

Amazon GameLift FlexMatch で マルチプレイヤーゲームを創造する

石井 宇大

技術統括本部 ソリューションアーキテクト
Amazon ウェブ サービス ジャパン合同会社

About Me

石井 宇大 (Takahiro Ishii)

Game Developer Relations

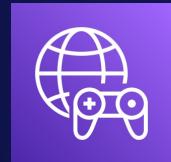
AWS Game Tech

 @takahiroishii_



ゲーム開発者のみなさんにAWSから沢山の選択肢を提供します

好きなAWS Services:



Amazon GameLift



Amazon DynamoDB

Agenda

AWS Game Tech

Multiplayer Games and Matchmaking

Amazon GameLift FlexMatch

Rule Set Samples

Key Takeaway

様々なマルチプレイヤーゲームのニーズに合うFlexMatchを使うことで
新たな楽しいゲームが創りだされます

AWS Game Tech



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Behind great games, there's AWS Game Tech

AWSはゲーム開発スピードを加速させ
あなたのゲームをよりスマートに動かし
そして成長させます

なぜ世界中の90%以上の代表的ゲーム会社が
AWSを利用しているのか確かめてみてください



プレヤー体験向上のために





Trusted infrastructure

“Ubisoftはオリジナルで記憶に残るゲーム体験によってプレーヤーライフを豊かにする事に取り組んでいます。For HonorというゲームではAmazon GameLiftのゲームサーバーによってこの取り組みを続ける事ができます。また、FleetIQのSpotインスタンスを活用することで、円滑でスケーラブルなプレーヤー体験を提供できています。”

Laurent Chouinard
Online Programming Lead, Ubisoft



and more ...



Multiplayer Games and Matchmaking

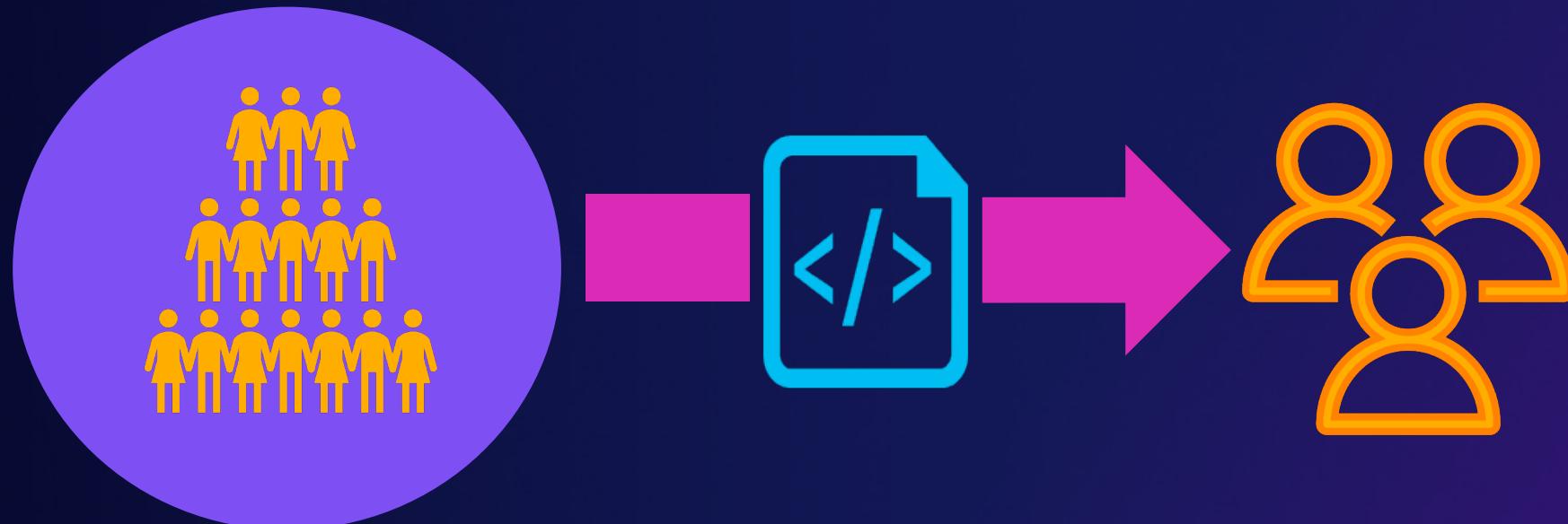
Multiplayer Games



- 他のプレイヤーと対戦したり協力したりするゲーム
- ゲームによって同時に遊ぶ人数や役割が違う
- バトルロイヤルゲーム
- チーム対戦ゲーム
- 1対1ゲーム
- ...

Matchmaking

特定のルールに基づき沢山のプレイヤーがいるプレイヤー
プールから一緒に遊ぶプレイヤーグループを作りだす作業



Fun Multiplayer Games



ストレス無く遊べること
自分のペースで遊べること



マッチングの速さ・公平さ
マッチングの選択肢・種類

Keys in Matchmaking

マルチプレイヤーゲームを楽しくする鍵はマッチメイキング



- 速いマッチメイキング ✓
- 公平性の高いカスタムルール ✓
- 一つのゲームに複数のマッチメイキング ✓
- 種類の異なるマッチメイキング ✓

Amazon GameLift FlexMatch



Amazon GameLift FlexMatch

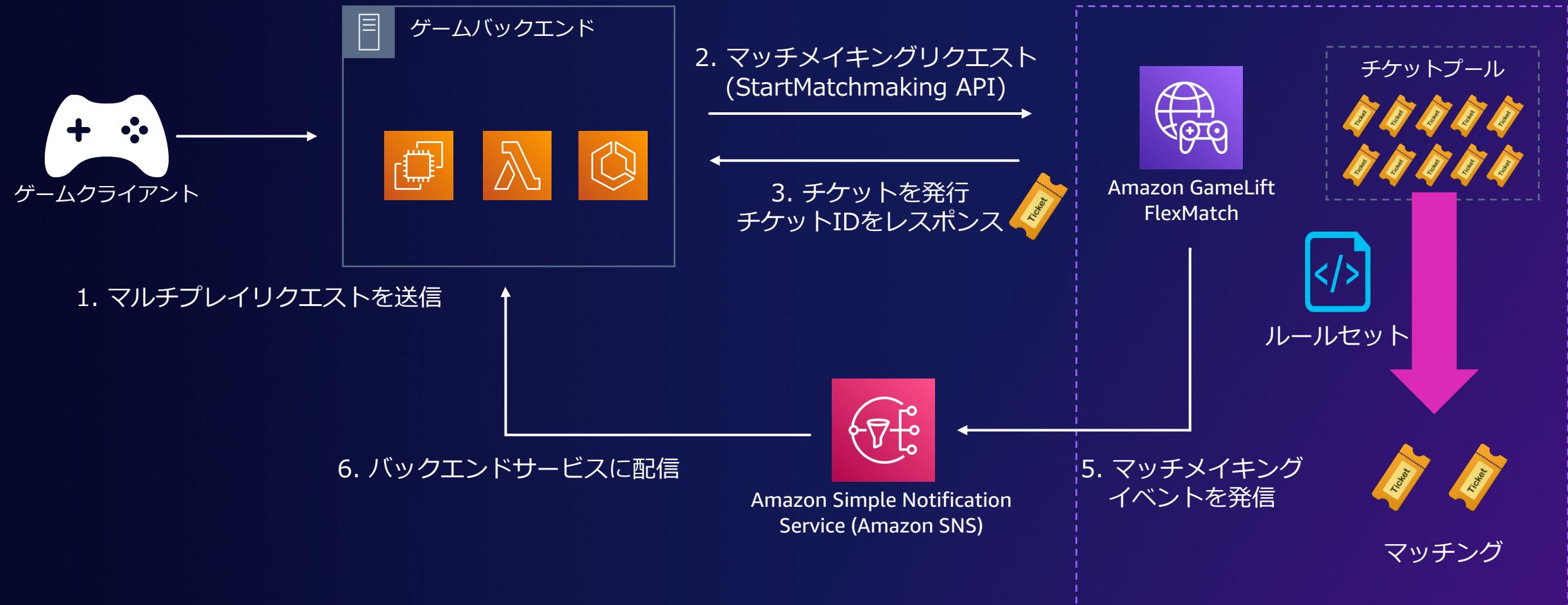
カスタマイズ可能なマッチメイキングサービス
フレキシブルなマッチングによってマルチプレイヤーゲームをさらに楽しく！

- ・簡単にゲームにあったカスタムルールを複数作れる
- ・チームマッチング対応
- ・サーバーとの間のレイテンシーを基準にマッチング可能
- ・時間と共にルールの条件を緩和
- ・マッチングされた後に明示的に受託可能
- ・マッチメイキングのメトリクスを取得
- ・Standaloneとして単独で利用可能
- ・キューを使用してゲームセッションを効率的に配置



Amazon GameLift FlexMatch

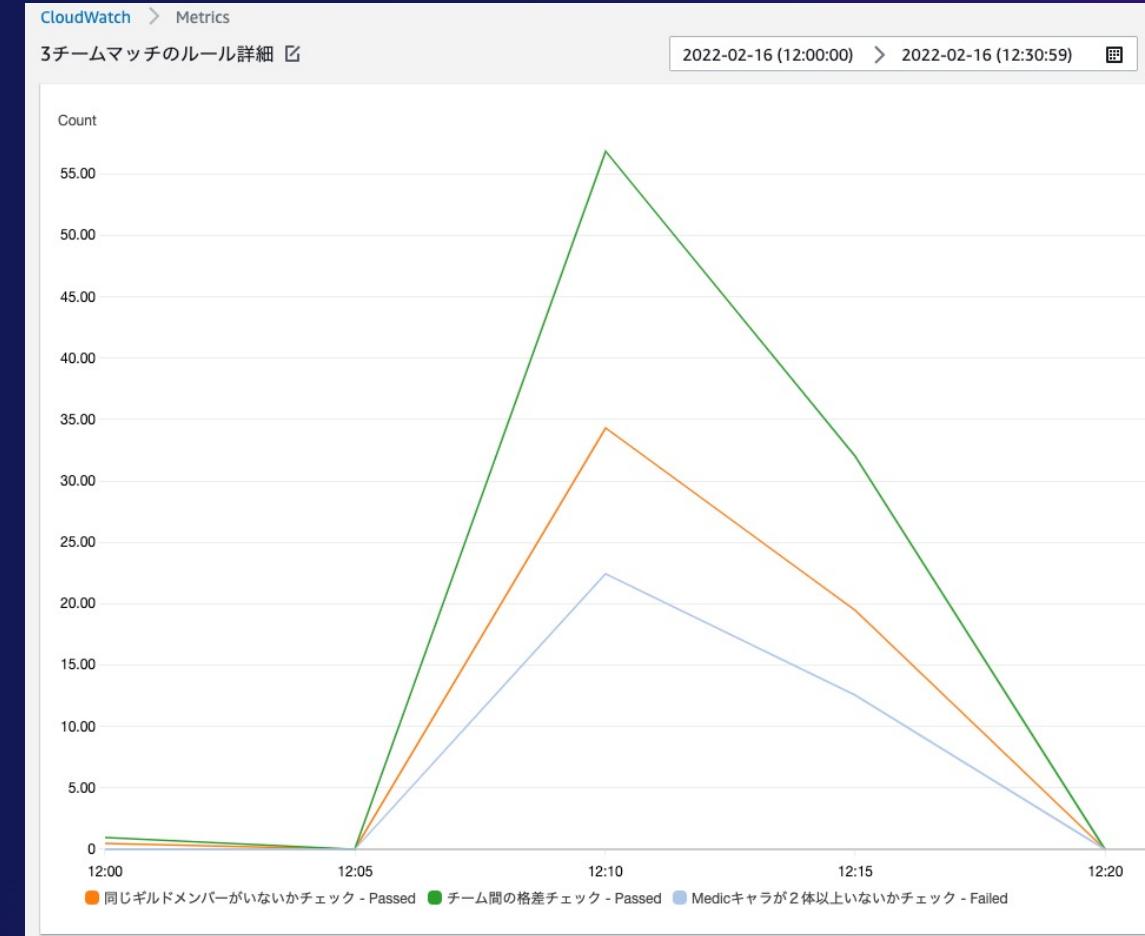
マッチメイキングの流れ



Make It Fast

様々なメトリクスを見ることでゲームに合った最も速いルールを選別可能

- 開発段階で複数のルールを作成し比較する
- ライブゲームの詳細情報を確認しすぐにルールを変更できるようにしておく
- 同じゲームモードに対しても複雑度別でルールを準備しておく
- 時間経過によるルールの制限緩和を活用する



Make It Fair

より面白くするためのゲームに合ったルールを作る

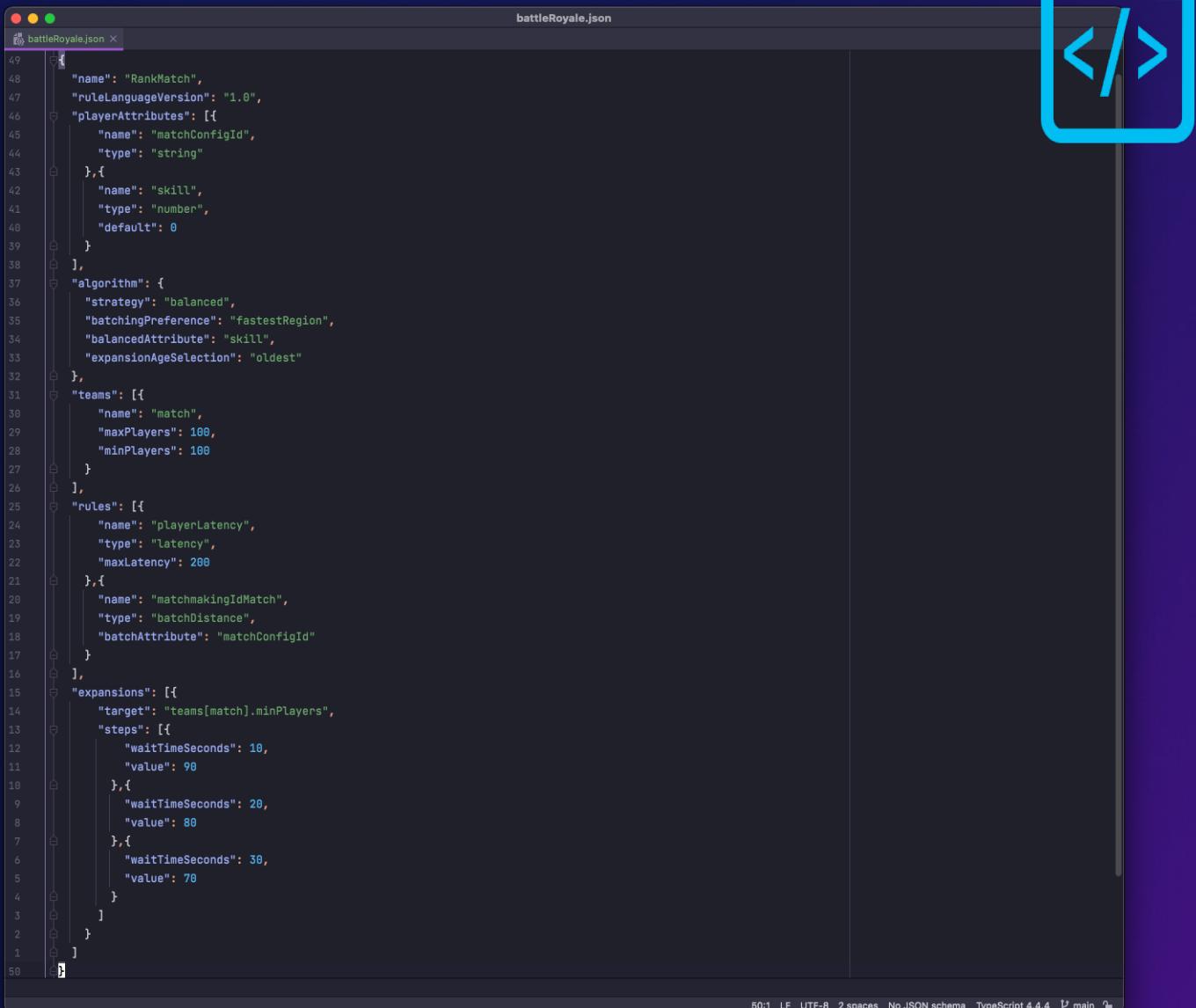
- プレイヤーのスキルレベルをマッチングに使いたい
 - 2つのチームを 4~8 人で構成したい
 - 4つのチームのスキルレベルの平均の差が10を超えないようにマッチングさせたい
 - もしマッチングしないならスキルレベルの差の上限を緩めたい
 - ..



Rule Set Samples

Rule for BattleRoyale

100人対戦のLarge Match
チーム制では無い
レイテンシーでグループ分け
レイテンシーの上限
時間経過で人数を緩和



```
battleRoyale.json
{
  "name": "RankMatch",
  "ruleLanguageVersion": "1.0",
  "playerAttributes": [
    {
      "name": "matchConfigId",
      "type": "string"
    },
    {
      "name": "skill",
      "type": "number",
      "default": 0
    }
  ],
  "algorithm": {
    "strategy": "balanced",
    "batchingPreference": "fastestRegion",
    "balancedAttribute": "skill",
    "expansionAgeSelection": "oldest"
  },
  "teams": [
    {
      "name": "match",
      "maxPlayers": 100,
      "minPlayers": 100
    }
  ],
  "rules": [
    {
      "name": "playerLatency",
      "type": "latency",
      "maxLatency": 200
    },
    {
      "name": "matchmakingIdMatch",
      "type": "batchDistance",
      "batchAttribute": "matchConfigId"
    }
  ],
  "expansions": [
    {
      "target": "teams[match].minPlayers",
      "steps": [
        {
          "waitTimeSeconds": 10,
          "value": 90
        },
        {
          "waitTimeSeconds": 20,
          "value": 80
        },
        {
          "waitTimeSeconds": 30,
          "value": 70
        }
      ]
    }
  ]
}
```



Rule for BattleRoyale

100人対戦のLarge Match

チーム制では無い

レイテンシーでグループ分け

レイテンシーの上限

時間経過で人数を緩和



```
battleRoyale.json
{
  "name": "RankMatch",
  "ruleLanguageVersion": "1.0",
  "playerAttributes": [
    {
      "name": "matchConfigId",
      "type": "string"
    },
    {
      "name": "skills"
    }
  ],
  "teams": [
    {
      "name": "match",
      "maxPlayers": 100,
      "minPlayers": 100
    }
  ],
  "latency": [
    {
      "type": "batchDistance",
      "batchAttribute": "matchConfigId"
    }
  ],
  "expansions": [
    {
      "target": "teams[match].minPlayers",
      "steps": [
        {
          "waitTimeSeconds": 10,
          "value": 40
        },
        {
          "waitTimeSeconds": 20,
          "value": 60
        },
        {
          "waitTimeSeconds": 30,
          "value": 70
        }
      ]
    }
  ]
}
```



Rule for BattleRoyale



100人対戦のLarge Match

チーム制では無い

レイテンシーでグループ分け

レイテンシーの上限

時間経過で人数を緩和



battleRoyale.json

```
{  
  "name": "RankMatch",  
  "ruleLanguageVersion": "1.0",  
  "playerAttributes": [  
    {  
      "name": "matchConfigId",  
      "type": "string"  
    },  
    {  
      "name": "skill",  
      "type": "number",  
      "default": 0  
    }  
  ],  
  "algorithm": {  
    "strategy": "balanced",  
    "batchingPreference": "fastestRegion",  
    "balancedAttribute": "skill",  
    "expansionAgeSelection": "oldest"  
  },  
  "teams": [  
    {  
      "name": "match",  
      "maxPlayers": 100,  
      "minPlayers": 100  
    }  
  ],  
  "batching": {  
    "name": "batchingIdMatch",  
    "type": "batchDistance",  
    "batchAttribute": "matchConfigId"  
  },  
  "expansions": [  
    {  
      "target": "teams[match].minPlayers",  
      "steps": [{  
        "waitTimeSeconds": 10,  
        "value": 90  
      },  
      {  
        "waitTimeSeconds": 20,  
        "value": 80  
      },  
      {  
        "waitTimeSeconds": 30,  
        "value": 70  
      }  
    }  
  ]  
}
```

Rule for BattleRoyale



100人対戦のLarge Match
チーム制では無い
レイテンシーでグループ分け
レイテンシーの上限
時間経過で人数を緩和



```
battleRoyale.json
{
  "name": "RankMatch",
  "ruleLanguageVersion": "1.0",
  "playerAttributes": [
    {
      "name": "matchConfigId",
      "type": "string"
    },
    {
      "name": "skill",
      "type": "number",
      "default": 0
    }
  ],
  "algorithm": {
    "strategy": "balanced",
    "batchingPreference": "fastestRegion",
    "balancedAttribute": "skill",
    "expansionAgeSelection": "oldest"
  },
  "teams": [
    {
      "name": "match",
      "maxPlayers": 100
    }
  ],
  "rules": [
    {
      "name": "playerLatency",
      "type": "latency",
      "maxLatency": 200
    },
    {
      "value": 90
    },
    {
      "waitTimeSeconds": 30,
      "value": 70
    }
  ]
}
```



Rule for BattleRoyale



100人対戦のLarge Match
チーム制では無い
レイテンシーでグループ分け
レイテンシーの上限
時間経過で人数を緩和



```
battleRoyale.json
{
    "name": "RankMatch",
    "ruleLanguageVersion": "1.0",
    "playerAttributes": [
        {
            "name": "matchConfigId",
            "type": "string"
        },
        {
            "name": "skill",
            "type": "number",
            "default": 0
        }
    ],
    "algorithm": {
        "expansions": [
            {
                "target": "teams[match].minPlayers",
                "steps": [
                    {
                        "waitTimeSeconds": 10,
                        "value": 90
                    },
                    {
                        "waitTimeSeconds": 20,
                        "value": 80
                    },
                    {
                        "waitTimeSeconds": 30,
                        "value": 70
                    }
                ]
            }
        ]
    }
}
```



Rule for BattleRoyale



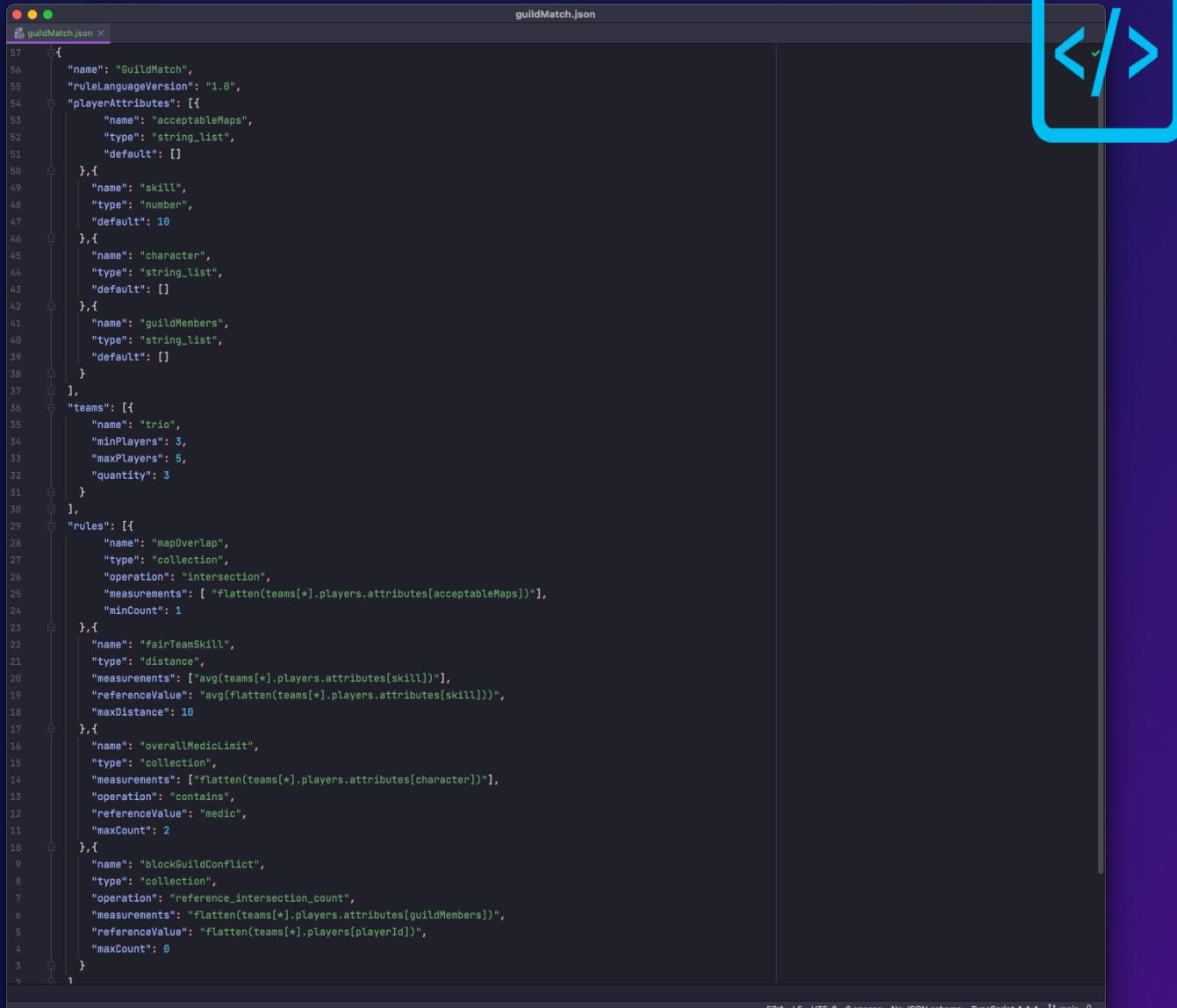
- 100人対戦のLarge Match ✓
- チーム制では無い ✓
- レイテンシーでグループ分け ✓
- レイテンシーの上限 ✓
- 時間経過で人数を緩和 ✓

```
battleRoyale.json
{
  "name": "RankMatch",
  "ruleLanguageVersion": "1.0",
  "playerAttributes": [
    {
      "name": "matchConfigId",
      "type": "string"
    },
    {
      "name": "skill",
      "type": "number",
      "default": 0
    }
  ],
  "algorithm": {
    "strategy": "balanced",
    "batchingPreference": "fastestRegion",
    "balancedAttribute": "skill",
    "expansionAgeSelection": "oldest"
  },
  "teams": [
    {
      "name": "match",
      "maxPlayers": 100,
      "minPlayers": 100
    }
  ],
  "rules": [
    {
      "name": "playerLatency",
      "type": "latency",
      "maxLatency": 200
    },
    {
      "name": "matchmakingIdMatch",
      "type": "batchDistance",
      "batchAttribute": "matchConfigId"
    }
  ],
  "expansions": [
    {
      "target": "teams[match].minPlayers",
      "steps": [
        {
          "waitTimeSeconds": 10,
          "value": 90
        },
        {
          "waitTimeSeconds": 20,
          "value": 80
        },
        {
          "waitTimeSeconds": 30,
          "value": 70
        }
      ]
    }
  ]
}
```



Rule for Trio Team Match

3つのチームによる対戦
同じマップを選んだチーム
チーム間格差の制限
キャラクター被りを防ぐ
同じギルドに所属していない



```
guildMatch.json
{
  "name": "GuildMatch",
  "ruleLanguageVersion": "1.0",
  "playerAttributes": [
    {
      "name": "acceptableMaps",
      "type": "string_list",
      "default": []
    },
    {
      "name": "skill",
      "type": "number",
      "default": 10
    },
    {
      "name": "character",
      "type": "string_list",
      "default": []
    }
  ],
  "teams": [
    {
      "name": "trio",
      "minPlayers": 3,
      "maxPlayers": 5,
      "quantity": 3
    }
  ],
  "rules": [
    {
      "name": "mapOverlap",
      "type": "collection",
      "operation": "intersection",
      "measurements": [ "flatten(teams[*].players.attributes[acceptableMaps])" ],
      "minCount": 1
    },
    {
      "name": "fairTeamSkill",
      "type": "distance",
      "measurements": [ "avg(teams[*].players.attributes[skill])" ],
      "referenceValue": "avg(flatten(teams[*].players.attributes[skill]))",
      "maxDistance": 10
    },
    {
      "name": "overallMedicLimit",
      "type": "collection",
      "measurements": [ "flatten(teams[*].players.attributes[character])" ],
      "operation": "contains",
      "referenceValue": "medic",
      "maxCount": 2
    },
    {
      "name": "blockGuildConflict",
      "type": "collection",
      "operation": "reference_intersection_count",
      "measurements": "flatten(teams[*].players.attributes[guildMembers])",
      "referenceValue": "flatten(teams[*].players[playerId])",
      "maxCount": 0
    }
  ]
}
```





Rule for Trio Team Match

3つのチームによる対戦
同じマップを選んだチーム
チーム間格差の制限
キャラクター被りを防ぐ
同じギルドに所属していない

```
"playerAttributes": [  
    {  
        "name": "acceptableMaps",  
        "type": "string_list",  
        "default": []  
    }, {  
        "name": "skill",  
        "type": "number",  
        "default": 10  
    }, {  
        "name": "character",  
        "type": "string_list",  
        "default": []  
    }, {  
        "name": "guildMembers",  
        "type": "string_list",  
        "default": []  
    }  
,
```

Rule for Trio Team Match

3つのチームによる対戦



同じマップを選んだチーム

チーム間格差の制限

キャラクター被りを防ぐ

同じギルドに所属してない

```
guildMatch.json
{
    "name": "BuildMatch",
    "ruleLanguageVersion": "1.0",
    "playerAttributes": [
        {
            "name": "acceptableMaps"
        }
    ],
    "teams": [
        {
            "name": "trio",
            "minPlayers": 3,
            "maxPlayers": 5,
            "quantity": 3
        }
    ],
    "teamOperations": [
        {
            "name": "Intersection",
            "type": "intersection",
            "measurements": [
                "#flatten(teams[*].players.attributes[acceptableMaps])"
            ],
            "minCount": 1
        },
        {
            "name": "FairTeamSkill",
            "type": "distance",
            "measurements": [
                "avg(teams[*].players.attributes[skill])"
            ],
            "referenceValue": "avg(flatten(teams[*].players.attributes[skill]))",
            "maxDistance": 10
        },
        {
            "name": "OverallMedicLimit",
            "type": "collection",
            "measurements": [
                "#flatten(teams[*].players.attributes[character])"
            ],
            "operation": "contains",
            "referenceValue": "medic",
            "maxCount": 2
        },
        {
            "name": "BlockBuildConflict",
            "type": "collection",
            "operation": "reference_intersection_count",
            "measurements": [
                "#flatten(teams[*].players.attributes[guildMembers])"
            ],
            "referenceValue": "#flatten(teams[*].players[playerId])",
            "maxCount": 0
        }
    ]
}
```

Rule for Trio Team Match



3つのチームによる対戦
同じマップを選んだチーム →
チーム間格差の制限
キャラクター被りを防ぐ
同じギルドに所属していない

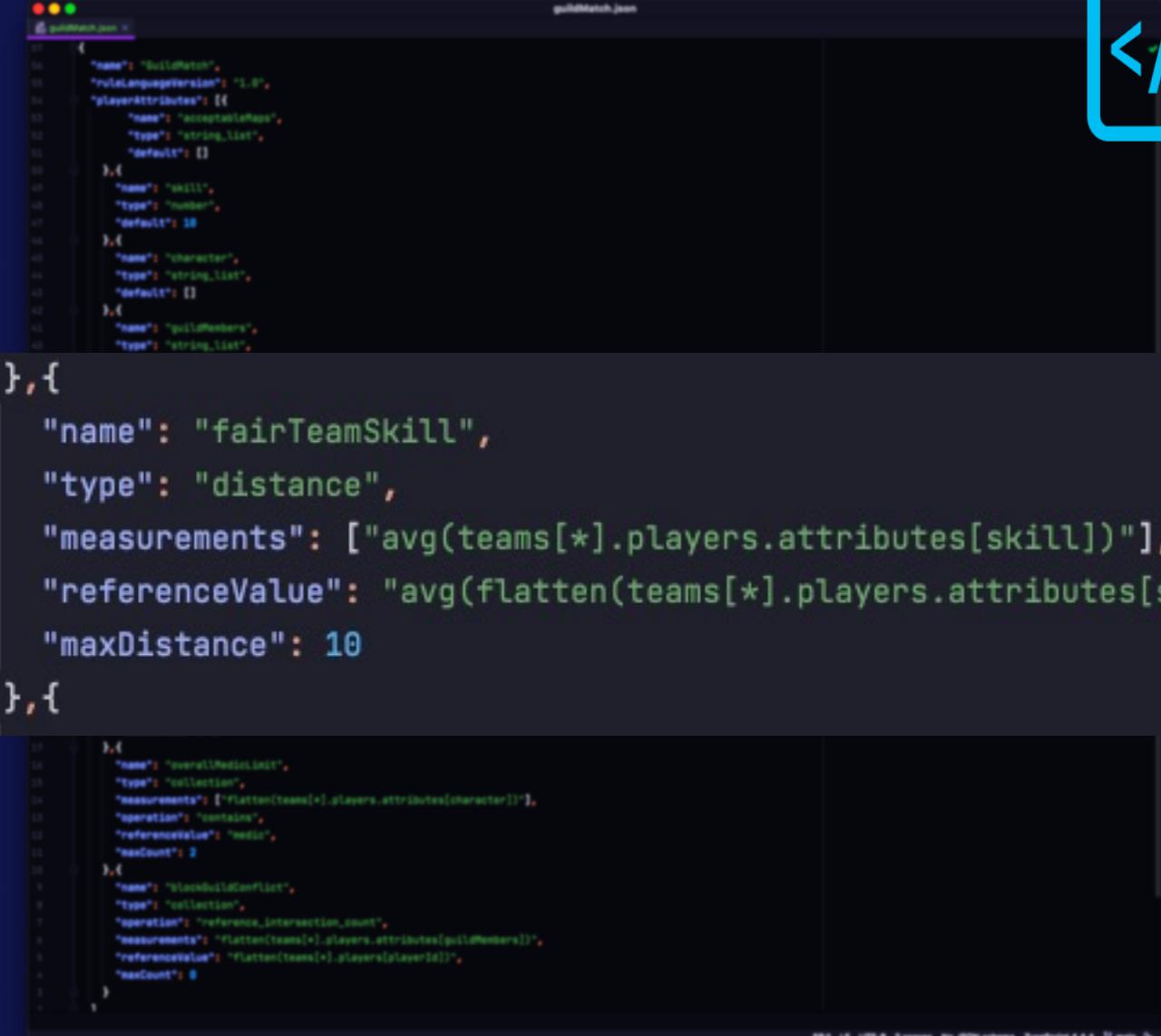
```
guildMatch.json
{
    "name": "BuildMatch",
    "ruleLanguageVersion": "1.0",
    "playerAttributes": [
        {
            "name": "acceptableMaps",
            "type": "string_list",
            "default": []
        },
        {
            "name": "skill",
            "type": "Number",
            "default": 10
        },
        {
            "name": "character",
            "type": "string_list",
            "default": []
        }
    ],
    "rules": [
        {
            "name": "mapOverlap",
            "type": "collection",
            "operation": "intersection",
            "measurements": ["flatten(teams[*].players.attributes[acceptableMaps])"],
            "minCount": 1
        },
        {
            "name": "fairTeamSkill",
            "type": "collection",
            "measurements": ["avg(teams[*].players.attributes[skill])"],
            "referenceValue": "avg(flatten(teams[*].players.attributes[skill]))",
            "maxDistance": 10
        },
        {
            "name": "overallMedicLimit",
            "type": "collection",
            "measurements": ["flatten(teams[*].players.attributes[character])"],
            "operation": "contains",
            "referenceValue": "medic",
            "maxCount": 2
        },
        {
            "name": "blockBuildConflict",
            "type": "collection",
            "operation": "reference_intersection_count",
            "measurements": ["flatten(teams[*].players.attributes[guildMembers])"],
            "referenceValue": "flatten(teams[*].players[playerId])",
            "maxCount": 0
        }
    ]
}
```



Rule for Trio Team Match



3つのチームによる対戦
同じマップを選んだチーム
チーム間格差の制限 →
キャラクター被りを防ぐ
同じギルドに所属していない



```
guildMatch.json
{
    "name": "BuildMatch",
    "ruleLanguageVersion": "1.0",
    "playerAttributes": [
        {
            "name": "acceptableHeads",
            "type": "string_list",
            "default": []
        },
        {
            "name": "skillList",
            "type": "Number",
            "default": 10
        },
        {
            "name": "character",
            "type": "string_list",
            "default": []
        },
        {
            "name": "guildMembers",
            "type": "string_list"
        }
    ],
    "teamRules": [
        {
            "name": "fairTeamSkill",
            "type": "distance",
            "measurements": ["avg(teams[*].players.attributes[skill])"],
            "referenceValue": "avg(flatten(teams[*].players.attributes[skill]))",
            "maxDistance": 10
        },
        {
            "name": "overallMedicLimit",
            "type": "collection",
            "measurements": ["flatten(teams[*].players.attributes[character])"],
            "operation": "contains",
            "referenceValue": "medic",
            "maxCount": 2
        },
        {
            "name": "blockBuildConflict",
            "type": "collection",
            "operation": "reference_intersection_count",
            "measurements": ["flatten(teams[*].players.attributes[guildMembers])"],
            "referenceValue": "flatten(teams[*].players[playerId])",
            "maxCount": 0
        }
    ]
}
```



Rule for Trio Team Match



3つのチームによる対戦

同じマップを選んだチーム

チーム間格差の制限

キャラクター被りを防ぐ

同じギルドに所属していない



```
guildMatch.json
```

```
{
  "name": "BuildMatch",
  "ruleLanguageVersion": "1.0",
  "playerAttributes": [
    {
      "name": "acceptableMaps",
      "type": "string_list",
      "default": []
    },
    {
      "name": "kills",
      "type": "Number",
      "default": 10
    },
    {
      "name": "character",
      "type": "string_list",
      "default": []
    },
    {
      "name": "guildMembers",
      "type": "string_list",
      "default": []
    }
  ],
  "teams": [
    {
      "name": "overallMedicLimit",
      "type": "collection",
      "operation": "contains",
      "measurements": ["flatten(teams[*].players.attributes[character])"],
      "referenceValue": "medic",
      "maxCount": 2
    },
    {
      "name": "blockBuildConflict",
      "type": "collection",
      "operation": "reference_intersection_count",
      "measurements": ["flatten(teams[*].players.attributes[guildMembers])"],
      "referenceValue": "flatten(teams[*].players[playerId])",
      "maxCount": 0
    }
  ]
}
```



Rule for Trio Team Match



3つのチームによる対戦
同じマップを選んだチーム
チーム間格差の制限
キャラクター被りを防ぐ
同じギルドに所属していない



```
guildMatch.json
```

```
guildMatch.json
```

```
}, {  
    "name": "overallMedicLimit",  
    "type": "collection",  
    "operation": "contains",  
    "measurements": ["flatten(teams[*].players.attributes[character])"],  
    "referenceValue": "medic",  
    "maxCount": 2  
}, {  
    "name": "overallAssaultRequirement",  
    "type": "collection",  
    "operation": "contains",  
    "measurements": ["flatten(teams[*].players.attributes[character])"],  
    "referenceValue": "assault",  
    "minCount": 4  
}, {  
    "name": "teamBalance",  
    "type": "compound",  
    "statement": "or(overallMedicLimit, overallAssaultRequirement)"  
}, {
```

再掲部分

Rule for Trio Team Match

3つのチームによる対戦
同じマップを選んだチーム
チーム間格差の制限
キャラクター被りを防ぐ
同じギルドに所属していない



```
guildMatch.json
{
  "name": "TrioTeamMatch",
  "ruleLanguageVersion": "1.0",
  "playerAttributes": [
    {
      "name": "acceptableMaps",
      "type": "string_list",
      "default": []
    },
    {
      "name": "skills",
      "type": "Number",
      "default": 10
    },
    {
      "name": "character",
      "type": "string_list",
      "default": []
    },
    {
      "name": "guildMembers",
      "type": "string_list",
      "default": []
    }
  ],
  "teams": [
    {
      "name": "trio",
      "minPlayers": 3,
      "maxPlayers": 3,
      "quantity": 3
    }
  ],
  "rules": [
    {
      "name": "blockGuildConflict",
      "type": "collection",
      "operation": "reference_intersection_count",
      "measurements": ["flatten(teams[*].players.attributes[guildMembers])"],
      "referenceValue": "flatten(teams[*].players[playerId])",
      "maxCount": 0
    }
  ]
}
```

Rule for Trio Team Match



- 3つのチームによる対戦 ✓
- 同じマップを選んだチーム ✓
- チーム間格差の制限 ✓
- キャラクター被りを防ぐ ✓
- 同じギルドに所属していない ✓

```
guildMatch.json
{
  "name": "GuildMatch",
  "ruleLanguageVersion": "1.0",
  "playerAttributes": [
    {
      "name": "acceptableMaps",
      "type": "string_list",
      "default": []
    },
    {
      "name": "skill",
      "type": "number",
      "default": 10
    },
    {
      "name": "character",
      "type": "string_list",
      "default": []
    },
    {
      "name": "guildMembers",
      "type": "string_list",
      "default": []
    }
  ],
  "teams": [
    {
      "name": "trio",
      "minPlayers": 3,
      "maxPlayers": 5,
      "quantity": 3
    }
  ],
  "rules": [
    {
      "name": "mapOverlap",
      "type": "collection",
      "operation": "intersection",
      "measurements": [ "flatten(teams[*].players.attributes[acceptableMaps])" ],
      "minCount": 1
    },
    {
      "name": "fairTeamSkill",
      "type": "distance",
      "measurements": [ "avg(teams[*].players.attributes[skill])" ],
      "referenceValue": "avg(flatten(teams[*].players.attributes[skill]))",
      "maxDistance": 10
    },
    {
      "name": "overallMedicLimit",
      "type": "collection",
      "measurements": [ "flatten(teams[*].players.attributes[character])" ],
      "operation": "contains",
      "referenceValue": "medic",
      "maxCount": 2
    },
    {
      "name": "blockGuildConflict",
      "type": "collection",
      "operation": "reference_intersection_count",
      "measurements": "flatten(teams[*].players.attributes[guildMembers])",
      "referenceValue": "flatten(teams[*].players[playerId])",
      "maxCount": 0
    }
  ]
}
```

And More...

comparison, <=!> operation, distanceSort, absoluteSort, compound, exhaustiveSearch, backfill, expansionAgeSelection, and more...

ゲームに合った組み合わせのルールを作り、より早く、より公平なマッチメイキングを生み出し新たな楽しいゲームを創れます

https://docs.aws.amazon.com/ja_jp/gamelift/latest/flexmatchguide/match-ruleset-schema.html

What customers are saying?

ガンホー様: Ninjala

“サーバーサイドやマッチングシステム部分は AWS にお任せできるようになりました。そのため、私たちはお客様により優れたゲーム体験を提供することに注力できるようになっています”

CTO
菊池 貴則

https://aws.amazon.com/jp/solutions/case-studies/gunho_2022/

AWS 導入事例: ガンホー・オンライン・エンターテイメント株式会社 導入事例 アーキテクチャ ご利用中の主なサービス PDFダウンロード 国内導入事例TOP



世界中で遊べるマルチプレイゲームのインフラに Amazon GameLift を採用
高度なマッチメイキングにより、優れたゲーム体験を創出

2022

スマートフォン向けパズル RPG『パズル&ドラゴンズ（以下「パズドラ」）』などでおなじみのガンホー・オンライン・エンターテイメント株式会社（以下「ガンホー」）。同社は、パズドラのリリース当初からゲームのインフラとしてAmazon ウェブ サービス（AWS）を利用しています。2020年6月にリリースしたオンライン対戦アクションゲーム『Ninjala（ニンジャラ）』では、マルチプレイに対応したゲームサーバー開発向けフルマネージドサービスである Amazon GameLift を採用。世界中のプレイヤーが対象となる本タイトルの高度なマッチメイキング要件を満たしながら、導入・運用コストを削減しています。



“

サーバーサイドやマッチングシステム部分は AWS にお任せできるようになりました。そのため、私たちはお客様により優れたゲーム体験を提供することに注力できるようになっています。自分たちでサーバーを用意する方法と比べると、構築コストは半分ほどで済んでいます。また、運用コストについても毎月 4 人月は削減できています

菊池 貴則 氏
ガンホー・オンライン・エンターテイメント株式会社
CTO



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Takahiro Ishii

 @takahiroishii_



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.