

# 「マイ-ジャンプ」

## 1. 概要説明

これから、横スクロールアクションゲームのサンプルプログラムをご説明いたします。  
7つのステップがあります。

### ステップ

- ・ キャラクターを動かそう
- ・ コースを作ってみよう
- ・ リングを配置しよう
- ・ タイムとスコアを表示しよう
- ・ ゲームオーバーを作ろう
- ・ ゲームクリアを作ろう
- ・ 説明画面を作成しよう

### ゲームの内容

まずは、「0.ゲームをプレーしよう!」を実行し、どの様なゲームをこれから作るのかをご覧ください。

#### 【遊び方】

キャラクターを左右に動かし、ジャンプを駆使してゴールを目指しましょう！  
コースの途中には、動くブロックや落とし穴が仕掛けられています。足を踏み外して水に落ちるとゲームオーバーなので、慎重に進んでください。

#### 【操作方法】

キーボード：A キー(左移動)、D キー(右移動)、スペースキー(ジャンプ)

タッチ操作：ジョイパッド左(左移動)、ジョイパッド右(右移動)、A キー(ジャンプ)

コントローラー：コントローラー左(左移動)、コントローラー右(右移動)、コントローラー A ボタン(ジャンプ)

#### 【スコアとクリア条件】

リング：回転しているリングをゲットすると、スコアが加算されます。

スター：最後にスターを取ればゴールです！

## 2. キャラクターを動かそう

### 目的

- ・ カメラ、背景、ゲーム内で使用するキャラクターを配置します。
- ・ 配置したキャラクターに左右移動やジャンプの動作を作ります。
- ・ タッチ、ゲームコントローラー、キーボードの操作に対応します。

### 使用する変数

名前	種類	説明
ゲームステータス	変数	ゲーム画面の種類を管理する。 1: ゲームスタート

### 使用するオブジェクト

名前	カテゴリ	説明
カメラ	-	ゲーム内カメラアングルを指定
プレーン (草)	-	-
プログラム	モデル/プログラム	初期化 ゲームステータス管理等
マイ	モデル/人	操作キャラクター
ジョイパッド	ツール	左右移動
A ボタン	ツール	ジャンプ

## 作成手順

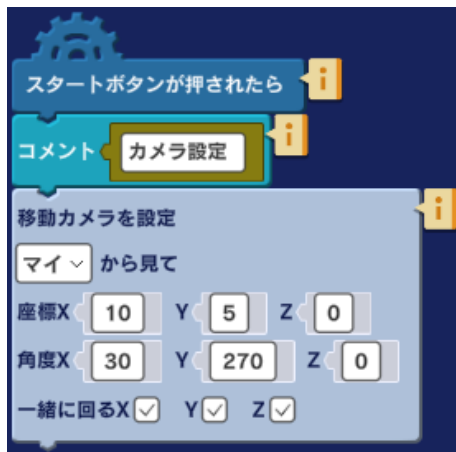
### 主人公を追加します

- ・ 主人公である、「マイ」を追加しましょう。
- ・ オブジェクト追加から、カテゴリー：人よりマイを選択して追加してください。



### カメラを設定しましょう

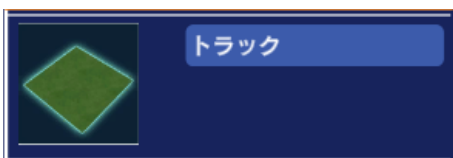
- ・ オブジェクト：カメラに下記のように、カメラ設定を配置します。



#### オブジェクト：カメラ

- ・ イベント：「スタートボタンが押されたら」を配置してください。
- ・ Mind Render 起動時に1度実行。
- ・ コメント文
- ・ カメラ：「移動カメラを設定」
- ・ マイから見た座標、角度を指定し、全身が映るように設定。

## 背景に指定します



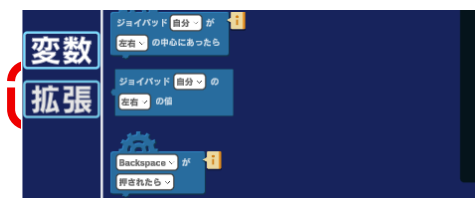
- ・ トラックは背景を指定するオブジェクトです。
- ・ 今回は、標準のプレーン（草）を使用します。

## 変数を追加します

- ・ 変数設定にて今回使用する「ゲームステータス」を追加します。
- ・ プログラムエディターに移動します。



- ・ 画面の左下の変数ボタンをタップし、変数設定画面に移動します。



## 変数設定

- ・ 変数設定に移動して、テキストボックスに「ゲームステータス」を追加し設定。

変数



## ジョイパッドを追加

- オブジェクト一覧のツール：ジョイパッドを選択し追加します。



- シーンエディターに移動します。



- 下記のように座標と大きさを設定します。

ジョイパッド			
座標	120.00	120.00	0.00
角度	0.00	0.00	0.00
大きさ	2.00	2.00	2.00

- プログラムエディターに移動します。



- 左右操作のプログラムを行います。

- 本オブジェクトに、左右操作のプログラムを作ります。ジョイパッド、ゲームコントローラー、キーボードで操作出来るように実装します。

- [ジョイパッド向け]



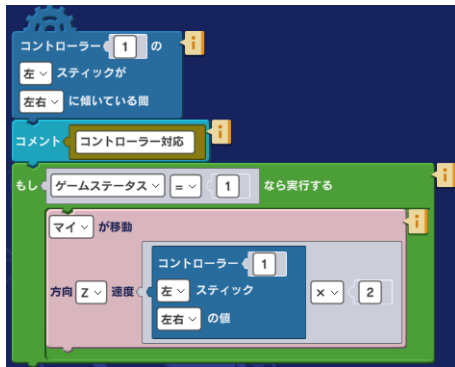
オブジェクト：ジョイパッド



[左移動]

- イベント：「[A]が[押されている間]」を配置
- iF：「もし[ゲームステータス]=1ならば実行する」を配置
- アクション：「[マイ]が移動」を配置して、-2の値をZ方向に与え移動します。

・ [コントローラー向け]

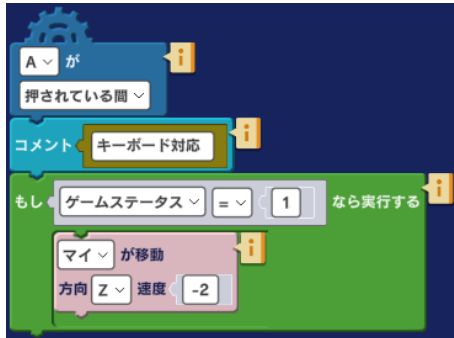


オブジェクト：ジョイパッド



- ・ イベント：「ジョイパッド[自分]が[左右]に傾いている間」を配置
- ・ iF：「もし[ゲームステータス]=1ならば実行する」を配置
- ・ アクション：「[マイ]が移動」を配置して、ジョイパッド[自分]の[左右]の値×2の値をZ方向に与え移動します。

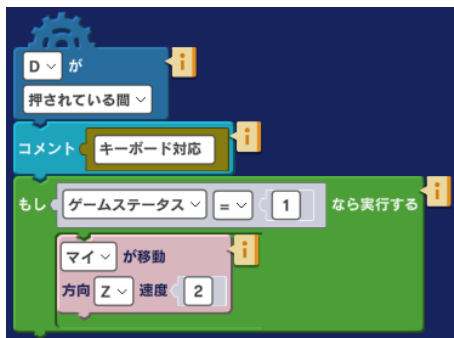
・ [キーボード向け]



オブジェクト：ジョイパッド



- ・ イベント：「コントローラー[1]の[左]スティックが[左右]に傾いている間」を配置
- ・ iF：「もし[ゲームステータス]=1ならば実行する」を配置
- ・ アクション：「[マイ]が移動」を配置して、コントローラー[1]、[左]スティック[左右]の値×2の値をZ方向に与え移動します。



オブジェクト：ジョイパッド

[右移動]

- ・ イベント：「[D]が押されている間」を配置
- ・ iF：「もし[ゲームステータス]=1ならば実行する」を配置
- ・ アクション：「[マイ]が移動」を配置して、2の値をZ方向に与え移動します。

## A ボタンを追加

- ・ オブジェクト一覧のツール : A ボタンを選択し追加します。



- ・ シーンエディターに移動します。



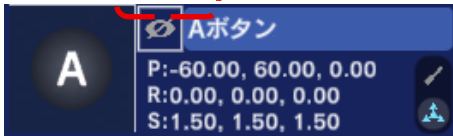
- ・ 下記の様に座標と大きさを設定します。

Aボタン			
座標	-60.00	60.00	0.00
角度	0.00	0.00	0.00
大きさ	1.50	1.50	1.50

- ・ プログラムエディターに移動します。



- ・ プログラムで表示の ON/OFF を行うため、オブジェクト一覧にて非表示に設定します。



- ・ ジャンプ操作の実装

- ・ 本オブジェクトに、ジャンプ操作のプログラムを作ります。ジョイパッド、ゲームコントローラー、キーボードで操作出来るように実装します。

・ [コントローラー対応]



オブジェクト : A ボタン

☆  
A  
Aボタン

[右移動]

- ・ イベント : 「[D]が押されている間」を配置
- ・ iF : 「もし[ゲームステータス]=1 ならば実行する」を配置
- ・ アクション : 「[ドンポンプ]が移動」を配置して、2 の値を Z 方向に与え移動します。

・ [キーボード対応]



※注 : 条件に「接地していること」を指定し、必ず着地している状態からしか、ジャンプ出来ないようにしています。無限にジャンプし続けることを制限しています。

プログラムを追加します

・ ゲームの状態を管理するためのプログラムを作ります。

- ・ このオブジェクト内に、プログラムを管理するブロックを配置します。
- ・ オブジェクト追加より、モデル/プログラムを指定し、プログラムを選択し、オブジェクト一覧に追加します。



- ・ プログラムエディターに移動します。

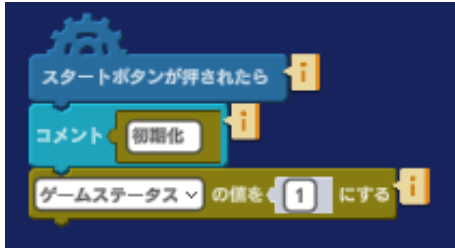


オブジェクト : A ボタン

☆  
A  
Aボタン

- ・ イベント : 「コントローラー[1]の[A]ボタンが[押されたら]」を配置
- ・ iF : 「もし[ゲームステータス]=1 ならば実行する」を配置
- ・ iF : 「もし「[ドンポンプ]が接地している」ならば実行する」を配置
- ・ アクション : 「[ドンポンプ]を押す [世界]からみて」を配置して、Y 方向に 2500 の力で押します。

- ・ ゲームステータスが「1」に設定します。



- ・ ゲームステータスが 1:ゲームスタート時の処理を作成します。



### オブジェクト：プログラム



- ・ ゲームステータスが変更され、値が1のときにプログラム。
- ・ 「ゲームステータスが変更されたら」を配置。
- ・ テクニック：「[ジョイパッド]の表示[ON]
- ・ テクニック：「[A ボタン]の表示[ON]
- ・ アクション：「[ドンポン]の座標を設定」で値を指定して初期化。
- ・ アクション：「[マイ]の角度を設定」で値を指定して初期化。
- ・ テクニック：「[マイ]の表示[ON]



ト時の処理を作成します。

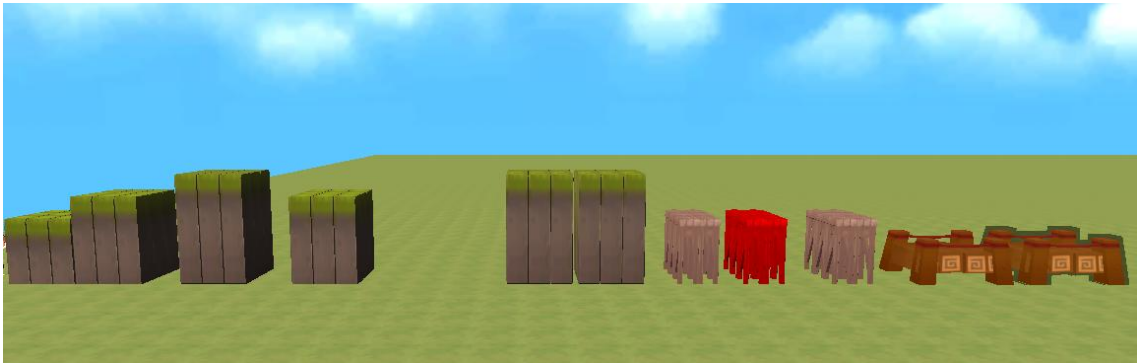
### オブジェクト：プログラム

- ・ 「スタートボタンが押されたら」を配置。
- ・ Mind Render 起動時に1度実行。
- ・ 変数：ゲームステータスを1に設定する。
- ・ 「1:ゲーム中」に指定。

## 2 コースを作ってみよう

### 目的

- ・ ゲームの土台となるコースを作成します。
- ・ シーンエディターで必要なオブジェクトを配置し、配置されたオブジェクトに対してプログラムを実装します。



### 使用する変数

名前	種類	説明
ゲームステータス	変数	ゲーム画面の種類を管理する。 1:ゲーム中

### 使用するオブジェクト

名前	カテゴリ	説明
カメラ	-	ゲーム内カメラアングルを指定
プレーン (草)	-	-
プログラム	モデル/プログラム	初期化 ゲームステータス管理等
マイ	モデル/人	操作キャラクター
ジョイパッド	ツール	
A ボタン	ツール	
草ブロック (高)_1-6	モデル/素材	
木ブロック_1-3	モデル/素材	
橋_1-2	モデル/素材	

## 作成手順

コースを作りましょう。

草ブロックを配置。

- ・ オブジェクト追加より、モデルを指定。「草ブロック(高)」を選択し、オブジェクト一覧に追加します。



- ・ シーンエディターに移動します。



- ・ 下記の様に座標と大きさを設定します。

草ブロック(高)_1			
座標	0.00	3.00	-96.76
角度	0.00	0.00	0.00
大きさ	1.00	1.00	1.00

- ・ オブジェクト一覧からオブジェクト詳細画面に遷移します。



- ・ オブジェクト一覧にて、オブジェクトコピーを実行します。

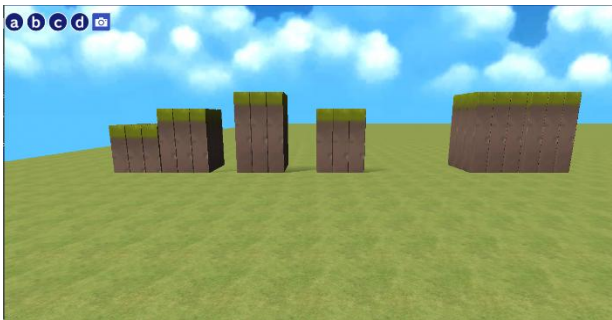


- ・ 草ブロックは6つ必要のため、5回コピーを実行します。
- ・ コピー後は名称を変更します。



- ・ 草ブロック(高)\_2~6の配置
  - ・ 草ブロック(高)\_1と同じ要領で、下記の座標に配置します。

(5) (6)



名前	座標
(1)草ブロック(高)_1	0.00, 3.00, -96.76
(2)草ブロック(高)_2	0.00, 4.00, -93.73
(3)草ブロック(高)_3	0.00, 5.00, -89.00
(4)草ブロック(高)_4	0.00, 4.00, -83.96
(5)草ブロック(高)_5	0.00, 5.00, -74.24
(6)草ブロック(高)_6	0.00, 5.00, -71.00

### 木ブロックの配置



- ・ シーンエディターに移動します。



- ・ 下記の様に座標と大きさを設定します。

木ブロック_1			
座標	0.00	2.99	-67.00
角度	0.00	0.00	0.00
大きさ	1.00	1.00	1.00

- ・ オブジェクト一覧からオブジェクト詳細画面に遷移します。



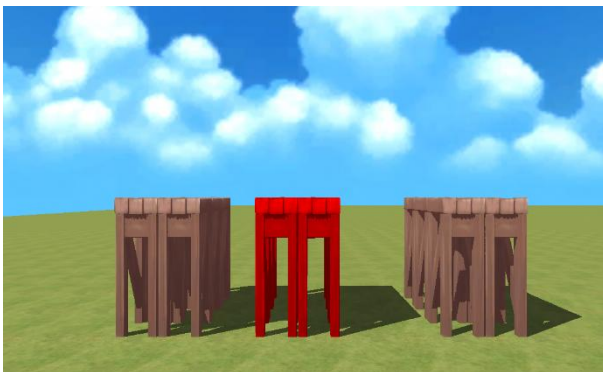
- ・ オブジェクト一覧にて、オブジェクトコピーを実行します。



- ・ 木ブロックは3つ必要なため、2回コピーを実行します。
- ・ コピー後は名称を変更します。



- ・ 木ブロック\_2~3の配置
  - ・ 木ブロック\_1と同じ要領で、下記の座標に配置します。



名前	座標
(1)木ブロック_1	P:0.00, 2.99, -67.00
(2)木ブロック_2	P:0.00, 2.99, -64.00
(3)木ブロック_3	P:0.00, 2.99, -59.98

(3)

## 橋の配置



- ・ シーンエディターに移動します。



- ・ 下記の様に座標と大きさを設定します。

橋_1			
座標	0.00	1.38	-55.00
角度	0.00	0.00	0.00
大きさ	1.00	1.00	1.00

- ・ オブジェクト一覧からオブジェクト詳細画面に遷移します。



- ・ オブジェクト一覧にて、オブジェクトコピーを実行します。



- ・ 橋ブロックは2つ必要なため、1回コピーを実行します。
- ・ コピー後は名称を変更します。



- ・ 橋ブロックの配置
  - ・ 木ブロック\_1と同じ要領で、下記の座標に配置します。



名前	座標
(1)橋_1	P:0.00, 1.38, -55.00
(2)橋_2	P:0.00, 1.38, -50.27

(2)

## プログラムの追加

### 上下に動く床の設定

- ・ 草ブロック(高)\_3 に、プログラムを実装し、自動的に上下する様に設定します。



#### オブジェクト：草ブロック(高)\_3



- ・ イベント：「スタートボタンが押されたら」を配置。
- ・ Mind Render 起動時に 1 度実行。
- ・ iF：常にくり返す
- ・ アクション：「[自分]の[座標]を変える」を配置。
- ・ 2 秒間隔で、Y 座標を 0<—>5 に変速させて変化を指定。

### 左右に動く床の設定

- ・ 草ブロック(高)\_5 に、プログラムを実装し、自動的に左右に動く設定します。



#### オブジェクト：草ブロック(高)\_5




- ・ イベント：「スタートボタンが押されたら」を配置。
- ・ Mind Render 起動時に 1 度実行。
- ・ iF：常にくり返す
- ・ アクション：「[自分]の[座標]を変える」を配置。
- ・ 2 秒間隔で、Z 座標を-74<—>-80 に変速させて変化を指定。

## 落ちる床の設定

- ・ 物理設定を変更して、主人公が上乗ると、バランスを崩してしまう設定に変更します。
- ・ オブジェクト詳細画面で、下記の様に設定を変更します。



オブジェクト：木ブロック\_2



- ・ オブジェクト一覧から「木ブロック\_2」のオブジェクト詳細を選択し、左図のように設定を行いましょう。

### 3 リングを配置しよう

#### 目的

- ・ コース内に座標を指定して各リングを配置するプログラムを作ります。
- ・ リングは一定間隔で回転します。
- ・ マイがリングに接触するとエフェクトを表示するプログラムを作成します。

#### 使用する変数

名前	種類	説明
ゲームステータス	変数	ゲーム画面の種類を管理する。 1:ゲーム中

#### 使用するオブジェクト

名前	カテゴリー	説明
カメラ	-	ゲーム内カメラアングルを指定
プレーン (草)	-	-
プログラム	モデル/プログラム	初期化 ゲームステータス管理等
マイ	モデル/人	操作キャラクター
ジョイパッド	ツール	
A ボタン	ツール	
草ブロック (高)_1-6	モデル/素材	
木ブロック_1-3	モデル/素材	
橋_1-2	モデル/素材	
リング_1_6	モデル/小物	

## 作成手順

### リングを配置します

- ・ オブジェクト追加より、モデルを指定。「リング」を選択し、オブジェクト一覧に追加します。



- ・ オブジェクト一覧からオブジェクト詳細画面に遷移します。



## ブロックによる座標設定と回転表示を設定

- ・ リングをブロックにて座標を指定し、回転を行うプログラムを構築します。



オブジェクト：リング\_1

- ・ イベント：「[自分] が変更されたら」を配置。
  - ・ 「ゲームステータス」に変更。
  - ・ 自分の座標を設定。
  - ・ iF：常に繰り返す
    - ・ Y座標を基準に速度3で回転、

リング\_1 の座用

名前	座標
(1)リング_1	X: 0 Y: 5 Z: -96.76

## 「マイ」と接触した際の動きを設定

- ・ リングとマイが接触した際、爆発(1) エフェクトを表示し、消えるブロックを構築する。

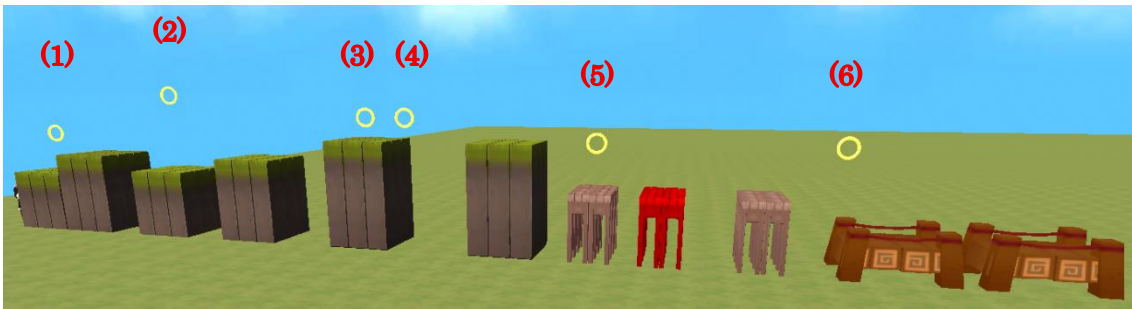


オブジェクト：リング\_1

- ・ イベント：「[自分] と[マイ]が接触したら」を配置。
  - ・ エフェクト：爆発（1）を表示。
  - ・ 自分を削除

## リンクを追加します

- ・ 下記のようにコース上にリングを配置します。



- ・ オブジェクト一覧にて、オブジェクトコピーを実行します。



- ・ リングは6つ必要なため、5回コピーを実行します。
- ・ コピー後は名称を変更します。

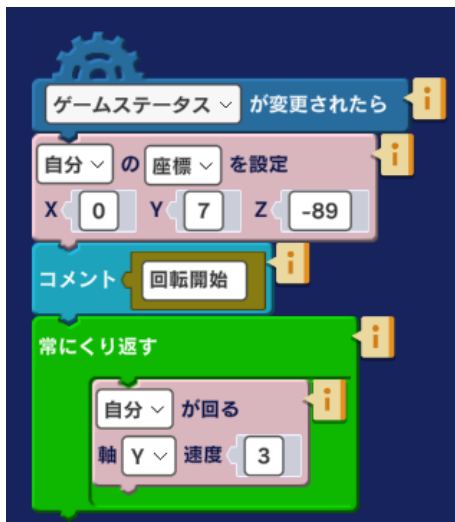


## オブジェクト：リング\_2～6へ設定

- ・ プログラムエディターに移動します。



- ・ オブジェクト一覧から、リング\_2～6を選択し、下記の座標をアクション：「[自分]の[座標]を設定する」に指定して、リングを配置します。



名前	座標
(2)リング_2	X: 0 Y: 7 Z: -89
(3)リング_3	X: 0 Y: 6 Z: -78
(4)リング_4	X: 0 Y: 6 Z: -76
(5)リング_5	X: 0 Y: 5 Z: -67
(6)リング_6	X: 0 Y: 5 Z: -57

## 4. タイムとスコアを表示しよう

### 目的

- ・ タイム計算と表示を作成します。
- ・ スコア計算と表示を作成します。
- ・ リンクを取得した際にスコアを 100 換算します。

### 使用する変数

名前	種類	説明
ゲームステータス	変数	ゲーム画面の種類を管理する。 1: ゲーム中
タイム	変数	ゲーム時間をカウントします。
スコア	変数	獲得したスコアをカウントします。

### 使用するオブジェクト

名前	カテゴリ	説明
カメラ	-	ゲーム内カメラアングルを指定
プレーン (草)	-	-
プログラム	モデル/プログラム	初期化 ゲームステータス管理等
マイ	モデル/人	操作キャラクター
ジョイパッド	ツール	
A ボタン	ツール	
草ブロック (高)_1-6	モデル/素材	
木ブロック_1-3	モデル/素材	
橋_1-2	モデル/素材	
リング_1_6	モデル/小物	
スコア表示	ツール/テキスト	タイトル(右)
タイム表示	ツール/テキスト	タイトル(右)

## 作成手順

スコア表示を作りましょう。



- ・ オブジェクト追加より、ツールを指定し、タイトル(右)を選択。



- ・ タイトル(右)の名称をスコア表示に変更。



- ・ シーンエディターに移動します。



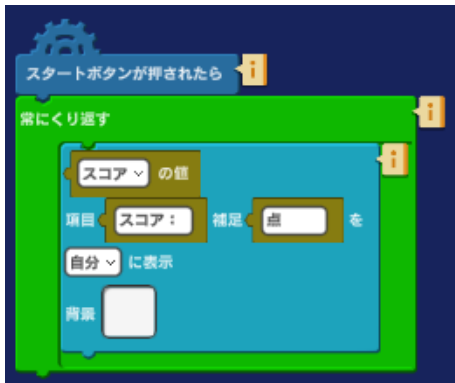
- ・ オブジェクト一覧にて、「スコア表示」を選択すると「グリッド移動」が表示されます。
- ・ 「グリッド移動」の「アンカー設定」にチェックしましょう。



- ・ 下記の様に座標を設定します。



- ・ プログラムを追加しましょう。
- ・ スコア表示「スコア：点」の表示を指定します。

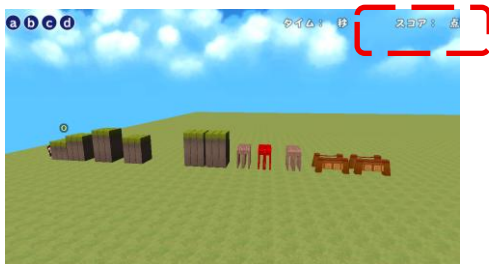


オブジェクト：スコア表示

☆  
A  
タイトル(右)

- ・ イベント：「スタートボタンが押されたら」を配置。
- ・ Mind Render 起動時に 1 度実行。
- ・ iF：常にくり返す
- ・ セリフ：「[Text] 項目[Text] 補足[Text]を自分に表示」

タイム表示を作りましょう。



- ・ オブジェクト追加より、ツールを指定し、タイトル(右)を選択。



- ・ タイトル(右)の名称をスコア表示に変更。



- ・ シーンエディターに移動します。



- ・ オブジェクト一覧にて、「スコア表示」を選択すると「グリッド移動」が表示されます。
- ・ 「グリッド移動」の「アンカー設定」にチェックしましょう。



- ・ 下記の様に座標を設定します。



- ・ プログラムを追加しましょう。
- ・ スコア表示「タイム： 秒」の表示を指定します。

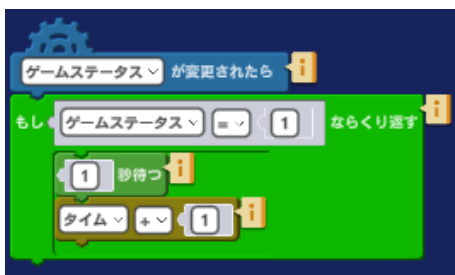


### オブジェクト：タイム表示

- ・ イベント：「スタートボタンが押されたら」を配置。
- ・ Mind Render 起動時に 1 度実行。
- ・ iF：常にくり返す
- ・ セリフ：「[Text] 項目 [Text] 補足 [Text] を自分に表示」



- ・ タイマーを作りましょう。(1 秒間に 1 カウントする)



### オブジェクト：タイム表示

- ・ イベント：「[自分] が変更されたら」を配置。
  - ・ 「ゲームステータス」に変更。
  - ・ 「ゲームステータス」が 0 から 1 などに変更された動作する様に指定。
- ・ 1 秒待つ、変数：タイムに 1 加算する。



## スコア加算処理を追加しましょう

- ・ リングに記載しているプログラム変更し、主人公と接触した際に、スコアを加算する処理を追加します。
- ・ 1つのリングを取得すると100点とします。

加算処理(100点)



オブジェクト：リング

- ・ イベント：「[自分] と[マイ]が接触したら」を配置。
  - ・ スコア加算処理を追加
  - ・ エフェクト：爆発（1）を表示。
  - ・ 自分を削除

- ・ この様に、リング\_2～リング\_6に対しても加算処理(100点)を追加します。

## 5.ゲームオーバーを作ろう

### 目的

- ・ 主人公が、コースから外れて、水オブジェクトに接触した場合、ゲームオーバーとする処理を実装します。
- ・ 接触した際、ゲームステータスを2に変更し、ゲームオーバーの状態とします。

### 使用する変数

名前	種類	説明
ゲームステータス	変数	ゲーム画面の種類を管理する。 1:ゲーム中
タイム	変数	ゲーム時間をカウントします。
スコア	変数	獲得したスコアをカウントします。

### 使用するオブジェクト

名前	カテゴリ	説明
カメラ	-	ゲーム内カメラアングルを指定
プレーン (草)	-	-
プログラム	モデル/プログラム	初期化 ゲームステータス管理等
水	モデル/自然	
マイ	モデル/人	操作キャラクター
ジョイパッド	ツール	
A ボタン	ツール	
草ブロック (高)_1-6	モデル/素材	
木ブロック_1-3	モデル/素材	
橋_1-2	モデル/素材	
リスタート	ツール	ボタン
リング 1_6	モデル/小物	
GameOver	ツール	
スコア表示	ツール	タイトル(右)
タイム表示	ツール	タイトル(右)
スコア結果表示	ツール	タイトル(中央)

## 作成手順

ゲームオーバー表示を作りましょう。

### 水オブジェクト配置

- ・ オブジェクト追加より、モデルを指定し、水を選択し、オブジェクト一覧に追加します。



- ・ シーンエディターに移動します。



- ・ 下記の様に座標と大きさを設定します。

水			
座標	0.00	0.10	-72.00
角度	0.00	0.00	0.00
大きさ	10.00	1.00	50.00

主人公にゲームオーバーの処理を追加しましょう。



オブジェクト：マイ

- ・ イベント：「[自分]と[自分]が接触したら」を配置。
  - ・ [自分]と[水]が接触したらに変更。
  - ・ 接触したら、マイの表示を OFF
  - ・ エフェクト：爆発(1)を表示。
  - ・ ゲームステータスを 2 に変更。

## ゲームオーバーオブジェクト配置

- ・ オブジェクト追加より、ツールを指定し、GameOver を選択。



- ・ シーンエディターに移動します。



- ・ オブジェクト一覧にて、「スコア表示」を選択すると「グリッド移動」が表示されます。
- ・ 「グリッド移動」の「アンカー設定」にチェックしましょう。



- ・ 下記の様に座標と大きさを設定します。



- ・ プログラムで表示の ON/OFF を行うため、オブジェクト一覧にて非表示に設定します。



スコア結果表示を作ります。

- ・ オブジェクト追加より、ツールを指定し、タイトル(中央)を選択。



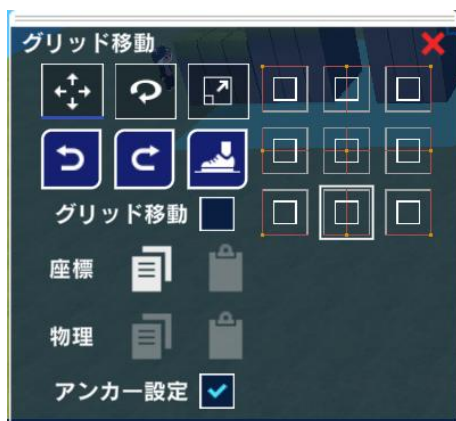
- ・ オブジェクト一覧よりタイトル(中央)を選択し、名称を「スコア結果表示」に変更。



- ・ シーンエディターに移動します。



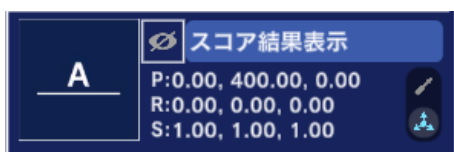
- ・ オブジェクト一覧にて、「スコア表示」を選択すると「グリッド移動」が表示されます。
- ・ 「グリッド移動」の「アンカー設定」にチェックしましょう。



下記の様に座標と大きさを設定します。

スコア結果表示			
座標	0.00	400.00	0.00
角度	0.00	0.00	0.00
大きさ	1.00	1.00	1.00

- ・ プログラムで表示の ON/OFF を行うため、オブジェクト一覧にて非表示に設定します。



リスタートボタンの処理を作りましょう。

ゲームオーバー時に、リスタートボタン、スペースキー、A ボタンをタップするとゲームステータスを 1 にして、ゲームスタートのステータスに変更します。

- ・ オブジェクト追加より、ツールを指定し、ボタンを選択。



- ・ シーンエディターに移動します。



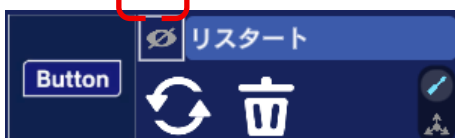
- ・ オブジェクト一覧にて、「リスタート」を選択すると「グリッド移動」が表示されます。
- ・ 「グリッド移動」の「アンカー設定」にチェックしましょう。



- ・ 下記の様に座標と大きさを設定します。

リスタート			
座標	0.00	200.00	0.00
角度	0.00	0.00	0.00
大きさ	3.00	3.00	3.00

- ・ プログラムで表示の ON/OFF を行うため、オブジェクト一覧にて非表示に設定します。



- ・ リスタート用のプログラムを記載します。



☆
Button

オブジェクト：リスタート
ボタン

- ・ タッチ操作対応
- ・ イベント：「[自分]が[タップされたら]」を配置。
  - ・ iF：もし、ゲームステータスが2の場合に実行
  - ・ ゲームステータスを1に変更



☆
Button

オブジェクト：リスタート
ボタン

- ・ キーボード操作対応
- ・ イベント：「[Space]が[押されたら]」を配置。
  - ・ iF：もし、ゲームステータスが2の場合に実行
  - ・ ゲームステータスを1に変更




☆
Button

オブジェクト：リスタート
ボタン

- ・ コントローラー操作対応
- ・ イベント：「コントローラー[1]の[A]ボタンが[押されたら]」を配置。
  - ・ iF：もし、ゲームステータスが2の場合に実行
  - ・ ゲームステータスを1に変更

ゲームステータス : 2 の処理を追加しましょう。

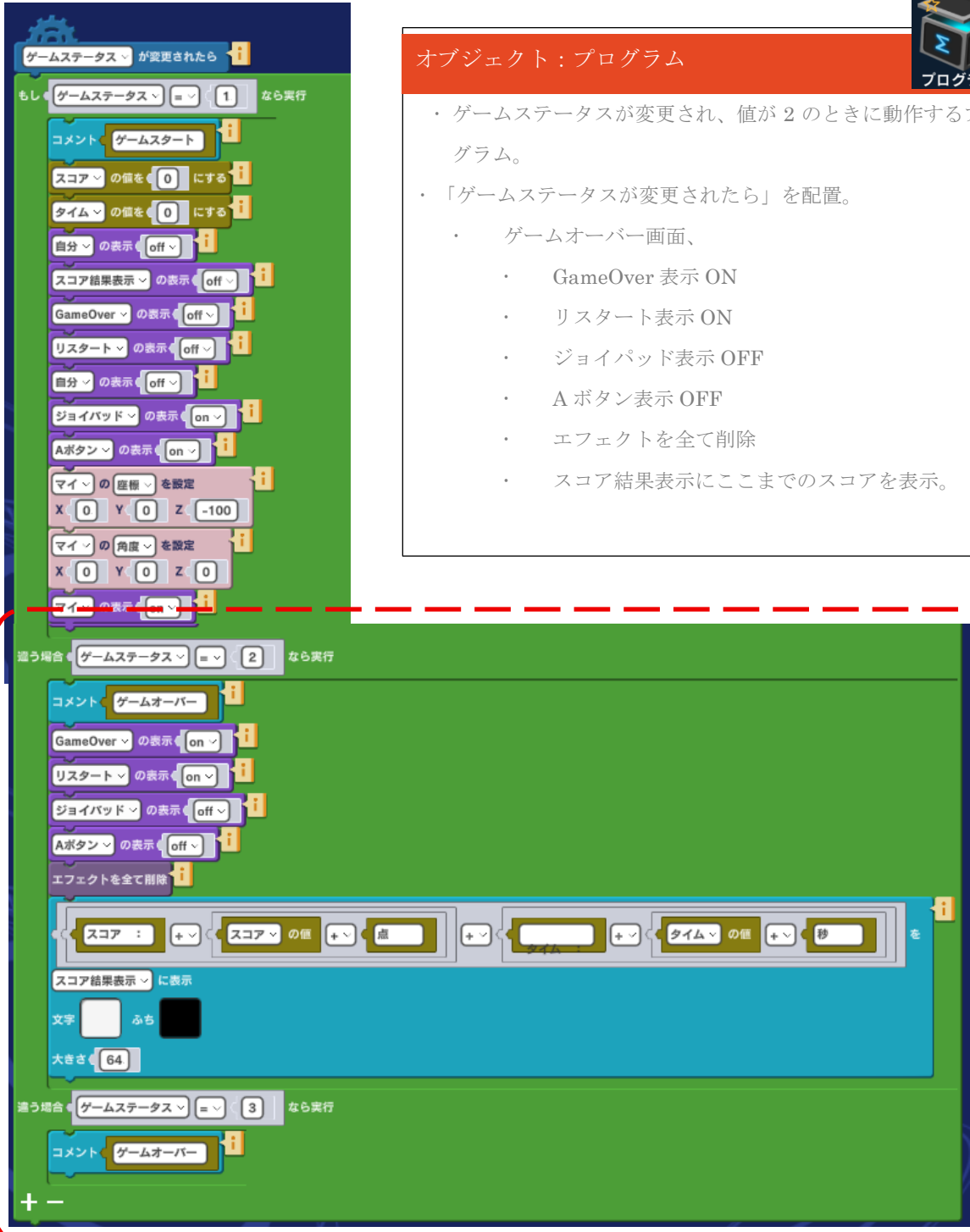
- ・ ゲームステータス 2 : ゲームオーバー時の処理を追加します。



プログラム

オブジェクト : プログラム

- ・ ゲームステータスが変更され、値が 2 のときに動作するプログラム。
- ・ 「ゲームステータスが変更されたら」を配置。
  - ・ ゲームオーバー画面、
    - ・ GameOver 表示 ON
    - ・ リスタート表示 ON
    - ・ ジョイパッド表示 OFF
    - ・ A ボタン表示 OFF
    - ・ エフェクトを全て削除
    - ・ スコア結果表示にここまでのスコアを表示。



ゲームステータス が変更されたら

もし ゲームステータス = 1 なら実行

コメント ゲームスタート

スコアの値を 0 にする

タイムの値を 0 にする

自分の表示 off

スコア結果表示の表示 off

GameOver の表示 off

リスタートの表示 off

自分の表示 off

ジョイパッドの表示 on

A ボタンの表示 on

マイの座標を設定  
X 0 Y 0 Z -100

マイの角度を設定  
X 0 Y 0 Z 0

マイの表示 on

違う場合 ゲームステータス = 2 なら実行

コメント ゲームオーバー

GameOver の表示 on

リスタートの表示 on

ジョイパッドの表示 off

A ボタンの表示 off

エフェクトを全て削除

スコア : + スコアの値 + 点 + タイムの値 + 秒 を

スコア結果表示 に表示

文字 数字 ふち

大きさ 64

違う場合 ゲームステータス = 3 なら実行

コメント ゲームオーバー

## 6. ゲームクリアを作ろう

### 目的

- ・ 主人公が、ゴールであるスターに接触した場合、ゲームクリアとする処理を実装します。
- ・ 接触した際、ゲームステータスを 3 に変更し、ゲームクリアの状態とします。

### 使用する変数

名前	種類	説明
ゲームステータス	変数	ゲーム画面の種類を管理する。 1: ゲーム中
タイム	変数	ゲーム時間をカウントします。
スコア	変数	獲得したスコアをカウントします。

### 使用するオブジェクト

名前	カテゴリ	説明
カメラ	-	ゲーム内カメラアングルを指定
プレーン (草)	-	-
プログラム	モデル/プログラム	初期化 ゲームステータス管理等
水	モデル/自然	
マイ	モデル/人	操作キャラクター
ジョイパッド	ツール	
A ボタン	ツール	
草ブロック (高)_1-6	モデル/素材	
木ブロック_1-3	モデル/素材	
橋_1-2	モデル/素材	
リング 1_6	モデル/小物	
スター	モデル/小物	
GameOver	ツール	
Victory	ツール	
スコア表示	ツール	タイトル(右)
タイム表示	ツール	タイトル(右)
スコア結果表示	ツール	タイトル(中央)

## 作成手順

ゲームクリア表示を作りましょう。

Victory 表示を作りましょう

- ・ オブジェクト追加より、ツールを指定し、Victory を選択。



- ・ シーンエディターに移動します。



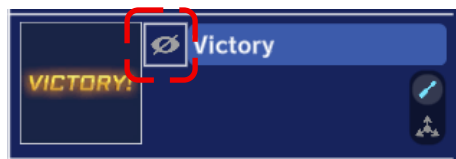
- ・ オブジェクト一覧にて、「スコア表示」を選択すると「グリッド移動」が表示されます。
- ・ 「グリッド移動」の「アンカー設定」にチェックしましょう。



- ・ 下記の様に座標と大きさを設定します。

Victory			
座標	0.00	-160.00	0.00
角度	0.00	0.00	0.00
大きさ	2.00	2.00	2.00

- ・ プログラムで表示の ON/OFF を行うため、オブジェクト一覧にて非表示に設定します。



スタートボタンの処理を作りましょう。

ゲームクリア時にタップするとゲームステータスを 1 にして、ゲームスタートのステータスに変更します。

- ・ オブジェクト追加より、ツールを指定し、ボタンを選択。



- ・ シーンエディターに移動します。



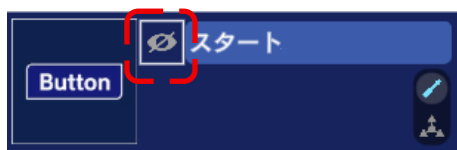
- ・ オブジェクト一覧にて、「リスタート」を選択すると「グリッド移動」が表示されます。
- ・ 「グリッド移動」の「アンカー設定」にチェックしましょう。



- ・ 下記の様に座標と大きさを設定します。

スタート			
座標	0.00	200.00	0.00
角度	0.00	0.00	0.00
大きさ	2.00	2.00	2.00

- ・ プログラムで表示の ON/OFF を行うため、オブジェクト一覧にて非表示に設定します。



スターを配置しましょう。

- ・ オブジェクト追加より、モデルを指定し、スターを選択。



- ・ 下記の様に座標と大きさを設定します。

スター			
座標	0.00	5.00	-50.00
角度	0.00	0.00	0.00
大きさ	1.00	1.00	1.00

ゴールした際の処理をくりましょう。

- ・ スターを回転させます。



オブジェクト：スター

- ・ イベント：「スタートボタンが押されたら」を配置。
  - ・ iF：常に繰り返す
  - ・ Y座標を基準に速度3で回転。

- ・ スターとマイが接触した際の処理を作ります。



オブジェクト：スター

- ・ イベント：「[自分]と[マイ]が接触したら」を配置。
  - ・ エフェクト：紙吹雪(爆発)
  - ・ スター表示を OFF
  - ・ ゲームステータスを3に変更

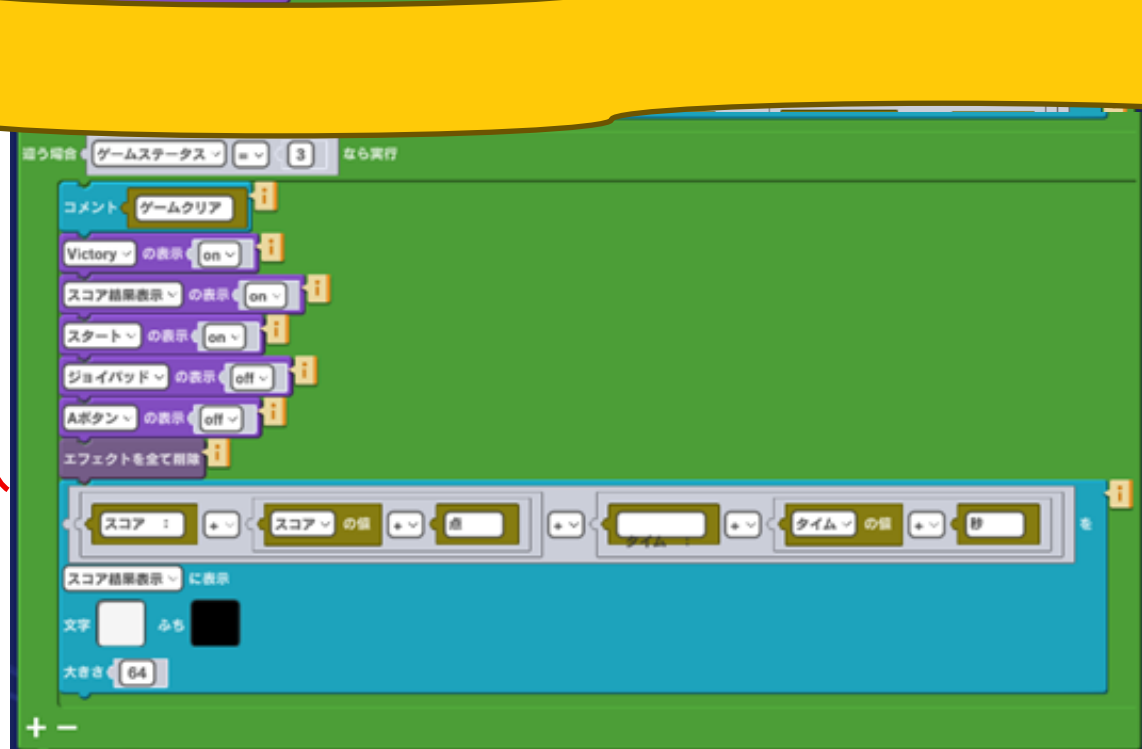
ゲームステータス : 3 の処理を追加しましょう。

- ・ ゲームステータス 3 : ゲームクリア時の処理を追加します。



### オブジェクト : プログラム

- ・ ゲームステータスが変更され、値が 3 のときに動作するプログラム。
- ・ 「ゲームステータスが変更されたら」を配置。
  - ・ ゲームクリア画面、
    - ・ Victory 表示 ON
    - ・ スコア結果表示 ON
    - ・ スタート表示 ON
    - ・ ジョイパッド表示 OFF
    - ・ A ボタン表示 OFF
    - ・ エフェクトを全て削除
    - ・ スコア結果表示にここまでのスコアを表示。



## 7.説明画面を作成しよう

### 目的

- ・ ゲーム起動時に表示する、ゲーム説明画面を作成します。
- ・ ゲームステータス : 0 タイトル表示を実装します。

### 使用する変数

名前	種類	説明
ゲームステータス	変数	ゲーム画面の種類を管理する。 1:ゲーム中
タイム	変数	ゲーム時間をカウントします。
スコア	変数	獲得したスコアをカウントします。

### 使用するオブジェクト

名前	カテゴリ	説明
カメラ	-	ゲーム内カメラアングルを指定
ブレーン (草)	-	-
プログラム	モデル/プログラム	初期化 ゲームステータス管理等
水	モデル/自然	
マイ	モデル/人	操作キャラクター
ジョイパッド	ツール	
A ボタン	ツール	
草ブロック (高)_1-6	モデル/素材	
木ブロック_1-3	モデル/素材	
橋_1-2	モデル/素材	
リング_1_6	モデル/小物	分身させて、コース内に複数配置する。
GameOver	ツール	
Victory	ツール	
スコア表示	ツール	タイトル(右)
タイム表示	ツール	タイトル(右)
スコア結果表示	ツール	タイトル(中央)
ダイアログ 1	ツール	

## 作成手順

説明画面を作りましょう。

- ・ オブジェクト追加より、ツールを指定し、ダイアログ 1 を選択。



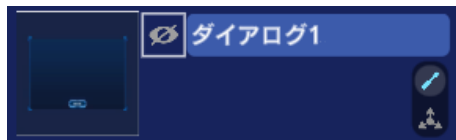
- ・ シーンエディターに移動します。



- ・ オブジェクト一覧にて、「スコア表示」を選択すると「グリッド移動」が表示されます。
- ・ 下記の様に座標と大きさを設定します。

ダイアログ1			
座標	0.00	0.00	0.00
角度	0.00	0.00	0.00
大きさ	1.00	1.00	1.00

- ・ プログラムで表示の ON/OFF を行うため、オブジェクト一覧にて非表示に設定します。



ゲームステータス : 0 の処理を追加しましょう。

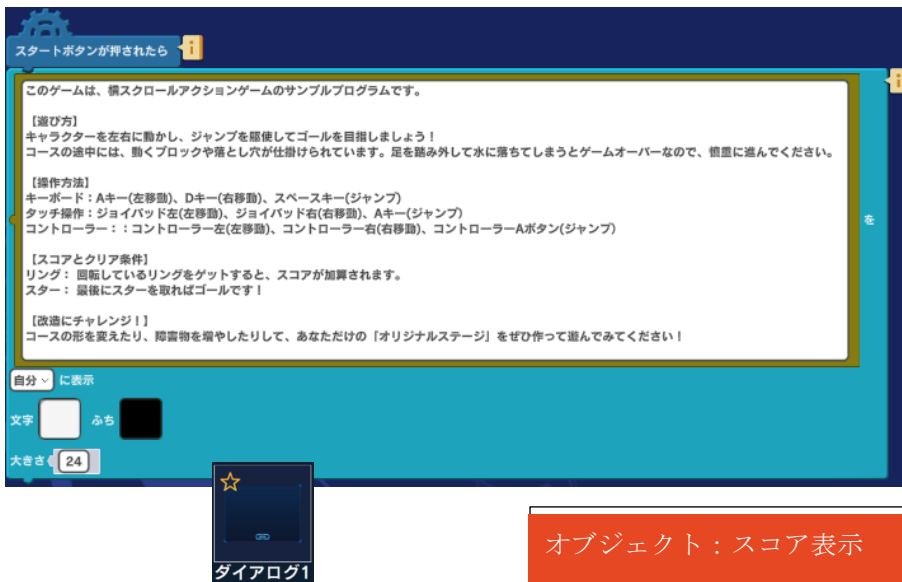
- ・ ゲームステータス 0 : タイトル表示の処理を追加します。



### オブジェクト : ダイアログ 1

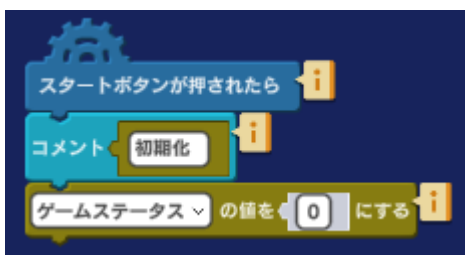
- ・ イベント : 「もし[ON]なら実行」を配置。
  - ・ テクニック : 「[自分]が表示されている」と演算 : [ON]ではなかったら」組合せ、「表示していなかったら」を作成。
  - ・ iF : ゲームステータスが 0 でなければ
    - ・ 自分を OFF
    - ・ ゲームステータスを 1 に変更

- ・ 表示内容を設定。



### オブジェクト : スコア表示

- ・ イベント : 「スタートボタンが押されたら」を配置。
- ・ セリフ : 「[Text][自分]に表示」にゲーム説明内容を設定。



### オブジェクト : プログラム

- ・ イベント : 「スタートボタンが押されたら」を配置。
  - ・ タイトル画面が完成したため、初期化時のゲームステータスを 0 に設定する。



## 7.完成！

これで、完成です。

コースの形を変えたり、障害物を増やしたりして、あなただけの「オリジナルステージ」をぜひ作って遊んでみてください！

