

研修発表

Unityを用いて

「月着陸船ゲーム」を作ったことについて

三浦 裕貴

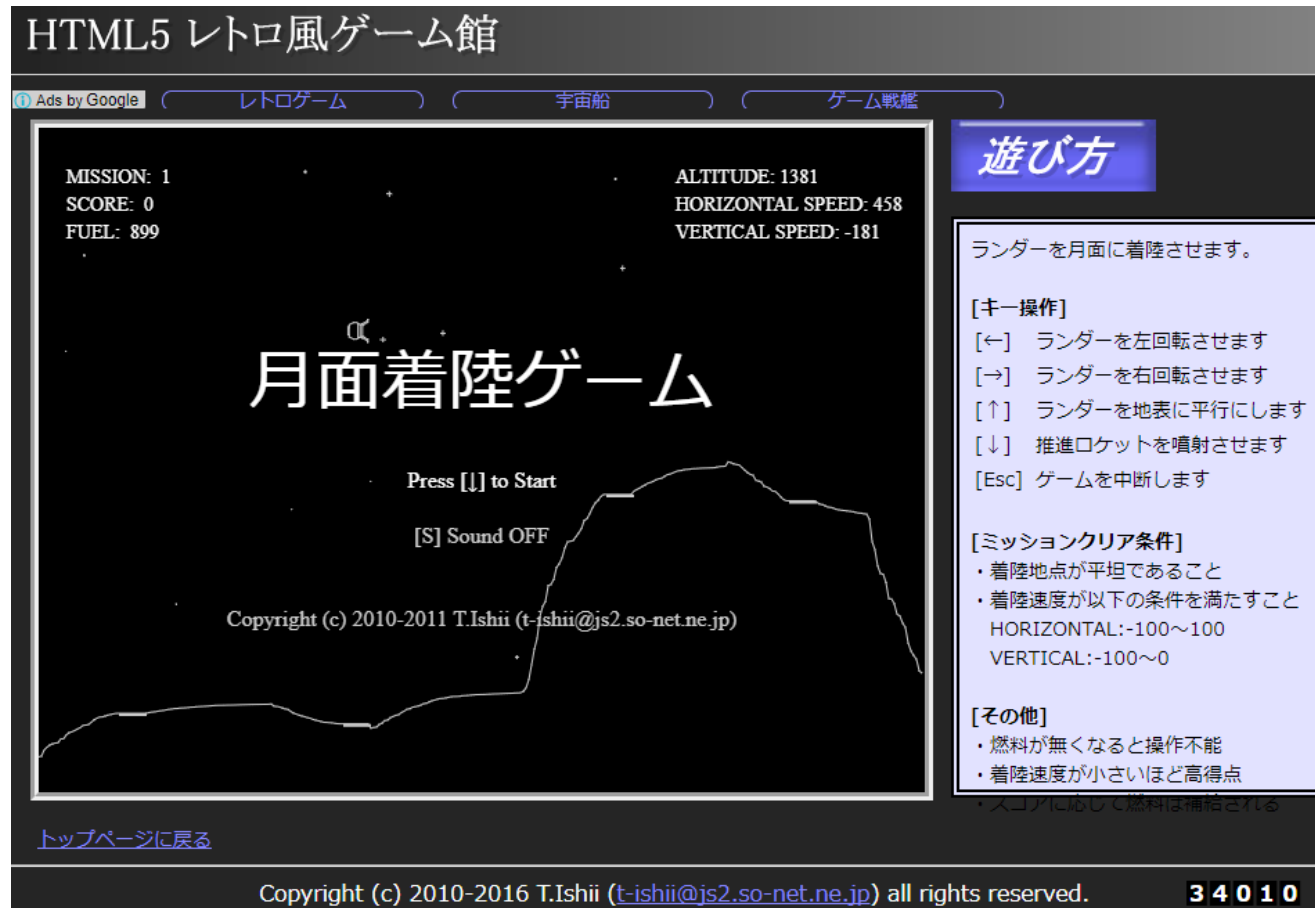
目次

1. どんなゲームを作るか。
2. 作っていく
3. 時間をかけたところ
4. 感想と反省

1. どんなゲームを作るか。

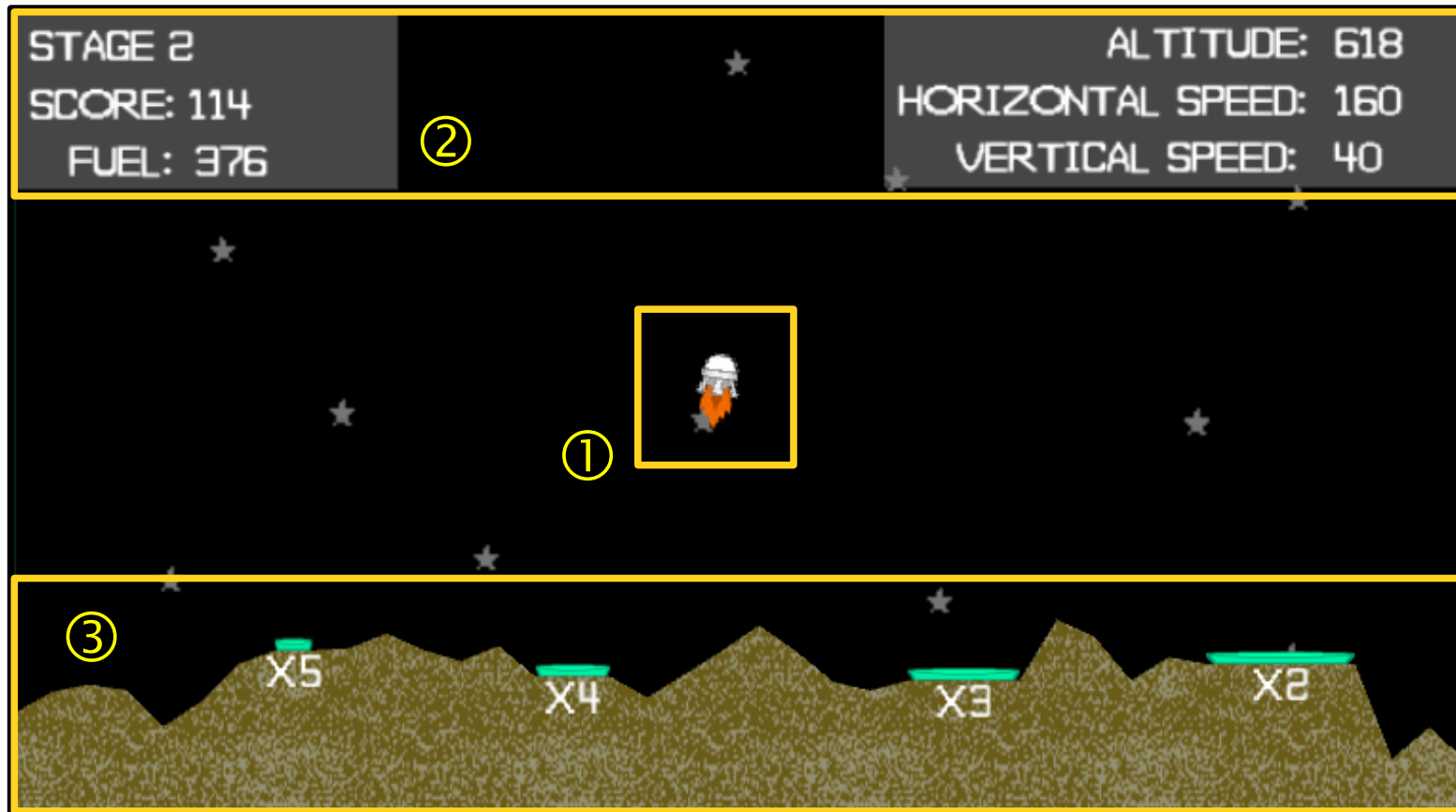
○参考にした月着陸船ゲーム

<http://t-ishii.la.coocan.jp/hp/HTML5/landing/index.html>



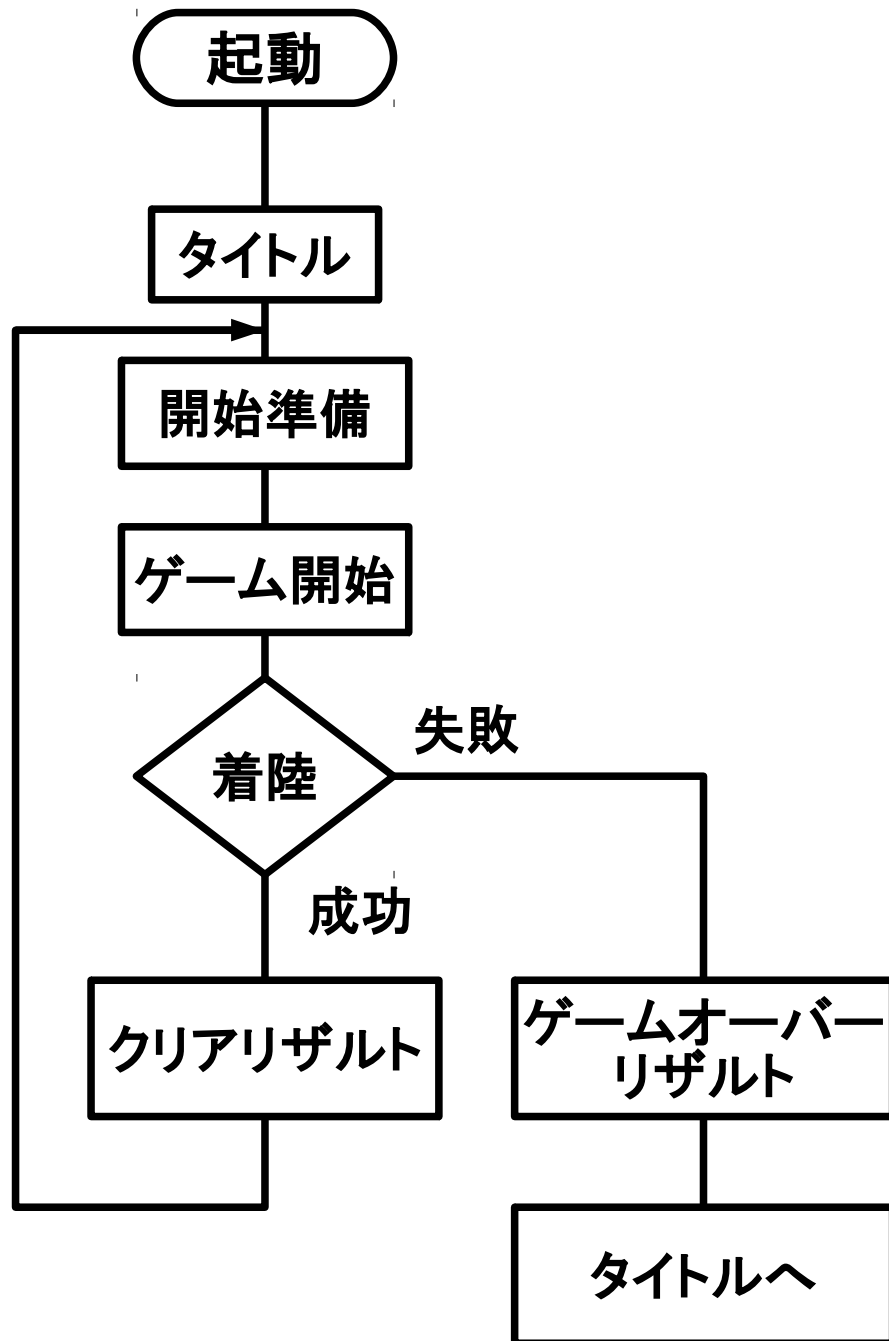
ランダーの制御が難しく、まったく着陸できなかった。
月着陸船ゲームがどのようなものか理解できた。

○画面イメージ



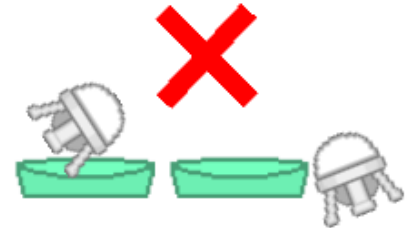
- ① ランダー : プレイヤー操作オブジェクト
- ② 月面ステージ : 4つの平坦な着陸ポイントがある。
- ③ 情報UI : ステージ番号/総合スコア/残り燃料
高度/水平の速度/垂直の速度

○ゲームの流れ



【着陸成功条件】

- ・着陸した場所が着陸地点である。
- ・ランダーが水平状態である。回転値-1~1
- ・ランダーの速度が低速である。速度-100~100



【着陸成功したら】

- ・ステージクリアとなる。
- ・ランダーの速度と着陸地点の倍率によって
スコアとボーナス燃料が得られる。
- ・残りの燃料とボーナス燃料を合わせた燃料で、
次のステージへ進む。

【着陸失敗したら】

- ・クリアステージ数と総合スコアを表示する。
- ・ゲームオーバーとなり、タイトルシーンへ戻る

○必要なオブジェクト・コンポーネント

タイトル



- ・カメラ
 - ・ゲームタイトル
 - ・メッセージ
(点滅処理)
 - ・漂うランダー
(動き)
 - ・背景
- } **uGUI**

- ・状態管理オブジェクト
(入力処理)

メインゲーム



- ・カメラ
(動き)
 - ・ランダー
(ステータス管理、入力処理
(**Collider2D**、**Rigidbody2D**))
 - ・ステージ
(着陸地点、自動生成、**Collider2D**)
 - ・情報UI
(情報更新処理、メッセージ管理)
 - ・背景
 - ・進行管理オブジェクト
(オブジェクト初期化、ゲーム開始終了判定)
- } **uGUI**

2. 作っていく

○Gitを利用しプロジェクトを管理

developブランチから

作業ごとにfeatureブランチを作成していくこと。



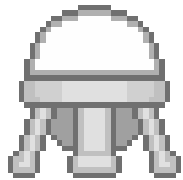
他の作業への影響がない。試験的な処理を作っても安心

○作業の手順

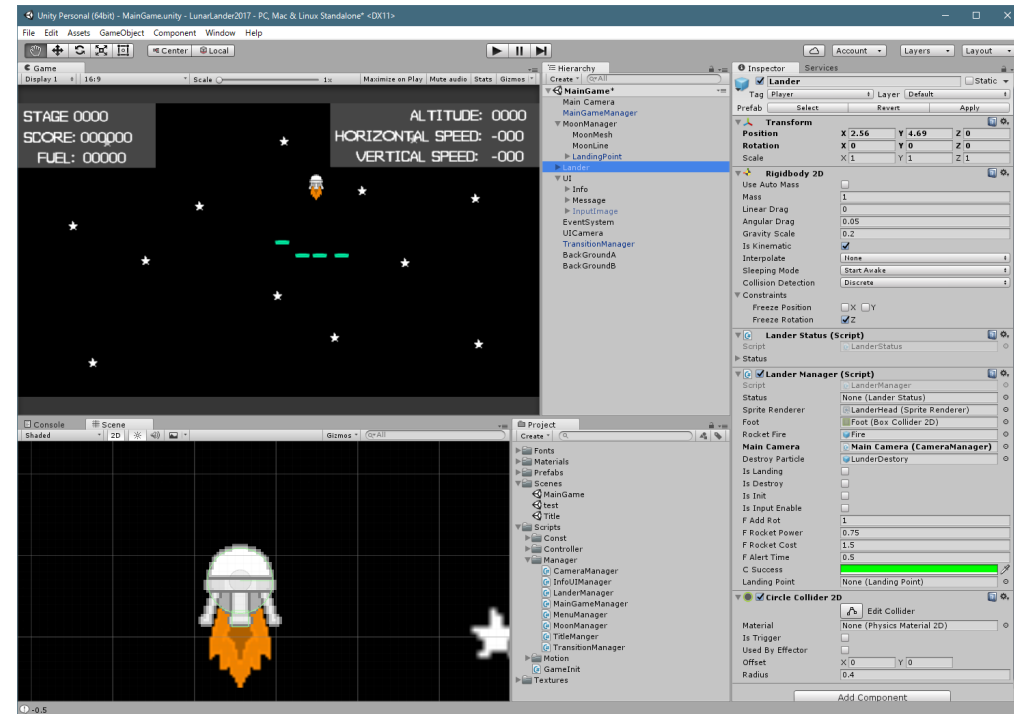
1. シーン管理オブジェクトを作成
 2. タイトルシーンを作成
 3. メインゲームシーンを作成
 4. 月面ステージ(仮)を作成(自動生成以外)
 5. ランダーを作成
 6. カメラの動きを作成
 7. メインゲームのUI・ルール・開始終了処理を作成
 8. 月面ステージの自動生成処理を作成
 9. タイトルシーンの装飾(ランダーや背景)
 10. 各種画像とフォントを作成
 11. 全体のデバッグと調整
- } シーンの遷移

○Unityの利用

- ・ 視覚的に操作してオブジェクトを配置
- ・ Collider2DやRigidbody2Dなどのコンポーネント



Collider2D
Rigidbody2D
SpriteRenderer
LanderManager(スクリプト)
LanderStatus(スクリプト)





【オブジェクトの作成】

- ・ 仕様イメージからシーンに必要なオブジェクトを配置する。
- ・ 各オブジェクトに必要なコンポーネントを付けていく。
- ・ 動きや処理をスクリプトで作っていく。

Unityを利用することで工程を省略できる。
画像の差し替えなどレイアウトの変更が容易。

3. 時間をかけたところ

○作業の手順

1. シーン管理オブジェクトを作成
2. タイトルシーンを作成
3. メインゲームシーンを作成
4. 月面ステージ(仮)を作成(自動生成以外)
5. ランダーを作成 
6. カメラの動きを作成
7. メインゲームのUI・ルール・開始終了処理を作成
8. 月面ステージの自動生成処理を作成 
9. タイトルシーンの装飾(ランダーや背景)
10. 各種画像とフォントを作成
11. 全体のデバッグと調整

○ランダー操作感の調整

【操作】

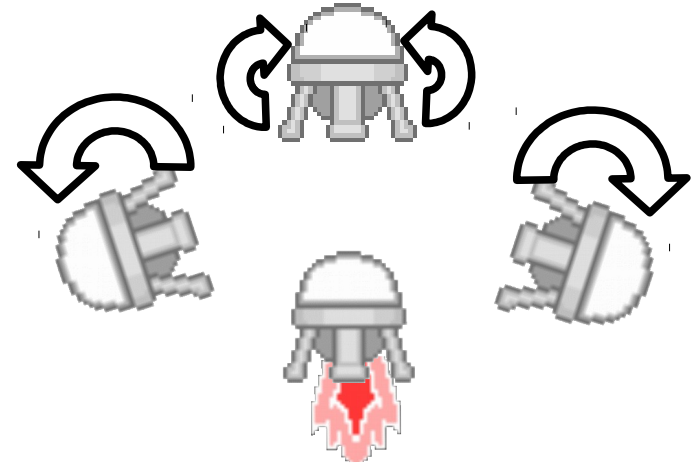
キーボード: ↑ ↓ ← →

↑ :ランダーの回転をリセットする。

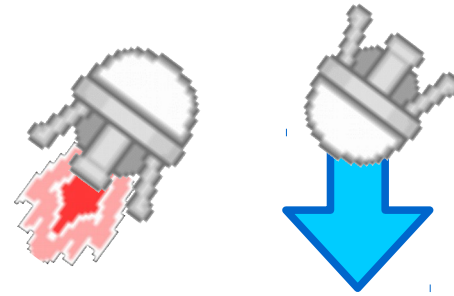
↓ :推進ロケットを噴射させる。

← :ランダーを左回転させる。

→ :ランダーを右回転させる。



- ロケットのパワー
- 重力影響度



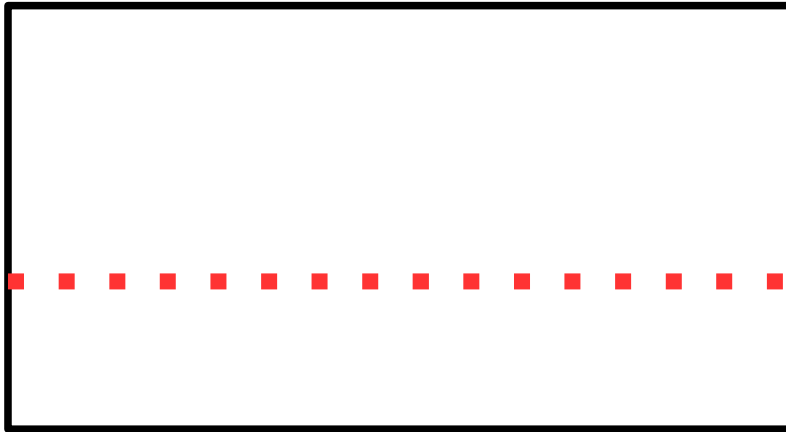
ロケットのパワーが強すぎると
飛びすぎてランダーの位置調整がしづらい。
重力影響度が強すぎると
ロケットを使っても上昇することができない。

程よい数値を見つけ出す。
体で覚えるレベルの操作感にする。

○ステージ自動生成

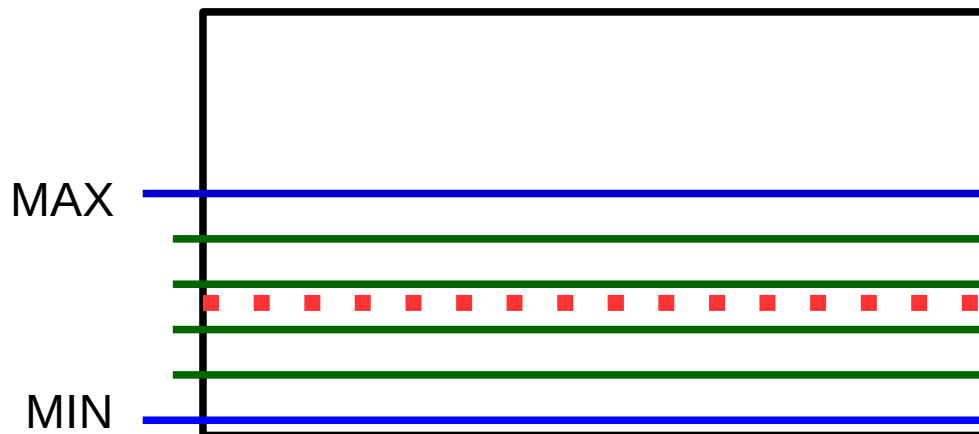
【ステージ生成手順】

1. 頂点を等間隔に配置する。



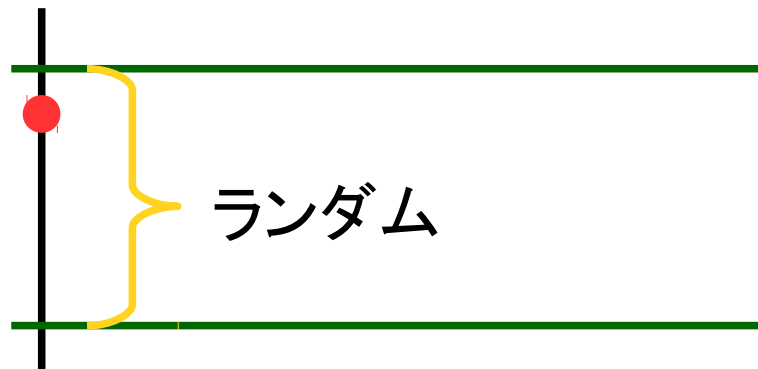
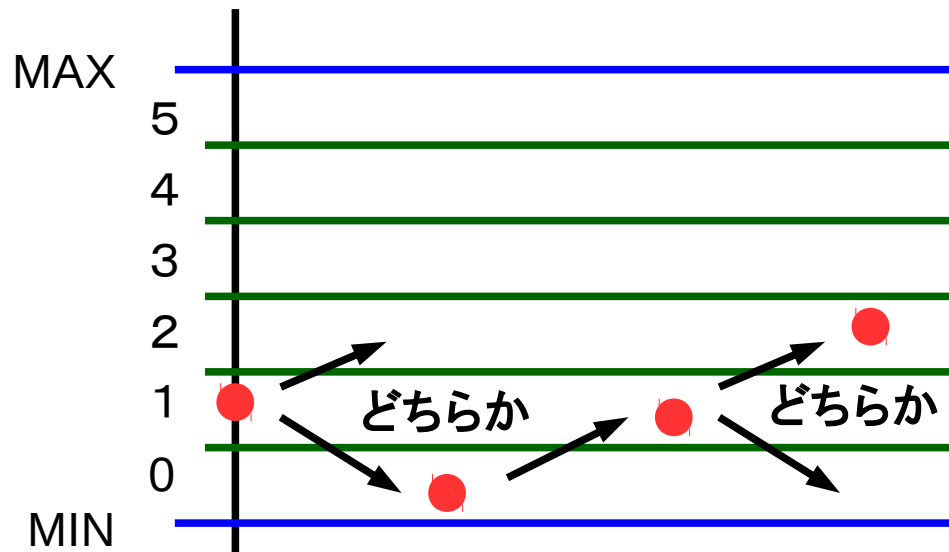
頂点数は設定する。
この時点で**頂点のX座標**が決まる。

2. Y軸の範囲を分割する。



Y軸の範囲の**MAX値**と**MIN値**と
分割数は設定する。

3. 頂点の左から順に設定するY軸の範囲を決め、
範囲内でランダムにY座標設定する。



【範囲を決めるルール】

- ・1つ前の頂点の範囲より
1つ上 or **1つ下** の範囲
- ・1つ前の頂点の範囲が
最も上 の範囲なら **1つ下** の範囲
最も下 の範囲なら **1つ上** の範囲
- ・1つ前の頂点が存在しないなら
ランダムに範囲を決める。

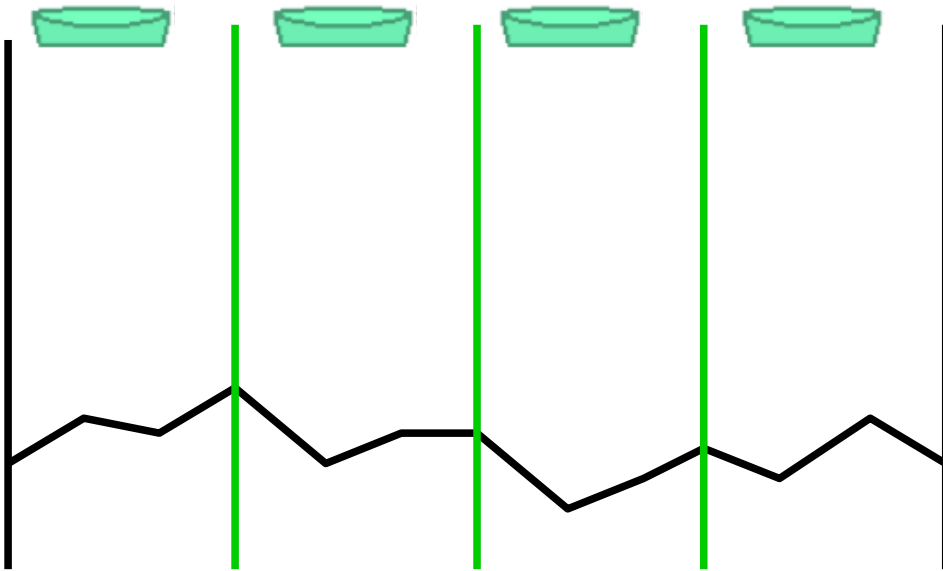
範囲が決まったら範囲内でランダムにY座標を設定する。

範囲の分割数を多く
ずらしていくことで急激なY座標の変化を抑える。

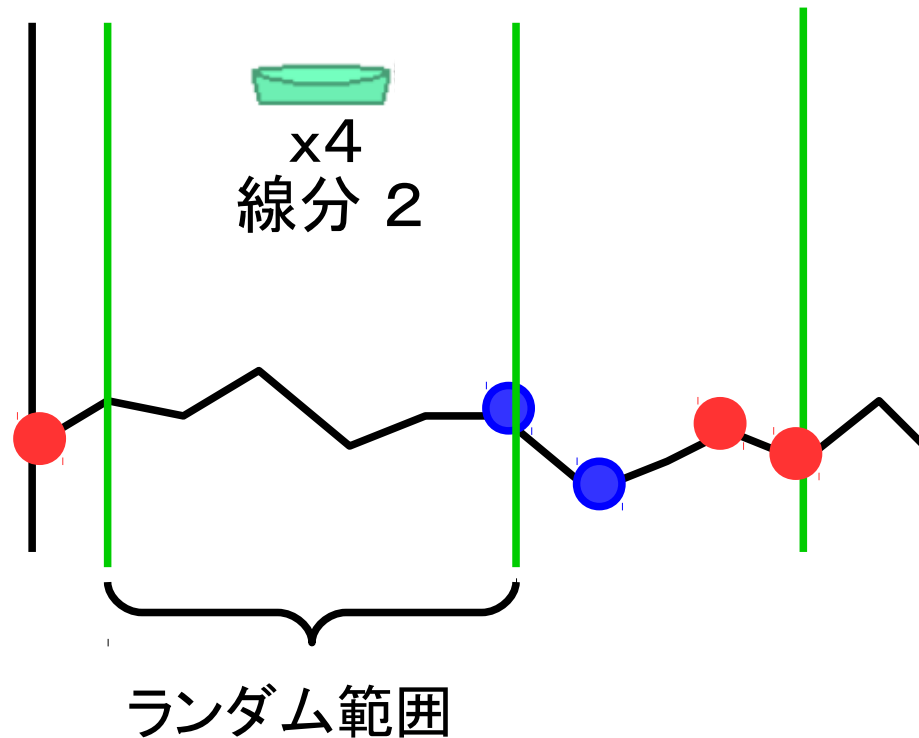
4. 4つの着陸地点に倍率をランダムに振り分ける。



5. 着陸地点を4つ配置するためにX軸を4つに分割する。

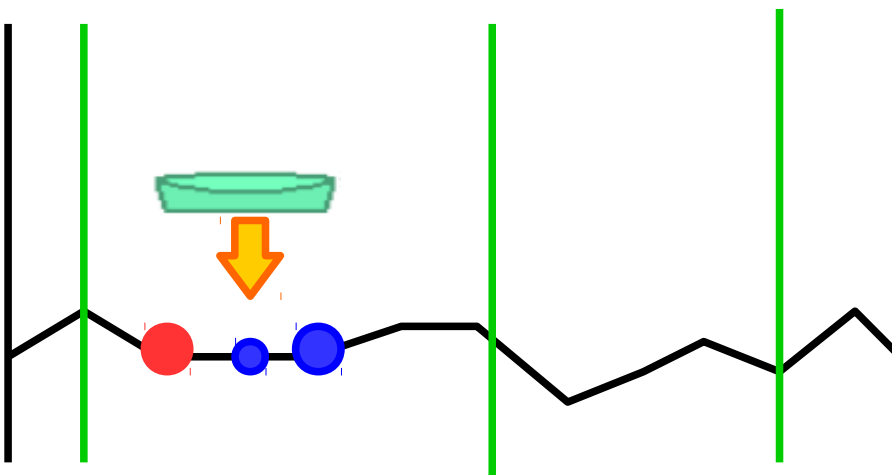


6. 範囲内から着陸地点の始点を決める。



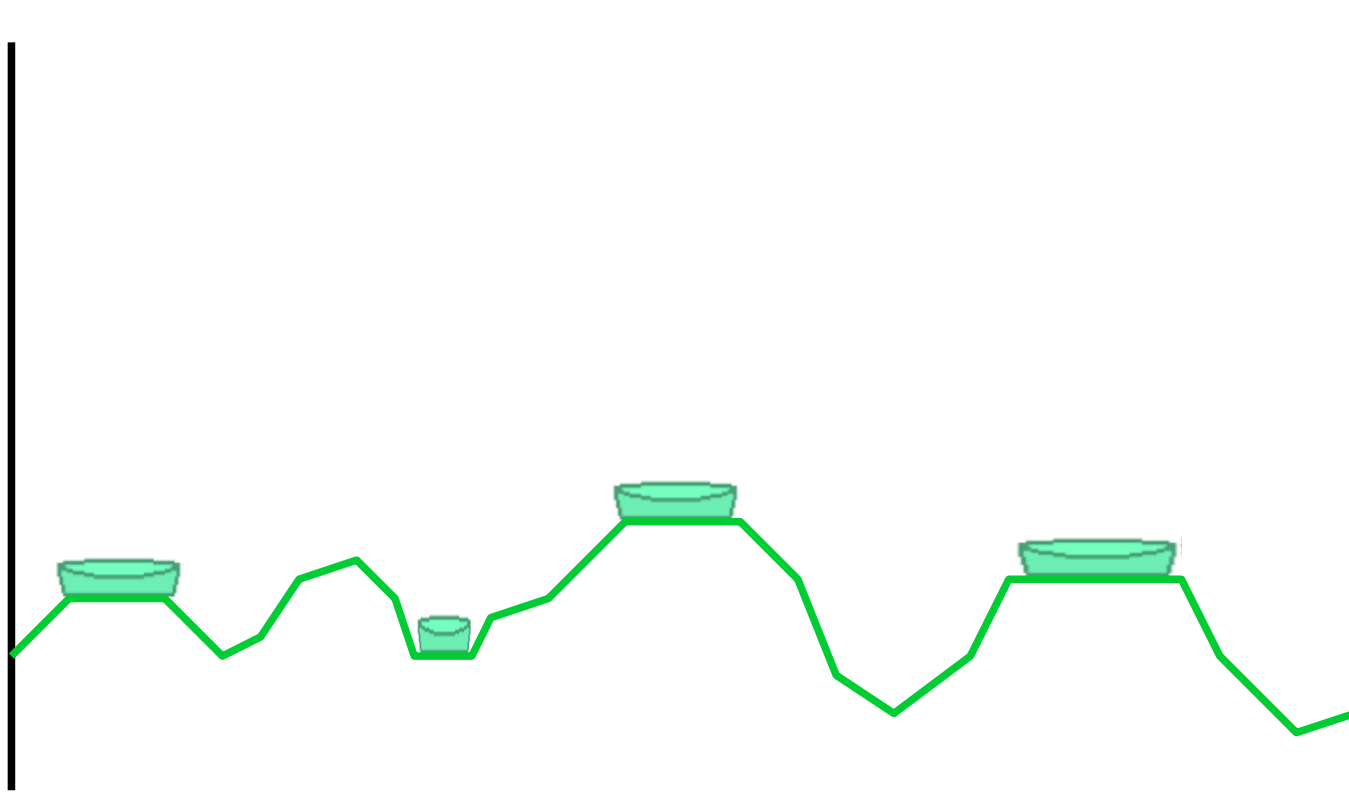
- 【始点のルール】
- 範囲の一番左の頂点と範囲の一番右の頂点と一つ前の頂点は除外する。
 - 着陸地点に設定した倍率によって線分の数が変わる。
 - ・ 線分の数分範囲の右側から縮小する。

7. 終点を決めて、着陸地点を配置する。



- 【平坦にする】
- 始点のY座標に
終点とその間の頂点のY座標
を合わせる。
- 着陸地点のサイズを設定し
配置する。

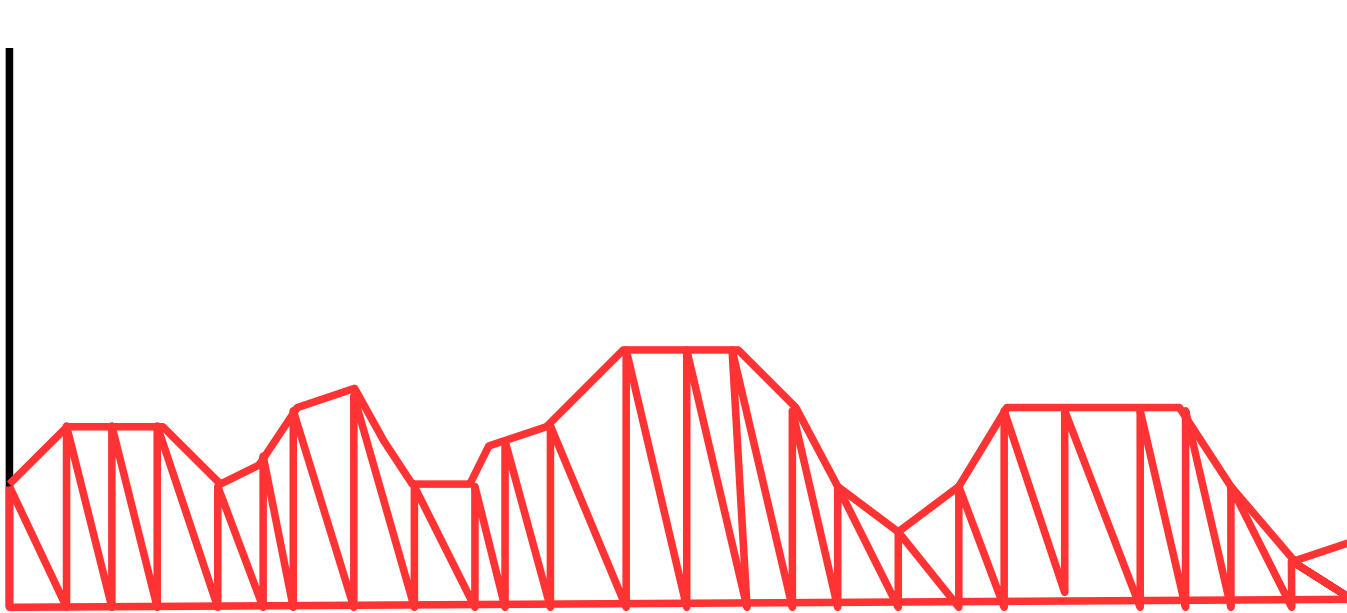
8. 出来上がった頂点データをEdgeCollider2Dに渡す。



この段階で月面ステージの出来上がり

9. 月面ステージのMeshを作成する。

- ・月面オブジェクトにMeshFilterとMeshRendererコンポーネントを追加する。



- ・各頂点の真下に頂点を追加する。
- ・インデックスを作成する。
- ・UV値を各頂点に設定する。
- ・頂点とインデックス、UV値からMeshデータを作成する。
- ・MeshデータをMeshFilterに渡すとMeshRendererによって描画される。

4. 感想と反省

- ・完成までスムーズに作業を進めることができた。
- ・実際に遊んでもらって出た意見から
見えた修正点について

ご清聴ありがとうございました。