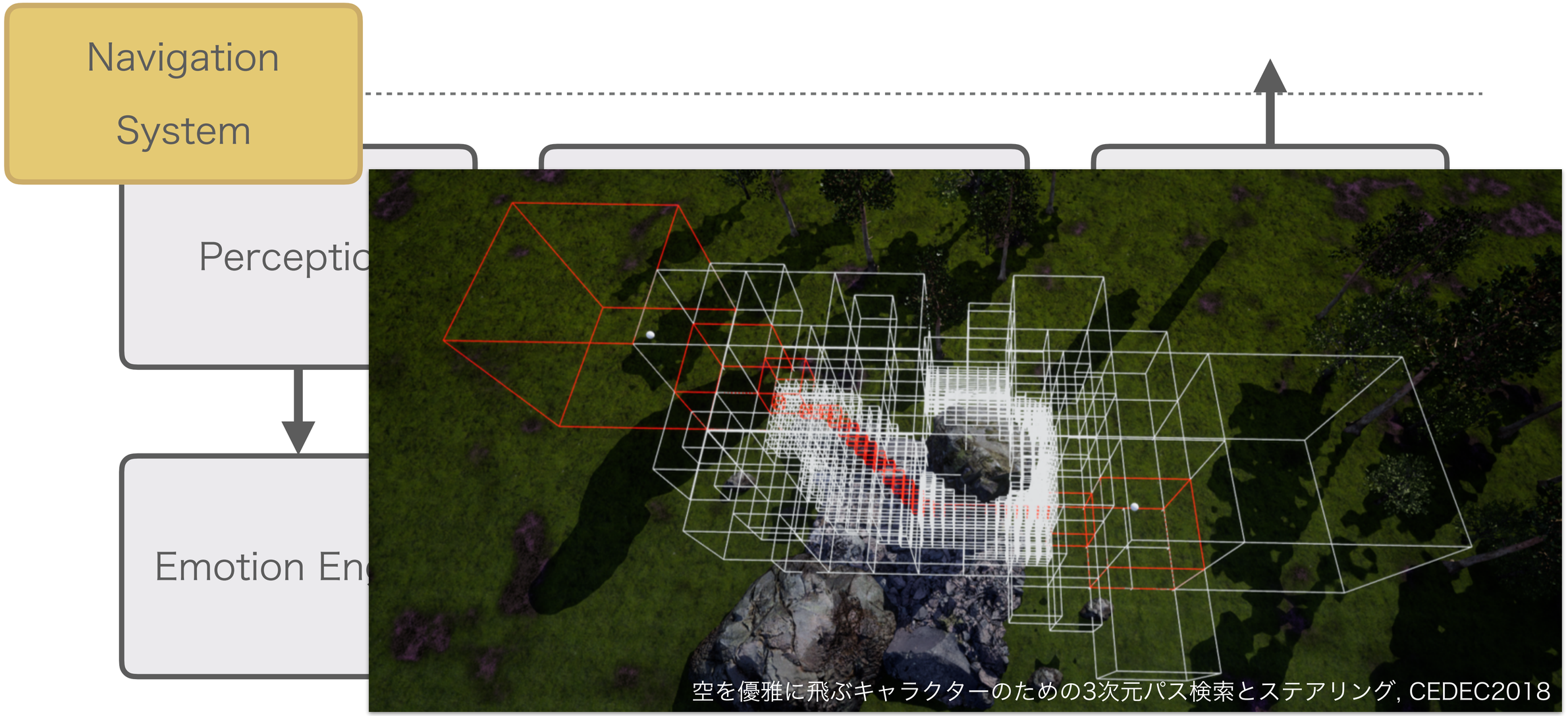
Munchkin Game AI Engine

エージェントアーキテクチャ



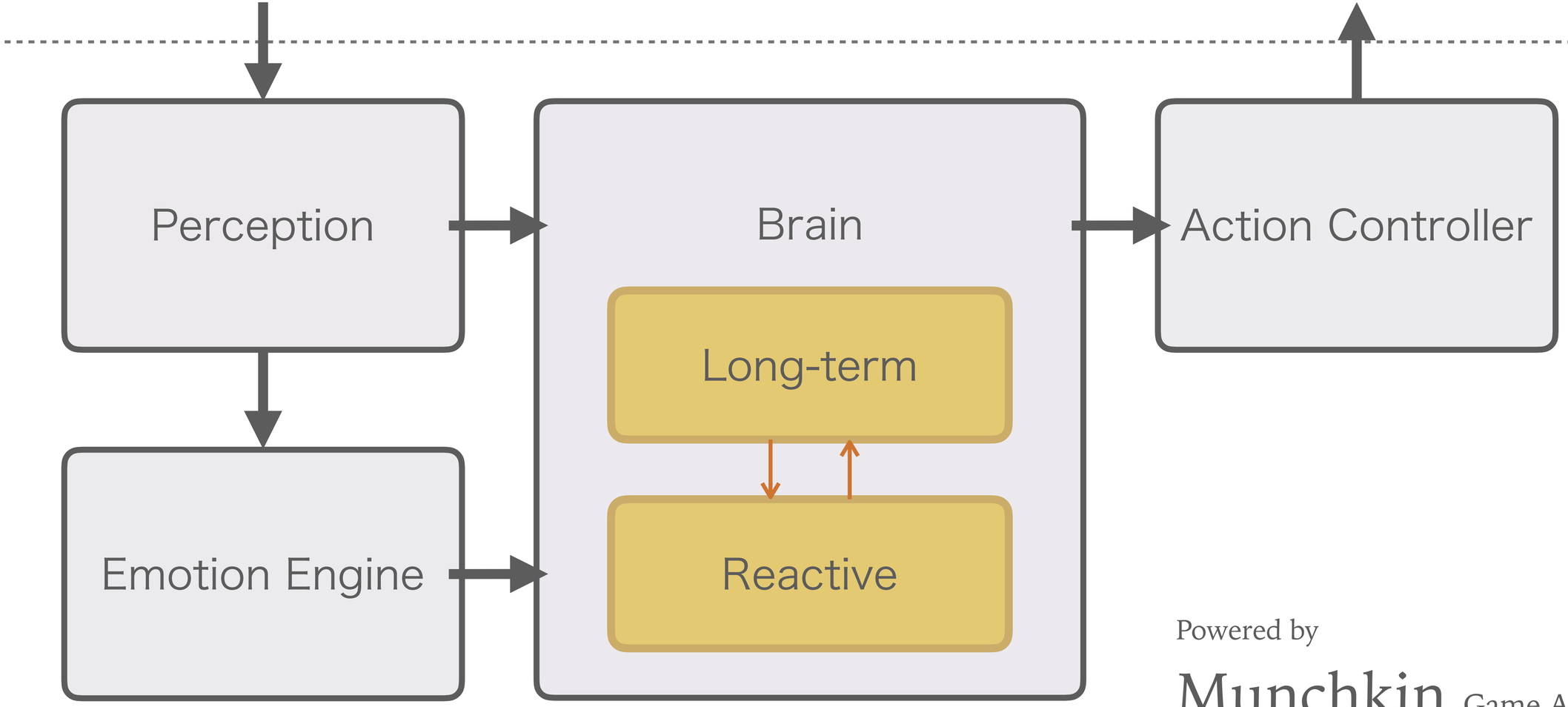
Perception Tree ～Behavior Treeを応用したお手軽、柔軟な環境認識システム～, CEDEC2017

エージェントアーキテクチャ



エージェントアーキテクチャ

BANDAI NAMCO Studios



Powered by

Munchkin Game AI Engine

意思決定アーキテクチャ

背景

複雑な環境で動くキャラクターを表現する上で
異なる時間スケールへの対応の重要性が増している

現実世界の人間の場合

- 会社へ行く
- 喉が渇いたのでコンビニに寄って水を買う
- 信号が変わりそうなので走って渡る
- 飛び出してきた子供を避ける

} 複数の目標が同時に発生する

意思決定アーキテクチャ

BANDAI NAMCO Studios

背景

複雑な環境で動くキャラクターを表現する上で
異なる時間スケールへの対応の重要性が増している

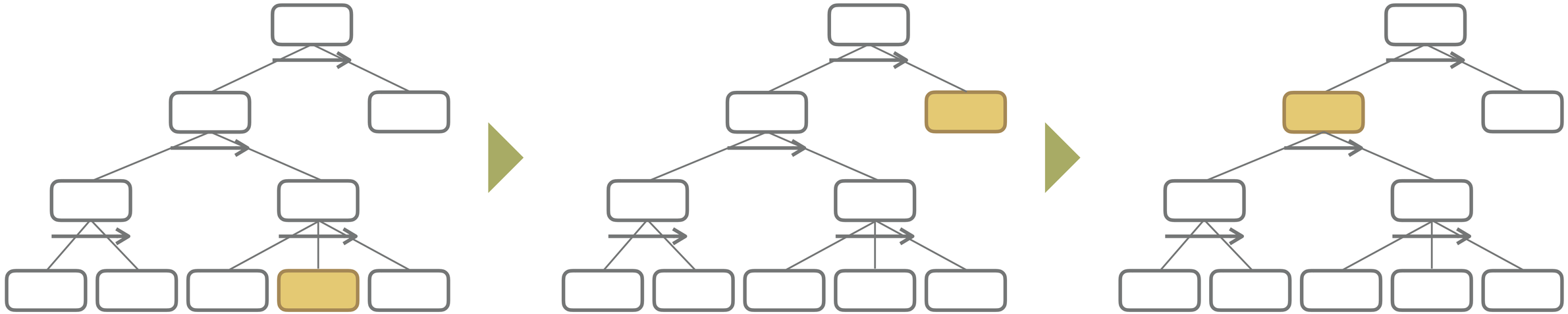
アクションゲームの場合

- 目標に対して一貫した行動が必要
- 展開が早く、目まぐるしく変わる状況に対して臨機応変な対応も必要

▶ 一つのシステムの中で両方を扱おうとすると無理が出る

意思決定アーキテクチャ

例：Behavior Tree のみで割り込み行動



何してたっけ？

以前実行していたノードの情報を持たないので
違う行動が選ばれる可能性がある

意思決定アーキテクチャ

BANDAI NAMCO Studios

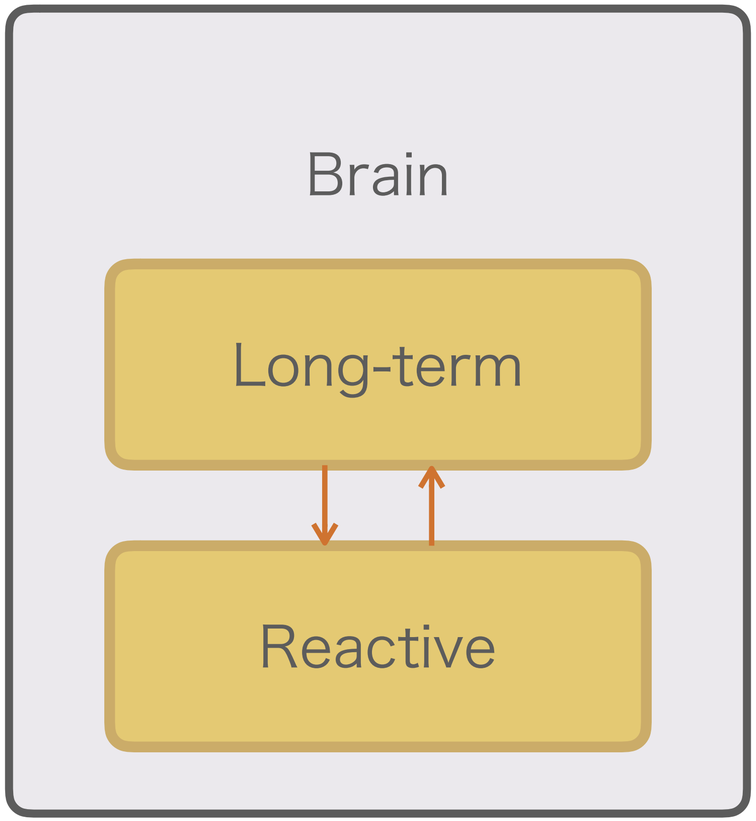
問題（一つのシステムで扱った場合）

- 行動の一貫性を持たせることが難しい
- 割り込み行動を簡単に追加できない

方針

- 異なる時間スケールごとにシステムを並列に動かす
- 各システムが決定した結果を後で調停する

意思決定アーキテクチャ



Long-term

現在の状態にあった目標を選択し、その目標を達成するある程度の長さの行動のシーケンスを扱うメインロジック

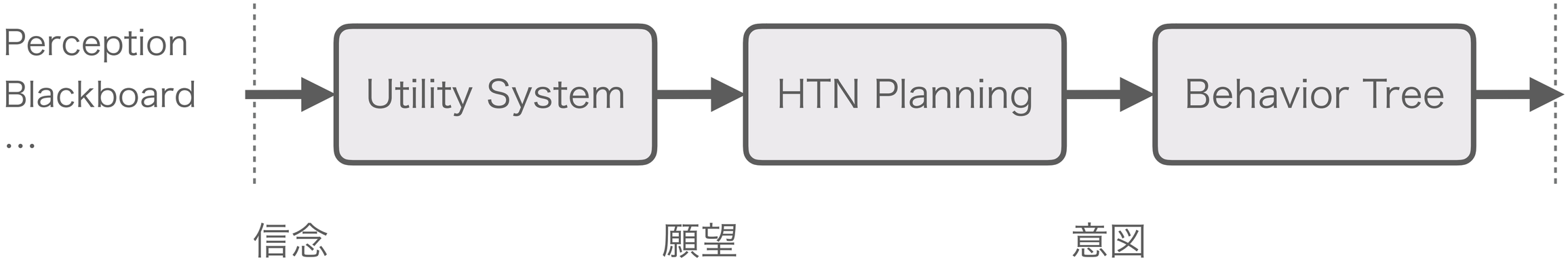
Reactive

短期の課題に対応する即応的なリアクション

基本的には Reactive が優先されるが、コンテキストによっては Long-term が Reactive を抑制する

メインロジック

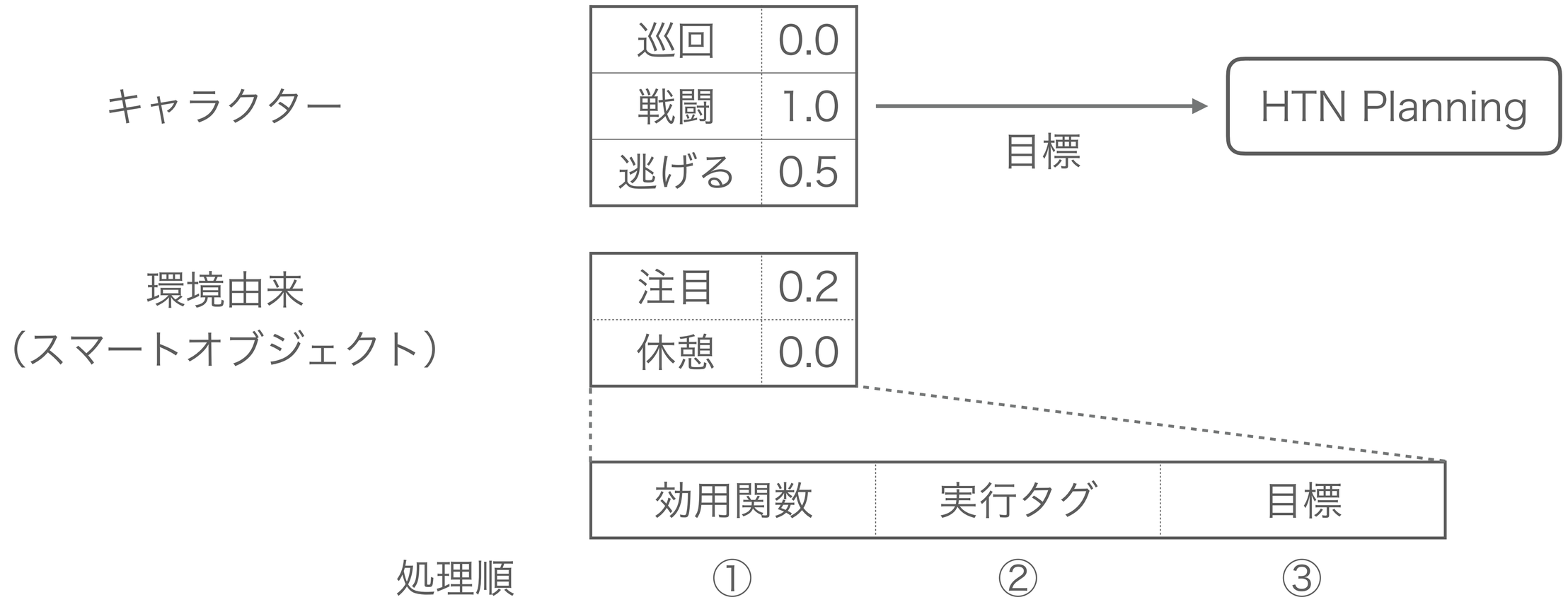
BDIベース



- 生物らしい振る舞いを実現できる
- それぞれが扱う問題が限定されて簡略化される
- 部分的な特殊化が容易

Utility System

現在の状況をもとに達成すべき目標を決定する



Utility System

実行タグ：行動のタイプをカテゴライズしたもの

実行に必要なタグ | 制限するタグ

巡回	1.0
注目	0.5

Main | Main, Attention ⟶ **制限タグ**：なし

Attention | Attention

巡回	1.0
注目	0.5

Main | Main, Attention

Attention | Attention ⟶ **制限タグ**：Main, Attention

Utility System

実行タグ：行動のタイプをカテゴライズしたもの

実行に必要なタグ | 制限するタグ

巡回	0.5
注目	1.0

Main | Main, Attention

Attention | Attention ⟶ **制限タグ**：なし

巡回	0.5
注目	1.0

Main | Main, Attention ⟶ **制限タグ**：Attention

Attention | Attention

… 矛盾しない場合は複数の願望が同時に発生し得る

Utility System

BANDAI NAMCO Studios

よそ見をするノンアクティブモンスター

HTN Planning

BANDAI NAMCO Studios

- 目標である抽象的なタスクをより具体的なタスクに分解していくことで
  目標の達成に必要な行動とその順序を見つける
- 将来の状態の変化をシミュレートしながら問題を解くので
  合理的な計画を得られる
- 人が事前知識を使うことで現実的な時間で計画を立てる思考と同じため
  他のプランニング技術に比べ直感的に構築がしやすい

例：旅行問題

あらゆる手段、ルートをしらみつぶしに調べるには探索空間が大きすぎるので
事前知識からこのルートは電車を使うなど部分問題に分割して解いている

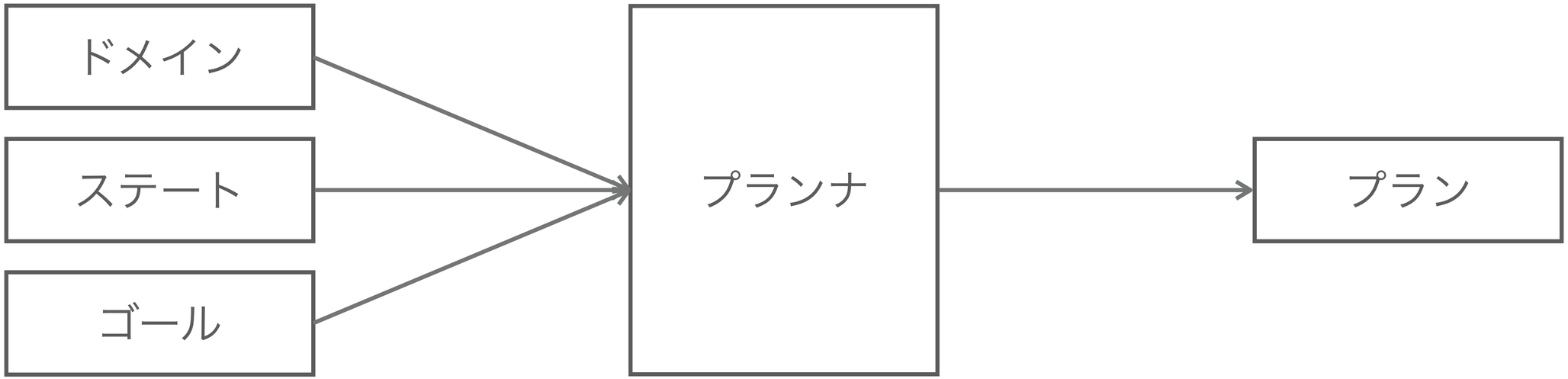
HTN Planning

ドメイン：問題を解くための手段が記述されたもの

（プリミティブタスクと複合タスクのリスト）

ステート：自身や環境の状態を表したもの

ゴール：達成すべき目標



HTN Planning（例：旅行問題）

BANDAI NAMCO Studios

プリミティブタスク：そのタスクを実行する時の事前条件と事後条件のセット

切符を買う	**事前条件**：所持金 >= 切符代	**事後条件**：所持金 ← 所持金 - 切符代

複合タスク：どういうサブタスクに分割するか（メソッド）のリスト

xからyまで移動する

- 飛行機（航空券を買う、xから空港に行く、飛行機に乗る、空港からyに行く）
- 新幹線（xから駅に行く、切符を買う、新幹線に乗る、駅からyに行く）
- ⋮

HTN Planning（例：旅行問題）

東京の家から福岡駅に行く

↓ 分割

航空券を買う、羽田空港に行く、飛行機に乗る、福岡駅に行く

↓ プランに追加、所持金：100,000円 – 50,000円 = 50,000円

航空券を買う、羽田空港に行く、飛行機に乗る、福岡駅に行く

↓ 分割

航空券を買う、東京駅に行く、切符を買う、電車に乗る、羽田空港に行く、
飛行機に乗る、福岡駅に行く

⋮

航空券を買う、東京駅まで歩く、切符を買う、電車に乗る、飛行機に乗る、切符を買う、電車に乗る

ステート

所持金：100,000円

HTN Planning（例：旅行問題）

東京の家から福岡駅に行く

↓ 分割

福岡駅まで歩く

↓ プランに追加、所持金は変化なし

福岡駅まで歩く

ステート

所持金：100円

詳しくは、
プレイヤーに反応するだけのAIはもう古い！ゲームAIへのプランニング技術の導入, CEDEC2016

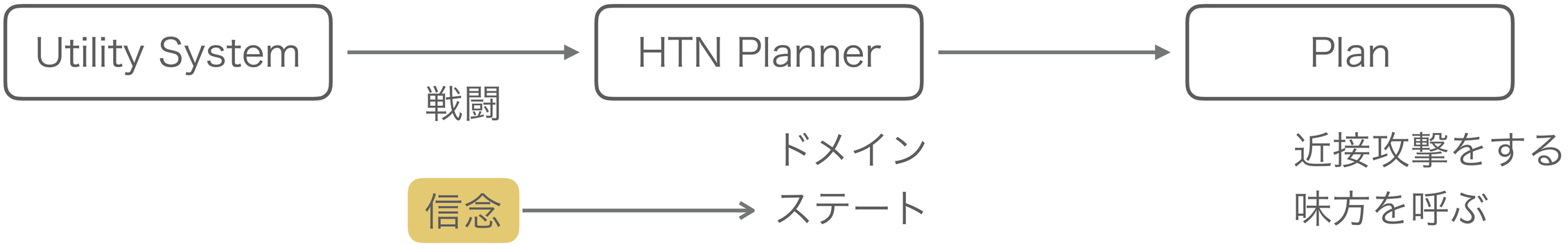
HTN Planning

ドメイン：キャラクターごとに個別のものを構築

ステート：現在の状況に合わせて信念情報をもとに更新

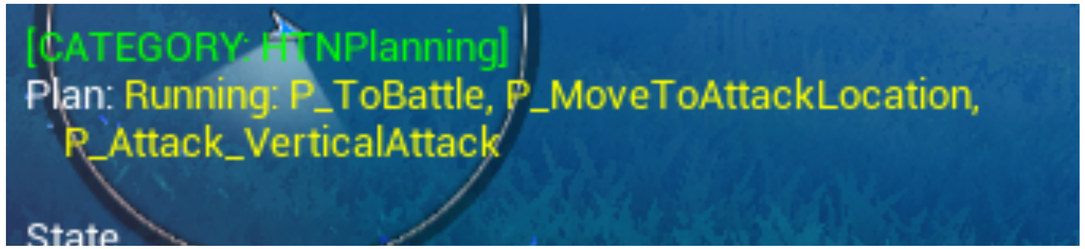
ゴール：Utility System から渡される

同じ目標に対してもキャラクターごとにアプローチが異なる

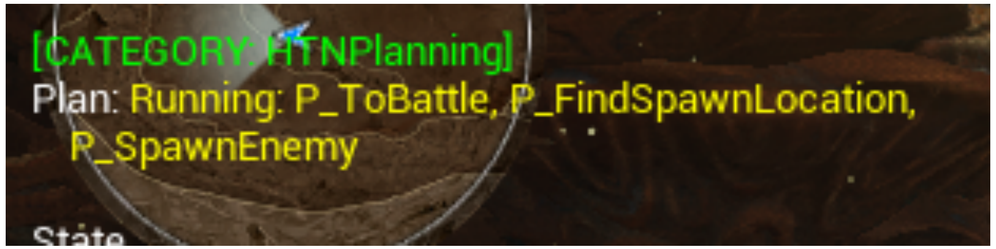


戦闘

ゴブリン（通常の敵）



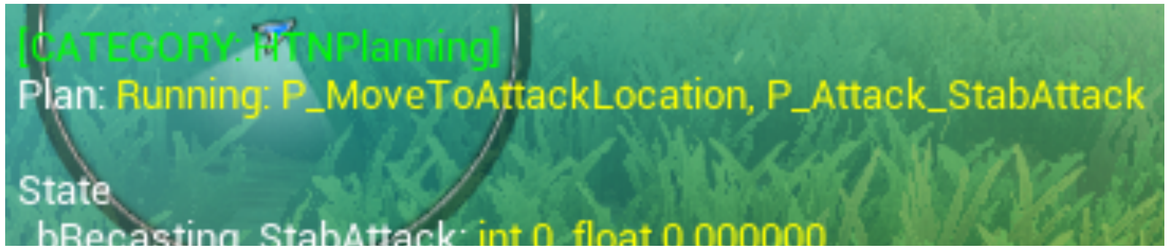
ミーンの巣（移動不可、攻撃不可）



中距離攻撃

BANDAI NAMCO Studios

遠くから



近くから

中距離攻撃

格闘攻撃

攻撃A（攻撃が当たる場所へ移動、攻撃Aを実行）

移動

歩行（...）

バックステップ（...）

サイドステップ左（...）

サイドステップ右（...）

ステップあり

状況に応じて
適切なアクションを選択

中距離攻撃

格闘攻撃

攻撃A（攻撃が当たる場所へ移動、攻撃Aを実行）

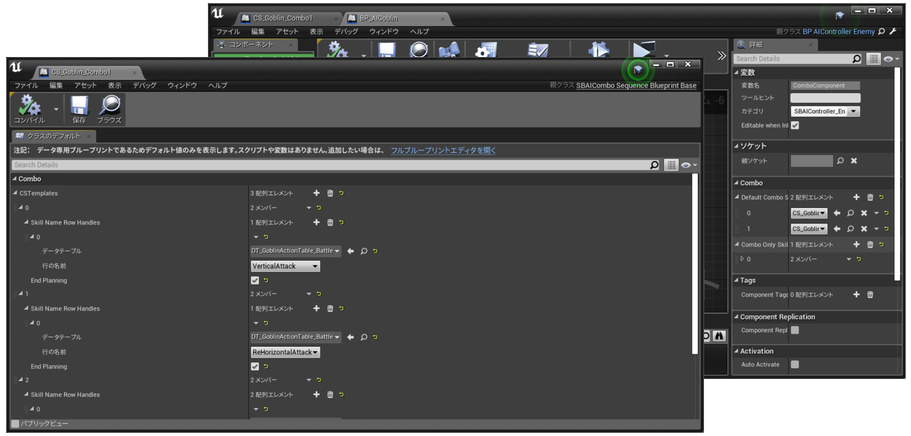
移動

歩行（...）

ステップなし

必ず歩いて移動する

コンボ



コンボ設定

（攻撃Aを実行、コンボ）

コンボ

攻撃A（継続条件、攻撃Aを実行、コンボ）

攻撃B（継続条件、攻撃Bを実行、コンボ）

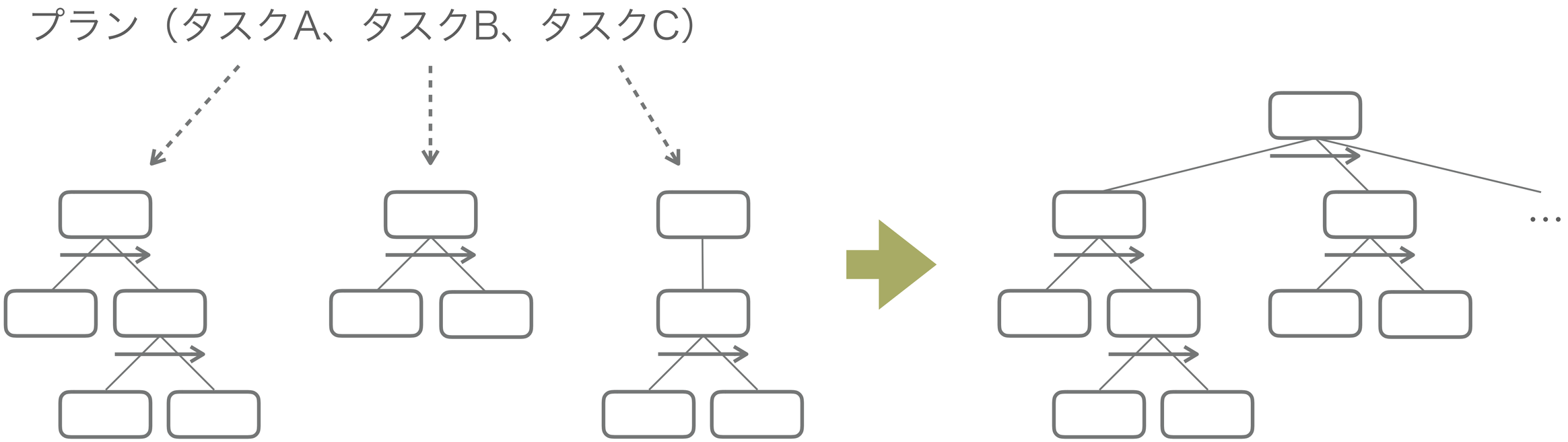
（攻撃Aを実行、継続条件、攻撃Bを実行、コンボ）

コンボ

BANDAI NAMCO Studios

アクション

求まったプランを Behavior Tree に変換し実行する

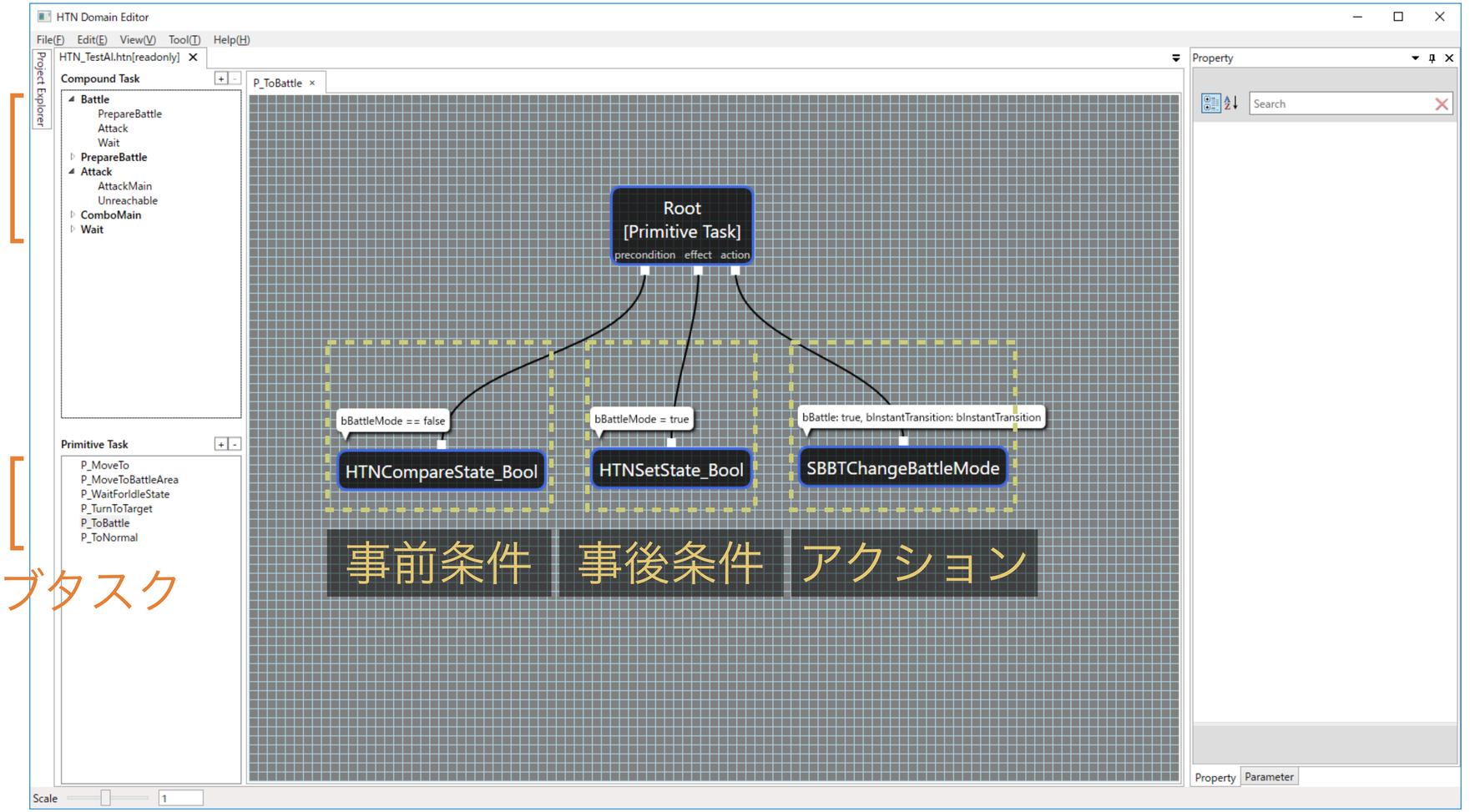


HTNドメインエディタ

HTNドメインとアクションとして実行される Behavior Tree を一緒に定義

複合タスク

プリミティブタスク

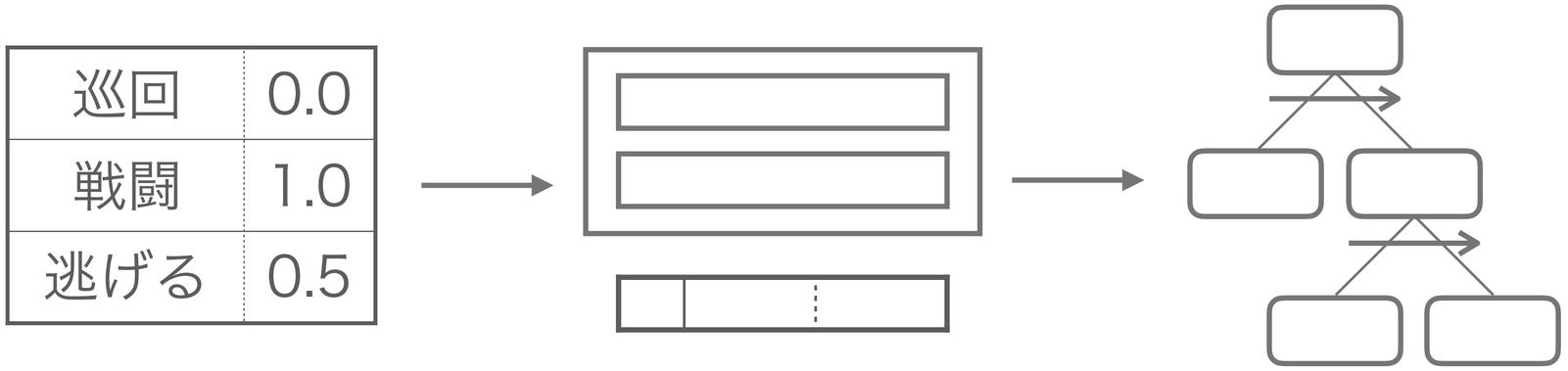


メインロジック

BDIベース

Perception
Blackboard
…

信念　　願望　　意図



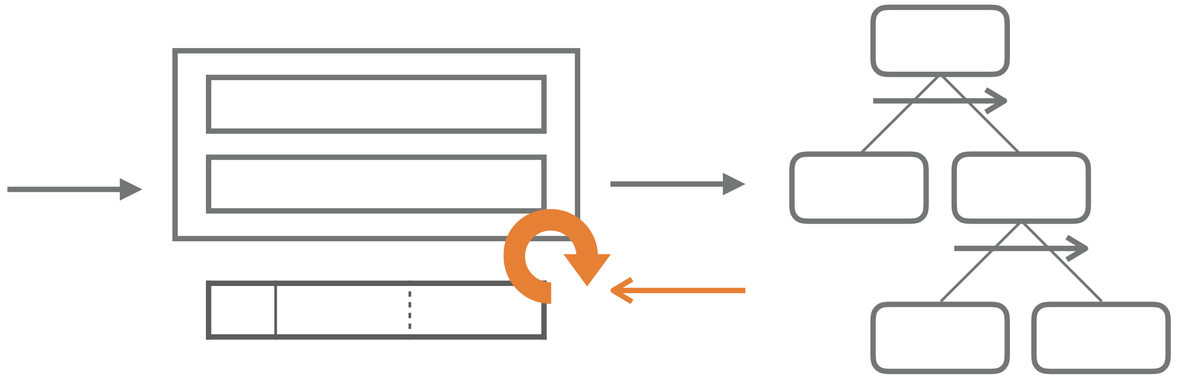
メインロジック

BDIベース

Perception
Blackboard
…

信念　　願望　　意図

巡回	0.0
戦闘	1.0
逃げる	0.5



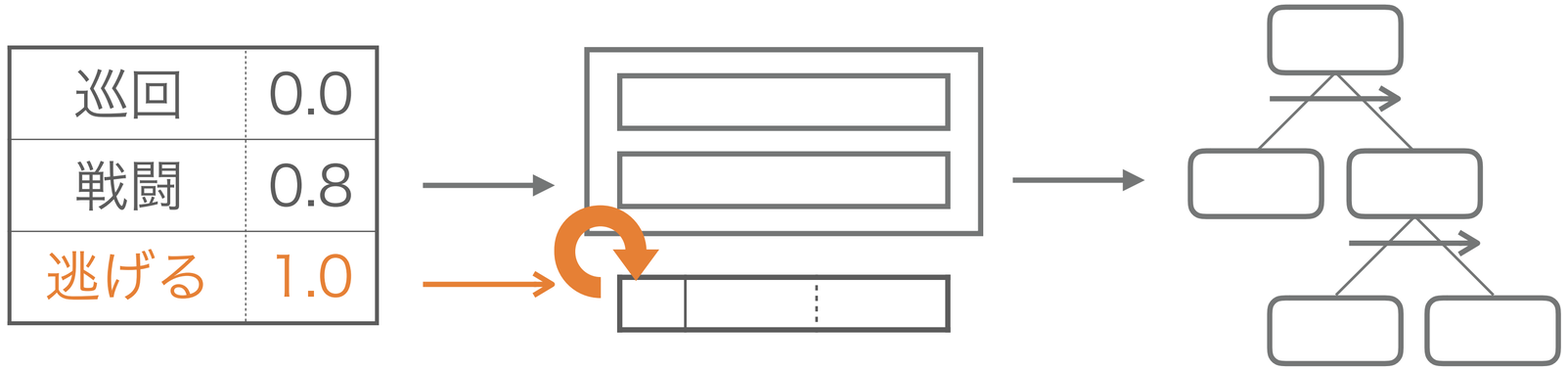
プランの成功、失敗、時間経過でリプランニング

メインロジック

BDIベース

Perception
Blackboard
…

信念　　願望　　意図



ゴールが変わった場合は実行中のプランを中断して新たなゴールでプランニングする

Scripted Behavior

- 演出的な振る舞いは
  Utility System に差し込んで実行する
- ワンモーション再生するだけなど
  単純な演出の場合は
  プランニングを経ずに直接BT実行や
  アクションのリクエストをすることも可能

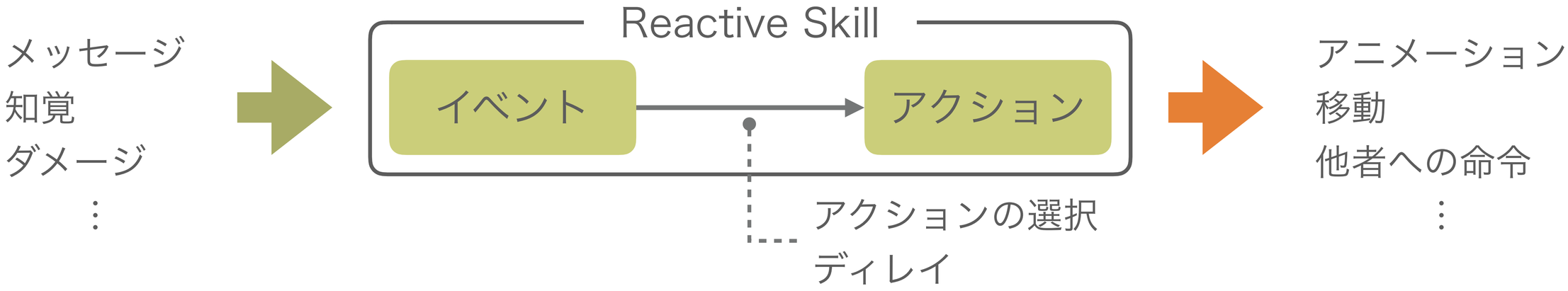
効用関数	実行タグ	BT

効用関数	実行タグ	アクション

例：登場演出

リアクション

- 思考によらない反射的な行動を扱う（例：回避、カウンター）
- 反応するイベントとそれに対応するアクションのセットで定義される
- イベントを受信すると付随するパラメータからアクション自体の選択やそのパラメータ、実行までのディレイなどを計算しリクエストする



回避行動

プレイヤー

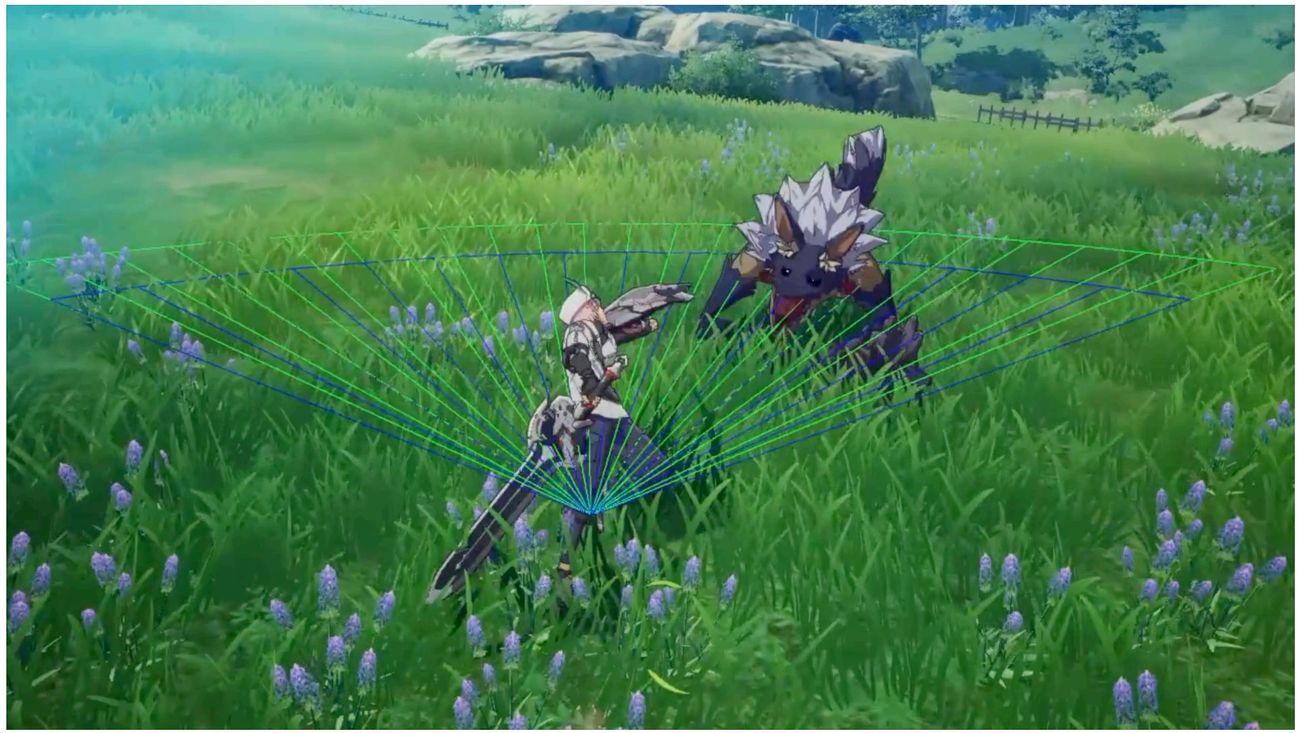
攻撃通知（攻撃範囲、ヒットが出るまでの時間）

Reactive Skill

- 使用できる回避アクションの種類
- アクションのサンプリングデータ
  （移動量、時間など）

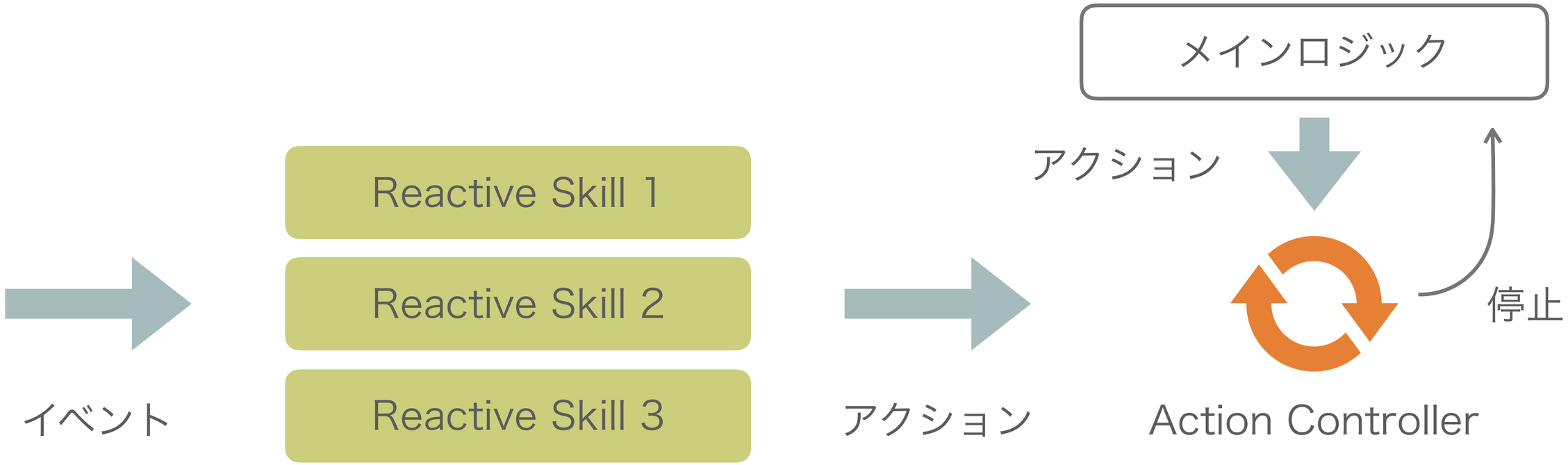
リクエスト

アニメーションの再生



リアクション

- 複数の Reactive Skill が並列で動いている
- 意思決定層はしたいことを決めるだけで、アクション層が調停し実行する
- リクエストされたアクション次第ではメインロジックを止める



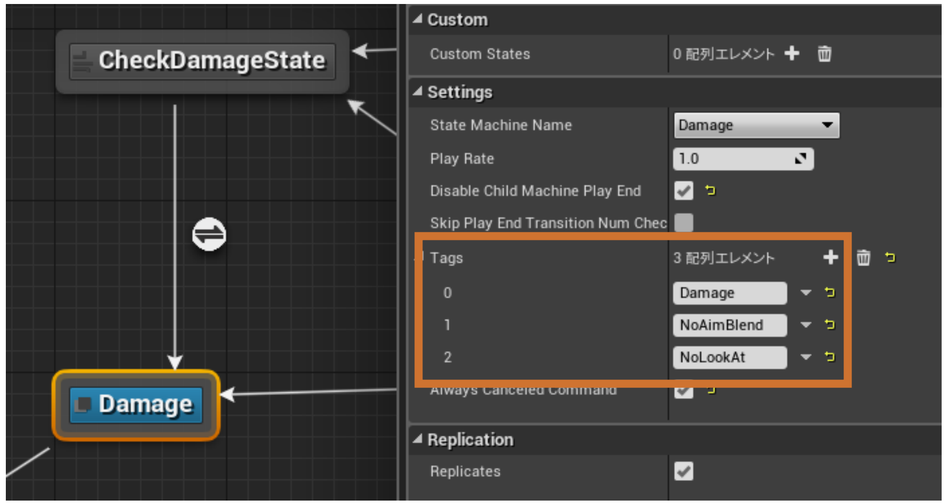
回避行動

BANDAI NAMCO Studios

外部要因による状態変化

外部要因（攻撃など）によって引き起こされた強制的な状態の変化は
身体側からの通知を Reactive Skill が受け取る

例：ダウンや状態異常による行動不能



身体（アニメーションステートマシン）

AI

Reactive Skill

メインロジック

停止

アクションリクエスト
（最高優先度、停止属性）

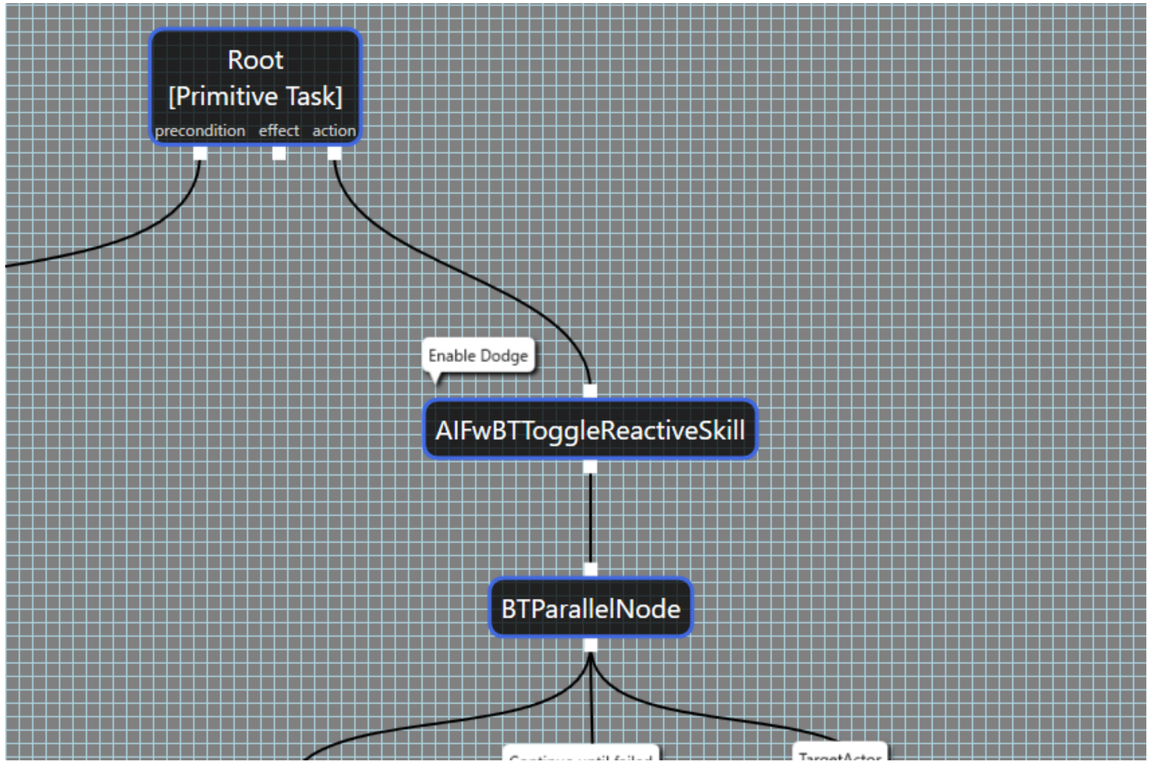

Main Logic: Battle
Reactive Skill: Behavior Restriction
Reactive Skill: Dodge

コンテキスト

ゲーム都合や振る舞いの自然さを求めるために
特定の文脈では反応して欲しくないことがある

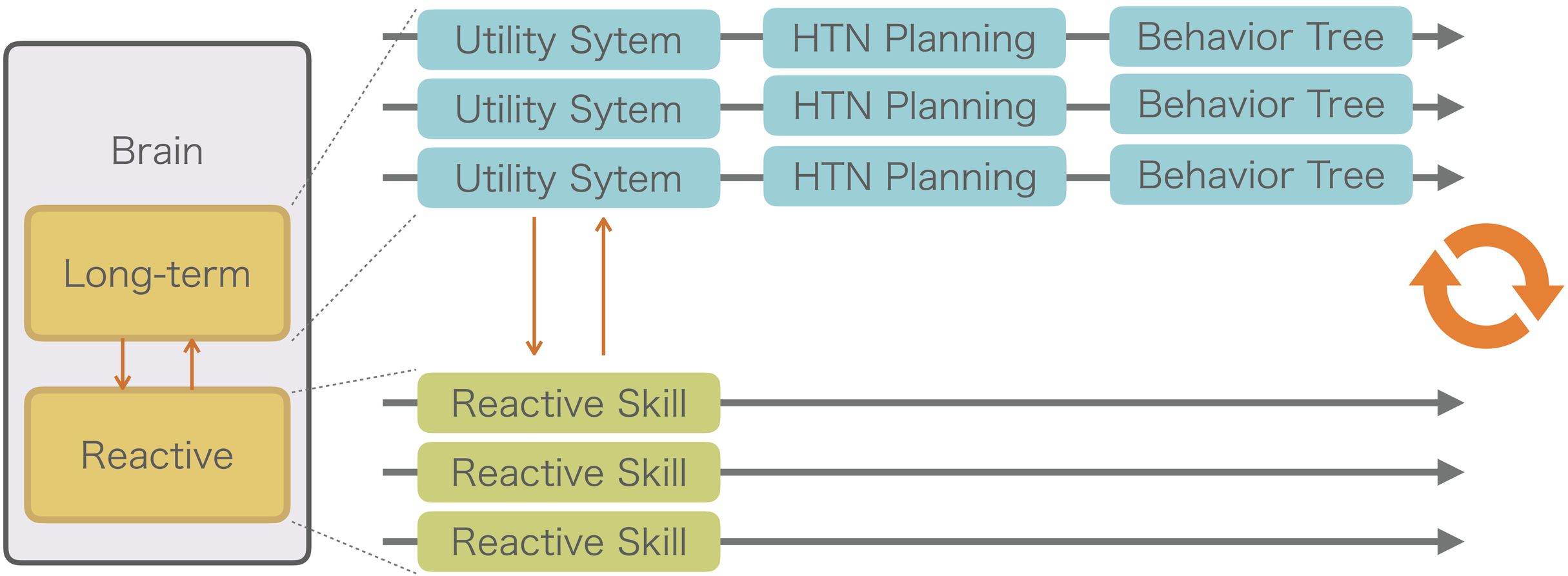
例：攻撃中は回避させたくない

メインロジックから
特定のリアクションを抑制 or 許可する




Main Logic: Battle
Reactive Skill: Behavior Restriction
Reactive Skill: Dodge

ここまでのまとめ


Brain
Long-term
Reactive
Utility Sytem
HTN Planning
Behavior Tree
Utility Sytem
HTN Planning
Behavior Tree
Utility Sytem
HTN Planning
Behavior Tree
Reactive Skill
Reactive Skill
Reactive Skill

ここまでのまとめ

BANDAI NAMCO Studios

問題	行動の一貫性を持たせることが難しい 割り込み行動を気軽に追加できない
方法	長期の目標を扱うシステムと短期の反射行動を扱うシステムを並列で動作 それぞれの結果を調停してアクションを実行する
結果	行動の一貫性と即応性を両立できるようになった 複数の目標を同時に志向することができるようになった

アジェンダ

BANDAI NAMCO Studios

- ☐ アーキテクチャ
- ☑ 個性づけ／量産
- ☐ パーティバトル
- ☐ まとめ

よくある疑問

- たくさんのシステムが組み合わされていると管理が大変そう
- 一体あたりの開発工数が大幅にかかりそう

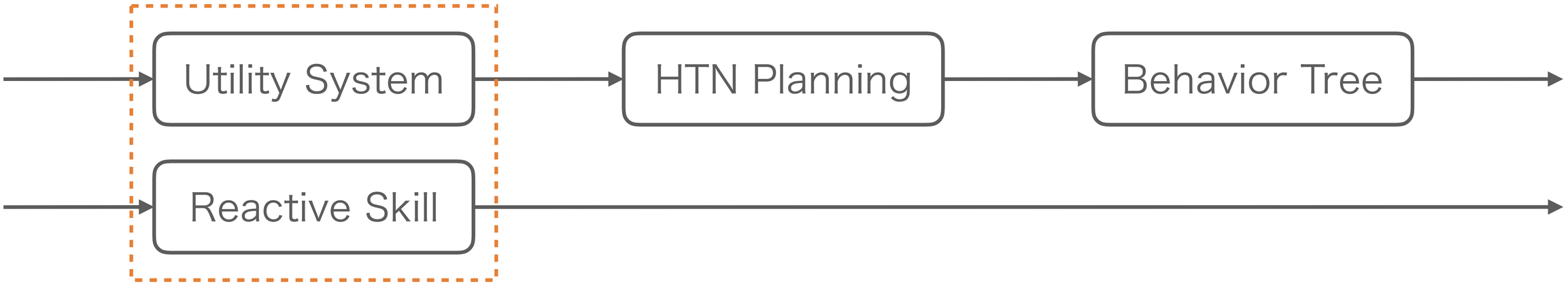
ある意味では正解

… 一つの Behavior Tree で完結するようなシンプルなAIならその方がいい

複雑な環境で動く複雑なAIの場合、部分問題に特化させたシステムを
組み合わせて個々を単純化させた方が効率的（背景で説明した通り）

とはいえ、工数は可能な限り減らさないと運営を含む量産をこなせない

BLUE PROTOCOLの例



BLUE PROTOCOLのキャラクターとして共通の性質

- 大枠の遊びが同じ以上、
  キャラクターごとに大きく変わることはない
- 構造がシンプルなので、
  固有のものが必要になってもすぐに対応可能

BLUE PROTOCOLの例

キャラクターごとの個性

- 具体的なキャラクターの行動を作る部分
- アクションはキャラクターごとに異なるので
  ドメインもキャラクターごとに固有になる
- コスト高になりがち

BLUE PROTOCOLの例

BANDAI NAMCO Studios

課題

- 個性を出すにはキャラクターごとに専用のドメインを作らないといけない
- でも、キャラクターごとにドメインを１から作るのはコストがかかりすぎる

方針

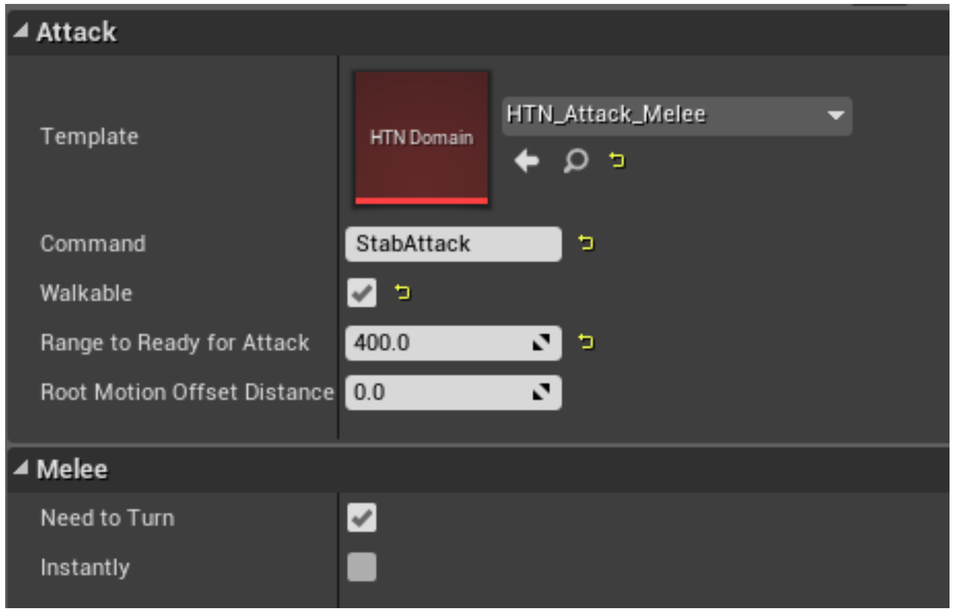
- ドメインをアクションごとにサブドメインに分割し、
  幾つものサブドメインを組み合わせて一つのドメインを作成する
- サブドメインは多くのキャラクターで共有できるようにする

Tactical Skill

BANDAI NAMCO Studios

- キャラクターが行う特定の行動について抜き出して実装したもの
- 攻撃、移動、待機などのあらゆる行動がスキルとして実装される

例：攻撃アクション



サブドメインのテンプレート

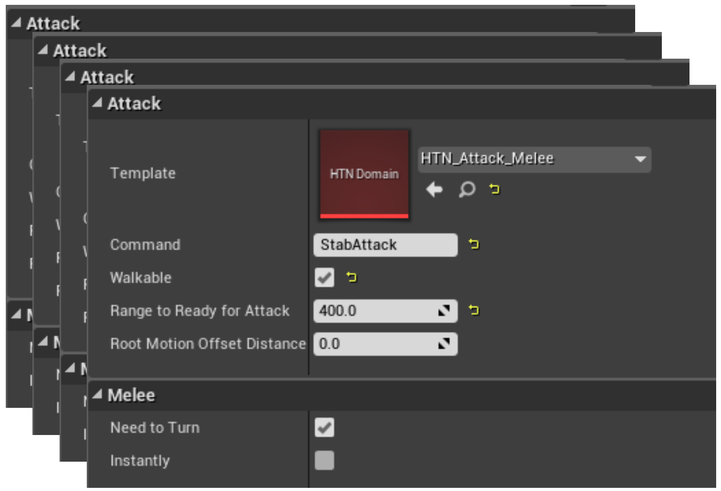
ドメインへ渡すパラメータ

Tactical Skill

- 汎用スキル
- テンプレートスキル
- 固有スキル

ほとんどの行動がこの２つで賄える

固有のものが必要になれば
その部分のみサブドメインを作成する



マージ

ドメイン

ドメインのマージポリシー

BANDAI NAMCO Studios

プリミティブタスク

$$A + B \neq B + A$$

- **既存タスク**：既存タスクをオーバーライド
- **新規タスク**：タスクを追加

複合タスク

- **既存タスクの既存メソッド**：既存タスクの既存メソッドをオーバーライド
- **既存タスクの新規メソッド**：既存タスクにメソッドを追加
- **新規タスク**：タスクを追加

ロジックの追加だけでなく、既存ロジックのオーバーライドも可能！

ガードあり

Tactical Skill

ガードなし

BANDAI NAMCO Studios

Tactical Skill

ここまでのまとめ

BANDAI NAMCO Studios

問題	キャラクターごとにドメイン（固有の思考ルーチン）を作る必要がある ドメインの作成コストが高い
方法	ドメインをサブドメイン＋パラメータ（Tactical Skill）に分割 スキルの組み合わせでキャラクターのAIを作成
結果	技の追加・削除を容易に行える（デバッグにも有用） 開発工数の大幅削減

これを活用するためには解決しなければいけない課題が…

HTNプランニングの課題

BANDAI NAMCO Studios

1. 一番最初に見つかったプランを返す

- 目標を達成するプランの候補が複数ある場合にプランの品質を考慮できない
  （一番いい攻撃手段の選択など）
- 設定の順序によってマージ結果が変わるので行動にも影響が出る

2. 計画に必要な情報を全てドメインに埋め込んでおかないといけない

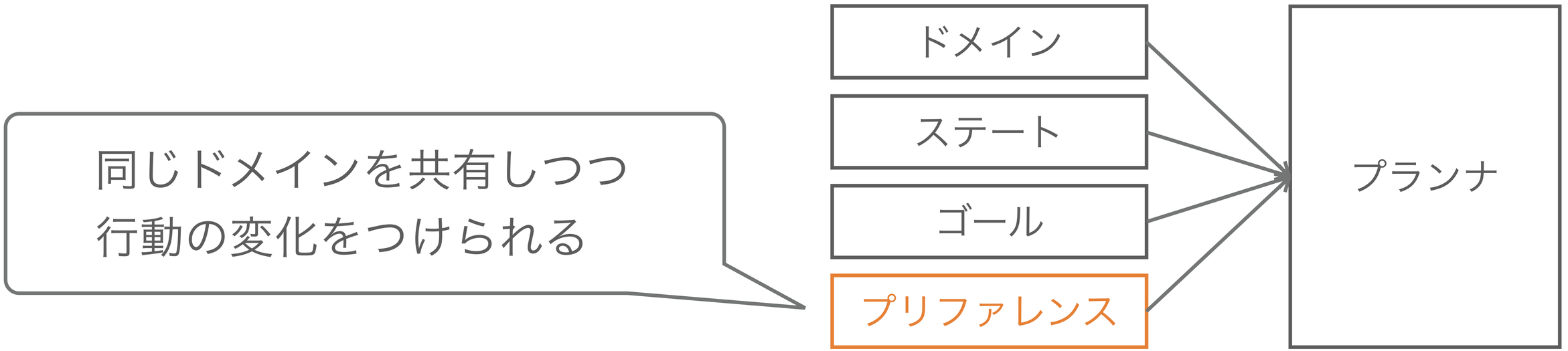
- 柔軟性が低く、多くのキャラクターでサブドメインを共有する場合に問題になる

  例：あるキャラクター用の対応が別のキャラクターの行動に影響してしまう

Preference-based HTN Planning

BANDAI NAMCO Studios

個人の嗜好に基づいた計画を立てる Preference-based Planning の HTN版



例：旅行問題

- 飛行機には乗りたくない：Always(Not(Occur(飛行機に乗る)))
- 費用と移動時間のバランスがいい交通手段を使いたい：Goal(Max(残金 - 移動時間))

Preference-based HTN Planning

BANDAI NAMCO Studios

制約の強さ

- ソフト制約
- ハード制約

制約の種類

- プリコンディションプリファレンス
- ゴールプリファレンス
- トラジェクトリープリファレンス

Preference-based HTN Planning

BANDAI NAMCO Studios

ソフト制約

- 可能な限り満たしてほしい条件を表す
- プラン中に満たされた数によって品質を計算する

ハード制約

- 事前条件と同様に満たすべき条件を表す
- 違反したプランはその時点で候補から削除される

Preference-based HTN Planning

BANDAI NAMCO Studios

プリコンディションプリファレンス

- タスクやメソッドに設定し、実行前の状態に対して評価される
- プラン中に複数回使用される場合はプリファレンスも複数回評価される

ゴールプリファレンス

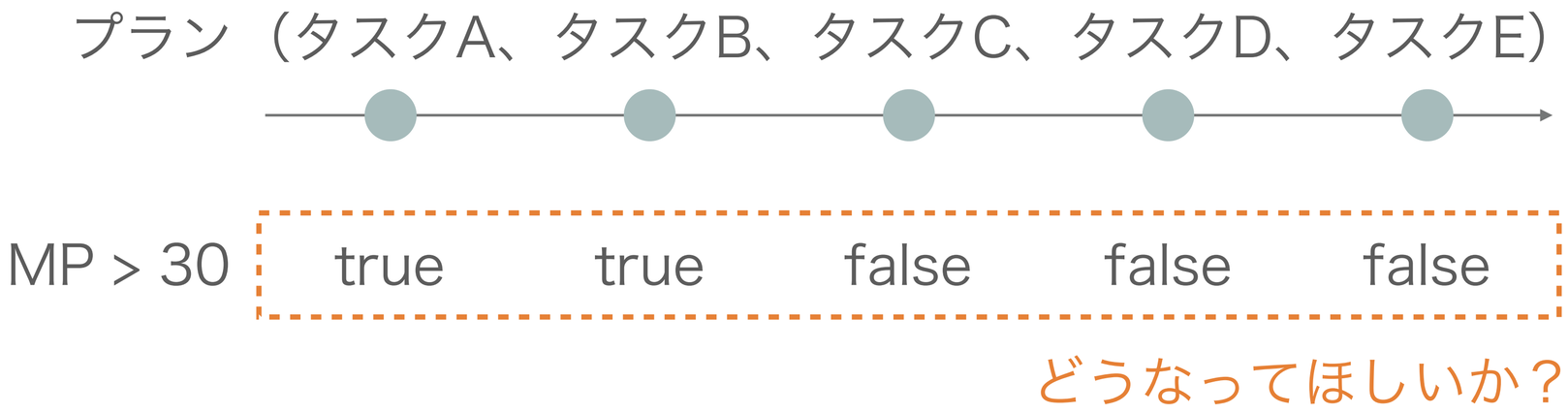
- プランが見つかった時にそのプランの最終状態に対して計算される
- ゴールプリファレンス専用のオペレータで
  プランの長さを評価することも可能

Preference-based HTN Planning

BANDAI NAMCO Studios

トラジェクトリープリファレンス

- プラン全体を通した条件の時間的変化に対して計算される
- 線形時相論理を基にしたオペレータを使用して評価する



Preference-based HTN Planning

BANDAI NAMCO Studios

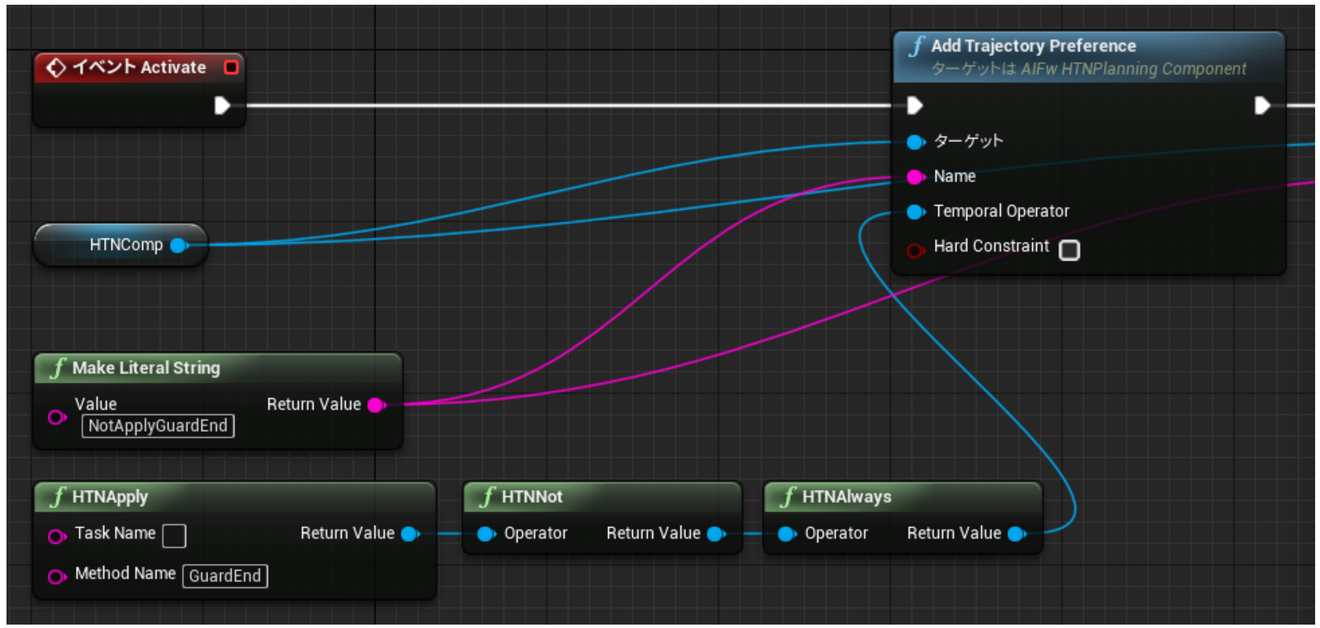
LTLオペレータ	説明
Always(A)	常にAを満たす
AlwaysBefore(A, B)	Aが満たされるまで、常にBを満たす
AlwaysAfter(A, B)	Aが満たされた後、常にBを満たす
Sometime(A)	任意の１箇所以上でAを満たす
SometimeBefore(A, B)	Aが満たされるまでに１箇所以上でBを満たす
SometimeAfter(A, B)	Aが満たされた後、１箇所以上でBを満たす

オペレータ	説明
Occur(task)	taskが発生したか
Apply(method)	methodが適用されたか
Apply(task, method)	taskのmethodが適用されたか

Preference-based HTN Planning

BANDAI NAMCO Studios

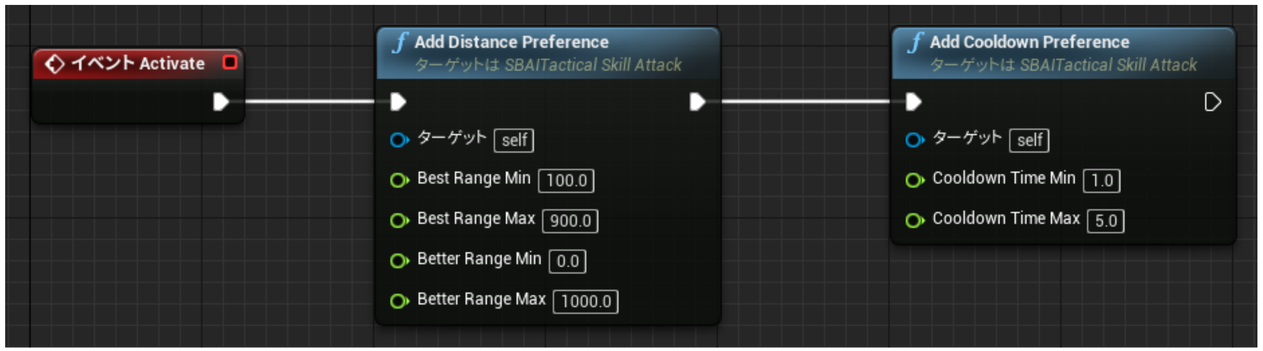
- プリファレンスはプログラム、ブループリントから設定
- デザイナはプログラマが用意した関数やプロパティを経由して設定



攻撃

攻撃アクションのメソッドに
プリコンディションプリファレンスを設定し
今の状況にどの程度適しているか評価する

例：ターゲットまでの距離、角度、
そのアクションを前回使ってからの経過時間



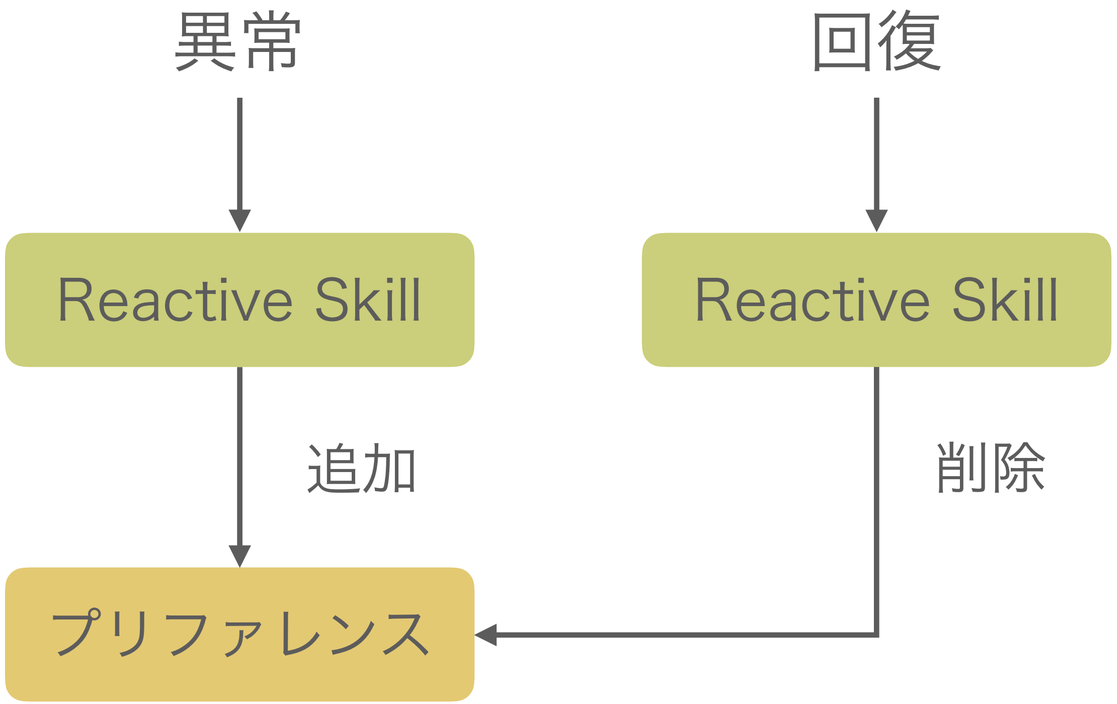
攻撃

ターゲットまで移動せずにその場で使ってほしい攻撃の場合は
トラジェクトリープリファレンスで移動を制限する

例：AlwaysBefore(Occur(攻撃A), Not(Occur(移動))) as hard constraint

状態変化

- 部位破壊やシールドブレイクによる行動の制限
- 状態異常による行動優先度の変化



Always(Not(Occur(ガード)))

固定砲台

レベルデザイナのオーダー

1. 高台の上などレベルで指定した場所から遠距離攻撃をしてきてほしい
2. プレイヤーが近づいたときはそこから離れて通常通り動いてほしい

スマートオブジェクト

Tactical Skill

- サブドメイン
- プリファレンス

固定砲台

サブドメイン

固定砲台

（スマートオブジェクト(SO)まで移動、遠距離攻撃）

プリファレンス

- 通常の行動より優先してほしい：Sometime(Occur(固定砲台))
- 相手が近いと使わない：AlwaysAfter(Occur(SOまで移動), DistToTarget > 500)
- 場所を移動しない：AlwaysAfter(Occur(SOまで移動), Not(Occur(移動)))

青：ソフト制約、赤：ハード制約

ここまでのまとめ

BANDAI NAMCO Studios

問題	プランの品質を考慮できない プランニングに必要な条件を全てドメインに記述する必要がある
方法	嗜好を考慮して計画を立てる Preference-based HTN Planning を導入
結果	ドメインを変更せずに振る舞いを変えることができるようになった 動的な振る舞いの変化も容易に行えるようになった

アジェンダ

BANDAI NAMCO Studios

- ☐ アーキテクチャ
- ☐ 個性づけ／量産
- ☑ パーティバトル
- ☐ まとめ

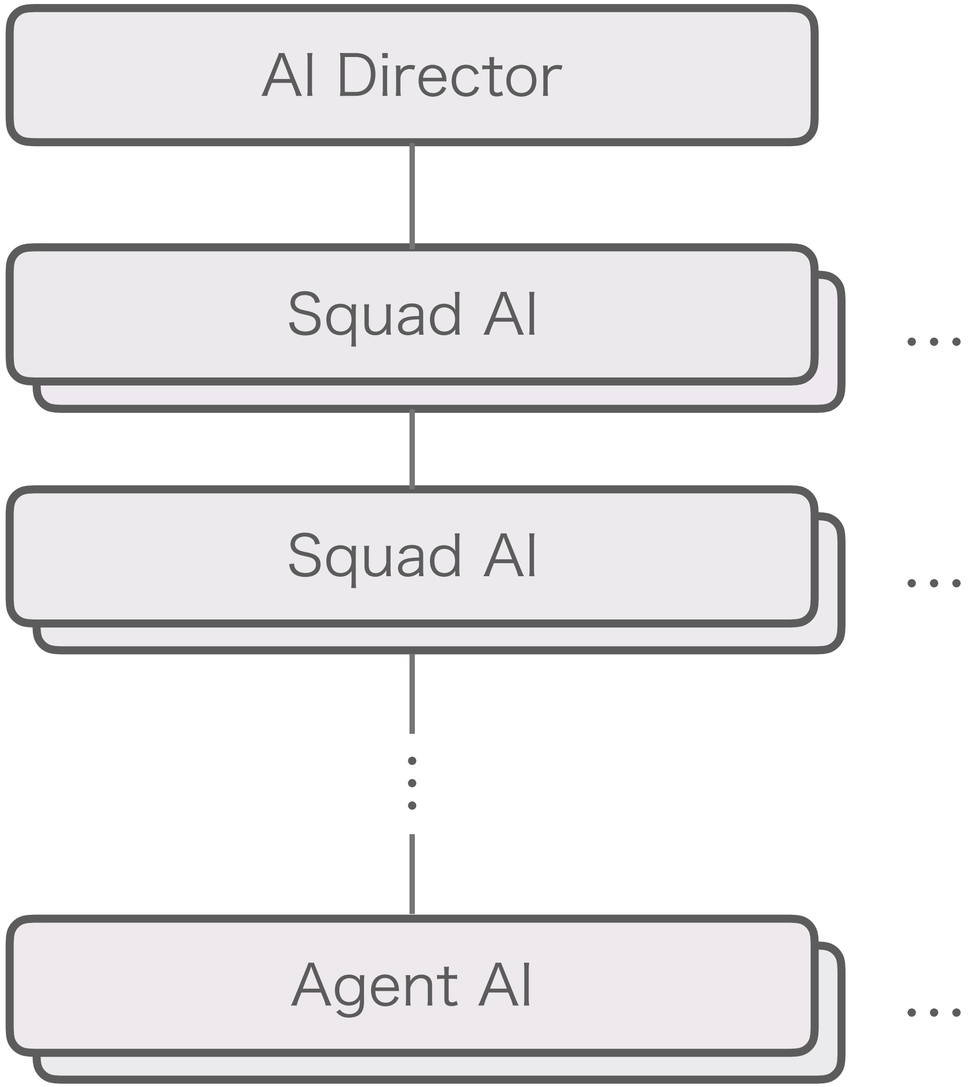
注意事項

BANDAI NAMCO Studios

ここで紹介するものは
パーティバトルを実現する
一要素であり
これが全てではありません

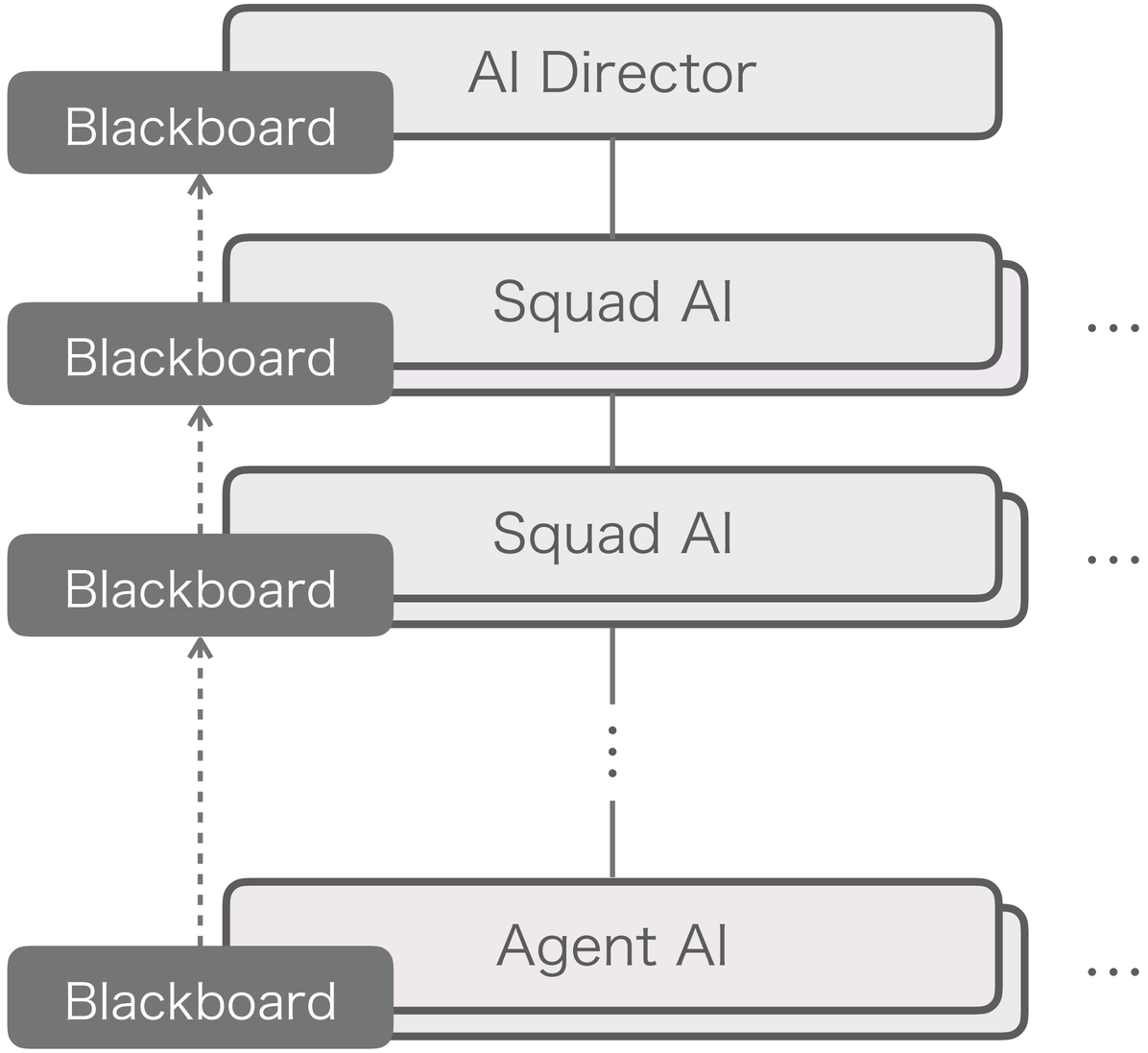
AIヒエラルキー

- キャラクターを操作する個々のAIを
  集団を指揮するAIが階層的に管理
- 基本的なアーキテクチャは共通



AIヒエラルキー

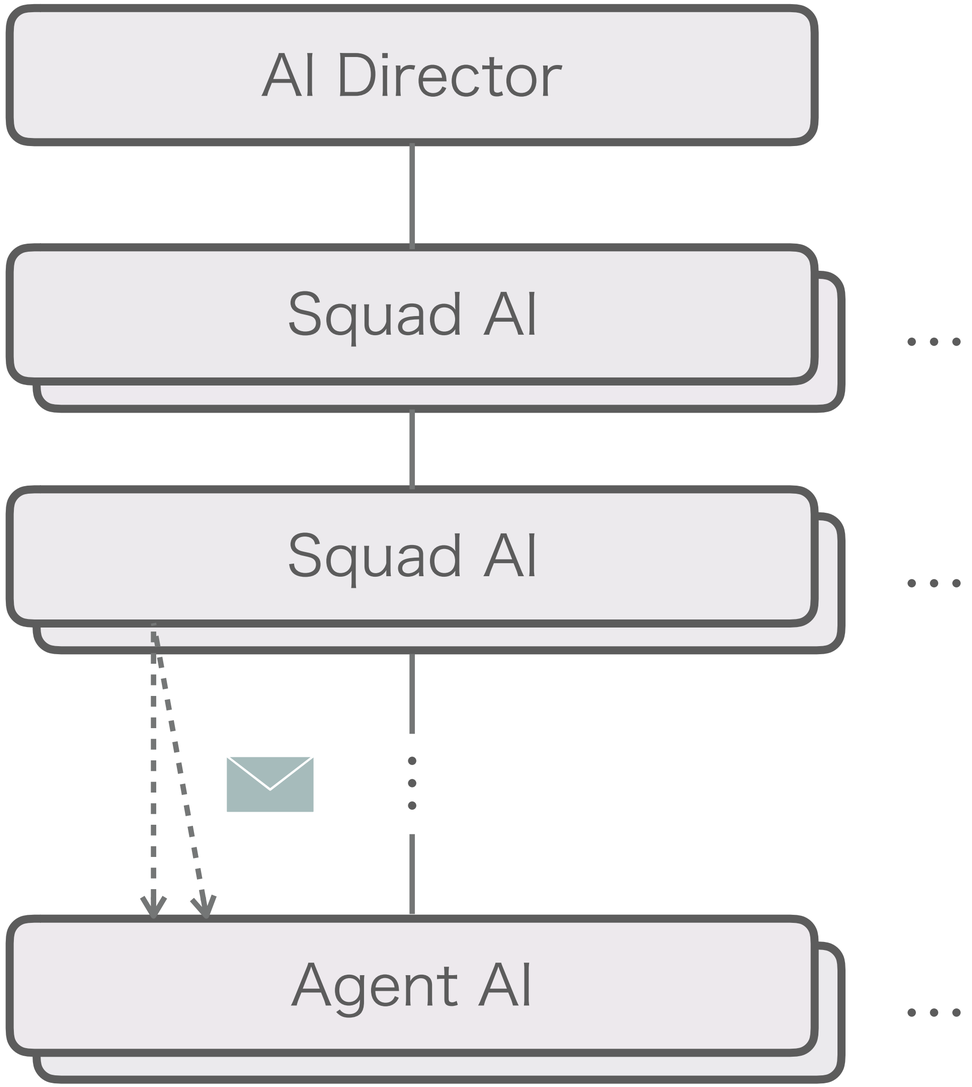
- キャラクターを操作する個々のAIを
  集団を指揮するAIが階層的に管理
- 基本的なアーキテクチャは共通
- Blackboardの親子付けによる
  親の情報への透過的なアクセス



AIヒエラルキー

- キャラクターを操作する個々のAIを
  集団を指揮するAIが階層的に管理
- 基本的なアーキテクチャは共通
- Blackboardの親子付けによる
  親の情報への透過的なアクセス
- メッセージのブロードキャスト
  - メッセージに反応する Reactive Skill を
    使用してコンテキストに応じた返信が可能

    例：ブロードキャストで救援要請を送り
    救援が可能なエージェントのみ返信する

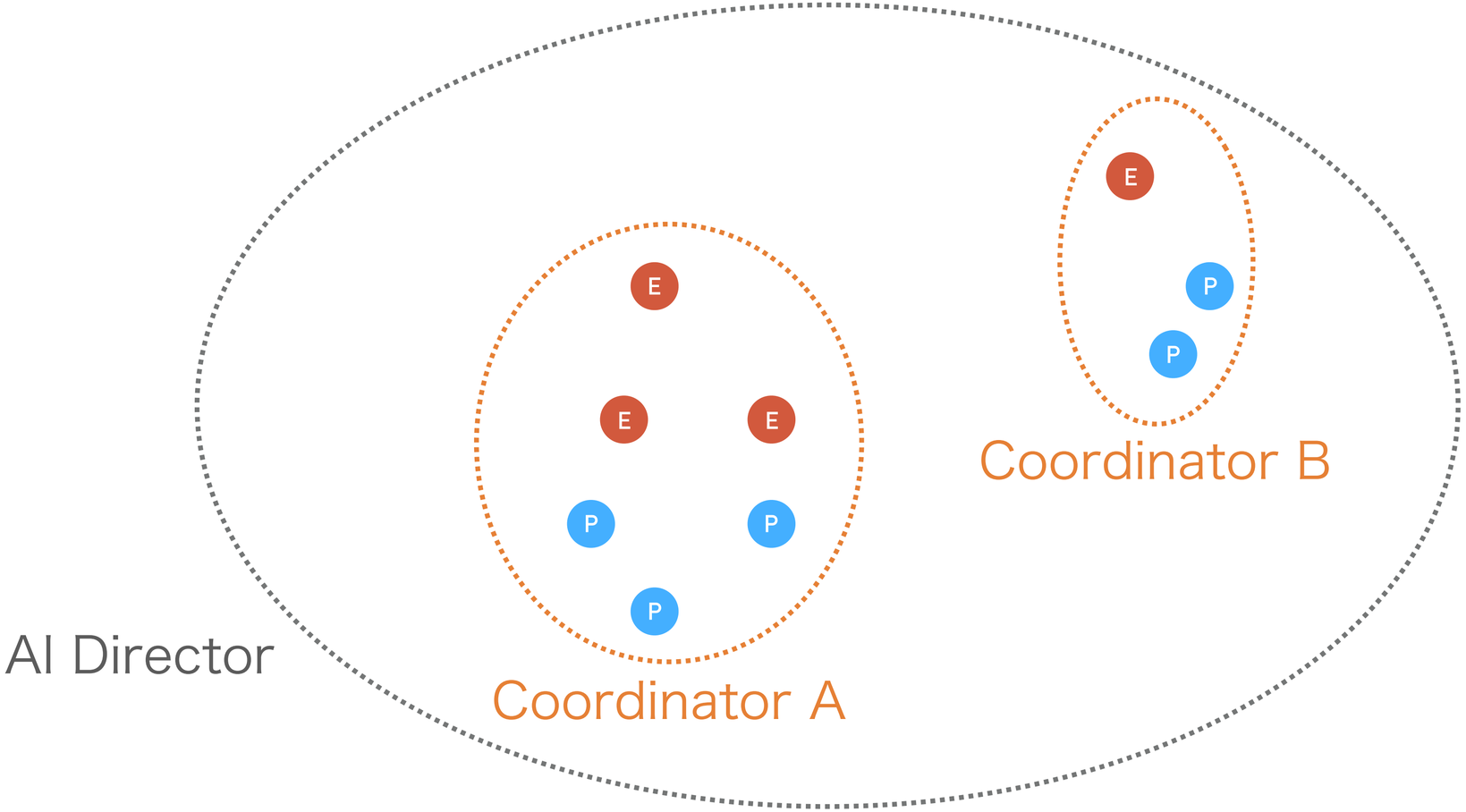


バトルコーディネート

もっともエージェントAIに近いところには
プレイヤー対エネミーの一つの戦闘単位を管理するAIがある

- 攻撃権の管理
- ターゲットの分散
- **ロール**の割り振り

⋮



ロール

戦闘におけるキャラクターの役割を定義したもの

例：サポーターは後ろからバフをかける、ディフェンダーがヒーラーを守る

要件

状況に応じてロールを変えてほしい

- ➤ 盾が壊れたらディフェンダーからアタッカーになる
- ➤ 前衛、後衛が可能なユーティリティアタッカーは足りないところに回る

コーディネータ側からロールの変更を要求する場合もある

- ➤ 攻撃役のキャラクターが足りない

目標：どんなキャラクターもロールの最低要件さえ満たしていればどんなロールにもなれる

ロール

ロール設定

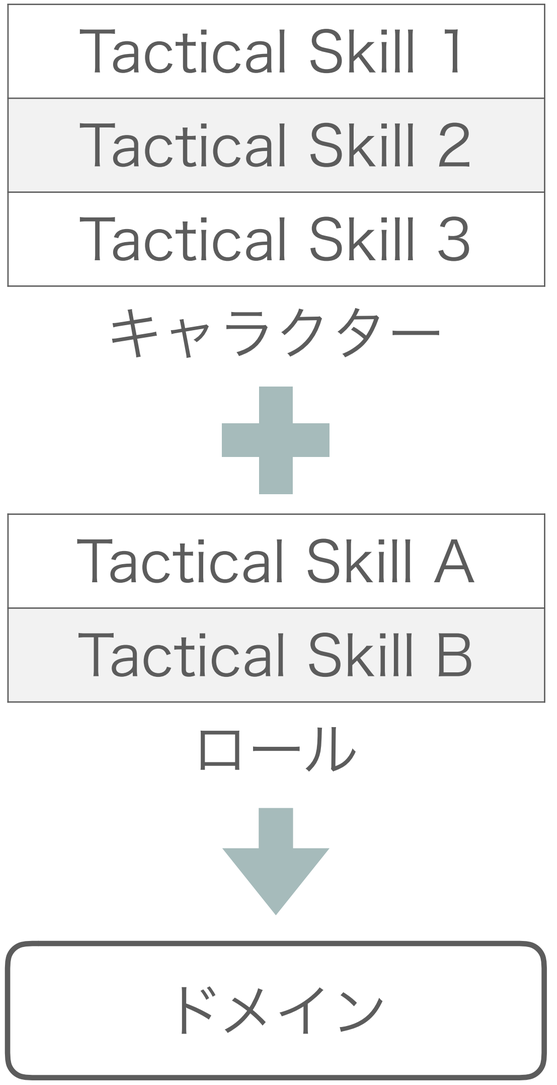
- ロール専用の立ち回りやアクションは
  Tactical Skill で追加

例：ディフェンダーのヒーラーを守る行動

- 行動の優先度の変更や抑制はプリファレンスで制御

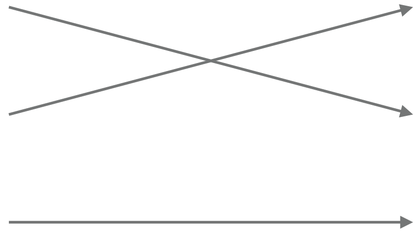
ロールの切り替え

- 専用行動：ハード制約のプリファレンスで使用を禁止
- 行動優先の変更：プリファレンスを削除



ロールアサイン

	優先度	数
アタッカー	1.0	1
ディフェンダー	0.8	1
フリー	0.0	∞



	前衛	後衛	ディフェンダー
キャラクターA	0.9	0.0	1.0
キャラクターB	0.0	1.0	0.0
キャラクターC	0.6	0.4	0.0

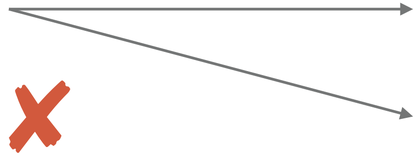
コーディネータ：必要ロール

キャラクター：ロールの希望

- コーディネータ側で必要としているロールから割り当てていく

ロールアサイン

	優先度	数
アタッカー	1.0	2
ディフェンダー	0.8	1
フリー	0.0	∞



	前衛	後衛	ディフェンダー
キャラクターA	0.9	0.0	1.0
キャラクターB	0.0	1.0	0.0
キャラクターC	0.6	0.4	0.0

コーディネータ：必要ロール

キャラクター：ロールの希望

- コーディネータ側で必要としているロールから割り当てていく
- 割り当てられるキャラクターがいない場合はスキップ


近接アタッカー
ディフェンダー
ヒーラー
ロールの設定以外は全て同じ

アジェンダ

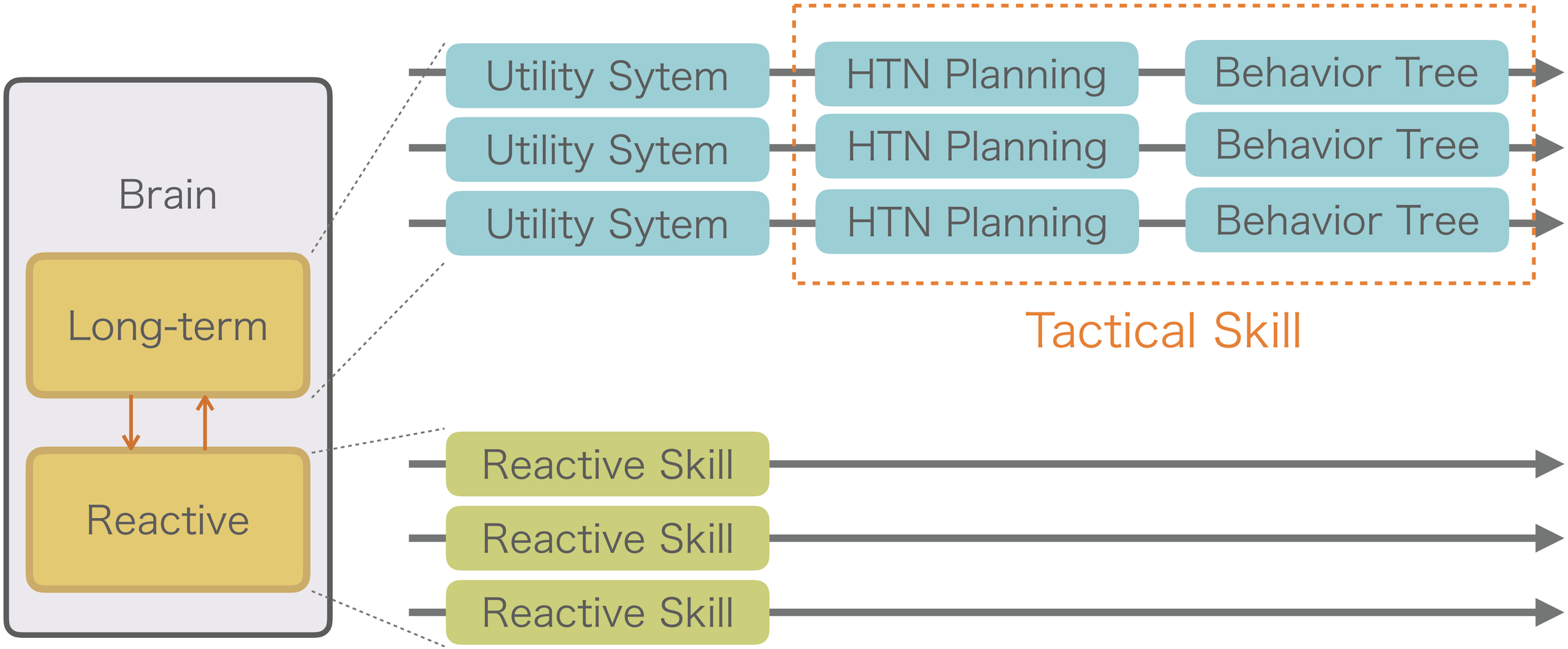
BANDAI NAMCO Studios

- ☐ アーキテクチャ
- ☐ 個性づけ／量産
- ☐ パーティバトル
- ☑ まとめ

まとめ

BANDAI NAMCO Studios

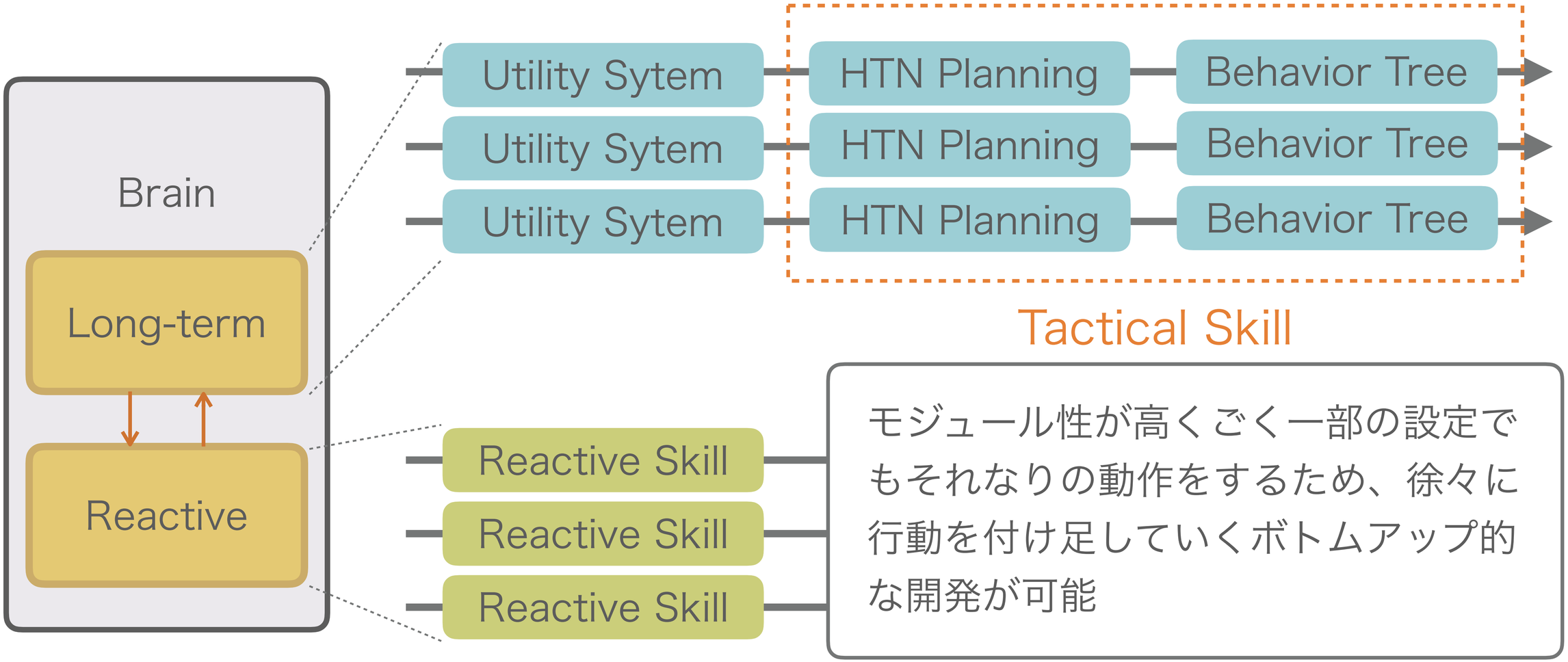
複数の振る舞いが並列で動く分散型アーキテクチャ



まとめ

BANDAI NAMCO Studios

複数の振る舞いが並列で動く分散型アーキテクチャ



まとめ

1. 多種多様なキャラクターの表現

2. どのような
   成立するル

3. 運営時の
   対応できる柔軟性

4. 工数削減

- 固有スキルを追加することで拡張できる
- プリファレンスを使うことで同じスキルを共有してても微妙な行動の変化をつけられる

まとめ

1. 多種多様なキャラクターの表現

2. どのようなキャラクターの組み合わせでも
   成立するパーティバトル

3. 運営時の
   対応できる

4. 工数削減

- キャラクターに依存せず設定可能なロール
- コーディネータによる状況に応じた
  動的なロールのアサイン

まとめ

1. 多種多様な

2. どのような
   成立する

3. 運営時のキャラクターやコンテンツの追加に
   対応できる柔軟性

4. 工数削減

- スキルの追加による高い拡張性
- プリファレンスを使うことで既存ロジックを
  変更せずに行動に変化をつけられる

まとめ

1. 多種多様なキャラクターの表現

2. どのようなキャラクターの組み合わせでも
   成立するパーティバトル

3. 運営時の
   対応でき

4. 工数削減

- スキルの組み合わせによってAIを構築できる
- 高いモジュール性を生かした反復型開発

副産物（副作用？）

当初作成の予定がなかった通常のエネミーから技の構成を変更した
リーダー系エネミー、レアエネミーが
（プログラマの知らないところで）作られる

これだけは覚えて帰って

BANDAI NAMCO Studios

プリファレンスはすごい！

思考ルーチン（ドメイン）をいじらずに行動を変えられる

- 事前にキャラクターAIに仕込んでおかなくても
  コーディネータ側から自由に命令を与えられる

抽象的な行動を指定できる

- 特定のキャラクターに依存しない