



【JPPGB】 ゲーム作成コンテスト #2

湧夢(ユーム)

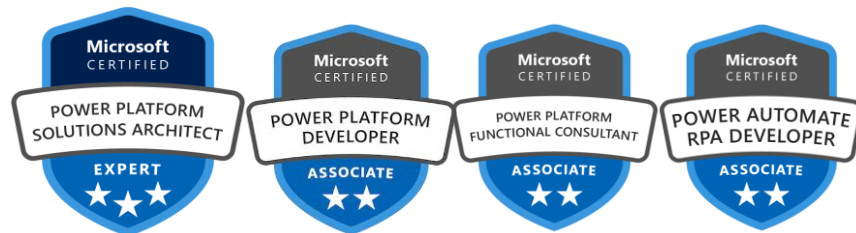
X: @yumuers

自己紹介



AKKODiSコンサルティング株式会社 湧夢(ユーム) X : @yumuers

- 2013 ● 大学生時代にスカウト
ファッションモデルとして活動開始
大学卒業後就職せず、ファッションモデルとして働く
- 2017 ● 転職 ハードウェアエンジニアへ
自動車部品BMS(バッテリーマネジメントシステム)設計開発
- 2022 ● コロナ中でDXが進み、Power Platformに出会う
- 2023 ● Power Platformを用いて業務改善の経験から教育部門へ人事異動
- 2024～ ● Power Platform Technical Evangelist



資格 : PL-600、PL-400、PL-200
PL-500所持



作成したゲーム紹介

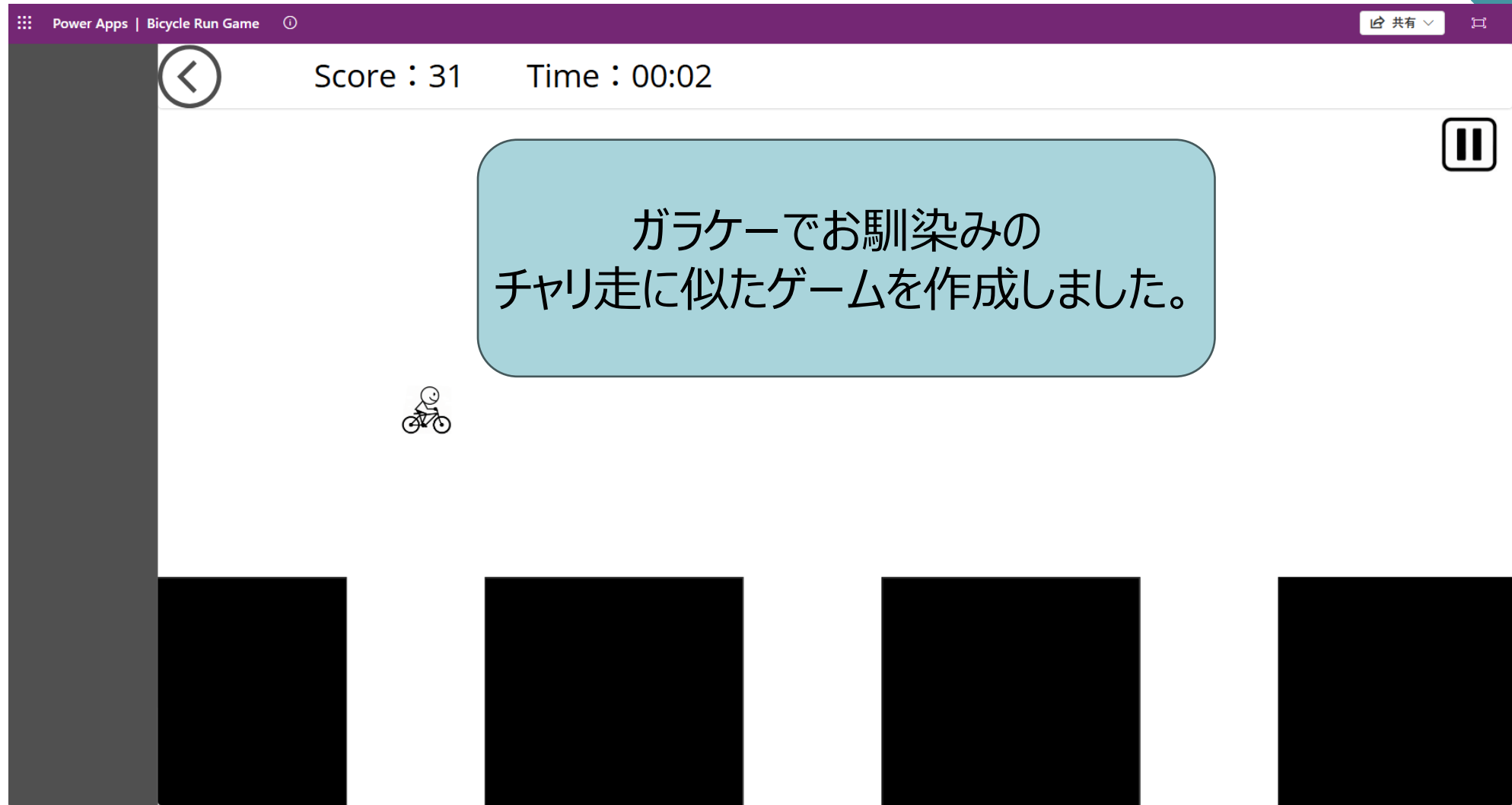
懐かしいゲーム

皆さん
ガラケーを所持していた時代に
どんなゲームが流行っていましたか？？



ガラケーゲームを再現

懐かしいガラケーで流行っていたゲームをPower Appsで再現してみました。

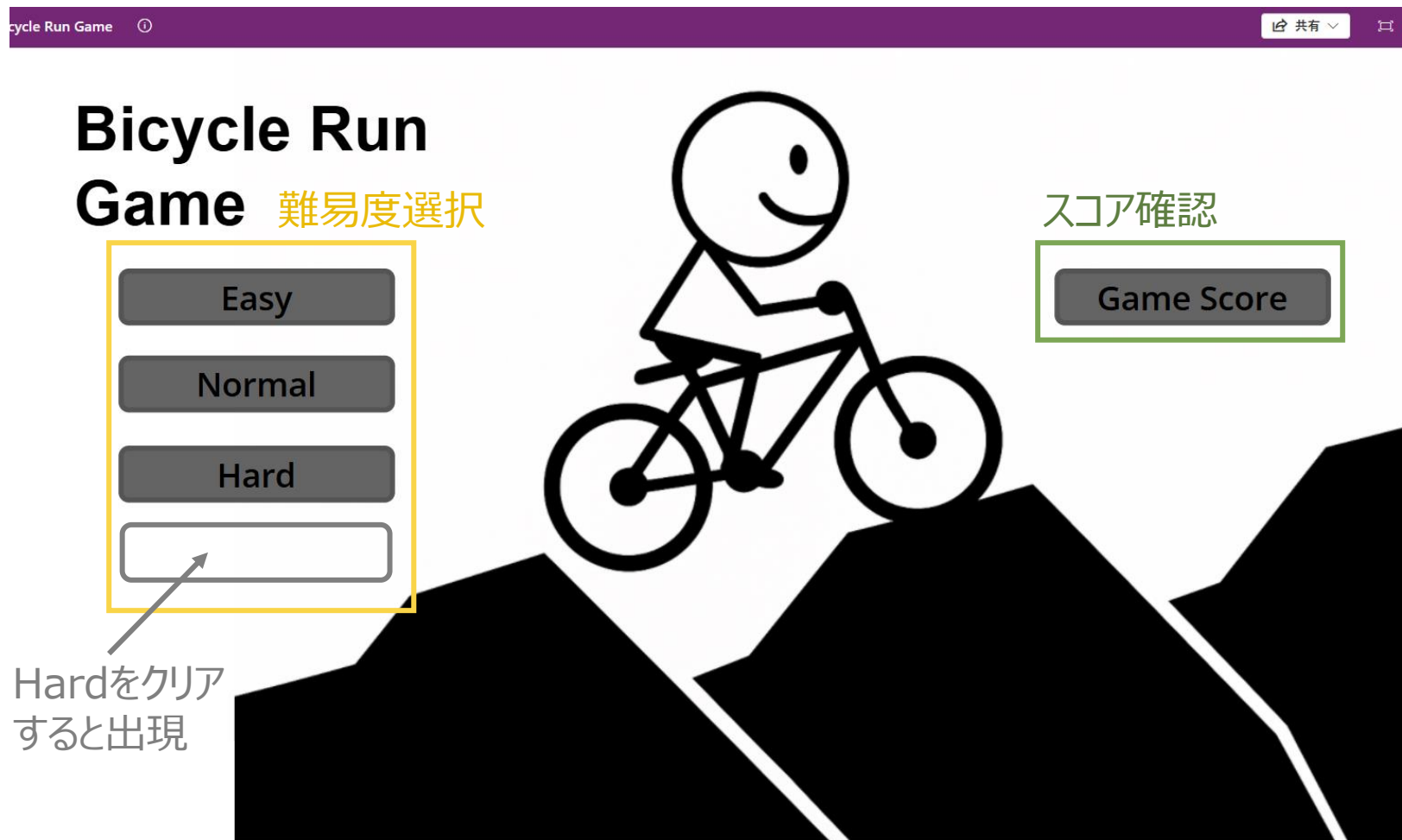




画面説明

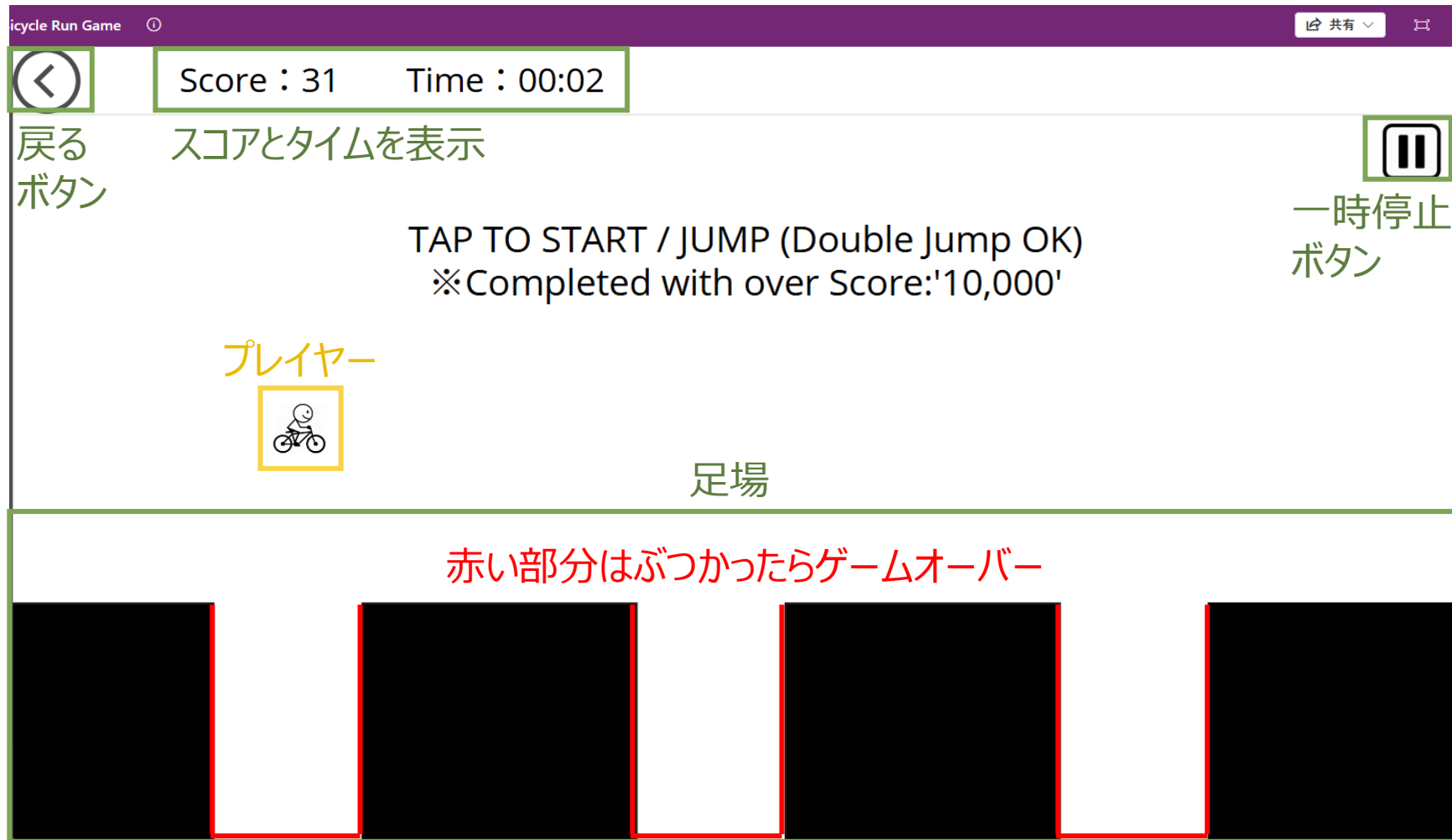
画面説明 Home画面

Home画面では難易度4種類（1種類はHardクリア後出現）選択とスコアを確認



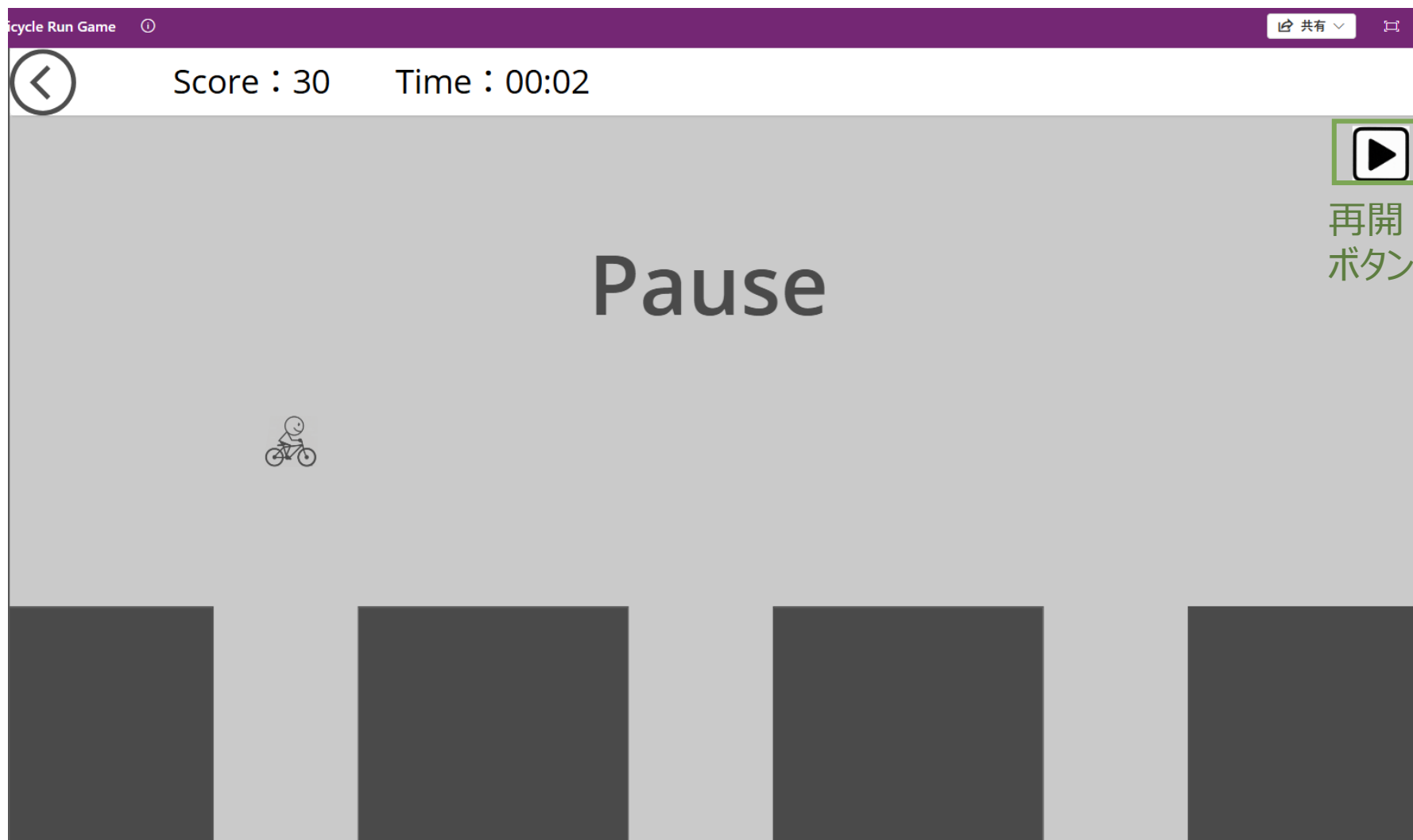
画面説明_ゲームプレイ画面

画面をタップするとスタート -> 穴に落ちない、足場の側面にぶつからないようにジャンプ
(ジャンプは2回まで) -> スコアを伸ばしていきます ※スコアが10,000いくとゲームクリア



画面説明_一時停止画面

途中で停止しても問題ないように一時停止と再開ボタンでストップ/スタートすることが可能



画面説明_スコア画面

スコア画面には3種類COMPLETE、GAME OVER、GAME SCOREを用意しています
スコアを確認することができます

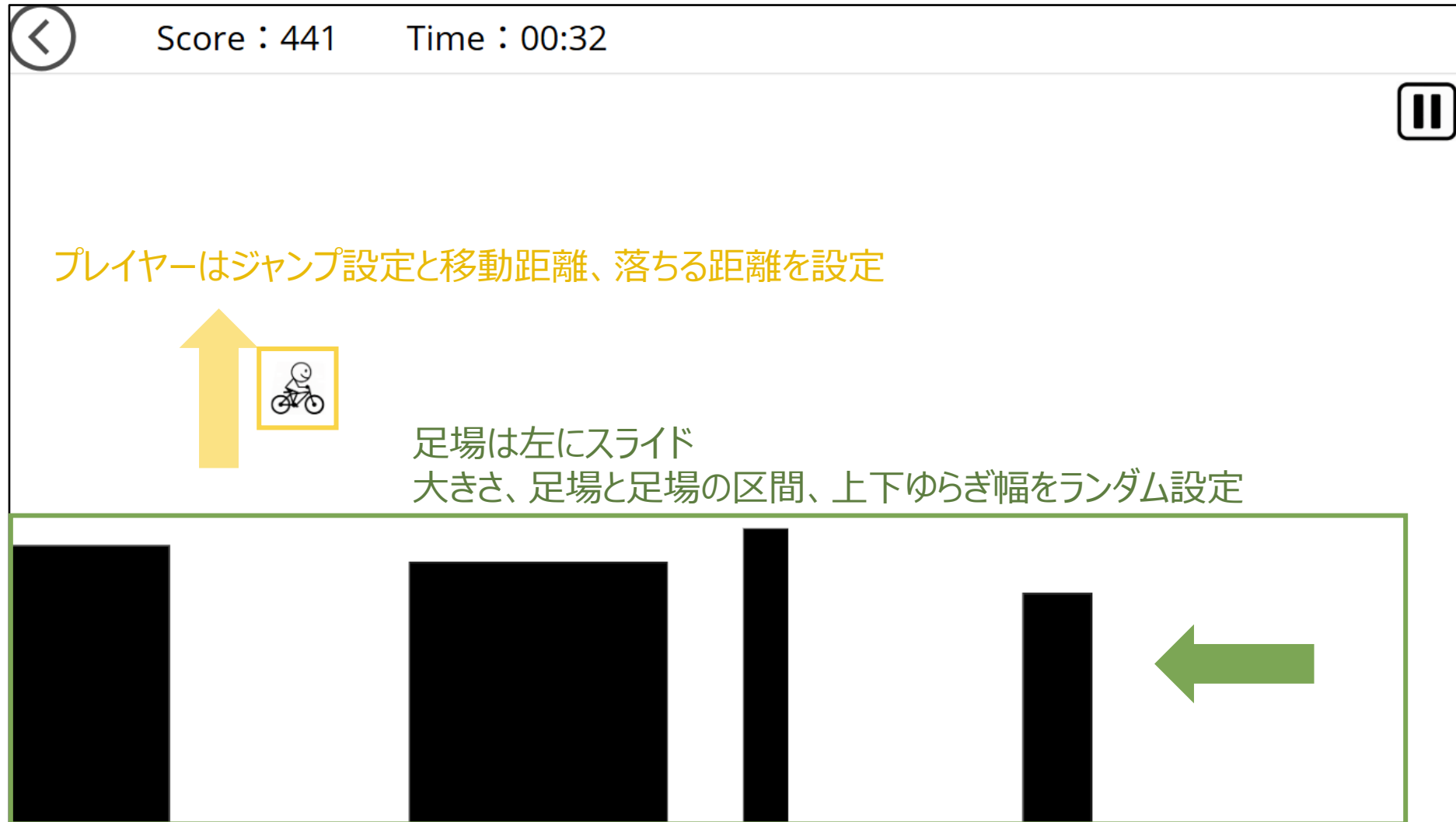




機能紹介

プレイヤーと足場設定

各設定内容



プレイヤー設定詳細

プレイヤーはジャンプ2回まで、ジャンプによる移動距離、落ちる距離を設定しました。また、足場の上面に着地設定とそれ以外に落ちた、ぶつかったなどはゲームオーバーにする設定を加えています。

```
// 物理・プレイヤー
Set(deathLine, viewH);
Set(gravity, 1.2);
Set(jumpPower, 14);

Set(playerW, 50); Set(playerH, 50);
Set(playerX, Round(viewW * 0.18, 0));
Set(playerY, Round(viewH * 0.70, 0) - playerH);
Set(vy, 0); Set(jumpCount, 0);
```

```
If(
    running,
    If(jumpCount < 2,
        Set(vy, -jumpPower);
        Set(jumpCount, jumpCount + 1)
    ),
    // 開始 (初回は justSpawned で吸着させる)
    Set(running, true);
    Set(justSpawned, true)
);
```

```
// ---- 重力・移動 ----
Set(vy, vy + gravity);
Set(prevY, playerY);
Set(playerY, playerY + vy);
Set(footY, playerY + playerH);
Set(tol, Max(6, Abs(vy)));

// ---- 着地 (トンネル対策: 前フレーム→今フレームで“上面を横切ったか”) ----
With(
{
    crossHit:
        Filter(
            Platforms As p,
            // 横方向が重なっている
            playerX + playerW > p.x && playerX < p.x + p.w &&
            // 下向きに移動している
            vy > 0 &&
            // 前フレームの足裏は上面の上、今フレームの足裏は上面の下
            prevY + playerH <= p.top &&
            playerY + playerH >= p.top
        )
    },
    If(
        CountRows(crossHit) > 0,
        With({ landY: Min(crossHit, top) }, // 一番高い上面に着地
            Set(playerY, landY - playerH);
            Set(vy, 0);
            Set(jumpCount, 0)
        )
    )
);
```

足場設定詳細

足場は難易度によって変更しています。足場の最小幅と最大幅及び足場と足場の最小間隔と最大間隔、上下ゆらぎ幅をランダム設定しています。

```
// ランダム設定の範囲
Set(minGap, 30); // 足場と足場の最小間隔
Set(maxGap, 250); // 最大間隔（広くすると難しく）
Set(minW, 100); // 足場の最小幅
Set(maxW, 320); // 最大幅
Set(topJitter, 0); // 上下ゆらぎ幅（0なら一定高さ、例えば40にすると±40px）
```

```
// ---- 足場スクロール（再出現時に幅/間隔/高さをランダム化） ----
Set(curMaxRight, Max(Platforms, x + w)); // 現在いちばん右端

UpdateIf(
    Platforms As p, true,
    {
        // 画面外に出たら“右端+ランダム間隔”へ移動
        x: If(p.x + p.w <= 0,
            curMaxRight + RandBetween(minGap, maxGap),
            p.x - p.speed
        ),

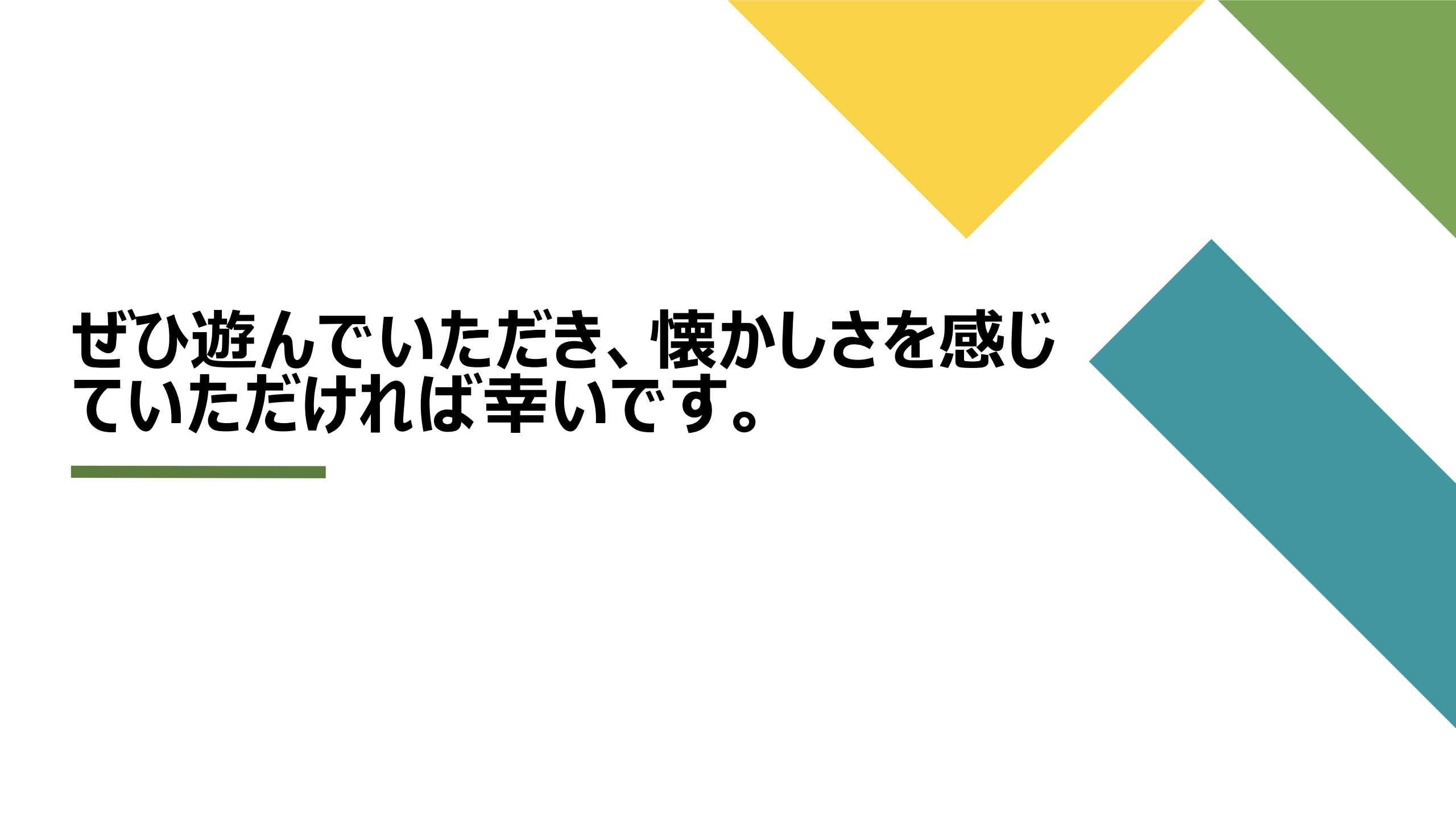
        // 幅もランダムで付け替え
        w: If(p.x + p.w <= 0, RandBetween(minW, maxW), p.w),

        // 上下ゆらぎ（不要なら topJitter=0 のままでOK）
        top: If(p.x + p.w <= 0,
            platformTop + RandBetween(-topJitter, topJitter),
            p.top
        )
    }
);
```

最後に

今回初めてゲームを作成しました。とてもシンプルな内容ですが、実際に作ってみると想像以上に大変で、思い通りに動かすまでに多くの試行錯誤がありました。それでも完成したときの達成感は大きく、ものづくりの面白さを改めて感じました。





**ぜひ遊んでいただき、懐かしさを感じ
いただければ幸いです。**
