



M i n d R e n d e r 2

すぐできる！ 使い方マニュアル



目次

1. Mind Render の特徴	6
2. Mind Render の基本的な使い方	8
メニュー画面	8
ラボ	8
プログラム編集画面	10
シーン編集画面	11
グリッド移動	12
BG 設定	12
複数選択／範囲選択	13
オブジェクトのグループ化	13
カメラ操作	14
実行画面	15
ロッカー画面	16
オブジェクト一覧画面	17
ヒント画面	18
3. Mind Render ではこんなことができる！	19
あなたがプログラミング初心者なら	19
あなたが Scratch をやったことがあれば	20
あなたプログラミングを教える立場なら	20
4. ストーリーを作ろう 1：フォレストガール	21
1. 村の全景	22
2. おばあちゃんが・・・	23
3. お見舞いの品	24
4. 出発	26
5. ウサギと出会う	27
6. 薬草を集めよう	28
7. オオカミが！	30
8. 助けて猟師さん	30
9. 空から何かが	32

10. 助けたお礼	33
11. 金の卵	34
12. おばあちゃんの家	34
13. 薬を作ろう	35
14. おしまい	36
 5. ストーリーを作ろう 2：スペースボーイ	 37
1. 何か落ちてきたぞ？	37
2. 見に行ってみよう	38
3. 初めまして	40
4. 町へ行こう	41
5. 大変だ！	42
6. 魔法を解くには・・・	42
7. 魔法を解こう	45
8. 逃げろ！	46
9. 海賊のことを知ろう	47
10. 海賊を倒せ	49
 6. ストーリーを作ろう 3：「探偵ストーリー」	 51
1. 依頼の手紙	51
2. 予告状	52
3. 美術館へ行こう	53
4. 秘宝が！	55
5. 謎の男	56
6. 逃げた理由	57
7. 真犯人？	59
8. 本物の秘宝は・・・	60
9. 秘宝の秘密！	62
 7. ゲームを作ろう	 63
1. 宇宙船着陸	63
2. ドライブ 1	66
12. Game Maker	68
 8. 動画と一緒に作ろう	 69

一話完結動画.....	69
シリーズ動画.....	69
9. ムービーを作ろう.....	70
映画監督になってショートムービー作りに挑戦！.....	70
1. 惑星を目指す.....	72
2. 大気圏に突入.....	77
3. 着陸.....	81
4. 戦いが始まる.....	84
10. ちょっとお勉強 1：「ドローンを動かしてみよう」.....	89
1. 町.....	89
2. 人命救助.....	93
3. ドローンゲーム.....	96
4. しかけを作る.....	98
5. 軌跡を描く.....	101
11. ちょっとお勉強 2：「3D で日本を知ろう」.....	105
1. 納豆の消費量.....	105
2. グラフの作り方.....	107
3. インスタントラーメン消費量.....	111
4. グラフの切り替え表示.....	115
5. 世界人口推移.....	121
12. ちょっとお勉強 3：「サーキットを走ろう」.....	128
1. 車の操作.....	128
2. 記録と再生.....	130
3. 自動運転.....	132
4. 車の操作（雪山）.....	139
5. コースづくり.....	140
13. ちょっとお勉強 4：「大砲を撃ってみよう」.....	143
1. 砲弾を当てる 1.....	143
2. 3 つのターゲット.....	147
3. 横風の影響.....	151

4. 砲弾を当てる 2.....	153
5. 高い壁.....	155
14. 外部とつなごう.....	158
micro:bit	158
1.micro:bit で遊んでみよう	158
2.micro:bit をつないでみよう	158
レゴ SPIKE	159
1.車を運転しよう (Block).....	159
2.車を運転しよう (Python)	159
3.SPIKE を操作しよう	160
M5StickC	160
Raspberry Pi	160
1.車を運転しよう	161
2.ロボットアームを動かそう	161
インターネット対戦	161
1.ターンバトルをしよう	162
2.スピードバトルをしよう	162
対戦相手の探し方 (マッチング)	162
15. 遊び場 を使って自由に作品を作ろう	165
16. さらに Mind Render を学ぶための方法	166
17. Mind Render 2 を便利に使うヒント.....	167
1) ヒント動画	167
2) プログラムの保存	168
3) ロッカー登録	169
4) プログラムの共有	171
5) プログラムの公開	172
6) ランキング	173
7) プログラムのコピー.....	174
8) オブジェクトの追加.....	174
9) サウンドの追加	174

1. Mind Render の特徴

Mind Render（マインド レンダー）は、誰でも気軽に本格的な 3D 作品を作ることのできるプログラミングソフトウェアです。パソコンやタブレットで命令ブロックを動かして組み立てるだけでゲームやシミュレーション、VR コンテンツまで作ることができます。Mind Render は 2018 年に（株）モバイルインターネットテクノロジーによって開発され、無料ダウンロードで世界中の学校や個人がプログラム作品作りに活用しています。

Mind Render は発表以来、国立大学、有名な通信制大学、名門私立中高一貫校などでプログラミング教育や理数教育の現場で使われてきました。こうした教育機関や利用者の皆さんからの有益な改善のヒントをもとに Mind Render は進化しています。より使いやすく、より豊富な機能を備えてプログラミング初心者から上級者まで楽しみながらプログラム作品を作ることができるよう改良を重ねています。

Mind Render は以下のような優れた特徴を持っています。

- 直感的でわかりやすい画面デザインと操作性
- 美しい 3D グラフィクス
- プログラミング無しでキャラクターの動きを変えて楽しめるコンテンツが付属
- プログラミングもとっても簡単。マウス操作で命令ブロックを組んでいくだけでプログラムを作ることができる
- 楽しいキャラクター、宇宙船、お城や高層ビルの風景までさまざまな部品を自分で選んで、組み合わせる楽しい 3D 世界を自分で作ることができる
- 映画監督になったように、自由に視点を変えながら迫力ある場面の演出をすることができる
- 難しいプログラミング無しに限られた範囲で物理法則に沿ったものの動きを再現できる（物理エンジン）
- 自分のプログラム作品をクラウドに保存して友人に遊んでもらうことができる（クラウドロッカー）
- Mac、Windows、iOS、Android、Chromebook に対応

以上のような他にない優れた特徴を備えた Mind Render ですが、使い方を学ぶのもとても簡単にできるよう工夫されています。Mind Render にはラボと呼ばれる部屋が 8 つ用意されています。



それぞれのラボでは、作品づくりのヒントとなるようあらかじめいろいろなプログラムが作って置いてあります。ラボのプログラムを試しながら、少しずつ自分で作り替えていくことで自然にプログラミングのやり方を学ぶことができます。また、最初から作りたいときは「遊び場」を使います。「遊び場」にはサンプルプログラムは入っていないので、アイデアを膨らませて自分だけのプログラム

作品を作ることができます。

参考情報

「ちょっとお勉強」のラボについては Mind Render（2018 年版）用のテキストでも詳しく紹介しています。

「天才を育むプログラミングドリル—Mind Render で楽しく学ぶ」、白土良一他著、カットシステム刊 (ISBN-10 : 4877834362、ISBN-13 : 978-4877834364)

命令ブロックが変更されているため全く同じではありませんが、プログラムの考え方や大きな構成を理解いただく上では参考になります。本テキストでは同じラボを使った別の楽しみ方を紹介しております。

2. Mind Render の基本的な使い方

主な画面の機能や、操作方法を説明します。

メニュー画面

各画面へ遷移します。



つくる：ラボ一覧

動画：ラボ「動画と一緒に作ろう」

作品：公開された作品の一覧（要ログイン）

いいねランキング：公開された作品につけられた「いいね」ランキング（要ログイン）

ロッカー：ロッカーのログイン画面（要ログイン）

お知らせ：バージョンアップの内容などを確認できます。

設定：各種設定ができます。

続きから：前回作成していたプログラムが開きます。

Web：ブラウザで Web サイトが開きます。

ラボ

様々なサンプルプログラムが入っています。どのプログラムも自分で自由に変更できます。



新しい機能	バージョンアップで追加された新しいブロックや機能を使ったプログラム例を紹介します。
動画と一緒に作ろう	いろいろなゲームの作り方を動画で解説します。動画と一緒に作りながらゲーム作りのテクニックが学べます。
ゲームを作ろう	プログラミングなしで遊べるゲームがたくさん入っています。ゲーム作りに挑戦してみましょう。
ムービーを作ろう	映画の予告編のようなサンプルプログラムが入っています。映画監督になったつもりで迫力ある映像作りに挑戦してみましょう。
ストーリーを作ろう	ところどころでプログラミング問題を解きながら物語を進めていくサンプルプログラムが入っています。
ちょっとお勉強	ドローンを飛ばす、サーキットを走る、大砲を撃つなど色々な実験ができます。
外部とつなごう	micro:bit やレゴ SPIKE などの外部機器とつないだ実験や、インターネット対戦ゲームができます。
遊び場	サンプルプログラムは入っていません。アイデアを膨らませて自分だけのプログラム作品を一から作ることができます。

プログラム編集画面

ブロックを組んでプログラムを作る画面です。



左のアイコンをクリックするとプログラム編集画面が開きます。



シーン編集画面

オブジェクトの配置、座標や物理設定等のコピー、複数選択・範囲選択などができます。

 左のアイコンをクリックするとシーン編集画面が開きます。

グリッド線（緑色の格子）を上下に移動（プログラム実行には影響しません。）

（選択中のオブジェクトの）位置、角度、大きさの変更

（選択中のオブジェクトの）位置、方向、大きさを変更など（後述参照）

背景をクリックした位置にオブジェクトを配置（後述参照）

BG設定

角度	0.00	0.00	0.00
大きさ	1.00	1.00	1.00
グリッド	1.00	1.00	1.00

オブジェクトのグループ化（後述参照）

グリッド移動

選択中のオブジェクトは黄色に反転表示

ギズモ（赤、青、緑の矢印）



The screenshot shows the Mind Render scene editor interface. At the top, there are icons for 'ツール' (Tools), 'グリッド' (Grid), and 'BG設定' (BG Settings). The main 3D scene features a checkered floor and a character. A 'BG設定' dialog box is open, showing settings for angle, size, and grid. A 'グリッド移動' (Grid Move) dialog box is also open, showing movement options. A 'ギズモ' (Gizmo) is visible on the floor, indicating a selected object. The right sidebar shows a list of objects, including 'マイ' (Me), '爆発' (Explosion), and '岩ブロック' (Rock Block).

グリッド移動

左から、オブジェクトの移動、回転、大きさ変更 (ギズモによる操作)

左から、オブジェクト移動などの取り消し/やり直し

チェックを入れると、その時表示されているギズモに従って、オブジェクトが

- ・ 1 グリッドずつ移動
- ・ 15 度ずつ回転
- ・ 0.1 ずつ大きさ変更

します。



・ オブジェクトが空中にある場合に着地させます。
・ 埋もれている場合は、地上に出します。

座標のコピー & 貼り付け

1.座標をコピーしたいオブジェクト(A)を選択

2.コピー (📄 アイコンをタップ)

3.座標を貼り付けたいオブジェクト(B)を選択

4.貼り付け (📋 アイコンをタップ)

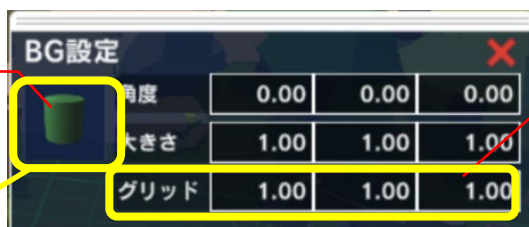
(上記操作で、オブジェクト(B)が、オブジェクト(A)の位置に移動します)

物理設定のコピー & 貼り付け

上記操作で、オブジェクト(A)の物理設定をオブジェクト(B)に貼り付けられます。

BG 設定

配置したいオブジェクトを選択します。



画面をタップすると、ここで指定した間隔をあけてオブジェクトが配置されます。



クリックした場所にオブジェクトが配置されます。

BG 設定でオブジェクトを配置する場合は、「グリッド移動」ウィンドウが表示されている場合は、閉じてください。

複数選択／範囲選択



オブジェクトリストで、離れた位置にあるオブジェクトを飛び飛びに複数選択する時に使用します。
主に、キーボードが無いタブレット用で、代わりに、キーボードの Ctrl (control) キーを押しながら選択することもできます。

オブジェクトリストで、連続した範囲のオブジェクトを選択したい時に押します。
主に、キーボードが無いタブレット用で、キーボードの Shift (⇧) キーを押しながら選択することもできます。

オブジェクトのグループ化



オブジェクトをグループに入れることができます。

エディター画面で、画面左下にある「拡張」ボタンからグループを作成します。



① グループに含めたいオブジェクトを選択します。

② 「グループ」ドロップダウンリストをクリックし、選択肢から上記で作成したグループ名を選択します。

①

カメラ操作

シーン編集画面では、以下の操作でズームイン／ズームアウトや、場所を移動することができます。

■ タッチスクリーン（スマホ、タブレット）の場合

- | | |
|--------------|--------------|
| 2 本指を動かしながら | 回転（上下／左右） |
| 3 本指を動かしながら | 移動（上下／左右） |
| 2 本指を広げる／狭める | ズームイン／ズームアウト |

■ マウスの場合

- | | |
|-----------------|------------------|
| マウスの右ボタンを押しながら | 回転（上下、左右） |
| マウスホイールを押しながら | 移動（上下、左右） |
| * または右ボタンを押しながら | WASD キーでも移動できます。 |
| マウスホイールを回しながら | ズームイン、ズームアウト |

■ mac の場合（トラックパッド）

- | | |
|---------------------------|-----------|
| option キーを押しながら 3 本指を動かす | 回転（上下、左右） |
| command キーを押しながら 3 本指を動かす | 移動（上下、左右） |
| 2 本指を広げる／狭める | 拡大・縮小 |

実行画面

作成したプログラムを実行します



録画した動画の保存場所は OS により異なります。「録画」ボタンについている「i」ボタンを押すと、保存場所が表示されます。

注意： Windows で録画機能を使用する場合

Microsoft ストアからダウンロードした Mind Render には録画機能がありません。Windows で実行画面を録画する場合は、以下の手順に従ってください。

1. ストアからダウンロードした Mind Render でプログラムを作成する
2. Mind Render ホームページから Mind Render をダウンロードする
3. 2 の Mind Render で実行画面を録画する

	プログラム作成	録画
ストアの Mind Render	○	×
ホームページの Mind Render	×	○

ロッカー画面

作成したプログラムの保存（クラウド）、共有、公開などが行えます。

The screenshot shows the 'Locker' interface in Mind Render. It features a top navigation bar with icons for saving, locking, sharing, searching, and user management. Below this, there are two program cards. Each card has a title, a share code input field, a share button, a public toggle, a play button, and a heart icon for likes. Callouts provide detailed instructions for each function.

プログラム保存
保存

公開プログラムの一覧と検索
共有ロッカー
検索
ユーザー

ユーザー名の設定（任意。プログラムを公開した時に表示されます。）

共有コードを入力して共有プログラムを表示

プログラムの説明を追加
プログラムへのコメント

タグを設定

共有コード（他の人とプログラムを共有したいときに設定します。）

共有リンク（共有コードを設定すると表示されます。クリックして表示される URL をクリックするだけでプログラムを実行することができます。（Mind Render のインストールが必要です。）

プログラムの公開（公開されると「検索」画面に表示され、だれでも実行できます。共有と異なり、プログラムの内容は公開されません。）

いいね数（公開した場合のみ）

プレイ数（公開した場合のみ）

左：ゲームのプレイ方法や自分用のメモなどを記載できます。（プログラムと一緒にコメントも公開されます。）

右：他の人からのメッセージを確認できます。

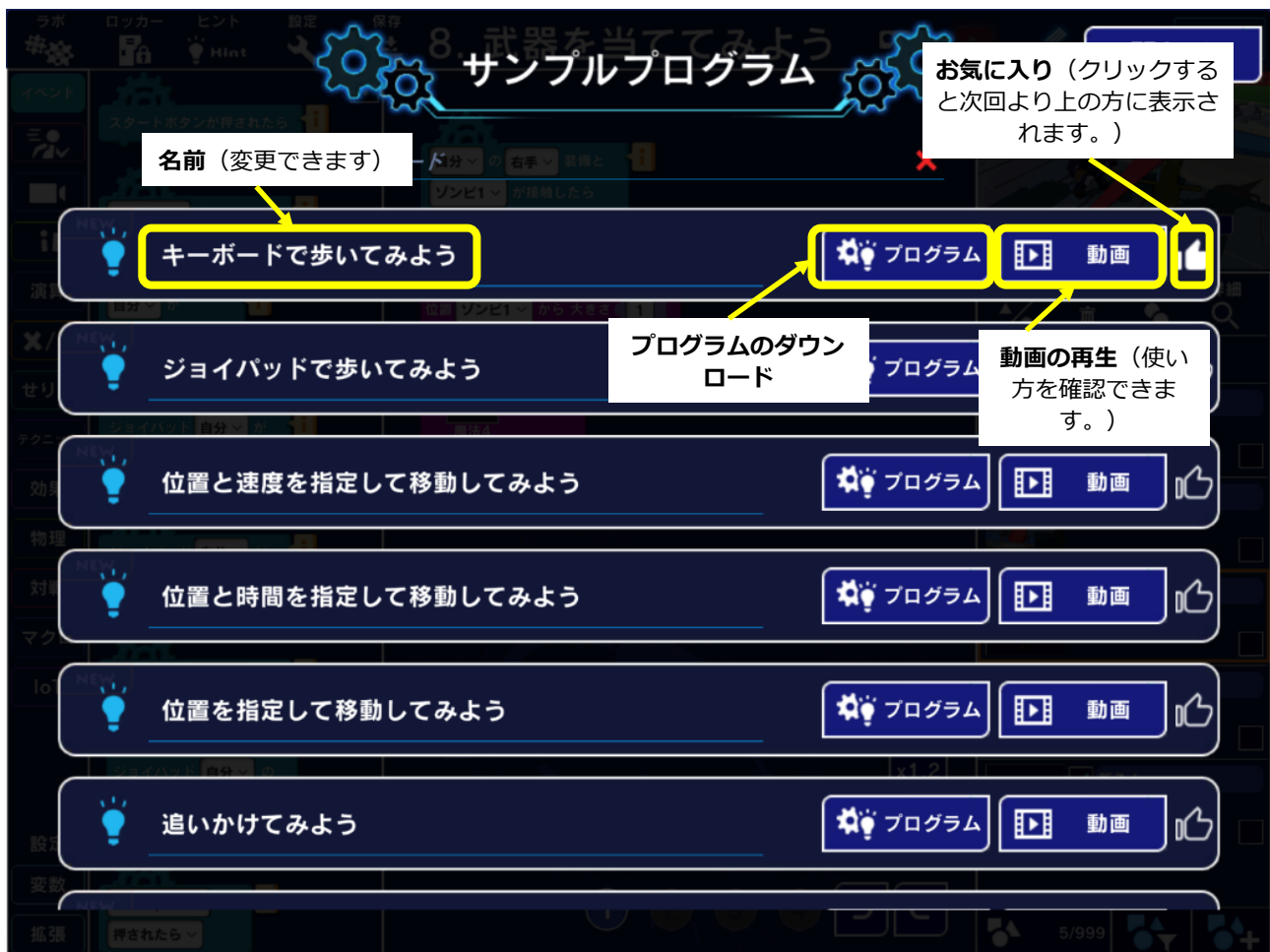
オブジェクト一覧画面

オブジェクト、ツール、サウンド、エフェクトを選ぶことができます。



ヒント画面

歩く、ジャンプなどの基本的なプログラムの作り方を動画で説明します。



3. Mind Render ではこんなことができる！

あなたがプログラミング初心者なら

初心者用のラボを使って、ゲーム感覚で楽しみながらプログラミングの基本を学ぶことができます。

「ストーリーを作ろう」の中にある「フォレストガール」、「スペースボーイ」などが初心者用のラボです。とても簡単なチャレンジから始めていきますのでこれまでプログラムをまったくやったことのない方でも安心して取り組みます。

フォレストガール



あかずきんちゃんのようなストーリー仕立てのコンテンツです。課題解決のヒントをうさぎに教えてもらったりオオカミに追いかけられたりと、わくわくしながらプログラミングでゴールを目指します。

スペースボーイ



宇宙から来た少年とネコを街の中に連れて行って海賊たちとのゲームに挑みます。海賊たちに追いかけられたりボールを発射して海賊たちを倒したりなど楽しいチャレンジが用意されています。

あなたが Scratch をやったことがあれば

ブロックプログラミングの基本を知っているあなたなら、「動画と一緒に作ろう」で「動画」を見ながらゲームを作ってみましょう。「シリーズ動画」は、Mind Render の基本的な使い方から始めて、一つのゲームの作り方を丁寧に説明しています。「一話完結動画」は、一動画で一ゲームを作っていきます。



また、好きなラボのプログラムをカスタマイズしてみましょう。

あなたプログラミングを教える立場なら

Mind Render のコンテンツはどれもアイデアと工夫で自由に作り変えることができます。学校や塾で基本的なプログラミング技術や理数科目の基礎知識を教える教材としても活用してみてください。

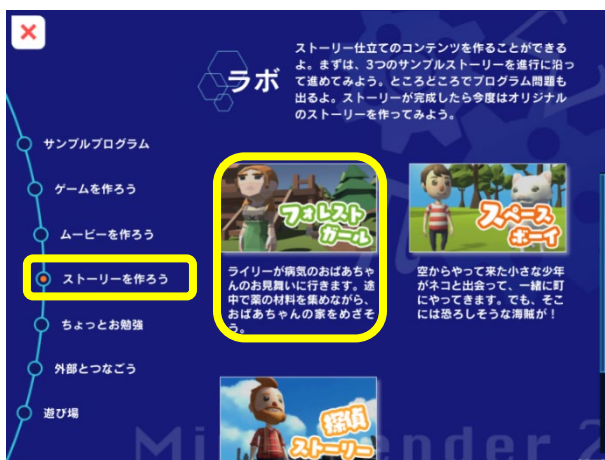
4. ストーリーを作ろう 1 : フォレストガール

「フォレストガール」はストーリー展開のある実験室です。画面やプログラムの中の説明に沿って進めて行きましょう。

ところどころでプログラム課題が出されますが、ヒントはたくさん置いてあります。必要な命令ブロックやサンプルプログラムが置いてあるので、参考にして作ることができます。さっそく始めてみましょう。



Mind Render を起動したら「つくる」をクリックします。



「ストーリーを作ろう」の中の「フォレストガール」を選択します。



「1 村の全景」の「最初から」を選択します。

* 始めてラボを開く時は「最初から」、前回終了したところから続けたい時は「続きから」を選びます。

1. 村の全景



下図が「村の全景」の起動画面です。

「実行」ボタンを押してフォレストガールのストーリーを始めましょう。



左図のように物語の進行に沿って説明が表示されますので何をすればよいか迷うことはありません。スタート地点から出発して、ゴール地点のおばあちゃんの家までお使いの用事をこなしながら向かいます。

次のステージ

アニメーションでストーリーの内容が説明されたら、左図の「次のステージ」ボタンをクリックして先に進めます。

発展課題

実行ボタンを押すと何度でもプログラムを実行することができます。繰り返し見てみましょう。

2. おばあちゃんが・・・



次のステージは「2 おばあちゃんが・・・」です。説明の画面が続きます。

次のステージ

説明が終わったら、「次のステージ」をクリックして先に進みます。

発展課題



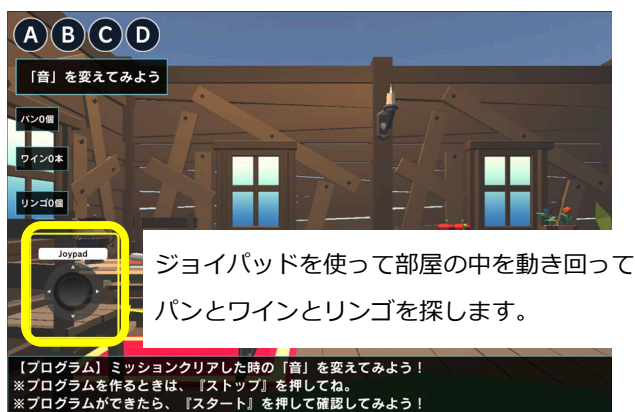
左図のボタンをクリックすると、シーン編集画面が開きます。シーン編集画面では、オブジェクトの位置を変更したり、シーンを色々な角度から見たりすることができます。



左図のようにズームアウトして外から部屋を見るにはどうしたらよいでしょうか。試してみましょう。

ズームアウトの操作は、「カメラ操作」を参考にしてください。

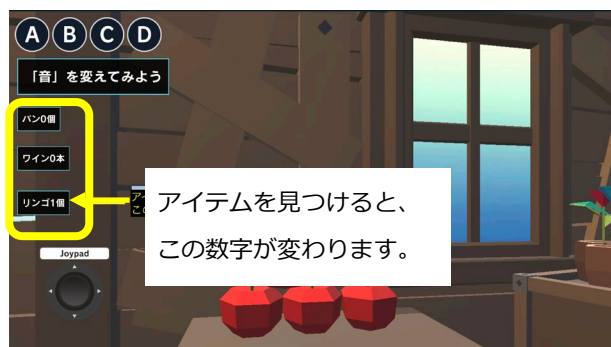
3. お見舞いの品



ステージ「3 お見舞いの品」では初めてのプログラミングに挑戦します。

プログラミングといってもすでに出来上がっているプログラムを見て、どんな動きになっているかを調べてミッションクリアした時の音を変えるために少しでもプログラムに手を加えるというものです。

ここでいうミッションとはパン、ワイン、リンゴの3つすべてを部屋の中で探して手に入れることです。このミッションをクリアしたら何か音が鳴ります。その音を変えてみよう、というのがここでのあなたの挑戦になります。まずはこのステージを実行してみましょう。部屋の中を動き回ってパンなどを探すには上図のジョイパッドを使います。



パン、ワイン、リンゴの3つすべてを見つけると音が鳴ります。この音を変えてみようというのがここでの挑戦です。音を変えるためにプログラムに手を加えるやり方を下に説明します。

- 1) スタートボタンをクリックしてプログラムの実行を停止します。



プログラムが実行中の時はスタートボタンが左図のように見えています。ボタンをクリックするとプログラムは停止します。

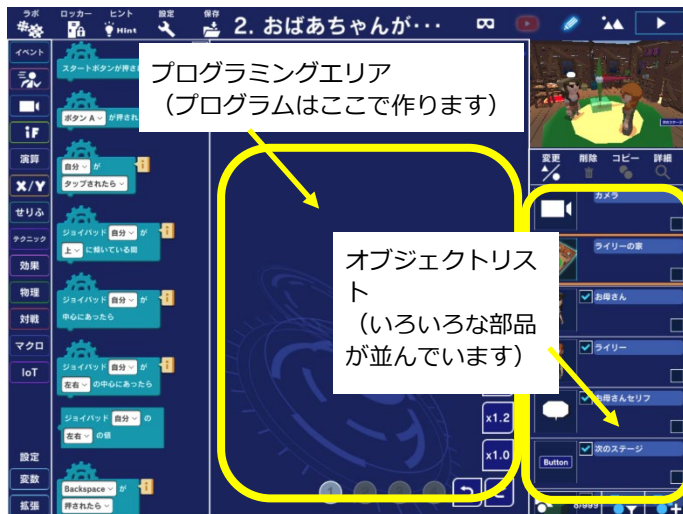


プログラムが停止している時はスタートボタンが左図のように見えています。ボタンをクリックするとプログラムは実行します。

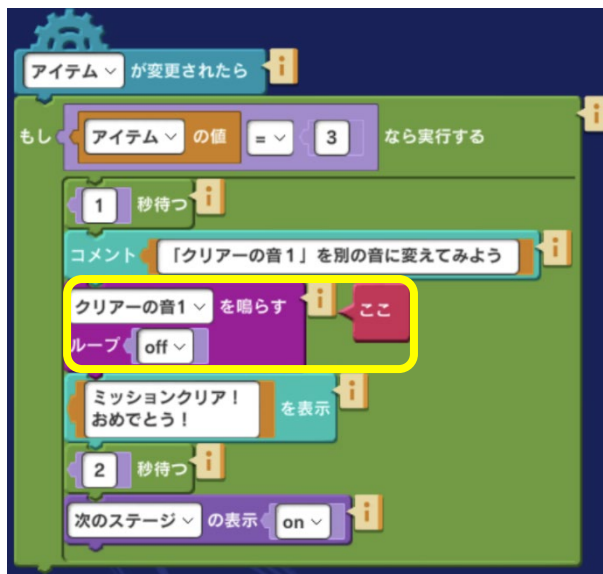
- 2) 鉛筆のボタンをクリックします。



- 3) 下図のような画面となります。オブジェクトリストで「プログラム」というオブジェクトを見つけてクリックします。

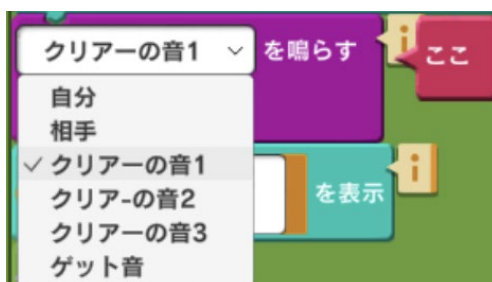


- 4) プログラミングエリアに下図のようなプログラムが表示されます。



図の中央あたりに赤いブロックで「ここ」となっているブロックがあって、それが示している紫色の命令ブロックには「(クリアーの音1)を鳴らす」と表示されています。文字が読みにくい時は赤い「ここ」ブロックを動かすことができます。

- 5) このステージでの挑戦は音を変えることでした。



(クリアーの音 1) のところをクリックすると左図のように他の音を選べるようになります。



3つあるクリアーの音から1つ選ぶ前にまず音を聴いてみたい場合には、オブジェクトリストからクリアーの音を探してアイコンをクリックしてみましょう。どんな音が鳴るか聴いてみるができます。

6) 音を選ぶにはオブジェクトリストから「プログラム」をクリックして、プログラムをもう1度表示させます。

7) 思い通りの音になったかどうかを確認めます。



スタートボタンをクリックしてプログラムを実行します。パン、ワイン、リンゴの3つすべてを見つけた時に鳴る音が自分が指定した音になっていれば成功です。

次のステージ

成功したら「次のステージ」ボタンをクリックして次のステージに進みましょう。

発展課題

プログラム中の「～秒待つ」の数字を小さくすると待つ時間が短くなります。大きくすると待つ時間が長くなります。数字を小さくしたり、大きくしてみましょう。

4. 出発

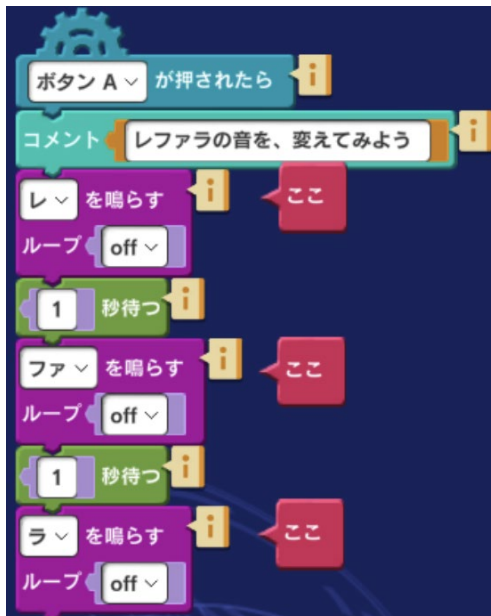
次のステージ「4 出発」はウサギとの出会いのステージです。ウサギと出会うことができたなら「次のステージ」をクリックしてステージ「5 ウサギと出会う」に進みましょう。

5. ウサギと出会う

ステージ「5 ウサギと出会う」ではウサギからのチャレンジを受けてみましょう。ウサギからのチャレンジとはウサギが鳴らした音と同じ音を鳴らすということです。



スタートボタンをクリックしてプログラムの実行を停止します。プログラミング画面が表示されます。



画面が切り替わると左図のようなプログラムが表示されます。

プログラムの中の「レ」、「ファ」、「ラ」のところをクリックすると他の音を選ぶことができます。さきほどステージ「5 ウサギと出会う」の中でウサギが鳴らした3つの音と同じになるように音と順番を選んでください。



うまくできたら左図の「できた！」ボタンをクリックしましょう。

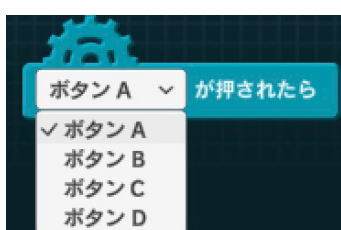


ウサギがこれから先の挑戦に必要なことを教えてくれます。おばあちゃんに届ける薬を作るには3色の薬草（赤、青、黄色）と金の卵が必要です。



「次のステージ」ボタンをクリックして「6 薬草を集めよう」に進みましょう。

発展課題 1



このプログラムではボタン A が押されたら音が鳴りますが、ボタン A をボタン B や C に変更することもできます。左図のようにボタン A のところをクリックしてプルダウンメニューを表示させて好きなボタンを選びます。

発展課題 2

ウサギを他のキャラクターに変更することができます。シーン編集画面（またはプログラム編集画面）に切り替えて下図の操作をしてウサギをネコに変えてみましょう。



①オブジェクトリストで「ウサギ」を選択します。



②「変更」を選択します。



③ネコを選びます。

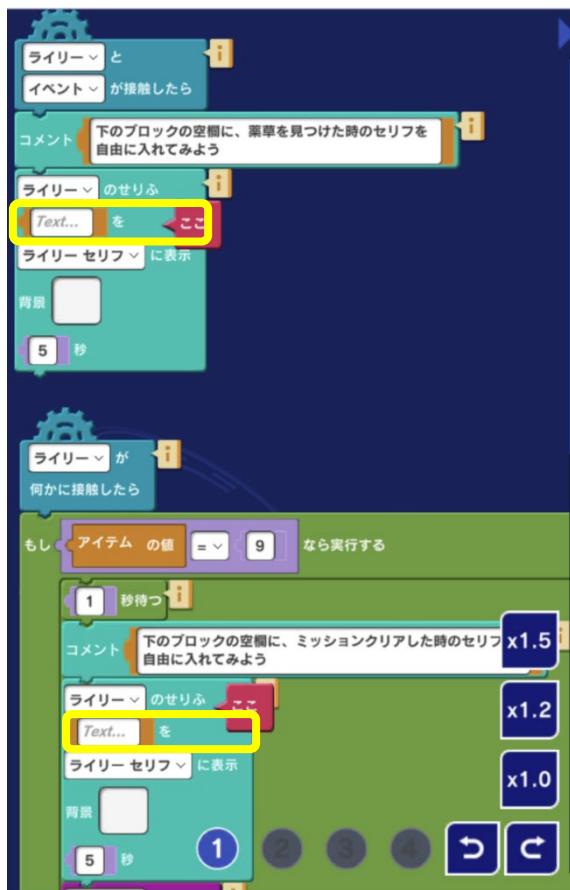
ウサギがネコに変わりました。

6. 薬草を集めよう

ステージ「6 薬草を集めよう」では3色の薬草（赤、青、黄色）を見つけます。もう一つの挑戦として「薬草を見つけた時のセリフを入力する」にチャレンジします。



スタートボタンをクリックしてプログラムの実行を停止します。プログラム編集画面が表示されます。



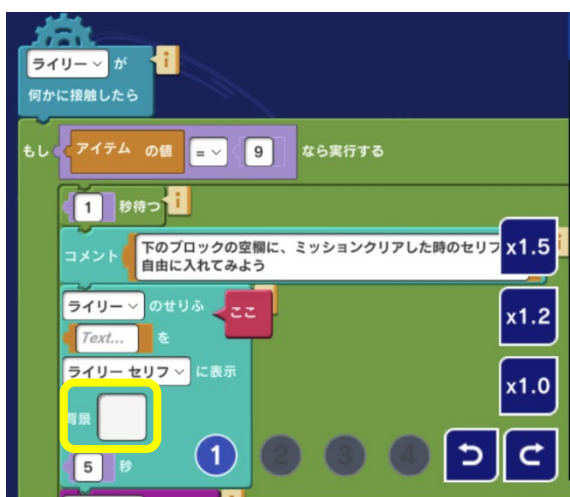
最初のプログラムには「ライリーのせりふ」ブロックのすぐ下の「Text..」となっているところに好きなセリフを入れます。例えば「見つけたよ!」と入力すると、最初に薬草を見つた時に「見つけたよ!」とセリフが表示されます。

2 番目のプログラムにはすべての葉草を必要な数だけ
見つけてミッションをクリアした時に表示するセリフ
を入力します。例えば「やったー！！ミッションをク
リアした！」などです。

次のステージ

うまくできたら「次のステージ」ボタンをクリックして「7 オオカミが！」に進みましょう。

発展課題



せりふの命令にある「背景」の四角をクリックすると色を選択することができます。好きな色に変更してみましょう。

7. オオカミが！



ステージ「7 オオカミが！」ではオオカミが襲ってきます。オオカミから逃げていくと、オオカミを倒してくれるという猟師に出会います。

次のステージ

猟師に出会ったら「次のステージ」ボタンをクリックして先に進みます。

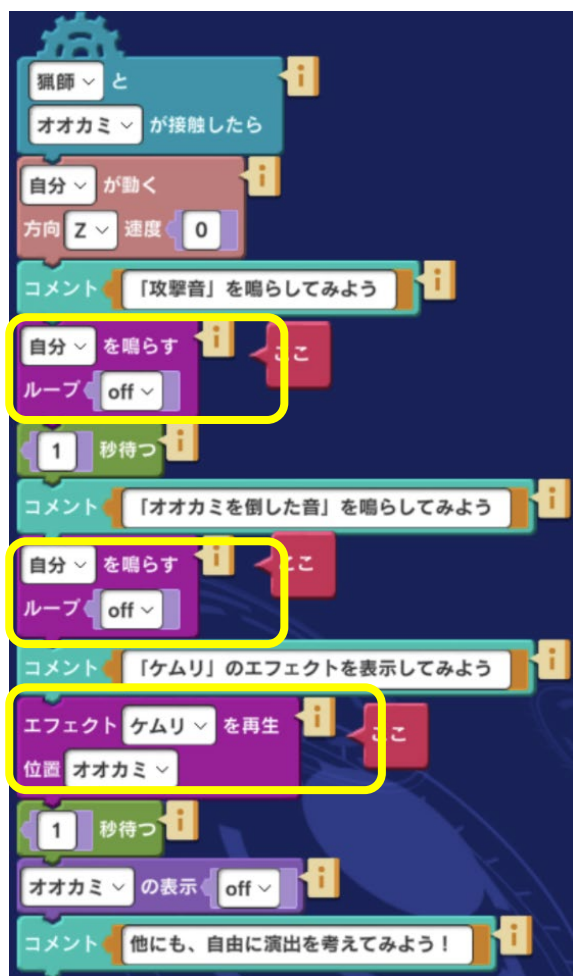
発展課題



上図では狼がどの程度離れたところにいるのかがわかりにくいですね。シーン編集画面で横方向から見た視点に切り替えてみましょう。

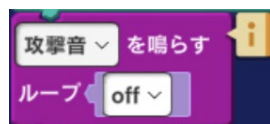
8. 助けて猟師さん

ステージ「8 助けて猟師さん」では音とエフェクトを使ったプログラミングに挑戦します。挑戦する内容は、「猟師とオオカミが接触したら、音やエフェクト出そう！」というものです。

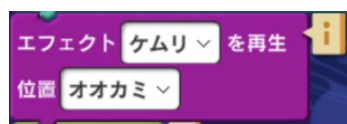


ステージ「3 お見舞いの品」、「6 薬草を集めよう」の説明を参考にして左図のプログラムを表示してみましょう。

「ここ」で示されている3箇所を変えていけばよさそうです。3箇所のそれぞれでどんな音を鳴らせばよいかは、「コメント」にヒントが書かれています。

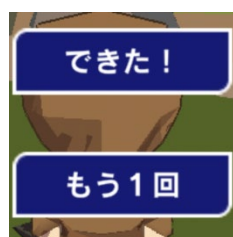


ヒントには「攻撃音を鳴らしてみよう」となっていますので左図のようにすればよいことがわかります。



注意：エフェクトの部分ではどんなエフェクトを表示するかに加えて、エフェクトを「どこで」表示するかを指定する必要があります。ここではケムリとともにオオカミが消えるようにしたいのでエフェクトの位置はオオカミとなります。

正しくプログラムを完成することができたら、自分で好きな音やエフェクトをいろいろ試してみましょう。



うまくいかなかった時は「もう1回」をクリックしてみましょう。自分で作ったプログラムのどこを直せばよいかを調べましょう。

思ったとおりにできたら「できた！」をクリックします。「次のステージ」ボタンが表示されるのでこれをクリックして次のステージに進みます。

発展課題



再生する音は自由に変更することができます。オブジェクト一覧を開いて好きな音を選んで再生してみましょう。

9. 空から何かが



ステージ「9 空から何かが」が始まると、いきなり空から何かが落ちてきて「たすけてー」と叫びます。ジョイパッドを使って行ってみると、落ちてきたのは神様でした。神様はお腹が空いて何か食べ物を分けて欲しいと話しかけてきました。

次のステージ

「次のステージ」をクリックして次のステージ「10 助けたお礼」に進みましょう。

発展課題



ステージ「9 空から何かが」のプログラムを見てみましょう。神様のプログラムの中に「(自分)の重量の影響 (on)」があります。これは重力の影響を有効にすることで、上から落ちてきた神様がその場に留まるようにするためです。この命令の (on) を (off) に変更してみましょう。神様は上から落ちてきて、地面にぶつかったら跳ね返ってそのままどこかへ飛んでいってしまいます。実際に試してみましょう。

10. 助けたお礼

神様が欲しいのはリンゴ、パン、ワインの中のどれかのようにです。3つのボタンを押してどれが神様の欲しいものなのかを見つけましょう。

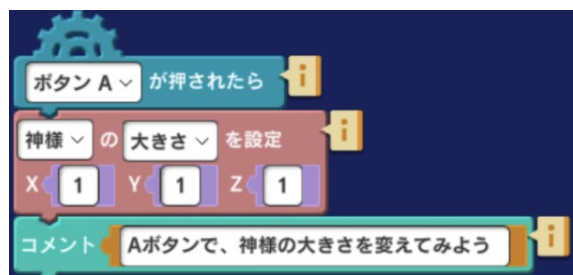


神様の欲しいものがわかったら、次はプログラミングに挑戦しましょう。ステージ「3 お見舞いの品」、「6 薬草を集めよう」の説明を参考にして下図のプログラムを表示してみましょう。

このプログラムを上から見てみるとワインが正解であることがわかります。

ワインを選んだ時の音が「正解の音」のはずなのに左図のプログラムでは「ハズレの音」となっています。また、パンやリンゴを選んだ時は「ハズレの音」を鳴らすはずなのに「正解の音」になっています。

ワイン、パン、リンゴがタップされた時の音をそれぞれ正しい音に変えましょう。



音を正しく変えることができたなら、A ボタンを押して神様の大きさを変えてみましょう。神様に限らずどんなオブジェクトでもプログラムの変更で大きさを変えることができます。



この命令では

X : 幅

Y : 高さ

Z : 奥行き

についてどれだけ大きくするかを指定することができます。例えばのっぽにしたければ（Y : 高さ）だけを大きさ数にすればよいのです。

次のステージ

このステージの最後では神様から金の卵を産むニワトリをもらえます。ニワトリをもらったら「次のステージ」をクリックして「11 金の卵」に進みましょう。

発展課題

上の神様の大きさを指定する説明を参考にして、神様を太らせたり、うんと小さくしたりしてみましょう。

11. 金の卵

このステージではニワトリが金の卵を産みます。

次のステージ

金の卵を手に入れたら「次のステージ」をクリックして「12 おばあちゃんの家」に進みましょう。

発展課題

「次のステージ」ボタンを押した時に何か音が鳴るようにしてみましょう。

12. おばあちゃんの家

ステージ「12 おばあちゃんの家」ではいよいよおばあちゃんの家へ向かいます。ジョイパッドを操作しておばあちゃんの家まで行きましょう。

発展課題



歩く速度を変えることができます。左図の速度の数字を変更することで速くしたり、遅くしたりすることができます。試してみましょう。

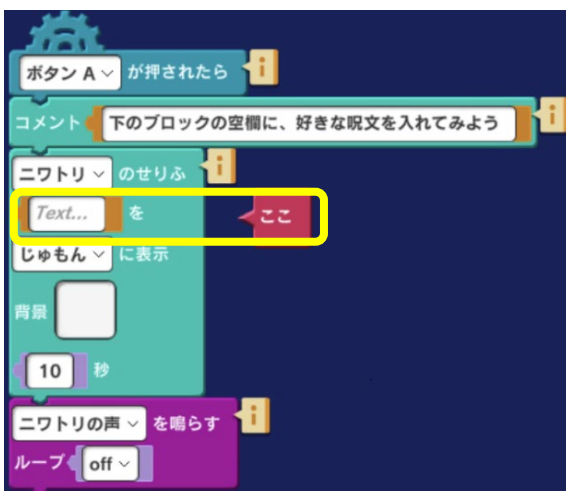


おばあちゃんの家に着いたら「次のステージ」をクリックしてステージ「13 薬を作ろう」に進みましょう。

13. 薬を作ろう



ステージ「13 薬を作ろう」ではニワトリに呪文を言わせるプログラムに挑戦します。ステージ「3 お見舞いの品」、「6 薬草を集めよう」の説明を参考にして下図のプログラムを表示してみましょう。



左図のプログラムで、「ニワトリのせりふ」の下の「Text...」に好きな呪文を入力してみましょう。



例えば「ビビデバビデブー」と入力すれば、A ボタンを押せばニワトリがこの呪文を唱えます。

次のステージ

薬が手に入ったら「次のステージ」をクリックして最後のステージ「14 おしまい」に進みましょう。

発展課題

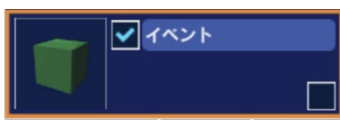
ニワトリが呪文を唱える時にエフェクトを同時に再生してより雰囲気盛り上げてみましょう。どんなエフェクトを入れたらよいか、いろいろなエフェクトを試してみましょう。

14. おしまい

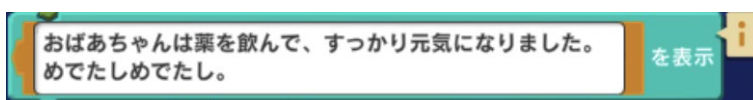


おばあさんは薬のおかげで元気になりました。これでフォレストガールの物語はおしまいです。どうでしたか？プログラムに少し手を入れるだけで物語を自分の好きなように変えることができましたね。

発展課題



上図の画面下に表示されている文字を変更するにはオブジェクトリストにある「イベント」オブジェクトのプログラムを変更します。テキストを変更して楽しんでみましょう。



5. ストーリーを作ろう 2 : スペースボーイ

「スペースボーイ」は、「フォレストガール」と同じようにストーリー展開のある実験室です。画面やプログラムの中の説明に沿って進めて行きましょう。

ところどころでプログラム課題が出されます。「フォレストガール」に比べてやや複雑なものもありますが、ヒントはたくさん置いてあります。必要な命令ブロックやサンプルプログラムが置いてあるので、参考にして作ることができます。



つくる> 「ストーリーを作ろう」 から「スペースボーイ」を選びます。

1. 何か落ちてきたぞ？



物語の始まりです。



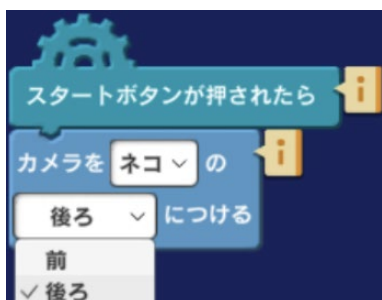
「スタート」ボタンをクリックして、ストーリーを見てみましょう。



アニメーションが終わったら、「次に進む」ボタンをクリックし

て、先に進めます。

発展課題



アニメーションを視点を変えて見てみましょう。カメラオブジェクトの中にあるプログラムでカメラの取り付け位置を変え、どのように見えるかを試してみましょう。

2. 見に行ってみよう



このステージではプログラミングに挑戦します。ジョイパッドを使って猫が落ちてきた玉のところまで歩いていくプログラムを作ります。

プログラムはあらかじめ作ってあるので、「スタート」ボタンをクリックして、ジョイパッドを動かし、猫を赤いエリアまで歩かせてみましょう。

・・・でも、何か変です。猫がまっすぐに歩きま

せん。プログラムを直してみましょう。



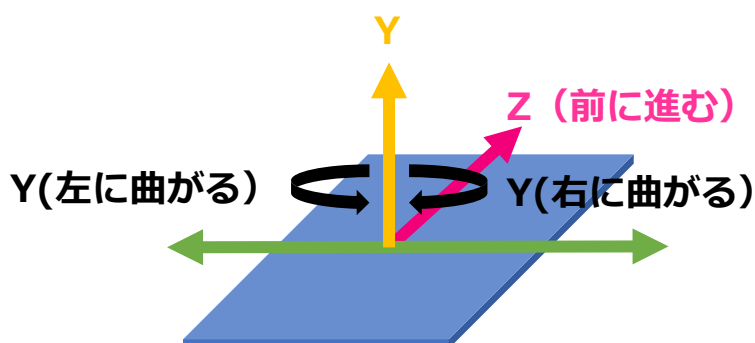
プログラムを作るときは、左図のボタンをクリックして、プログラミング画面を表示します。これ以降のステージでプログラムを作るときも同様です。



ジョイパッドのプログラムを作るので、オブジェクトリストをスクロールして「ジョイパッド」を探します。空白のエリアをクリックして茶色の枠を表示します。



プログラミングエリアに「ジョイパッド」のプログラムが表示されます。「ここ」と書いてある赤いブロックがヒントです。



Mind Render では前に進むときは「Z」、曲がる時（左右に方向を変える時）は「Y」軸を選ぶ必要があります。

Mind Render では 3D のプログラムを作るので、X（左右）、Y（上下）、Z（前後）の 3 つの軸で方向を指定しますが、ここではまず「前進は Z 方向」「左右の回転は Y 軸」と覚えておきましょう。

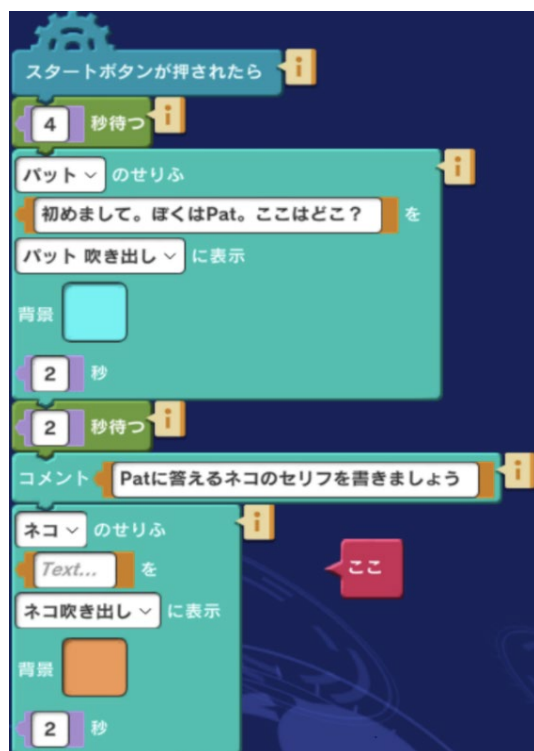
プログラムをよく見て、ネコが前に進めるようにプログラムを直してみましょう。



プログラムが出来たら、動作を確認してみましょう。スタートボタンをクリックして、プログラムを実行します。

ネコが赤いエリアまで来ると、玉の中から小さな少年が現われます。うまくできたら、「次に進む」ボタンをクリックして、次に進みます。

発展課題



ネコと玉の位置関係を知るために上から見てみましょう。シーン編集画面にしてオブジェクトリストのネコの画像をクリックします。

画面が切り替わり、ネコのすぐ後ろから見た景色にかわります。

操作方法は、[カメラ操作](#)を参照してください。

3. 初めまして

ネコと少年が挨拶を交わします。少年のセリフしか入っていないので、ネコのセリフを入れて、二人が会話をするプログラムを完成させましょう。



プログラムは「ネコ」の中に作ります。
少年のセリフを読み、「ここ」と書かれた赤いブロックのネコのセリフを考えます。
灰色で「Text・・・」と書かれている空欄をタップすると、セリフを入力できます。

ネコのセリフは2ヶ所あります。両方ともにセリフを入れましょう。

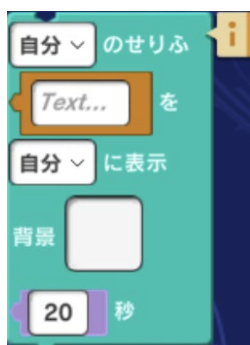
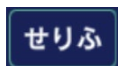
プログラムが出来たら、会話を見てみましょう。スタートボタンをクリックします。

うまくできたら、「次に進む」ボタンをクリックして、次へ進みます。

発展課題

命令を追加して、もっといろいろ話してみしましょう。命令を追加する手順は以下の通りです。

- 1) 命令カテゴリーで、「せりふ」をタップします。



- 2) 命令を下の方にスクロールします。下図の命令をプログラミングエリアにドラッグ&ドロップします。



- 3) 命令をプログラムに挿入します。例えば、繰り返し命令の前に挿入します。

- 4) せりふ命令ブロックにせりふを入力します。他の命令を参考に、誰のせりふか、何秒表示するかなどを指定します。

4. 町へ行こう



二人は町へ行くことになりました。ジョイパッドを操作して、町へ行きましょう。前方に見えている橋が町の入口です。



橋を渡り切ったところにある緑色のエリアまで行きます。

到着したら、「次に進む」ボタンをクリックして、次へ進みます。

発展課題



町への入り口は他にもあります。壁沿いにずっと歩いていってみましょう。少年とネコが通り抜けることのできる隙間が見つかります。入ってみましょう。

注意：この隙間から町に入った場合は「次に進む」ボタンが表示されないために次のステージに進むことができません。

次のステージに進むにはラボ「4 町へ行こう」をもう一度やり直して町の入り口から町に入ってください。

5. 大変だ！



町に到着すると、海賊たちがやってきました。パットとネコはどうなってしまっているのでしょうか。見てみましょう。

アニメーションが終わったら、「次に進む」ボタンをクリックして、次へ進みます。

発展課題



このラボで登場する海賊は同じ海賊のキャラクターの X, Y, Z 方向の大きさを変えることで大きさや縦横の比率の違う海賊を作っています。

シーン編集画面で X, Y, Z の大きさを変えて巨大な海賊や小さな海賊などに変更してみましょう。

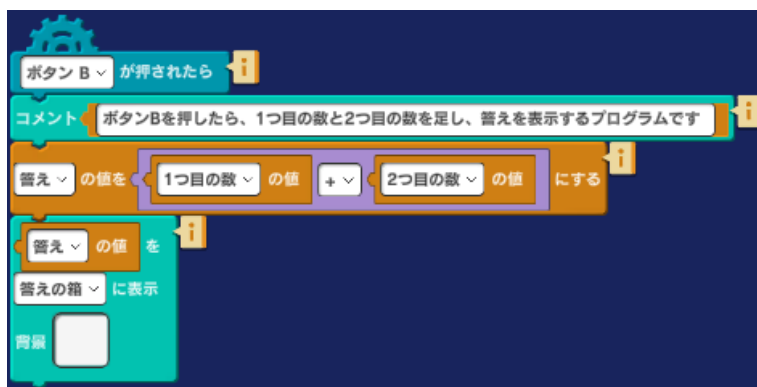
6. 魔法を解くには・・・

パットたちにかけられた海賊の魔法を解くにはプログラミング課題を解かなければなりません。このステージでは、どんなプログラムを作ったらよいかサンプルが表示されます。次のステージで実際にプログラムを作ってみます。



プログラミング課題は、ランダムに選ばれる 2 つの 2 桁の数を足し算するプログラムを作成することです。

プログラムは、「計算機」の中に作成します。



「計算機」の中を見ると、左図のようなプログラムが入っています。

「答え」、「1つ目の数」、「2つ目の数」は変数です。変数とは、中身を入れ替えたり、中身に数を足したり引いたりすることができる便利なしくみです。

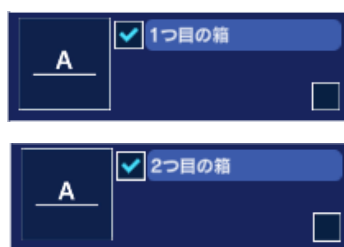


変数は、自分で名前を付けて作ります。

新しい変数を作るときは、画面左下、命令カテゴリーの下の方にある「変数」をクリックします。



このラボでは、変数がすでに作成されているので、新たに作成する必要はありません。右上の「閉じる」ボタンをクリックして画面を閉じます。



上のプログラムから、変数「答え」は、1つ目の数と2つ目の数の合計であることがわかります。

では、変数「1つ目の数」と「2つ目の数」にはどんな数が入るのでしょうか。オブジェクトリストの「1つ目の箱」「2つ目の箱」で確認できます。



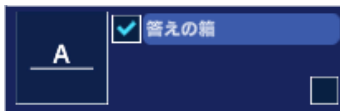
「1つ目の箱」は、画面上に1つ目の数を表示するための箱です。

「1つ目の箱」には左図のようなプログラムが入っています。

スタートボタンが押されたら、1つ目の数を「0」にします。また、箱の中に表示されていた古い数字をクリアします。

ボタン A が押されたら、1つ目の数には10～99までのいずれかの数が設定されます。そして、箱の中に表示されます。

「2つ目の箱」のプログラムも同様です。



「答えの箱」のプログラムも見ておきましょう。



スタートボタンが押されたら、答えの数を「0」にします。また、箱の中に表示されていた数字をクリアします。

プログラムと動作を確認したら、「次に進む」ボタンで次へ進みます。

7. 魔法を解こう



前のステージ「6 大変だ！」を参考に、ランダムに表示される2つの数を足し算するプログラムを作成しましょう。プログラムは「計算機」の中に作ります。

前のステージをもう一度確認したいときは、「ラボ選択」ボタンで、「6 大変だ！」を選択します。

このとき、作成途中のプログラムが失われないよう

に、「保存」ボタンを押して、保存しましょう。

プログラムがうまくできたら、「次に進む」ボタンで次へ進みましょう。

発展課題

ラボ「6 大変だ！」と「7 魔法を解くには」ではAボタンを押すたびに異なる2つの2桁の数が表示されました。このように毎回異なる数を作って表示するためには乱数という命令を使います。乱数はたとえば10～99のように範囲を決めて、その範囲の中からサイコロを振って決めるように1つの数を選び出す命令です。



例えば左図は足し算をするための2つの数の1番目の数を作って表示するプログラムです。この中で「乱数(10)から(99)」とあるのが乱数を作る命令です。この命令では10から99の範囲の中から数を1つ選び出してくれます。命令の中の数を変えることで乱数を作る範囲を変えることができます。

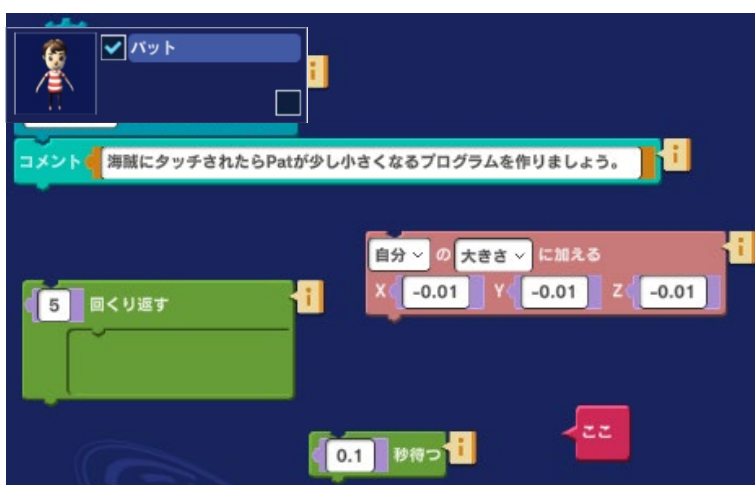


3桁どうしの計算をするためにはどうしたらよいでしょうか？やってみてください。

8. 逃げろ！



魔法がとけて、パットとネコが再び動けるようになりました。海賊が追いかけてくるので、急いで逃げましょう。パットとネコはジョイパッドで動かします。



海賊にタッチされるとパットが小さくなってしまいう演出を加えましょう。プログラムは「パット」の中に作ります。

「ここ」と書かれている赤いブロックを探します。

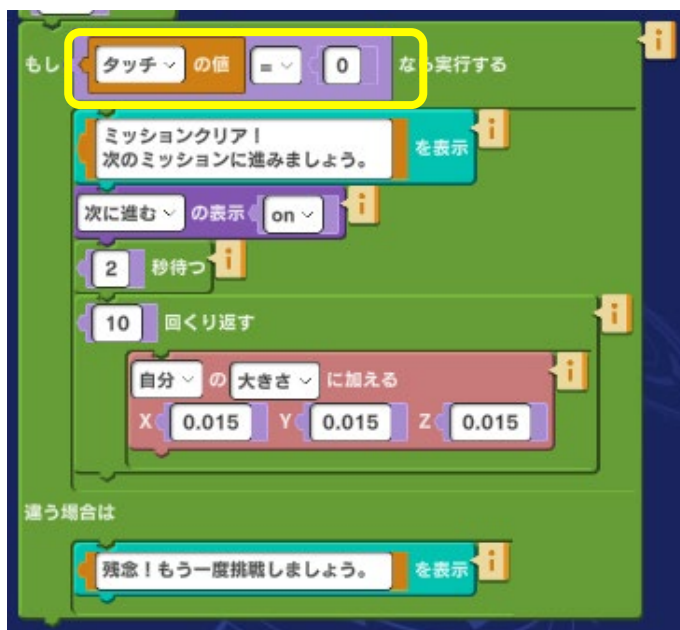
用意されている3つの命令ブロックを並べて、パットが少しずつ小さくなるようにします。

「-0.01」や「5回」などの値を変えるとどうなるか試してみましょう。

うまく逃げられたら、「次に進む」ボタンが表示されます。次へ進みましょう。

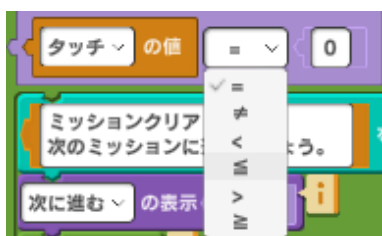
発展課題

このラボをクリアするためには30秒の間一度も海賊にタッチされないようにしなければなりません。この条件がちょっと厳しすぎると感じる場合には条件を変えることができます。



条件を設定しているプログラムはオブジェクトリストの中にある「残り時間」オブジェクトにあります。

左図が条件を設定している部分です。「タッチ」の値（タッチの回数）が 0 に等しいとなっています。



この命令の等号をクリックすると他のオプションが表示されます。



例えば、タッチの値が 3 より小さければとすることで 2 回まではタッチされてもよいという条件に変更することができます。「5 以下」、「2 以下」など自分で条件を設定してみましょう。

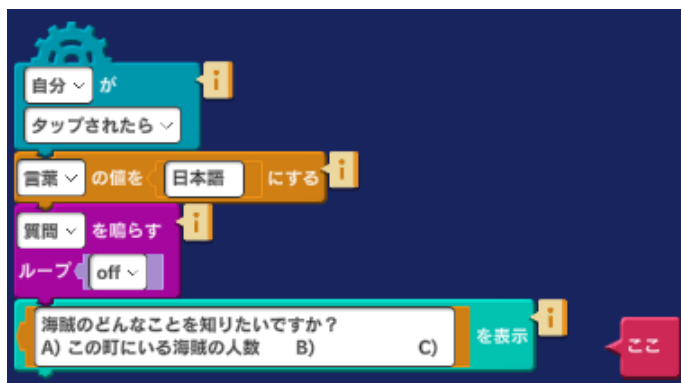
9. 海賊のことを知ろう



このステージでは、海賊について質問し、回答するプログラムを作成します。



質問を出すプログラムは「質問(日本語)」の中に作ります。



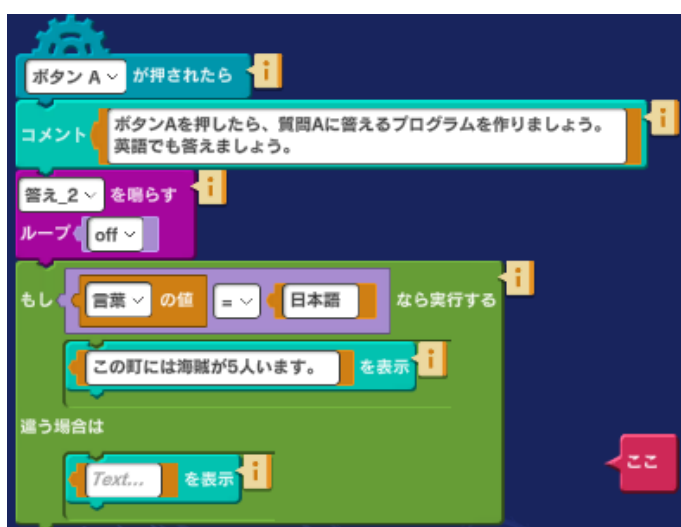
例として、海賊の人数をたずねる質問 A が書かれています。

質問 B、C を考えて書きましょう。

もちろん質問 A の内容を変えることもできます。



答えのプログラムはオブジェクト「答え」の中に作ります。



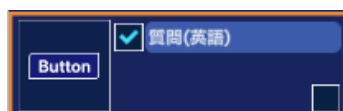
左図は、質問 A の答えを表示するプログラム例です。ボタン A を押したら質問 A の答えが表示されます。

答えの内容は自由に変更できます。

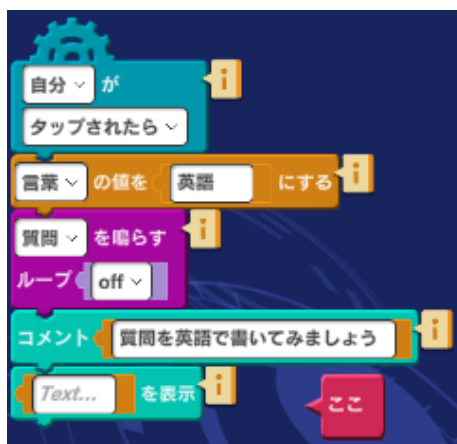
同じように、ボタン B でを押したら質問 B の答え、ボタン C を押したら質問 C の答えが表示されるようにプログラムを作ってみましょう。

発展課題

日本語で質問されたら日本語で答え、英語で質問されたら英語で答えるようにプログラムを変更してみましょう。



英語の質問は、「質問（英語）」の中に書きます。



左図は、質問（英語）の中のプログラムです。

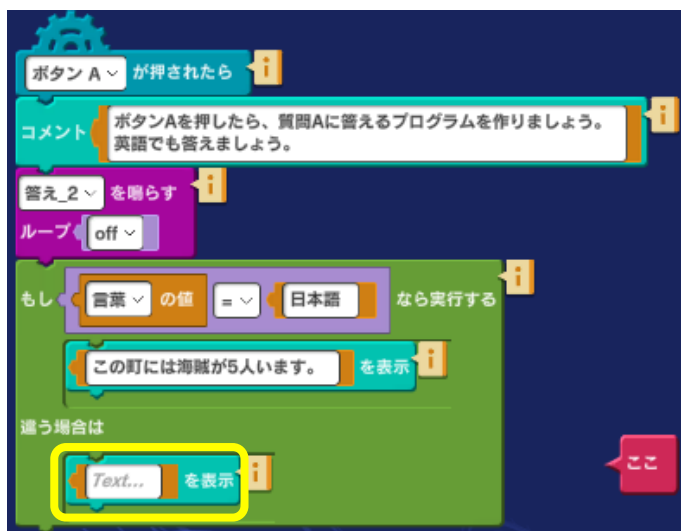
質問を英語で「Text…」と書かれている部分に入力します。

「質問(英語)」ボタンがクリックされると、変数「言葉」の値が「英語」になります。

これにより、答えを日本語で表示するか、英語で表示するかを判定します。



答えは、オブジェクト「答え」の中に書きます。



左図は、オブジェクト「答え」の中のプログラムです。

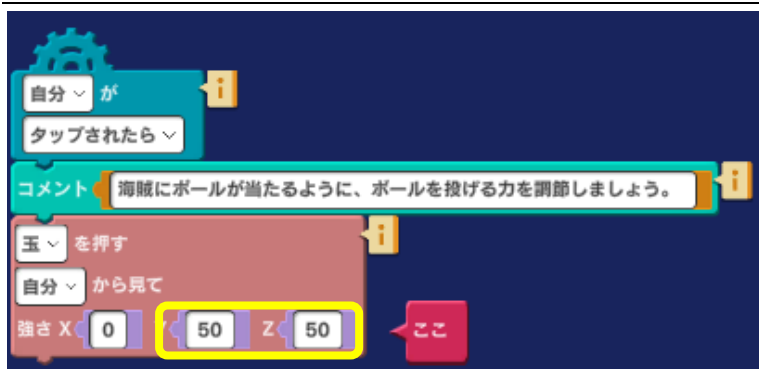
質問の答えを英語で「Text…」と書かれている部分に入力します。

「もし～」の命令によって、変数「言葉」の値が日本語なら日本語の答えが、日本語でなければ（＝英語）、英語の答えが表示されます。

10. 海賊を倒せ



最後のステージです。ここでは、海賊にボールを当てて反撃しましょう。時間内にすべての海賊に当たるとミッションクリアです。



オブジェクトリストの「発射 1」「発射 2」「発射 3」の中にあるプログラムでボールの飛距離を調整できます。

左図は、「発射 1」のプログラムです。

ボールは前方斜め上に飛ばしたいので、Y(下から加える力)と Z (後ろから加え

る力) に値を入力します。ボールの飛距離は、加える力の大きさによって変わります。

「『自分』から見て」とは、オブジェクト（この場合はボール）を基準にして、オブジェクトの横や後ろから力を加える指定方法です。

「発射 2」「発射 3」のプログラムも同様です。

これでスペースボーイの物語はおしまいです。どうでしたか？ストーリーを構成するプログラムは全てブロック命令で作成してあります。自分で自由にプログラムを変えて、もっと面白い冒険ストーリーを作ってみましょう。

発展課題

プログラミングに慣れてきたら、課題以外のプログラムも見てください。

プログラムを変更して、オリジナルストーリーを作ってみましょう。例えば、

- ・登場人物のキャラクターを入れ替えてみよう

注意：キャラクターの大きさ調整が必要となる場合があります。

- ・エフェクトや音を入れ替えてみよう
- ・自分で自由にストーリーを作り変えてみよう

6. ストーリーを作ろう 3：「探偵ストーリー」

ラボ「探偵ストーリー」は、「フォレストガール」や「スペースボーイ」と同じようにストーリー展開のある実験室です。

ある日、探偵のもとに依頼の手紙が届きます。依頼内容は「秘宝を守って欲しい」というものでした。探偵は無事秘宝を守ることができるでしょうか？そして秘宝とは何なのでしょう・・・？謎解きとともに進めて行きましょう。

登場人物



ラボ実験室メニューから「探偵ストーリー」を選びます。

1. 依頼の手紙



「スタート」ボタンをクリックして、ストーリーを見てみましょう。



探偵が依頼主の話を聞くために、依頼主マリーの家へ向かいます。マリーの家はもみの木が目印です。ジョイパッドを操作して探偵をマリーの家（地面が緑色になっている場所）まで歩かせてみましょう。マリーの家の前に来たらミッションクリアです。

次に進む

「次に進む」ボタンをクリックして、次のステージに進みます。

発展課題

ジョイパッドのプログラムを見て、歩くスピードを変えてみましょう。

2. 予告状



マリーの家で依頼の詳細を聞きましょう。

アニメーションが終わったら、「次に進む」ボタンをクリックして、次に進みます。

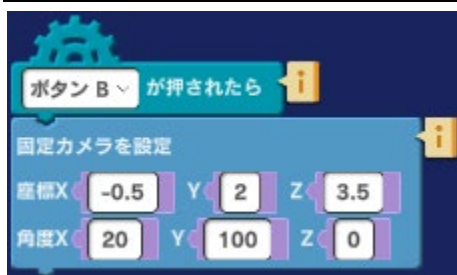
発展課題



マリーが秘宝について話をしている間、リングにカメラを向けるようにするにはどうしたらよいでしょうか？例えば、B ボタンを押すと左図のような視点に切り替わるようにしてみましょう。

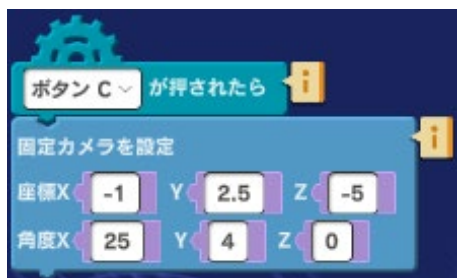


シーン編集画面で、左図のようなアングルになるように調整します。カメラの操作は、「カメラ操作」を参考にしてください。



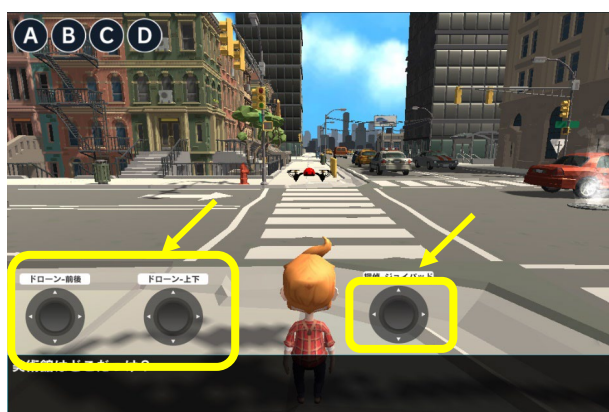
調整できたら、画面左上の座標と角度の値を、ブロック命令に入力します。

プログラム編集画面で、「カメラ」オブジェクトの中に左図のし、値を入力します。（数字は違っていても構いません。）マリーのセリフに合わせて、ボタン B を押してみましょう。



元のカメラ視点に戻すには左図のようなプログラムも作っておくと便利です。

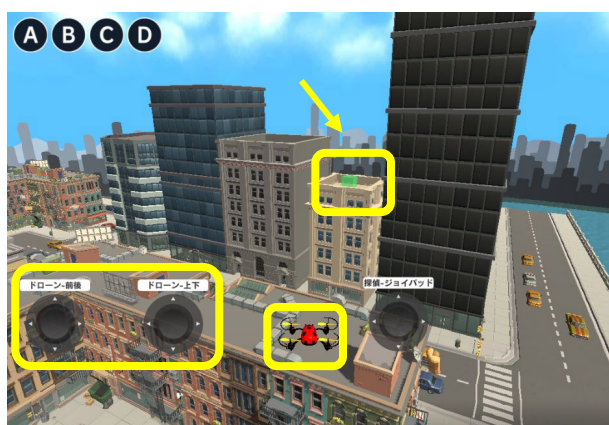
3. 美術館へ行こう



探偵がマリーから教えてもらった秘宝が展示されている美術館に行きます。このステージでは課題が2つあります。

1. ドローンで美術館の屋上に行く
2. 探偵が美術館に行く

1. ドローンで美術館の屋上に行く



画面中央に赤いドローンがあります。ドローンを操縦するジョイパッドは2つあります。美術館の建物の屋上には、目標となる緑色の箱が置いてあります。ジョイパッドでドローンを操縦して、緑色の箱を探しましょう。



ビルの屋上に緑の箱が見えてきました。

ドローンが緑の箱にタッチしたら、1つ目の課題クリアです。

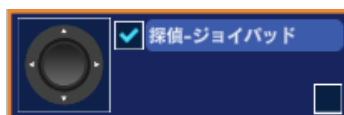
タッチすると、次の課題が表示されます。

次の課題は、ジョイパッドのプログラムを作って、探偵を美術館まで連れて行くことです。

2. 探偵が美術館に行く



プログラミング画面を表示します。



プログラムは、「探偵-ジョイパッド」の中に作ります。



必要なブロックはすでに置いてあります。ドローンのジョイパッドを参考に作ってみましょう。

前後の移動はZ、左右の回転はYを指定します。

左図はプログラムの一例です。



ジョイパッドを操作して美術館の前まで行きましょう。

美術館の前には、目印となる緑色のエリアがあります。

うまくできたら、「次に進む」ボタンをクリックして、次へ進みます。

発展課題

広い町の中を歩き回っていると目印に近づいているのか、それとも遠ざかっているのかがわからなくなることがあります。探偵と目印の間の距離を表示させることができれば便利ですね。



左図はプログラムの例です。このプログラムでは目印の X 座標と探偵の X 座標の差を常に表示します。差が小さくなる方向に動けば X 方向で、目印までの距離が短くなるとわかります。



上図のプログラムを実行すると左図のように表示されます。上図のプログラムを参考にして、Z 方向の距離の差も表示するようにしてみましょう。

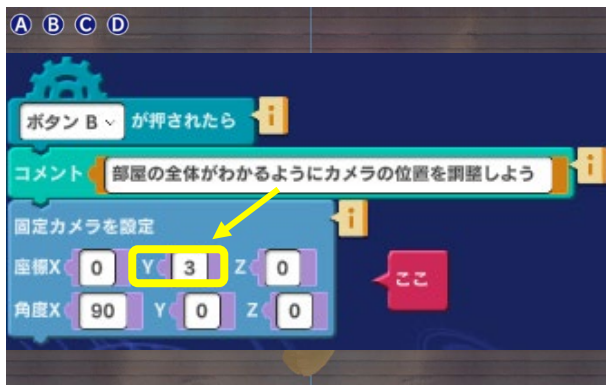
4. 秘宝が！



美術館の秘宝展示室にやってきました。警備員に話を聞いてみます。

しかし、警備員と探偵の目の前で秘宝が消えてしまいます。一体何が起きたのか、部屋の様子がよくわかるように、カメラ映像を切り替えてみましょう。

B B ボタンを押すとカメラが切り替わり、天井からの映像になります。

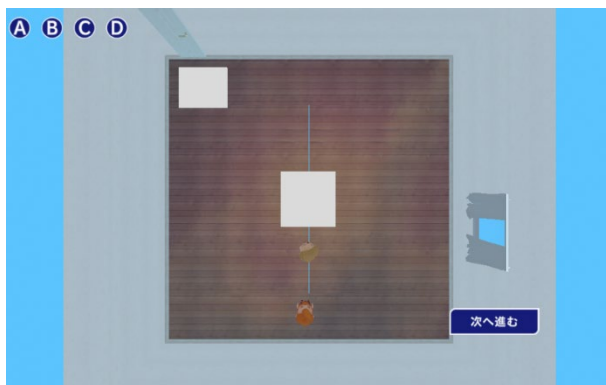


しかし、部屋の一部しか映っていません。部屋全体が見えるように、カメラのプログラムを調整します。



カメラのプログラムを表示します。

部屋全体を映すには、カメラをどこに置いたらよいか、色々試してみましよう。



例えば、このように見えるようにしてみます。

カメラの高さを変えるには、上下の軸である Y 軸の座標を変えます。

そのほかにも数字をいろいろ変えて、見え方がどう変わるか試してみましょう。

うまくできたら、「次に進む」ボタンをクリックして、次へ進みます。

発展課題

C ボタン、D ボタンを使って警備員の視点や探偵の視点に切り替えられるようにするにはどうすればよいでしょうか？ プログラムを作って視点切り替えができるようにしてみましょう。

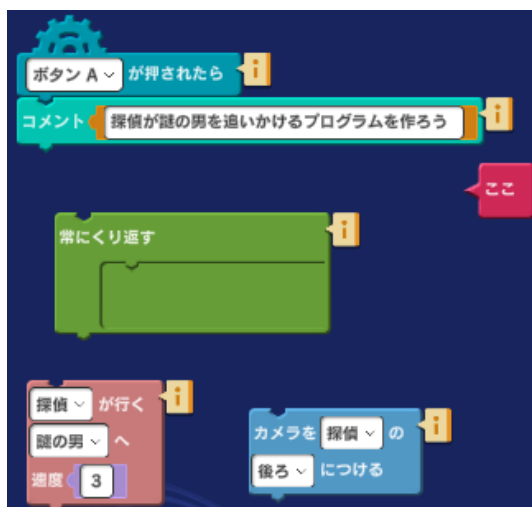
5. 謎の男



美術館から逃げていく男がいました。男を追いかけるプログラムを作りましょう。



プログラムは「探偵」の中に作ります。



使用するブロックが置いてあるので、組み合わせて作ります。

男から少し遅れて追尾できるように、速度に注意しましょう。

うまくできたら、「次に進む」ボタンをクリックして、次へ進みます。

発展課題

ラボ「3 美術館へ移行」の発展課題を参考にして、逃げていく男とドローンの距離の変化を表示するプログラムを作ってみましょう。

6. 逃げた理由



男に追いついた探偵は、男から事情を聞き出そうとします。男に話してもらうには、じゃんけんで勝たなければなりません。

A ボタンを押したら、探偵が必ず勝つじゃんけんプログラムを作りましょう。



プログラムは、「じゃんけんプログラム」の中に作ります。

じゃんけんプログラムの 1～4 ページに以下のようなプログラムが入っています。

1

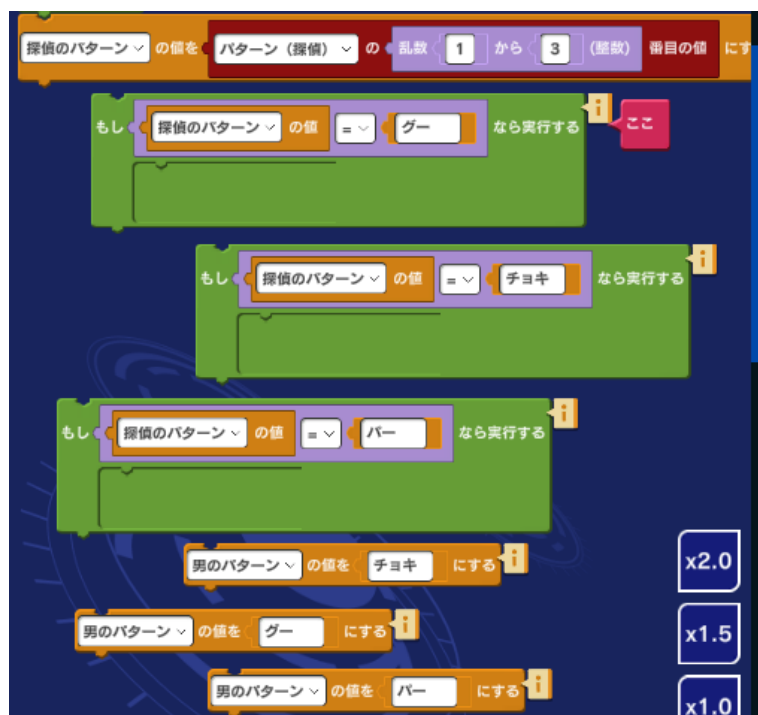
2～4 ページで共通に使用するリスト、マクロ、プログラム

(2～4 のプログラムが動作しなくなる場合があるため、変更しないでください。)

2 A ボタンを押したら探偵が必ず勝つプログラム（未完成）
プログラムを完成させましょう。

3 B ボタンを押したら勝敗がランダムに決まるプログラム（完成）
B ボタンを押すと動作します。

4 3 回勝負で勝敗を決定するプログラム（完成）
C ボタンを押すと動作します。



左図は2ページ目に入っているプログラムの一部です。探偵が常に勝つようにグー、チョキ、パーを組み合わせましょう。

3ページ（B ボタン）、4ページ（C ボタン）のプログラムも試してみましょう。

うまくできたら、「次に進む」ボタンをクリックして、次へ進みます。

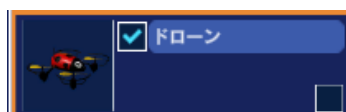
発展課題

「じゃんけんプログラム」の A ボタンが押された時のプログラムを参考にして、D ボタンが押された時に探偵が必ず負けるプログラムを作ってみましょう。

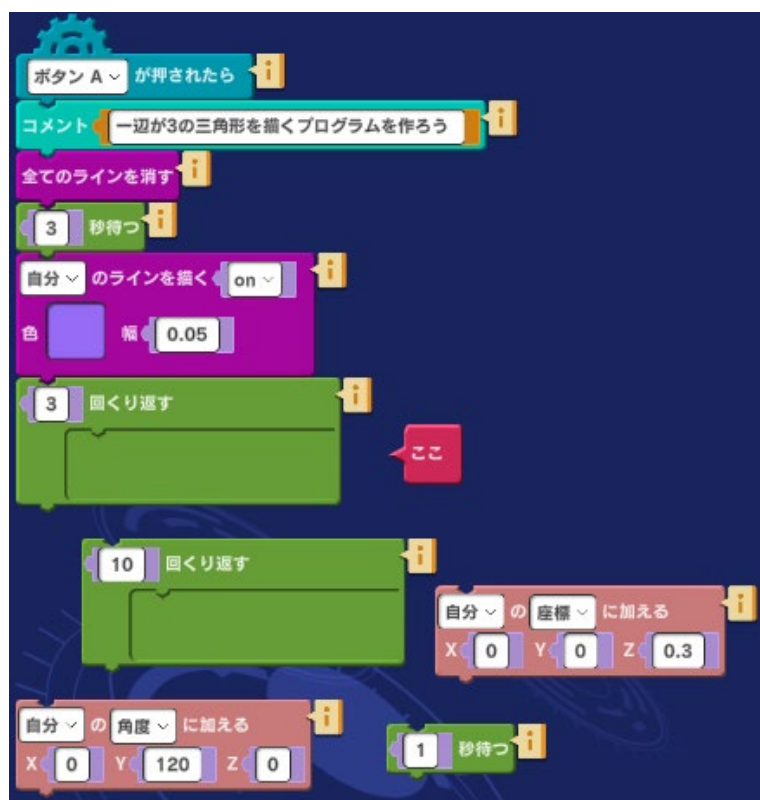
7. 真犯人?



男から聞き出した怪盗のボスはどこにいますでしょうか？ウサギが何か知っているようです。情報をもらうため、ウサギが要求する図形を描くプログラムを作ります。



プログラムは「ドローン」の中に作ります。



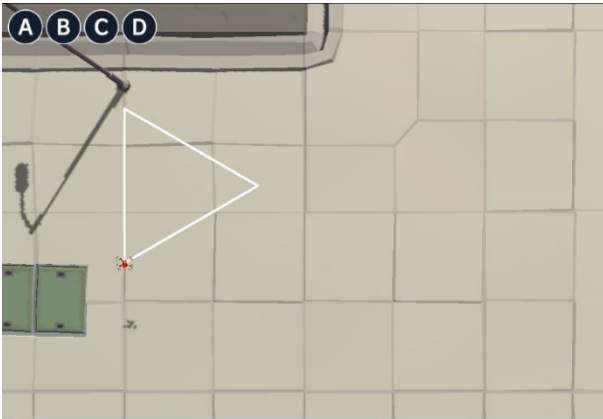
ドローンをアニメーションのように少しずつ動かして線を描くようにします。

うまく描けたら、B ボタンを押してウサギから怪盗のボスに関する情報を教えてもらいましょう。



ボスがいるらしい公園にやってきました。

「次へ進む」ボタンをクリックして、次に進みます。

発展課題

ドローンが描く図形を正六角形、正八角形など別の図形も試してみましょう。

また、円を描くにはどうしたらよいでしょう？

8. 本物の秘宝は・・・

怪盗のボスと対決です。怪盗のボスが、本物の秘宝を偽物（分身）の中に隠してしまいます。秘宝を取り戻すには、偽物の中から本物を探し当てなければなりません。



本物を見つけるプログラムを作りましょう。「秘宝」の中に作ります。



秘宝が偽物（分身）の場合は不正解音が鳴り、本物の場合は正解音が鳴るプログラムを作ります。

音が鳴る部分を作ったら、最後に、「2秒待つ」「降参」という命令を接続して完成です。

「降参」という命令は複数の命令を使って新しく作った命令です。このような命令をマクロと呼びます。



マクロによって実際に処理する内容は、オブジェクト「怪盗」の中にあるプログラムで定義されています。

本物が見破られた後の怪盗のせりふや動作が定義されています。

うまくできたら、「次に進む」ボタンをクリックして、次へ進みます。

発展課題

「探偵がリングに触れるのは3回まで」などのルールをプログラムに取り入れるにはどうしたらよいでしょうか？「秘宝」プログラムを変更して作ってみましょう。

9. 秘宝の秘密！



秘宝を取り戻した探偵は、再び依頼主マリーを訪れます。そこで探偵は秘宝に隠された秘密を教えてもらいます。

これで探偵ストーリーの物語はおしまいです。どうでしたか？ストーリーを構成するプログラムは全てブロック命令で作成してあります。自分で自由にプログラムを変えて、もっと面白い探偵ストーリーを作ってみましょう。

発展課題

プログラミングに慣れてきたら、課題以外のプログラムも見てみましょう。

プログラムを変更して、オリジナルストーリーを作ってみましょう。例えば、

- ・登場人物のキャラクターを入れ替えてみよう

注意：キャラクターの大きさ調整が必要となる場合があります。

- ・エフェクトや音を入れ替えてみよう
- ・自分で自由にストーリーを作り変えてみよう

7. ゲームを作ろう

「フォレストガール」などのストーリーは物語を楽しむコンテンツでした。Mind Render を使うと物語以外にも いろいろなコンテンツを作ることができます。「ゲームを作ろう」には Mind Render で作ったゲームが入っています。

ラボから「ゲームを作ろう」選んで最初のゲームを見てみましょう「ゲームを作ろう」には Mind Render で作ったゲームが入っています。どのゲームもそのまま遊べるようになっています。ここでは、最初の 2 つのゲームを遊んでみましょう。



「スタート」ボタンをクリックします。

1. 宇宙船着陸

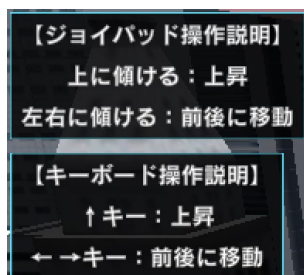


1 つ目のゲームは小型の宇宙船をビルの上から飛ばして、隣のビルの上に着陸させるゲームです。シンプルなゲームですが、慣れないとなかなかうまくいかないため夢中になって遊べます。

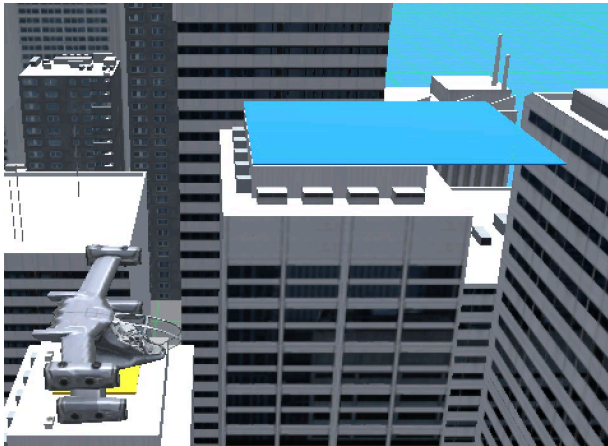


「スタート」ボタンを押すと左図のように難易度や操作方法の説明が表示されます。「簡単」から始めてみましょう。

難易度を選ぶとゲームが始まります。ゲームは宇宙船をジョイパッドまたはキーボードの操作で隣のビルの上にある青いエリアまで飛ばして着陸させることです。慣れるまで何度もやってみましょう。



宇宙船を操作するには、パソコンの場合はキーボードまたはジョイパッド、タブレットやスマートフォンの場合はジョイパッドを使います。



「簡単」レベルをクリアしたら、「普通」「難しい」レベルにも挑戦してみてください。「難しい」レベルではスタート地点と赤い屋根との間にはわずかなすきましかないので簡単には着陸することができません。楽しみながら成功するまで挑戦してみてください。

満足するまで遊んだら、プログラムに少し手を加えてもっと面白くなるようにしてみましょう。例えば、宇宙船の大きさを小さくして制御しやすくするには次のようにします。



まず、画面右上のボタンをクリックして、シーン編集画面を表示します。シーン編集画面では、オブジェクトの位置や大きさなどを変更することができます。

宇宙船			
座標	57.00	24.50	40.00
角度	0.00	0.00	0.00
大きさ	1.00	1.00	1.00

オブジェクトで宇宙船をクリックします。画面右上に宇宙船の座標、角度、大きさが表示されるので、「大きさ」の数字を変えます。例えば x、y、z それぞれ 0.2 にしたらどの程度小さくなるかやってみましょう。



(左) $x(1), y(1), z(1)$

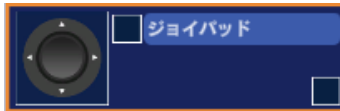
(右) $x(0.2), y(0.2), z(0.2)$

発展課題

宇宙船の噴射力を変えるにはどうすればよいでしょうか？宇宙船を操作するプログラムは「ジョイパッド」の中に作成されています。



プログラムを編集する場合は、画面右上のボタンをクリックして、プログラム編集画を表示します。

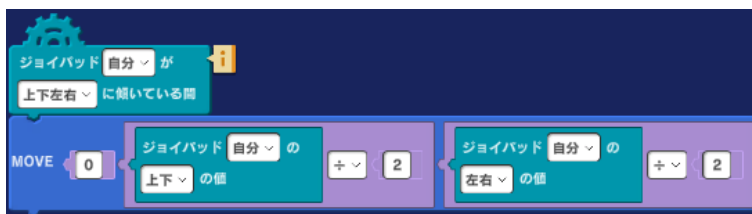


オブジェクトリストでジョイパッドをクリックします。



ジョイパッドで操作する場合は、左図のプログラムを変更します。

宇宙船の噴射力はジョイパッドをどれだけ上下および左右方向に動かしたかで決まります。



この値を小さくして噴射力を小さくしてみます。

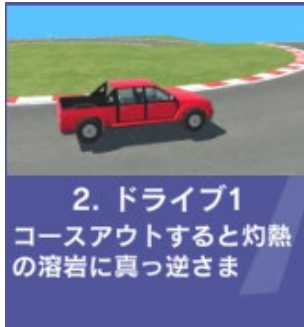
ここで割る数をあまり大きくすると、噴射力が小さくなり過ぎて宇宙船は重力に逆らって上向きに飛ぶことができ

なくなります。いろいろな数で試してみましょう。



キーボードで操作する場合は、左図のプログラムで数値を変更します。

2. ドライブ 1



「ドライブ 1」で遊んでみましょう。「OK」ボタンをクリックしてゲームを試してみてください。



説明を読んだら「OK」をクリックします。



「簡単」「普通」「難しい」の3つのレベルのどれかを選びます。



自動車がコースから外れないようにジョイパッドまたはキーボードの矢印キーで操作します。ブルーの橋を落ちないようにコースを周回すればクリアとなります。

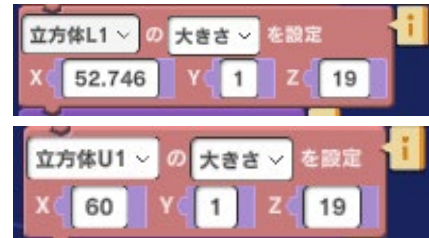
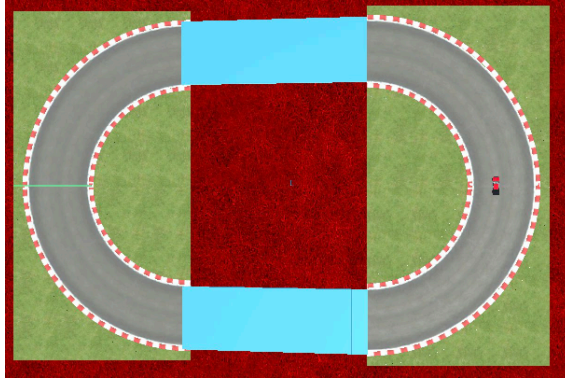
橋から落ちると自動車は爆発してしまいます。

「簡単」レベルをクリアできたら、「普通」「難しい」レベルも攻略してみましょう。

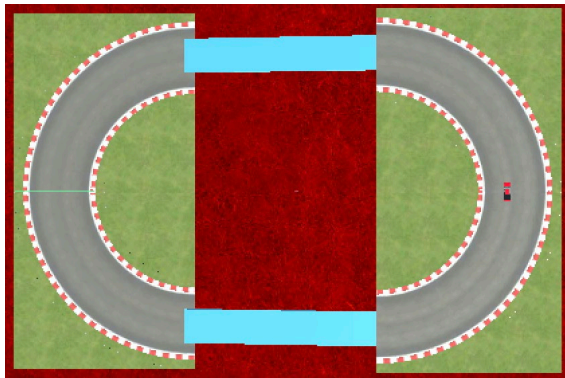
遊び終わったらさらにゲームを面白くしてみましょう。このゲームの難しくて面白いところは橋の場所や橋をかける場所が変えられるところです。

例として橋の大きさを変えるやり方を見てみましょう。橋の大きさやかける場所などはオブジェクトリストの「簡単」、「普通」、「難しい」の命令ブロックに見つかります。下にそれぞれのレベルのコースを上から見た図と対応するプログラムを示します。

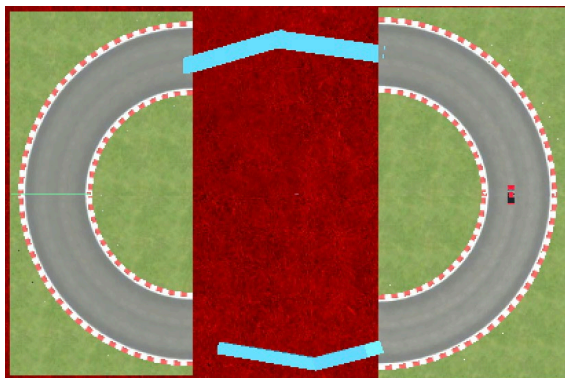
「簡単」



「普通」



「難しい」



発展課題

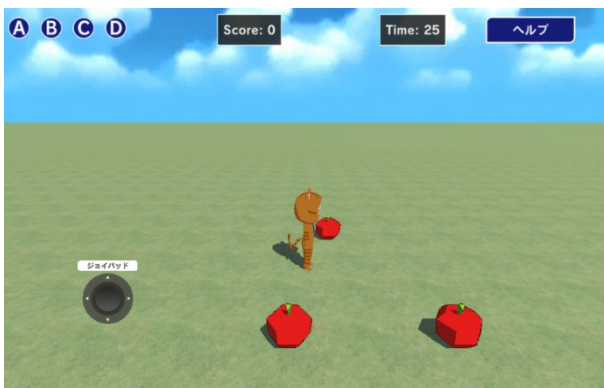
上で示されたプログラムの X、Y、Z の数字をいろいろ変えてみてゲームをもっと楽しくしてみましょう。どんな数字にすると簡単になるか、どんな数字にすると難くなるかいろいろな数字を入れて、実際に走行して試してみましょう。

12. Game Maker



「Game Maker」は、サンプルゲームを書き換えてオリジナルゲームが作れるようになっています。ゲームを構成する様々な要素（タイトル、プレイ開始前やプレイ中、プレイ終了後の処理、タイマーやスコアなど）のプログラムが説明付きで分かりやすく用意されています。

自分だけのオリジナルゲーム作りにトライしてみましょう。



8. 動画と一緒に作ろう

「動画と一緒に作ろう」は、いろいろなゲームの作り方、ゲームに欠かせない様々なパーツ、ゲーム作りのテクニックなどを動画で丁寧に説明します。動画で説明するのと同じプログラムも入っているので、動画を見ながら一緒に作ることもできます。

動画は、Mind Render の [YouTube チャンネル](#) でも公開しています。動画を見るだけでも楽しめます。是非ご覧ください。

一話完結動画



一話完結動画は、1 動画で 1 ゲームの作り方を説明します。動画は 16 あるので、16 種類のゲームの作り方がわかります。



ゲーム内で使うオブジェクトなどがあらかじめ入っているスタートプロジェクトが用意されています。

シリーズ動画



シリーズ動画 1、2 は、それぞれシリーズを通して 1 つのゲームの作り方を少しずつ説明します。



プレイヤーの動き、敵キャラの攻撃、ゲームオーバーやゲームクリアなど、ゲームに欠かせないパーツの作り方がわかります。

動画を見ながら一緒に作ってみましょう。

9. ムービーを作ろう

Mind Render には固定カメラや移動カメラを自由に使う命令が用意されています。カメラの座標、角度、動きをプログラムでコントロールすることで 映画のような迫力ある映像を作ることができます。

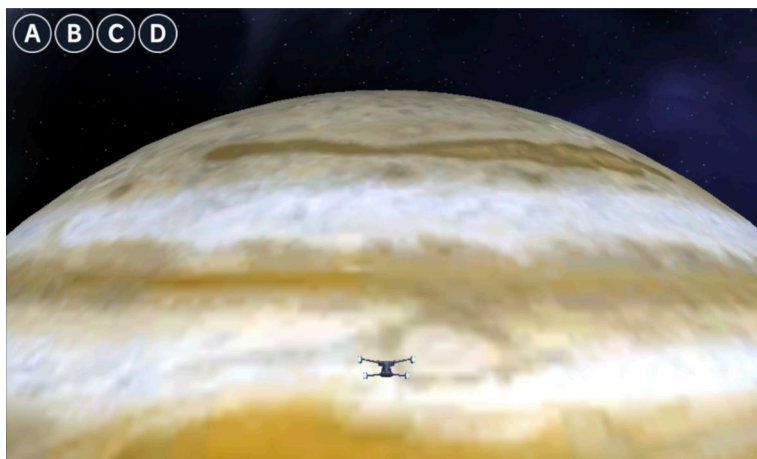
映画監督になってショートムービー作りに挑戦！

映画監督になった気分で、ショートムービーを作ってみましょう。「ムービーを作ろう」には、宇宙を舞台としたサンプルのショートムービーが入っています。



1. 惑星を目指す

宇宙船が惑星（木星オブジェクトを使用）に向かって飛んでいく様子、下から見上げるような角度にカメラを設定することで宇宙船の大きさを表現することができます。宇宙船の動きをゆっくりにすることでさらに迫力のあるシーンを作ることができます。



2. 大気圏に突入

宇宙船が惑星に近づいて、大気圏に突入していく様子です。カメラの座標は宇宙船の中に設定することもできるので、操縦席から見た巨大な惑星の姿を眺めるシーンを作ることができます。



3.着陸

カメラとオブジェクトの座標、角度、動きを工夫すると遠くの宇宙船が近づくにつれ大きくなっていく様子や同時進行でロボットなど他のオブジェクトが動く様子も同じシーンに納めることができます。



4.戦いが始まる

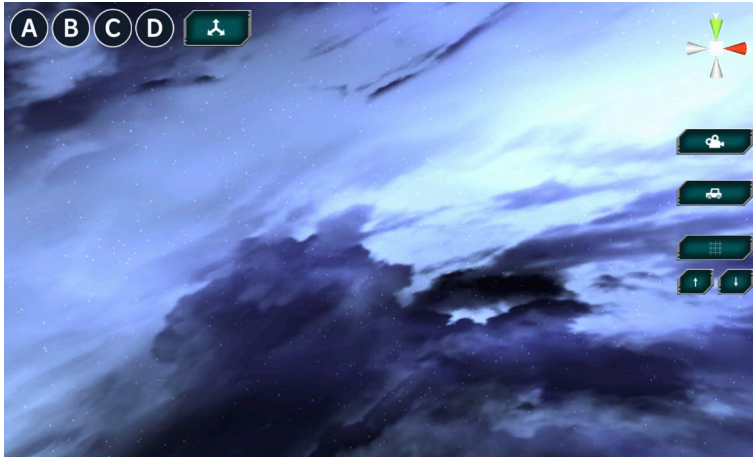
立っているキャラクターに背後からカメラが近づくシーンです。

映画でよく見るようなシーンを作るには、1コマ1コマの動きを考えてキャラクターとカメラの位置関係を調整していきます。映像効果をさらに高めるために、効果音を入れることもできます。Mind Render では「ホラー」、「アクション」、「戦闘」、「ファンタジー」などムービー作りに役立ついろいろなサウンドが用意されています。さらに「爆発」、「群衆の声」、「発射音」などの効果音を組み合わせることでよりリアルな表現となります。

次に具体的なシーン作りのためにどのようなプログラミングが必要となるかを見ていきましょう。以下で紹介するのは30秒ほどの短い動画の例です。プログラムは「ムービーを作ろう」に入っています。そちらも参考にしてください。

1. 惑星を目指す

最初に「1.惑星を目指す」について5枚のスクリーンショットで動画の流れを紹介します。その後でサンプルプログラムを見ていきます。



シーン1) カメラは広い宇宙のある方向を捉えています。ここからカメラを画面下方向にゆっくりと下げていきます。宇宙の雰囲気のある音が背景に鳴っています。



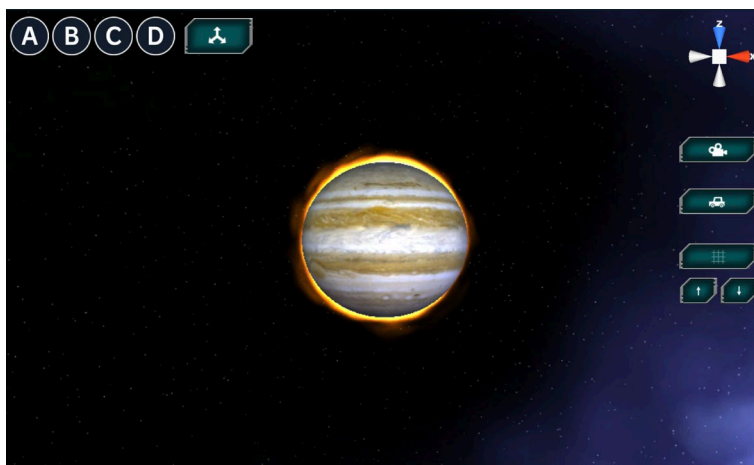
シーン2) やがて画面下の方から宇宙船の姿が見えてきます。宇宙船は画面奥に向かって進んでいますが、カメラは宇宙船の下まで回り込んで上を見上げるようにして宇宙船の底部を映します。



シーン3) カメラは宇宙船の底部を映しながら、前方の惑星へと向かって移動します。ゆっくりと宇宙船の底部を映していきます。



シーン4) カメラは宇宙船の最前部まで映して、宇宙船を追いつくようにしてさらに前へ進みます。



シーン5) やがて画面には惑星だけが映り、これから何かが始まりそうな不気味な音楽が流れます。

スクリーンショットだけでは動画の雰囲気が伝わらないかもしれません。以下に「1.惑星を目指す」から「3.着陸」までのサンプルプログラムを解説しますので、自分で作ってどのようなになるか実際に試してみてください。

最初に「1.惑星を目指す」の動画を作るためのプログラムを説明します。



「1.惑星を目指す」例1)を作るのに使われたオブジェクトの一覧を左に示します。



オブジェクトをこのシーンの流れに沿って動かすためにスタート時の各オブジェクトの座標を左図のようにしておきます。すべてのオブジェクトがY軸上に並んでいます。X座標とZ座標の初期値はすべて0となっています。太陽2は木星の後ろに置いて太陽フレアが木星の周囲に輝くようにして使います。

Y: 200 の座標に置いた太陽は木星の表面を明るく照らすためです。この太陽が無いと木星表面が暗く模様もあまりよく見えません。

次ページからオブジェクトごとに順番にプログラムを見ていきましょう。

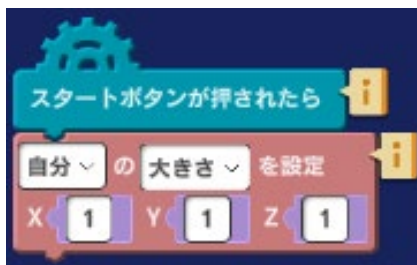


カメラオブジェクト

左図がカメラオブジェクトのプログラムです。

固定カメラの座標は X (0)、Y (100)、Z (0)、角度はすべて 0 にしてあります。

2 秒間待った後で、X 軸方向にカメラをゆっくり回転させていきます。カメラの回転が終わったところで「サスペンス」を鳴らし始めます。ループは「on」にしておきます。



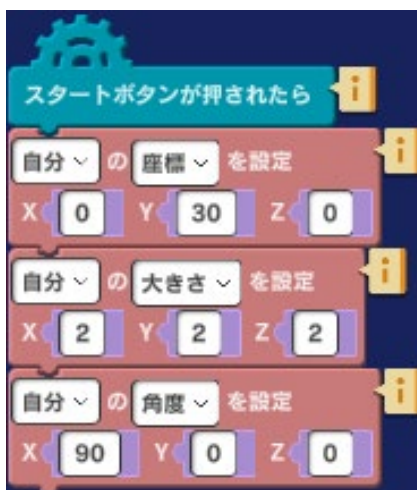
NewTown (背景)

遊び場の背景の初期設定は NewTown となっています。これを「宇宙 (青)」に変更しておきます。



太陽

太陽の大きさと座標は左図のように設定します。



木星

木星の座標と大きさは左図のように設定します。木星は見る角度によって模様が違って見えますので、ちょうど見せたい面がカメラに向くように X 座標の位置を調整します。



太陽_2

太陽_2の大きさは木星からはみ出さないように調整します。座標は左図のとおりです。



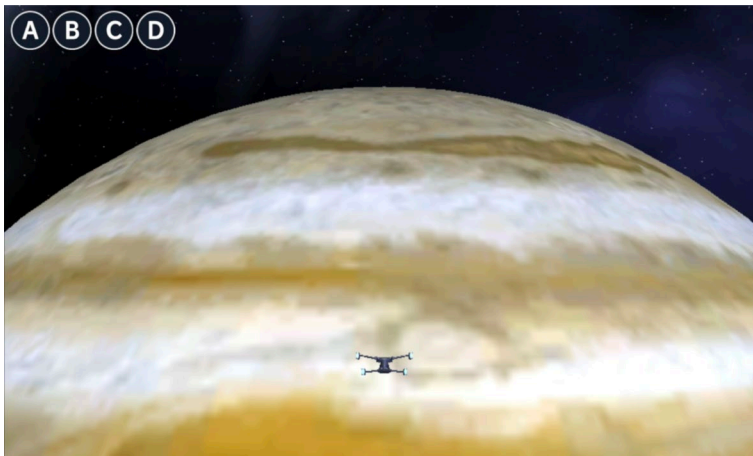
宇宙船

宇宙船の座標初期値は左図のとおりです。角度については木星に向かって先端部が向くようにX軸方向に90度回転させてあります。宇宙船のZ軸方向（進行方向）の座標値を0.01ずつ減らしていくことで、固定カメラの上を宇宙船がゆっくり後退するようにしてあります。繰り返しが終わってから10秒待つのは、木星を見ながらサウンドを聞かせるためです。

最後に次のシーンに切り替えるために宇宙大戦争-2へジャンプします。ジャンプによって他のプログラムへ切り替えるのは、背景が一度に1つしか存在できないので、背景を変えたいときは、別のプログラムとして作る必要があるためです。

2. 大気圏に突入

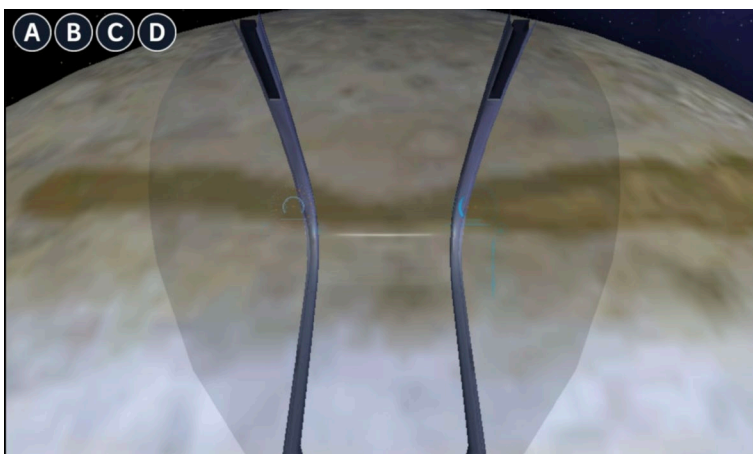
次に「2.大気圏に突入」のシーンを紹介します。



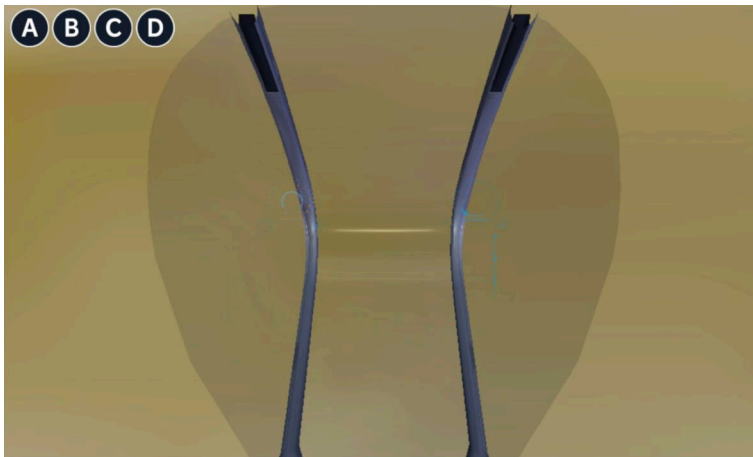
シーン1) は木星に近づく宇宙船を遠く眺めるシーンです。宇宙船を小さく、木星を大きく見せることで惑星の巨大さを印象づけます。



シーン2) 宇宙船の操縦席からの眺め。木星に次第に近づいていくシーンです。木星が次第に大きく見えて、同時に回転しています。



シーン3) 宇宙船は木星の大気に入っていきます。



シーン4) 木星の大気に入れた状態。視界が悪くなっています。

以下は「2.大気圏に突入」で使用するオブジェクトの一覧です。

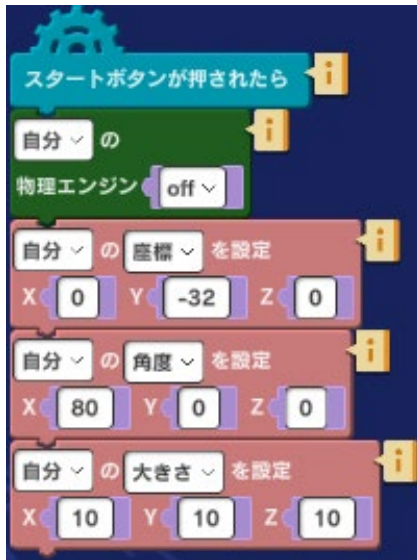


以下で「2.大気圏に突入」のプログラムについてオブジェクトごとに説明します。



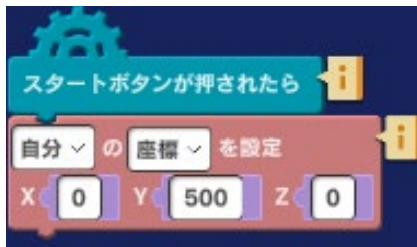
カメラ

スタート時点でのカメラの座標と角度は左図のようになっています。木星を正面に捉えて、カメラと惑星の間に宇宙船が小さく見えているような設定です。



木星

木星の初期設定は左図のとおりです。木星の大きさを X、Y、Z それぞれ 10 にして大きさを強調しています。



太陽

太陽はカメラの後ろに配置して、木星の表面を照らすことで木星表面の模様が明るく照らされるようにします。

宇宙船

スタートしたら「宇宙空間」と「サスペンス」のサウンドを鳴らします。宇宙船の座標、角度、大きさの初期設定は左図のとおりです。木星のすぐ手前に大きさを「0.25」と小さくして置くことで、木星の大きさとの対比を見せます。この状態で3秒待った後で固定カメラの座標をX(0)、Y

(0.8)、Z(0.2)、固定カメラの角度をX(0)、Y

(0)、Z(0)とします。この設定によりカメラは宇宙船の操縦席へ移動し、操縦席から見た木星が表示されます。

宇宙船とカメラの設定が決まったら、木星の角度と座標を少しずつ変更して、宇宙船が木星に次第に近づいていく様子を表現します。

サウンドを停止して次のシーンに移るために「宇宙大戦争3」にジャンプします。



3. 着陸

次に「3.着陸」のシーンを紹介します。



シーン1) 惑星の上空に宇宙船が出現して着陸態勢に入ります。画面右下と奥の方には大きなロボットが歩いています。

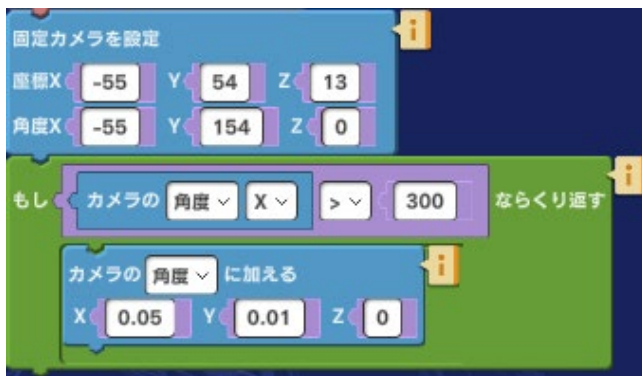


シーン2) 宇宙船が轟音を上げて着陸スペースに着陸します。その横を巨大ロボットが通過していきます。

「3.着陸」で使用したオブジェクトは以下のものです。



以下に「3.着陸」のプログラムをオブジェクトごとに説明します。



カメラ

固定カメラはのように座標と角度を設定してあります。初期設定の後でカメラの X と Y の角度をすこしずつ変えて、宇宙船の着陸の様子を追いかけていくようにします。(プログラムは New Town の中に入っています。)

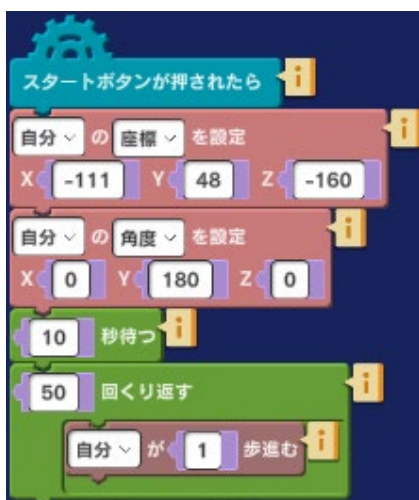
NewTown (背景)

「雪山」に変更しておきます。



ロボット (緑)

スタート時の座標と大きさを左図のようにします。この座標は着陸地点のすぐ横の道路上となります。宇宙船が降りてくるまで待ってから歩き始めます。



ロボット (赤)

スタート時の座標と大きさを左図のようにします。この座標は宇宙船の先の方にある道路の上となります。宇宙船が降りてくるのを待ってから歩き始めます。



宇宙船

スタートしたらエンジン音の代わりにサウンドオブジェクト「車の騒音」を繰り返し鳴らします。座標の初期値は上空で左図のような設定です。

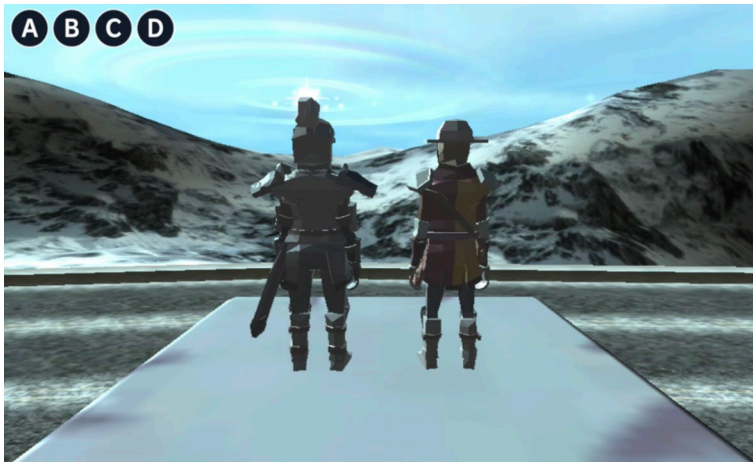
着陸態勢に入って次第に高度を落としていく様子を3段階に分けて表現してあります。最初の400回の繰り返しでは少し早めに降下、続く100回の繰り返しではYの減少する値を小さくしてゆっくり降下、最後の400回の繰り返しでYの減少する値をさらに小さくしてもっとゆっくり効果するように表現します。また着陸位置に近づくにつれ角度もわずかに下向き（Z軸方向）になるようにしてあります。

着陸と同時にエンジン音（「車の騒音」）を停止して、着陸音（「恐怖音」）とエンジン停止音（「パワーダウン」）を鳴らします。

5秒待ってから次のシーン「4.戦いが始まる」にジャンプします。

4. 戦いが始まる

次に「4.戦いが始まる」のシーンを紹介します。



シーン1) 2人の戦士が雪山を眺めています。雪山の向こう、遠くの方で大きな爆発音とともに閃光が繰り返し見られます。



シーン2) カメラが2人の戦士に向かって近づいていくと閃光とともに女王キャラクターがワープしたように登場します。



シーン3) 3人のキャラクターが揃ったところでカメラは背後まで近づいて遠くの戦場（爆発音と閃光の場所）を見つめる3人を捉え続けます。

このシーンで使うオブジェクトは以下ようになります。次ページから各オブジェクトごとにプログラムを説明します。



カメラ

カメラの座標と角度の初期設定は左図のようになっています。2人の戦士の後方から雪山を視界に入れます。スタートと同時にサウンド「戦闘」を繰り返し鳴らします。

次にカメラのZ軸方向の値を0.01ずつ大きくして戦士の方に近づいていきます。同時にカメラの角度をX(-0.05)だけ大きくしていきます。こうすることでカメラは戦士に近づくにつれ少しずつ上を向いていきます。これを200回繰り返したところで「アクション」というシグナルを送信します。このシグナルは、このタイミングで女王キャラクターを出現するためそのきっかけとして発信するものです。シグナルを送信したら、さらにカメラを戦士に近づけていきます。



岩ブロック

岩ブロックは2人の戦士を登場させる舞台として使います。岩ブロックの座標位置と大きさは左図のようにします。



騎士

騎士は大きさを X、Y、Z 方向に 2 倍の大きさにしておきます。座標は岩ブロックの上に立って見えるよう左図のようにします。



戦士

戦士も騎士と同じように大きさを X、Y、Z 方向に 2 倍大きくしておきます。座標は騎士のすぐ横に立つように左図の設定とします。



光 (大)

表示とともにサウンド「大砲」を鳴らします。このエフェクトは繰り返し再生しますが、X 軸方向（画面左右）でランダムな位置で再生されるよう X 座標の値を乱数で決めるようにしてあります。また繰り返しのタイミングがランダムとなるよう「待つ」命令でも乱数を使っています。

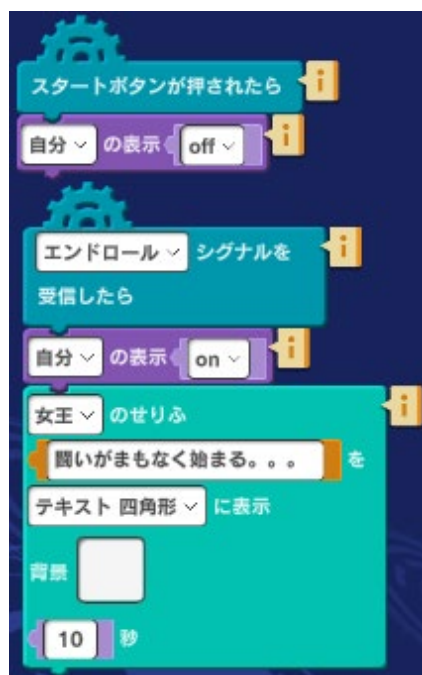


女王

女王のプログラムは2つあります。最初のプログラムでは初期条件を設定してあります。スタートしたらすぐに表示を off にします。大きさは騎士や戦士と同じように X、Y、Z 方向に 2 を指定して大きくしておきます。座標については、後でワープ表示させる位置をここで X (22)、Y (44)、Z (202) と決めておきます。

「アクション」シグナルを受信したら、ワープを表現するためのエフェクト「レベルアップ」を再生して、サウンド（魔法2）を鳴らします。その後で女王キャラクターの表示を on にして、女王を出現させます。

最後に「エンドロール」というシグナルを送ります。



テキスト 四角形

「エンドロール」シグナルを受け取ったら、テキストエリアを表示し、「戦いがまもなく始まる。。。という文字を表示します。

以上、「1.惑星を目指す」から「4.戦いが始まる」までのプログラムを見てきました。見栄えのするシーンを作るためにはオブジェクトの位置、角度、動きやカメラの設定、エフェクトや音をどのように映像に同期させるかなど考えることがたくさんあります。それでも実際の撮影と違って、同じシーンを少し条件を変えて何度も試行錯誤して作り直すことができますので最終的には思い描いたシーンを Mind Render で作り上げることができます。その作業はとても楽しく、作品を作った満足を得ることができます。

完成したプログラムを連続して実行すれば、ショートムービーを再生するように楽しむことができます。また実行画面を動画撮影し、映像編集ソフトウェアを使って字幕を入れたりして YouTube や SNS で人に見てもらう作品に仕上げていくこともできます。Mind Render にはたくさんのキャラクターや背景、音、エフェクトが揃っています。イマジネーションを使って作品づくりにチャレンジしてみてください。

10. ちょっとお勉強 1 : 「ドローンを動かしてみよう」

ラボ「ドローンを動かしてみよう」を使って、プログラミングでどんなことができるかをもう少し詳しく見ていきましょう。

1. 町



実験室メニューからラボ「1.町」をクリックして開けてみましょう。全画面にすると左図のようになっています。

このラボでは画面中央下にある赤いドローンを動かして楽しめます。

スタートボタンを押してあらかじめこのラボに用意されているプログラムを動かしてみましょう。



ドローンの操作には2つのジョイパッドを使います。ジョイパッドL（左側にある）を上下に倒すとドローンは前後に進みます。左右に倒すとドローンは左右に移動します。ジョイパッドR（右側にある）を上下に倒すとドローンは上下に移動します。左右に倒すと右方向、または左方向に回転します。

町中でドローンを飛ばしていろいろな場所に行ってみましょう。例えばジョイパッドの操縦で下図①、②の場所が見つけることができるか試してみましょう。

①



②

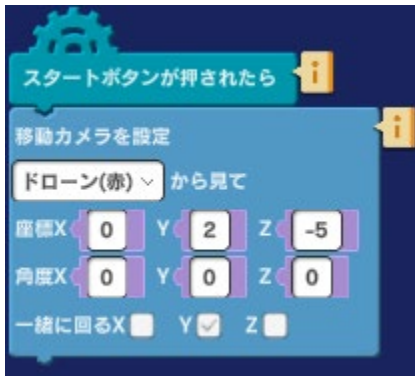


カメラの視点を切り替えるプログラミングを作ってみましょう。

プログラム編集画面を表示します。



オブジェクトリストからカメラをクリックして選びます。



Mind Render ではカメラをどこかに固定することも、何かに取り付けて移動するカメラとすることもできます。このラボの初期設定ではカメラの設定は左図のプログラムのようにになっています。

この設定では移動カメラをドローンの近くに取り付けます。取り付けるとことはドローンが移動すれば、カメラもそれにつれて一緒に移動するということになります。



上のような設定をした時のカメラ位置から見た様子は左図のようになります。ドローンの後ろ、少し高い場所からドローンを見下ろすようなアングルです。

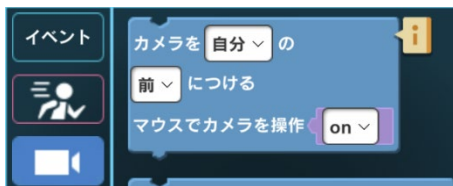
座標 X Y Z で指定するのはドローンを基準にして X Y Z それぞれの方向についてどれくらい離れたところに取り付けるかを指定するものです。この設定では X 方向はドローンの位置、Y はドローンより少し高く、Z 方向はドローンの後ろの場所を示しています。

す。

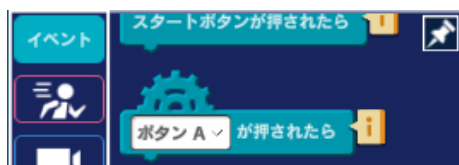
ここでは A から D まで 4 つ用意されているボタンを使ってカメラを切り替えるプログラムを作ってみます。



プログラムを作る場所はカメラオブジェクトです。オブジェクトリストから「カメラ」をクリックします。



まずカテゴリーの中から「イベント」をクリックします。



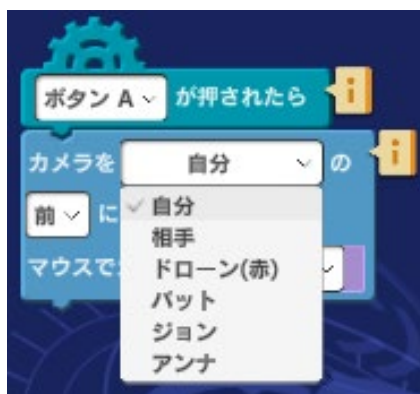
命令ブロックエリアに表示されたブロックから「ボタン A が押されたら」をドラッグ&ドロップしてプログラミングエリアに持っていきます。



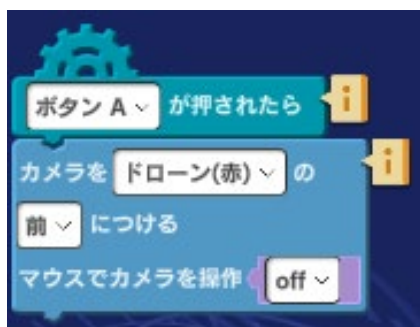
次にカテゴリーの中から「カメラ」アイコンをクリックして、「カメラを[自分]の[前]につける マウスでカメラを操作[on]」ブロックをドラッグ&ドロップしてプログラミングエリアに持っていきます。

この2つのブロックを左図のように組み合わせるとプログラムがで上がります。

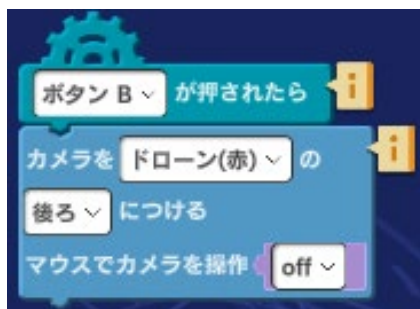
「マウスでカメラを操作」は、[off]にしてください。



カメラを誰に取り付けるかをドロップダウンメニューで選んで指定します。



左図のようにプログラムが完成したら、ボタン A を押すことでカメラの取り付け位置をドローンの前にすることができます。この時カメラはドローンの前方を向きます。

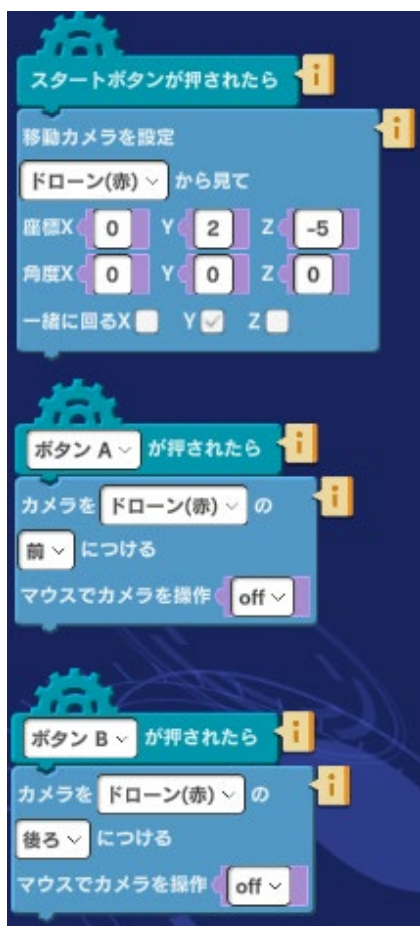


次に、左図のプログラムも作ります。



同じ様なプログラムを作るときは、コピー機能が便利です。コピーしたいブロックをマウスの左ボタン（または指）で長押しします。「Copy」と表示されたら、空いている場所をクリック（またはタップ）すると貼り付けられます。

このプログラムによって、ボタン B を押すことでカメラの取り付け位置をドローンの後ろに変えることができます。



上で作った2つのプログラムを加えると、カメラオブジェクトのプログラムは左図のようになります。



プログラムが完成したら、全画面に切り替えて実行してみてください。いつでも好きな時にカメラの向きを変えることができます。

こんなふうに人物に近づいてアップ画像を見ることができます。



また、空高く上がって、この町の全体を見ることもできます。

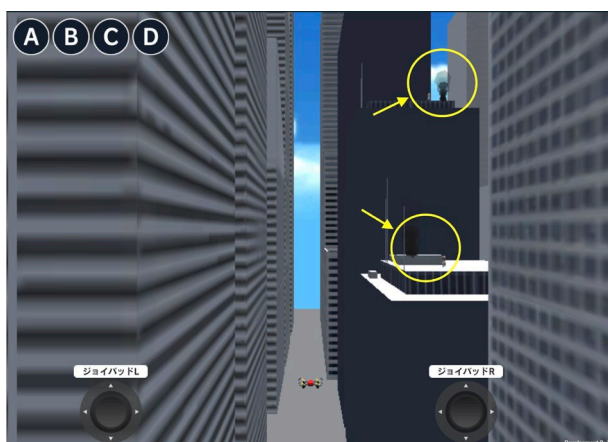


左図のような眺めを得るにはどこに行けばよいでしょう？ドローンで探検して探してみてください。

発展課題

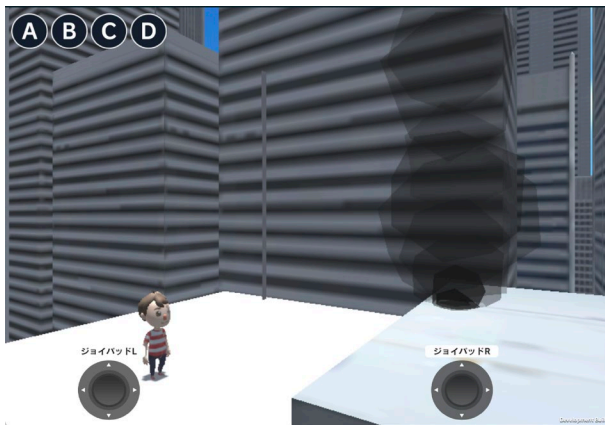
カメラはどのオブジェクトにも取り付けることができます。いろいろなオブジェクトにカメラを取り付け、角度を調整してドローンが飛ぶ様子を見てみましょう。例えば、立っている人の誰かの視点でドローンの飛行を見たらどう見えるか、街頭や家の屋根から見たらどのように見えるかなどいろいろなアイデアを試してみてください。

2. 人命救助



ラボ「2.人命救助」は高層ビル街でドローンを飛ばして人命救助に挑戦します。

ラボ「2.人命救助」をスタートしてプログラムを実行すると左図のようにビルの上に黒い煙が上がります。



煙の上がっているところには左図のように人がいて救助されるのを待っています。

このビル街では4人の人がビルの上で救助を待っています。煙が上がっているのを目印に人がいる4箇所を探してみましょう。

4人の人がいる場所をドローンが自動的に順次巡回していくようなプログラムを作ってみましょう。シーン編集画面で確認すると、4人がいる場所のX、Y、Zの座標は以下のようになっています。

パット			
座標	42.00	31.68	13.00
角度	0.00	0.00	0.00
大きさ	1.00	1.00	1.00
グループ			

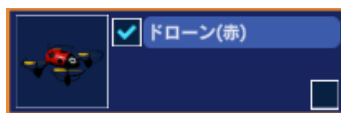
パットがいる場所 : X (42) 、 Y (31.68) 、 Z (13)

トムがいる場所 : X (46) 、 Y (52.46) 、 Z (30)

ジョンがいる場所 : X (40) 、 Y (76.88) 、 Z (55)

アンナがいる場所 : X (38) 、 Y (114.12) 、 Z (110)

ビルがたくさん立ち並んでいますのでぶつからないように移動する必要があります。そのためにはどのビルにもぶつからない高さ (Y) までいったん高度を上げておいてから X と Z の位置に向かうようにします。



プログラムは ドローンオブジェクトの中に作っていきましょう。オブジェクトリストからドローンをクリックして選びます。



プログラムの実行を開始した時のドローンの位置は左図のプログラムによって設定されています。

X (30) 、 Y (33) 、 Z (-16)



「ボタン A が押されたら」を自動巡回プログラムが実行されるきっかけにします。左図はドローンが初期の場所からパットのいる場所へ飛んでいくプログラムです。

パットのいる場所に到着したら 1 秒待って、次の場所に向かいます。

ドローンの現在地の Y はすでにパットのいる場所より高いので、左図では X と Z についてだけ移動しています。

移動する距離は現在の場所と目的地の X、Y、Z の差となります。Z を例に説明すると、ドローンの現在地（スタート時点）が -16、パットのいる場所の Z が 13 となるので差は 29 となります。左図では 1 ずつ移動するのを 29 回繰り返すことで 29 だけ移動しています。ドローンを瞬間的に移動することも可能ですが動く様子がアニメーションのように見えるように短い距離を繰り返し移動するようにしています。



左図はパットのいる場所から、次のトムがいる場所へ向かうためのプログラムです。移動する順序は他のビルに衝突するのを避けるため Y が最初、次に X と Z となっています。

同じ要領でトムからジョン、ジョンからアンナへ移動するプログラムを自分で作ってみましょう。

4 人全部を巡回できたらプログラムの最後に左図の命令を付けて加えて煙をすべて止めてみましょう。

発展課題

救助を待つ人のところまで行ったら、その人が救助されたことがわかるように次のようにしてみましょう。

- 1) 「助けてくれてありがとう！」と感謝のメッセージを表示する
- 2) 救助された人はもうその場にはいないこととして、表示を消す

メッセージを表示する方法として画面下部に表示する方法と吹き出しオブジェクトを使うやり方があります。どちらも試してみましょう。

3. ドローンゲーム



ラボ「3.ドローンゲーム」では、ドローンをターゲットに次々に命中させるゲームを作りましょう。

ラボ「ドローンゲーム」を実行すると左図のようになっています。ドローンを使って押してビル街の様子を調べてみましょう。



全部で5つの緑色の立方体がビルの上に乗っています。これがこのゲームのターゲットになります。



ドローンが緑の立方体を通り抜けると立方体の色が黒に変わります。

全部のターゲットの色をできるだけ短い時間で黒に変えるのがこのゲームです。手元にスマホのストップウォッチアプリなどを置いてどれくらい短時間でクリアできるか挑戦してみましょう。

ターゲットを通る度にポイントを獲得するプログラムを見てみましょう。

オブジェクトリストを見ると5つの立方体があります。それぞれの立方体の中にポイントを追加するプログラムがあります。例として「立方体1」のプログラムを見てみます。

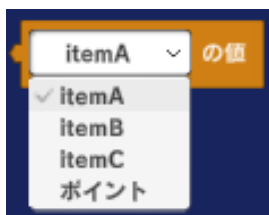
ポイントを計算するためには変数を使います。変数とは次々に中身を入れ替えたり、変数の中身に数を足したり引いたりすることのできる便利なしくみです。

変数

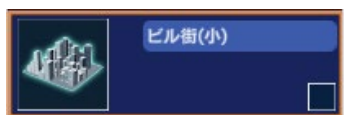
新しく変数を作るには命令カテゴリーの下の方にある「変数」をクリックします。



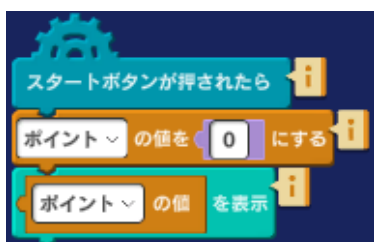
すでに「ポイント」という名前の変数が用意されています。



「x/y」カテゴリーで茶色の変数命令ブロックをプログラミングエリアにドラッグ&ドロップします。ドロップダウンメニューをクリックすると、「ポイント」が新しい変数として追加されていることがわかります。



ポイント変数はプログラムを起動した時にゼロに戻しておく必要があります。このことを初期化といいます。変数の初期化はビル街（小）オブジェクトの中に作っておきましょう。



左図のようなプログラムを作ってポイント変数の初期化をします。ポイント変数を表示するために「せりふ」カテゴリーから「～を表示」命令を追加して中に「ポイントの値」を入れておきます。



プログラムを実行すると左図のように画面下に表示エリア（黒い部分）が表示されて、ここにポイント変数の数字が表示されます。



「立方体1」のプログラムは左図のようになっています。このプログラムの意味はターゲットを通過するたびにポイントを1だけ増やして、表示エリアに現在のポイントを表示するということです。

ポイントが表示されてよりゲームらしくなってきました。他にもどのターゲットを追加しても1点とするかわりに遠くにあるターゲットやアクセスしにくい難易度の高いターゲットではポイントを増やすなどゲームが楽しくなる工夫をしてみましょう。

発展課題



ゲームをさらに面白くするために、ターゲットの大きさが大きくなったり、小さくなったり一定の距離を動き回ったりするようにしてみましょう。

4. しかけを作る

ラボ「しかけを作る」ではドローンが飛び回る背景の中にしかけを作っていきます。

カメラ			
座標	-2.00	2.50	-7.00
角度	0.00	10.00	0.00
大きさ	1.00	1.00	1.00
グループ	▼		

まずスタートボタンを押した時のドローンの位置を決めて設定しておきます。

シーン画面で入力します。左図は設定の例です。この通りでなくても自分でいろいろな位置を試して決めてください。移動カメラをドローンに取り付けてこの位置から見た様子が下図になります。

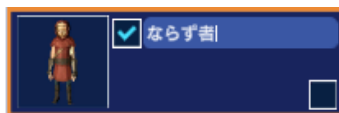


ジョイパッドは、ラボ「1.町」で使ったのと同じものです。「1.町」を参照しながら自分で作ってみましょう。中央にいるキャラクターはオブジェクトリストで追加したものです。

オブジェクトを追加する時はオブジェクトメニューの下にある左図矢印のアイコンをクリックします。



いろいろな種類のオブジェクトが表示されます。使いたいオブジェクトをクリックすれば追加されます。



追加したオブジェクトはオブジェクトリストに表示されます。

大勢のキャラクターが動き回る様子をドローンでいろいろな場所から見てみましょう。上で示した方法でオブジェクトはいくつでも追加することができますが、ここではプログラムによってもっと楽にすばやくキャラクターの数を増やす方法を試してみます。



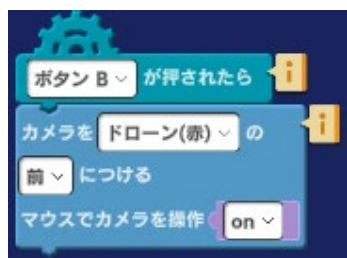
左図のプログラムを追加したキャラクターのところに作っておくと、ボタン A を押すたびに自分の分身を作っていきます。



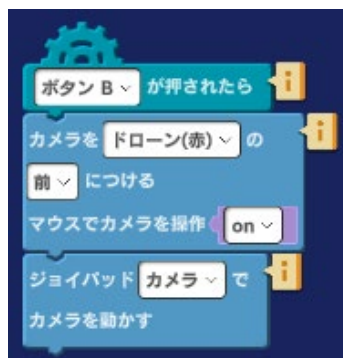
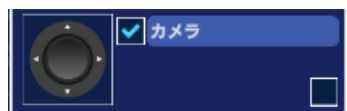
A ボタンを何度も押せば左図のようにたくさんのキャラクターの分身を作ることができます。分身はオリジナルのように歩き出します。



カメラの角度をプログラムで変更できるようにしておいていろいろなカメラワークを試してみましょう。



例えば左図のようなプログラムをカメラオブジェクトに加えておけば B ボタンをクリックしてカメラの取り付け位置を切り替え、マウス操作でカメラ角度を変えることができます。



マウスが無い場合は、カメラ操作のジョイパッドを追加し（左）、カメラオブジェクトまたはカメラ用ジョイパッドの中に左図のような命令を追加すると、ジョイパッドで同様の操作ができます。

ムービー制作に挑戦

ドローンを飛ばしながら映画のようなシーンの再現ができないかいろいろ試してみましょう。



実行画面の A～D ボタンの横にある「録画」ボタンで、プログラム実行中の画面を録画することができます。プログラムをスタートし、こ

のボタンを押すだけで録画が開始します。動画ファイルは、各 OS で指定されたフォルダに保存されます。

カメラを上手く操作して、ドローンの空撮や建物の脇を駆け抜けるなど迫力あるシーンを記録してみましょう。

注意： Windows で録画機能を使用する場合

Microsoft ストアからダウンロードした Mind Render には録画機能がありません。Windows で実行画面を録画する場合は、以下の手順に従ってください。

1. ストアからダウンロードした Mind Render でプログラムを作成する
2. Mind Render ホームページから Mind Render をダウンロードする
3. 2 の Mind Render で実行画面を録画する

発展課題

第 9 章の解説を見て、ショートムービーの作成にチャレンジしてみましょう。このラボで使った背景やオブジェクトを使って面白いショートムービーを作ってみてください。

5. 軌跡を描く

ラボ「5. 軌跡を描く」ではドローンの軌跡をペンで色をつけて見えるようにしてみましょう。その上で図形を空に描いてみることに挑戦します。



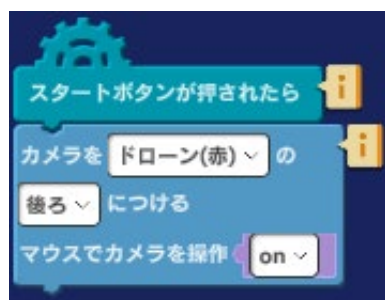
最初に使用するオブジェクトは左図のとおりです。

ドローン(赤)			
座標	5.00	20.00	0.00
角度	0.00	0.00	0.00
大きさ	1.00	1.00	1.00
グループ			

ドローンの位置は左図のように指定します。



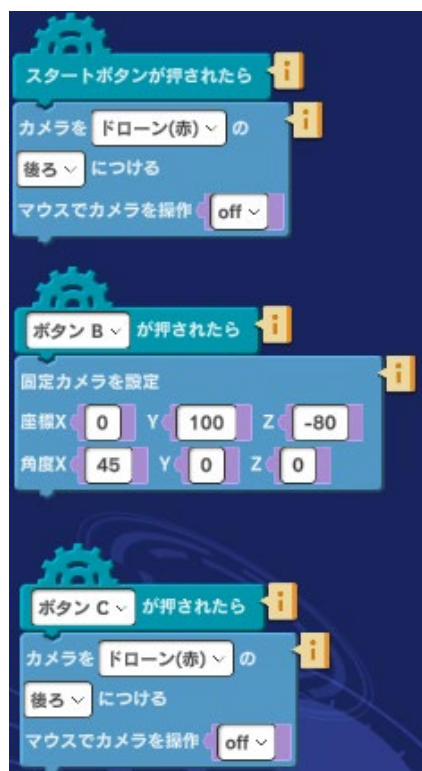
ドローンのプログラムは左図のようなものを用意します。



左図がカメラオブジェクトのプログラムです



プログラムを実行して、A ボタンを押すと左図のように線を描きます。

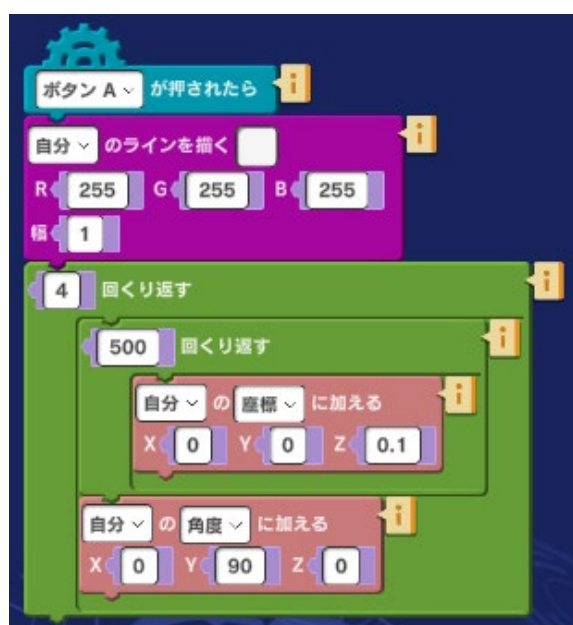


カメラをドローンの後ろにすると描いているラインの全体を見ることができないので左図のようにボタン操作でカメラの切り替えができるようにプログラムを作ります。



上空から見るとラインで描いた軌跡が見えます。

上空に四角形を描くにはどうしたらよいでしょうか？



左図はそのやり方を示しています。



上のプログラムを実行すると下図のように四角形を描くことができました。

発展課題

下図のような円を描くにはどうしたらよいでしょうか？



円を描くプログラムのサンプルが左図です。自分でもいろいろ条件を変えて試してみましょう。

11. ちょっとお勉強 2：「3D で日本を知ろう」

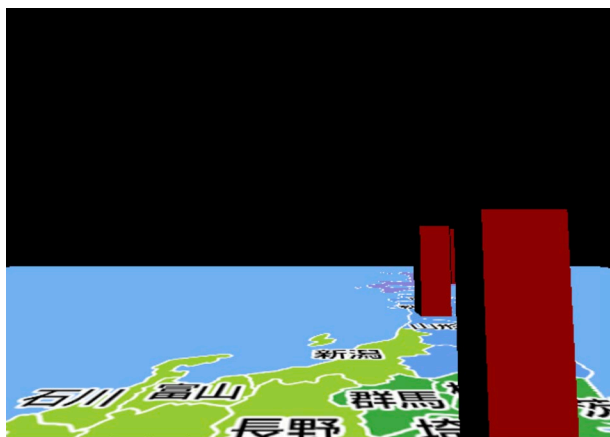
ラボ「3D で日本を知ろう」では 3D グラフの使い方を学ぶことができます。3D グラフとは円柱や四角柱の形をした 3D グラフです。下図は 3D グラフの例です。Mind Render では 3D グラフが伸びていく様子をアニメーションで表示することもできます。



では早速使ってみましょう。ラボメニューから「ちょっとお勉強」を、続いて「3D で日本を知ろう」を選びましょう。

1. 納豆の消費量

ラボ「納豆の消費量」をオープンしたら早速全画面表示にしてプログラムを実行してみましょう。



スタートボタンをクリックした後、A ボタンをクリックしてください。5 秒間待ってからプログラムがスタートすると左図のようにアニメーションで日本地図の上に四角柱の 3D グラフが伸びていく様子を見ることができます。

ここで 3D グラフが示しているのは日本各地（北海道、秋田、東京、大阪、福岡）の 1 世帯当たりの納豆の消費量です。数字を入れ替えて納豆の消費量ではどうなっているか見てみましょう。参考にした資料は公益財団法人日本特産農産物協会発行の「平成 30 年度豆類振興事業報告書」です。この資料はインターネットで誰でも見ることができます。報告書 175 ページにある「納豆に対する 1 人当たり年間支出額（都道府県庁所在市別）」には次のように各地で年間に 1 人が納豆購入にいくら使っているかがまとめられています。調査年は平成 30 年を選びました。単位は 円/人/年 となります。

北海道 : 1,553

秋田県 : 1,922

東京都 : 1,458

大阪府 : 964

福岡県 : 1,477

サンプルプログラムには上記の北海道から福岡県までの 5 つのデータがあらかじめ入力されています。

北海道			
座標	9.00	0.50	12.30
角度	0.00	0.00	0.00
大きさ	1.00	1.00	1.00
グループ			

3D グラフを描くには、「立方体」や「円柱」オブジェクトを使用します。

始めに立方体などのオブジェクトを地図上に配置します。シーン画面で地図上を移動させるか、座標を指定します。Mind Render で特定の場所を示すには 3 つの座標の値（X、Y、Z）を使ってを指定します。（9.00、0.50、12.30）は地図上の北海道がある場所の座標です。



オブジェクトはプログラムスタート時には非表示に設定してあるので、表示を[on]にします。

グラフの大きさは、Y の高さで指定します。

1553 のような大きな数を表示する場合は、割り算をして（ $1553 \div 100$ ）適当な大きさに収まるようにします。

グラフを描画する速さを指定します。ゆっくり描きたい時は大きな数を指定します。

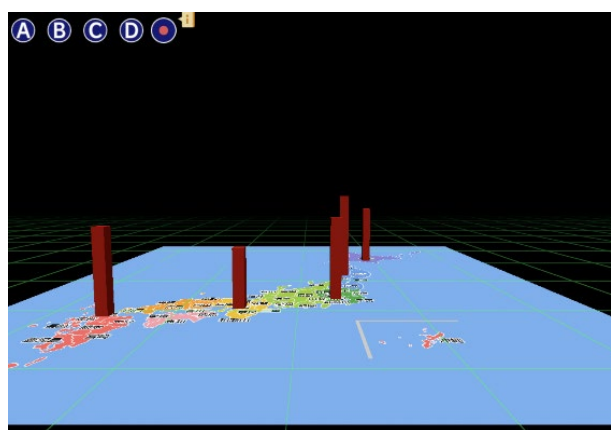
納豆消費量のデータでは北海道の値が最も大きく、1,553 となっています。上記例では、これを 100 で割っています。秋田、東京、大阪、福岡でも同じように元のデータの値を同じように 100 で割っています。プログラムを見て確かめてください。

このプログラムは VR でも楽しめるように、プログラムがスタートするとカメラが移動してアニメーションを動きながら眺められるようにしてあります。もし、日本地図全体を表示してグラフを見たい場合には次のような手順でカメラの位置を変えることもできます。



1) オブジェクトリストからカメラを選択する

2) ボタン A が押されたら のプログラムを左図のように変更する



変更完了後にプログラムを実行すると左図のように表示されます。

固定カメラの値を変えることで 3D グラフをいろいろな場所から見るができます。

発展課題

インターネット上では各県別にまとめられたいろいろな統計を見ることができます。一例として「なるほど統計学園 (<https://www.stat.go.jp/naruhodo/>)」では人口や農林、エネルギーなどさまざまな統計が紹介されています。このサイトで面白そうな統計を探して北海道、秋田、東京、大阪、福岡のデータを見つけ、ラボのプログラムに入れてグラフを表示してみましょう。

2. グラフの作り方

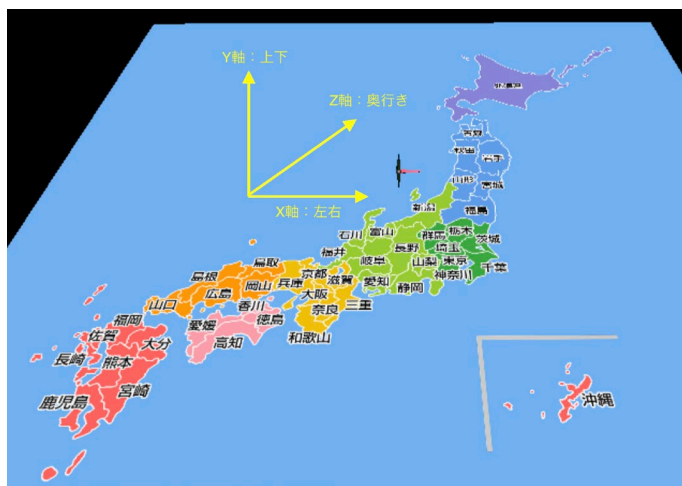
ラボ「2. グラフの作り方」にはあらかじめプログラムが入っていません。このラボでは日本地図の上で好きな場所にグラフを描かせるために座標の調べ方を学んでみましょう。



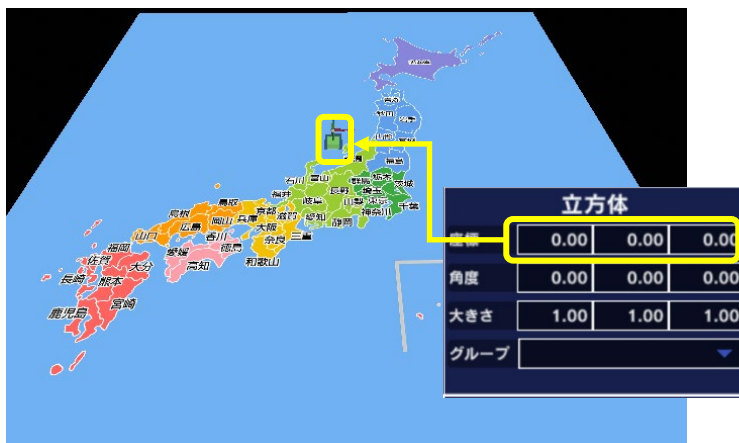
準備としてオブジェクトを1つ追加します。オブジェクトリストの下
のオブジェクト追加ボタンをクリック、左上のドロップダウンから「素
材」を選択してその中にある「立方体」（ここでは「立方体を使いま
すが、「円柱」でも OK です）をクリックしてください。



オブジェクトリストに「立方体」が追加されたら、画面左上に座
標が表示されています。Mind Render は 3D なので、座標は X
軸、Y 軸、Z 軸という 3 つの軸上のどこにあるかで決まります。
座標 : X 座標 (0.00)、Y 座標 (0.00)、Z 座標 (0.00)



立方体を追加した時に表示されている場所は
新潟県の少し上になります。この位置を座標
で表すと X(0.00)、Y(0.00)、Z(0.00)とな
ります。



日本地図上のどこかの場所の座標を調べ
るには立方体を動かして表示される座標
情報を見ればよいのです。立方体を動か
す方法は次のとおりです。



シーン編集画面で立方体をクリックすると左図のように赤、青、緑の3本の矢印が表示されます。赤い矢印をドラッグすると立方体をX軸に沿って動かすことができます。青色の矢印はZ軸、緑色の矢印はY軸に沿って立方体を動かします。

青色の矢印がつかみにくいときは、緑色の四角をつかんで動かすと、高さ(Y)を変えることなく、前後(Z)、左右(X)に動かすことがで

きます。



例題として岐阜県の座標を調べてみましょう。下図のように立方体を動かして岐阜県の上まで移動します。

立方体			
座標	-1.40	0.50	-6.50
角度	0.00	0.00	0.00
大きさ	1.00	1.00	1.00
グループ			

立方体が岐阜県の場所にある時、立方体の座標を確認すると左図のようになっています。X (-1.40)、Y (0.50)、Z (-6.50) あたりが岐阜県の座標であることがわかりました。

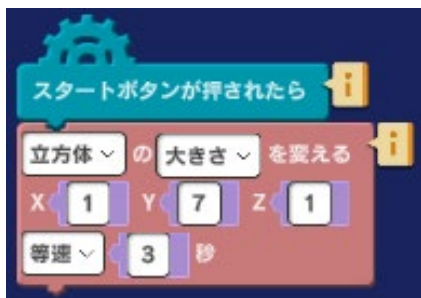
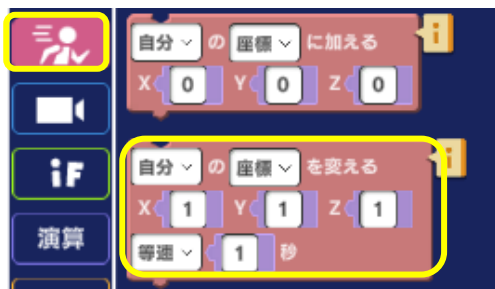
この座標情報を使えば、岐阜県の上に3Dグラフを描くことができます。やってみましょう。



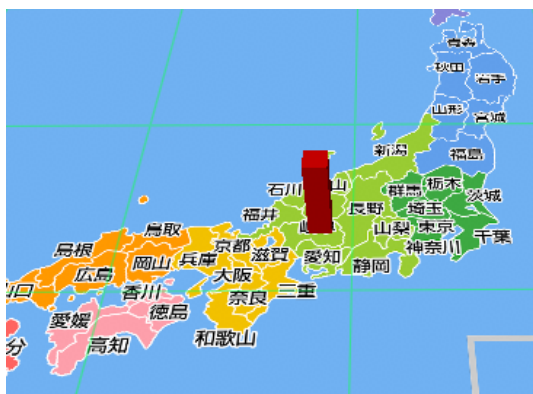
なお、立方体を少し上方に持ち上げてから、「靴」アイコンをクリックすると、立方体を地面に着地させることができます。

プログラム編集画面で、「日本地図」オブジェクトを選択します。ここにプログラムを作成します。

「移動」カテゴリから左図のブロックを選択します。



大きさ（高さ）として適当な値（例として7）スピード(例として3秒)を指定します。左図に例を示します。色は、オブジェクトリストで「立方体」を選択し、「詳細」ボタンをクリックして表示されるオブジェクトの詳細画面で指定します。



このプログラムを実行すると左図のように岐阜県の上に3D グラフが現れます。



描かれた 3D グラフはプログラムで消さないとそのまま残ってしまいます。左図のように「立方体の表示 [off]」、「立方体の表示 [on]」という命令ブロックを加えることで毎回実行するたびに 3D グラフを描くアニメーションを見ることができます。

これで日本地図上の好きな場所に 3D グラフを描く方法がわかりました。次のラボではインターネットで見つけた統計データを使ってグラフを描いてみます。

発展課題

グラフを描く際に効果音を鳴らしてみましょう。



「効果」 カテゴリーから左図のブロックを選択します。
アイコンを押すと、サウンドの一覧が表示されたたくさんの音が見つかります。



再生ボタンをクリックして音を聞いて使いたい音を選びましょう。



左図は音を鳴らすサンプルプログラムです。ここでは「注意音3」という音を使っています。このプログラムではグラフを描く前に音を鳴らしていますが、どのタイミングで音を鳴らせばより効果的になるかいろいろなパターンを試してみましょう。

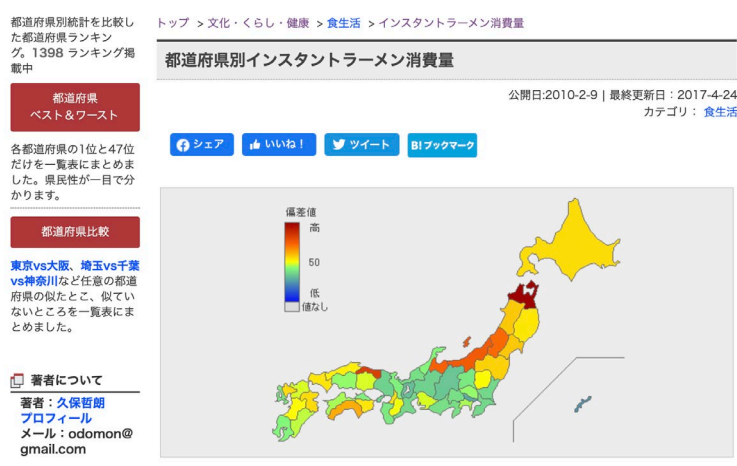
3. インスタントラーメン消費量

インターネットでは日本国民の生活を知るためのいろいろなデータを探すことができます。例として総務省統計局のホームページでは人口動態や企業動向などあらゆるデータを見つけることができます。他にも民間で統計情報をまとめたサイトもあります。

総務省統計局 : <https://www.stat.go.jp>



都道府県別統計とランキングで見る県民性 : <https://todo-ran.com/t/kiji/11811>



ここでは「都道府県別統計とランキングで見る県民性」ウェブサイトに掲載されている面白いデータを使って 3D グラフを描いてみましょう。

使うデータは「都道府県別インスタントラーメン消費量」です。このデータは、「都道府県別統計とランキングで見る県民性」ウェブサイトで見つけることができます。

実際のデータを見ると下図のように掲載されています（下図は一部を抜粋したもの）。ここで示されているデータは1世帯が1年間に購入しているインスタントラーメンの量を g（グラム）で表示したものです。このデータの元となっているのは総務省統計局の家計調査で 2016 年の調査結果です。

このデータから以下の青森県、富山県、佐賀県のデータを使って 3D グラフを描きます。

□ 都道府県別インスタントラーメン消費量

「並替」の右横の「北／南」「降順／昇順」をクリックすると並べ替え表示します。

順位	都道府県	消費量	偏差値
並替	北 南	降順 昇順	降順 昇順
1	青森県	9,227g	86.31
2	鳥取県	8,164g	73.83
3	新潟県	7,742g	68.87
4	富山県	7,634g	67.61
5	山形県	7,523g	66.30
6	高知県	6,950g	59.57
7	宮城県	6,711g	56.77
8	秋田県	6,679g	56.39
9	福島県	6,620g	55.70
10	佐賀県	6,560g	54.99
11	北海道	6,536g	54.71
12	大分県	6,484g	54.10
13	徳島県	6,472g	53.96
14	大阪府	6,438g	53.56
15	山口県	6,424g	53.40
16	岩手県	6,401g	53.13
17	島根県	6,315g	52.12
18	熊本県	6,232g	51.14
19	栃木県	6,227g	51.08
20	奈良県	6,148g	50.16
21	愛知県	6,102g	49.62

青森県 : 9,227g

富山県 : 7,634g

佐賀県 : 6,560g



ラボ「3.インスタントラーメン消費量」にはサンプルプログラムが用意されています。オブジェクトリストの「日本地図」を選択すると下図のようなプログラムが表示されます。



このプログラムを実行すると左図のように3つの県の上に3Dグラフがアニメーションで表示されます。



このプログラムの冒頭にある左図の命令は、プログラムの実行前にすでに描かれている3Dグラフがあればそれらを全て削除するためのものです。この命令がないと、実行前に描いた3Dグラフがそのまま残ってしまって不都合です。



プログラム本体を青森県を例にとって説明しましょう。左図では青森県のインスタントラーメン消費量である 9227(g)を 1000 で割った数をグラフの大きさとしています。他の県のデータも同様に値を 1000 で割ったものをグラフの大きさとしします。

グラフは 5 秒で描くように設定されています。値が小さければ早くグラフを描き、逆に値が大きければゆっくりグラフを描きます。

サンプルプログラムで次のように変更したい時はどうしたらよいでしょうか？実際にやってみてください。

- 1) 3D グラフの形を四角柱から円柱に変えたい
- 2) 3D グラフを素早く描きたい
- 3) 3D グラフの色を 3 つの県、それぞれに違う色にしたい

3D グラフ描画命令に慣れたら、ラボ「4. グラフの切り替え表示」に進みましょう。

発展課題

3 つの県で最も大きなグラフの色を変える、描画方法を「等速」以外にするなどして目立つようにしてみましょう。

4. グラフの切り替え表示

ここまでで基本的な 3D グラフの描き方はわかったと思います。ラボ「4. グラフの切り替え表示」では、多くのデータを切り替えながら表示するやり方について学んでいきましょう。

まず元データを用意しましょう。元データは下のサイトから 3 種類の都道府県別データを選んで使います。

「都道府県別統計とランキングで見る県民性」 (<https://todo-ran.com>)

3 種類のデータとは次のものです。

- 1) Facebook のユーザ数
- 2) コンビニ店舗数
- 3) 人口

以上の 3 種類のデータを 3 つの県（福島県、奈良県、高知県）についてまとめると下のようになります。下の表では座標データも加えてあります。

県別統計

	1 列目	2 列目	3 列目	4 列目	5 列目	6 列目	7 列目
	県名	Facebook ユーザ数	コンビニ 店舗数	人口	X座標	Y座標	Z座標
1 行目	福島県	400000	889	2040000	5.1	0	-2.2
2 行目	奈良県	310000	456	1399000	-3.9	0	-9.8
3 行目	高知県	150000	286	766000	-9.3	0	-11.6

このラボではこの表のデータ（座標データを除く）をボタンを押すことで次々に切り替えて表示できるようなプログラムを考えてみましょう。そのためには Mind Render で用意されているデータセットを使うのが便利です。データセットとは縦（行）と横（列）をもったデータを入れる容器のようなものと考えてください。

上の表を見ると例えば福島県のコンビニ店舗数が知りたければ 1 行目と 3 列目が交差するところの数がそれに対応するとわかります。このようにデータセットではあらかじめ行と列に整理して納めたデータをプログラムから自由に呼び出すことができます。

（例 1）高知県の人口が知りたい → 3 行目、4 列目

（例 2）奈良県のコンビニ店舗数が知りたい → 2 行目、3 列目



データセットを使うにはまず以下のようにして新しいデータセットを作ります。

- 1) 画面左側の命令カテゴリーから「変数」のボタンをクリックします。



2) 変数、リスト、データセットの3つの設定エリアが表示されるので、データセットのエリアで「県別統計」と入力してから「追加」ボタンをクリックします。これで新しく「県別統計」という名のデータセットが作られました。



データセットを使うためにはまずデータをデータセットに入れていきます。「x/y」カテゴリーをクリックしてデータセット関連の命令を探します。



データをデータセットに入れるには左図の命令ブロックを使います。上で示した「県別統計」という表を参照しながらデータを入れていきます。何行目の何列目にどのデータを入れるか、間違わないように気をつけてデータを入れていきましょう。

左図は福島県のデータを1行目1列目から4列目に入れたところです。同じようにして奈良県と高知県のデータも入れてみましょう。奈良県と高知県のデータを入れるには、福島県のプログラムをコピーして奈良県と高知県のデータを入れれば時間の短縮になります。命令ブロックをコピーするやり方は巻末の「Mind Render 2 を便利に使うヒント」に説明してあります。



グラフを見る人がこのプログラムを使う時に、ボタンを切り替えて自分の見たい統計を見ることができたら便利ですね。そのため次に4つのボタンをオブジェクトリストに追加します。左図を参考にしてください。ボタンを追加するにはオブジェクト追加ボタンをクリックしてから、ツールタブで追加してください。



それぞれのボタンは3つの県について特定のデータ（例：Facebook ユーザ数）を呼び出して、3D グラフを描き出します。例として下図の Facebook ユーザ数についてプログラムを見てみましょう。

このプログラムでは変数を使ってデータの呼び出しています。変数「行」は初期値を1として命令を実行していきます。変数「行」が1の時には、（県別統計1行目2列の値）/ 100000 の大きさのグラフが、3秒で描画されます。

1 行目の各列が何を示しているかはさきほどの県別統計表を参照すればわかります。以下に1列目だけを抜粋して再度示します。

	1 列目	2 列目	3 列目	4 列目
	県名	Facebook ユーザ数	コンビニ店舗数	人口
1 行目	福島県	400000	889	2040000



左図の命令により変数「行」の値に1が加算され、変数「行」の値が2になって、次の命令が実行されます。

今度は2行目の各列のデータを使って3Dグラフを描きます。2行目のデータは下のようになっています。

2行目	奈良県	310000	456	1399000
-----	-----	--------	-----	---------

変数「行」にまた1が加算され、今度は変数「行」の値が3になって、次の命令が実行されます。今度は3行目の各列のデータを使って3Dグラフを描きます。3行目のデータは下のようになっています。

3行目	高知県	150000	286	766000
-----	-----	--------	-----	--------

これでプログラムの実行は完了しました。左図はこのプログラムを実行した結果です。



3つの県のFacebookユーザー数の違いがわかりますね。

コンビニ店舗数、人口でもプログラムの基本のしくみはFacebookユーザー数と同じです。Facebookユーザー数のプログラムをコピーして数値を入れ替えて使えば時間短縮になります。

変更が必要な箇所は元データのある数で割ってグラフの大きさを調整するところです。元データは下図のように数値の範囲が違います。

2列目	3列目	4列目
Facebook ユーザー数	コンビニ 店舗数	人口
400000	889	2040000
310000	456	1399000
150000	286	766000

そのため3Dグラフの大きさを指定するところが以下のように違ってきます。



Facebookユーザー数



コンビニ店舗数



人口

上の3つの図を参考にしてコンビニ店舗数と人口の3D グラフを描くプログラムを完成させてください。



さて、もう1つ残っている Facebook ユーザの割合は3つの県の Facebook ユーザが県民 20 人当たり何人いるかを示すものにしたいと思います。

したがって、 $(\text{Facebook ユーザ数}) / (\text{人口})$ を計算してグラフの大きさとすればよいですね。下図を参考にしてプログラムを作ってください。



4つのボタンのプログラムの実行結果を下に示します。

Facebook ユーザ数



コンビニ店舗数



人口



Facebook ユーザの割合



Facebook ユーザ数、Facebook ユーザの割合の 2 つの 3D グラフからわかることは、Facebook ユーザ数の比較では高知県は他の 2 県に比べて数が少ないけれども、Facebook ユーザの割合で見ると他の 2 県とあまり差がないことがわかります。コンビニ店舗数の割合ではどうなるでしょうか？ 練習問題として自分でやってみて結果を調べてみましょう。

発展課題

「都道府県別統計とランキングで見る県民性」 (<https://todo-ran.com>) には他にも面白いデータがいろいろと紹介されています。自分で見つけたデータで 3D グラフを描いてみましょう。

5. 世界人口推移

このラボでは日本地図の代わりに世界地図を使っています。

世界の統計データはネットで数多く探することができますが、ここでは次のサイトにあるデータを使うことにします。

総務省統計局「世界の統計 2023」 (<http://www.stat.go.jp/data/sekai/0116.html>)

このサイトでは様々なデータが紹介されていますが、「世界人口の推移 (1950~2050)」を元データとして使ってみましょう。下の表は 1950 年から 2010 年までの各地域のデータです。単位は 100 万人です。「北アメリカ a」となっているのは米国やカナダなど北米の主要な地域を指します。

地域名	アジア	北アメリカ a	南アメリカ	ヨーロッパ	アフリカ	オセアニア
1950	1,405	173	169	549	228	13
1960	1,705	205	220	605	283	16
1970	2,142	231	287	657	363	20
1980	2,650	254	361	694	476	23
1990	3,226	280	443	721	630	27
2000	3,741	312	522	726	811	31
2010	4,210	343	591	736	1,039	37

表：世界人口の推移（1950～2010）

地図上でグラフを描くために必要な座標も必要です。各地域のおよその座標は下表のようになります。

地域名	アジア	北アメリカ a	南アメリカ	ヨーロッパ	アフリカ	オセアニア
x	20	-22	-14	2	0	30
y	0	0	0	0	0	0
z	2	4	-8	5	0	-10

表：座標データ

これから作るプログラムは、「世界人口推移」ボタンを押すことでグラフを描き始めて、「次に進む」ボタンを押すことで次々に地域別のデータが世界地図上にグラフに表示されるというものです。

最初に「世界人口の推移」のデータを入力します。ここまでのラボでは、グラフのデータは命令ブロックに直接手入力してきましたが、データの数が増えると大変です。そこで、外部のスプレッドシートに入力したデータを Mind Render に読み込むことで、入力の手間を省きましょう。

始めに、Google スプレッドシートで、以下のようなデータを入力します。

ファイル名：世界の人口推移

シート：シート 1

世界の人口推移 ☆ 📄 ☁

ファイル 編集 表示 挿入 表示形式 データ ツール 拡張機能 ヘルプ

🔍 🔄 🏠 📏 100% ▼ | ¥ % .0 .00 123 | デフォ... ▼ | - 10 + | B I

A12 ▼ | 🏠

	A	B	C	D	E	F	G
1	地域名	アジア	北アメリカ a	南アメリカ	ヨーロッパ	アフリカ	オセアニア
2	1950	1,405	173	169	549	228	13
3	1960	1,705	205	220	605	283	16
4	1970	2,142	231	257	657	363	20
5	1980	2,650	254	361	694	476	23
6	1990	3,226	280	443	721	630	27
7	2000	3,741	312	522	726	811	31
8	2010	4,210	343	591	736	1,039	37
9							
10							

+ ≡ シート1 ▼

「世界の人口推移」を共有 ⓘ ⚙

アクセスできるユーザー

 @gmail.com オナー

一般的なアクセス

 リンクを知っている全員 ▼
リンクを知っているインターネット上の誰もが閲覧できます 閲覧者 ▼

 **リンクをコピー** **完了**

次に、このシートを共有します。

「ファイル」メニューから、「共有」を選択、続いて「ほかのユーザーと共有」を選択します。

左図のような画面が開きます。「リンクを知っている全員」が閲覧できるようにし、「リンクをコピー」ボタンをクリックします。

「完了」ボタンで画面を閉じます。

データセット

Enter text...

人口の推移 ▼

設定

+ 追加

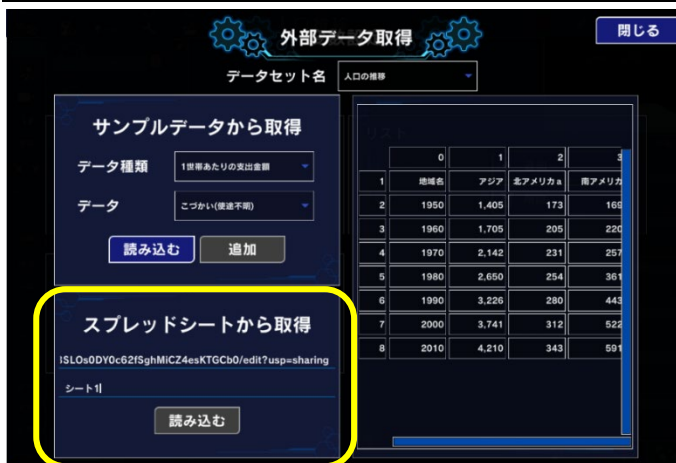
削除

次に、Mind Render で、「3D で日本を知ろう」の「5.世界人口推移」ラボを開きます。

画面左下の「変数」ボタンから変数設定画面を開きます。

データセット欄に、「人口の推移」というデータセットが

あらかじめ設定されているので、「設定」ボタンを押します。



外部データ取得画面が開きます。

「スプレッドシートからの取得」欄の「スプレッドシート URL」に、Ctrl+V で先ほどコピーした URL を貼り付けます。

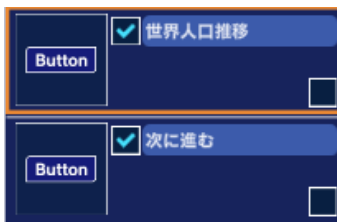
「シート名」には、データを入力したシートである「シート 1」と記入します。

「読み込む」ボタンをクリックすると、右側にスプレッドシートの内容が表示されます。

続いて、このデータを使用してグラフを表示します。

ユーザが便利に操作できるように次のような機能を取り入れます。

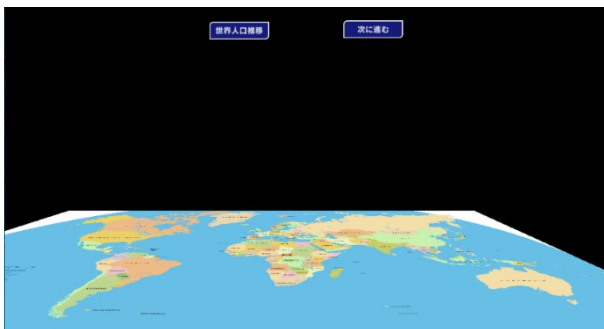
- ボタンを押すとグラフの描画をスタートする
- 別のボタンを押すと次のデータのグラフを表示する



以上の 2 つの機能のために左図の 2 つのオブジェクトを追加しておきます。

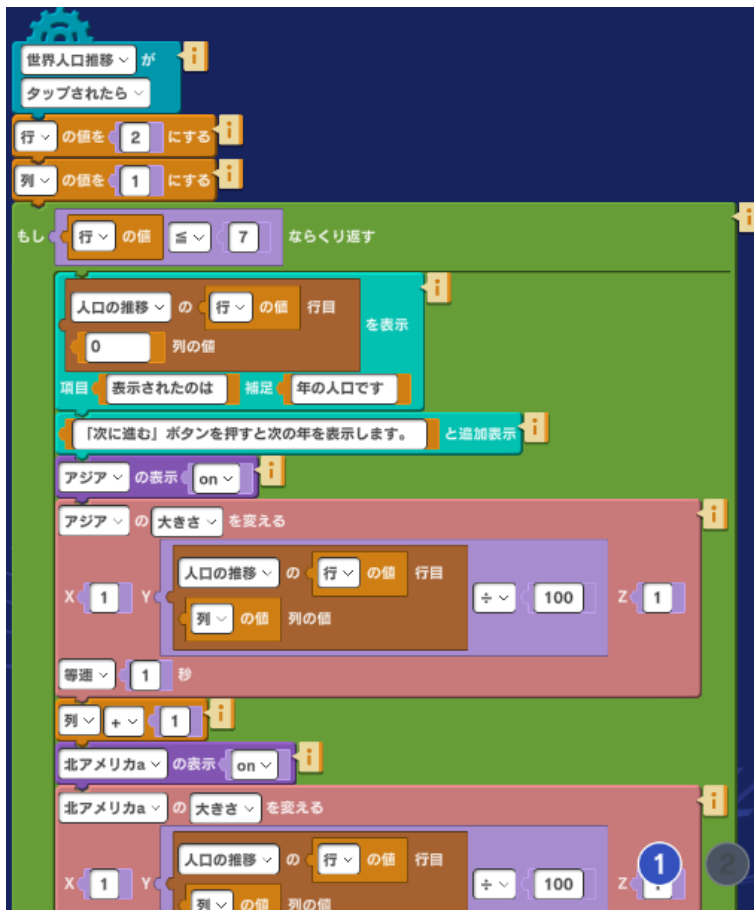
「世界人口推移」ボタンを押すと、世界人口の推移データを使ったグラフを描き出します。

「次に進む」ボタンを押すと、次のデータを使ってグラフを描きます。



ボタンとテキスト四角形の配置は左図のようにします。

グラフを描くプログラムは「世界人口推移」ボタンの中に作ります。



左図はプログラムの一部です。いくつかの部分に分けて以下に説明します。

プログラム全体は「世界人口推移」ボタンがタップされたら実行するようになっています。実行すると最初に2つの変数「行」と「列」の初期値を

「行」は2、「列」は1にします。この2つの変数はあらかじめ設定しておく必要があります。（変数の設定の仕方については、ちょっとお勉強>ドローンを動かしてみよう>3.ドローンゲーム ラボを参考にして下さい）



「行」の初期値を2にするのは元となるスプレッドシートの1行目が見出しとなっているからです。「列」の初期値を1にするのは、列の場合は0列目が見出しとなっているからです。

1 列目
↓

2 行目 →

	0	1	2	3
1	地域名	アジア	北アメリカ	南アメリカ
2	1950	1,405	173	169
3	1960	1,705	205	220
4	1970	2,142	231	257
5	1980	2,650	254	361
6	1990	3,226	280	443
7	2000	3,741	312	522
8	2010	4,210	343	591

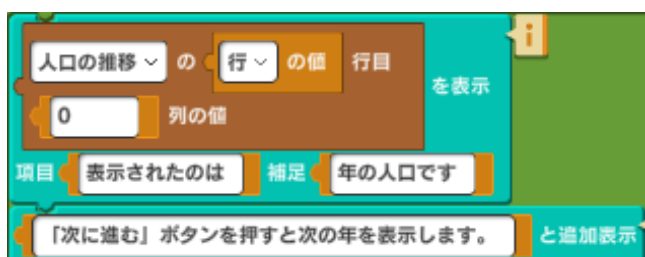
左図を見るとスプレッドシートの2行目、1列目が最初のデータのある場所であることがわかります。

サンプルプログラムでは 2 行目の 1950 年から 7 行目の 2000 年まで、変数「行」が 7 になるまでグラフを描く命令を繰り返します。



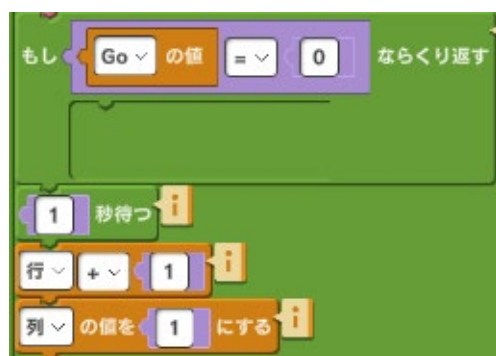
グラフ描画のプログラムは左図のようになっています。グラフの大きさを決めるデータは「人口の推移」データセット 2 行目を読み出します。読み出した数はグラフ描画の最小値 0 ～最大値 10 の範囲に収めるために適当な数で割ります。この数を見つけるには試行錯誤でいろいろな数を試してグ

ラフの大きさの差がわかりやすく、一番大きなデータが画面からはみ出さないようにします。ここでは 100 で割っています。



グラフを描き終えたら、数の命令を使って「表示されたのは () 年の人口です。」と画面に表示してユーザにわかりやすくしています。

(年) は「座標」データセットの行を呼び出して使っています。



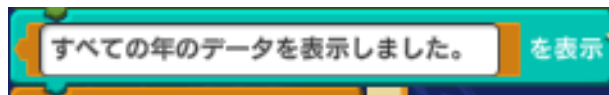
それぞれのグラフが表示された後、グラフの大きさを見る時間を作るために「次に進む」ボタンが押されるまでは先に進まないようにしてあります。変数 Go をあらかじめ設定しておいて、この変数の値が 1 になるまでは何もしないでただ繰り返しているようにしてあります。



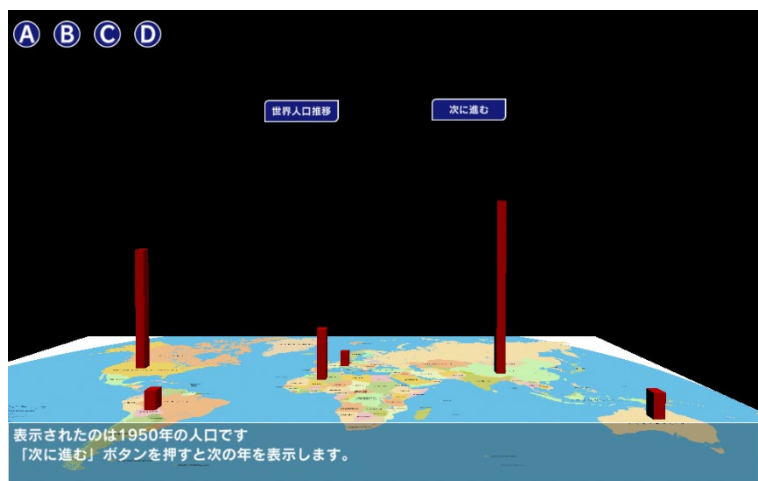
変数 Go の値を変えるプログラムは「次に進む」ボタンの中にあります。ボタンが押されたら 変数 Go の値を 1 にします。1 秒間待ってすぐに 値を 0 に戻しておきます。

変数 Go の値が 1 になったら、1 秒待ってから行の値を 1 つだけ大きくして繰り返しの最初に戻って繰り返し処理を実行します。

「行」の初期値は 2 なので、「行」は 2 から 7 まで繰り返しを実行します。



すべてのグラフを表示したら左図のようなメッセージを表示します。



プログラムを実行すると左図のようになります。

ネット上には世界統計の興味深いデータがたくさん見つかります。自分で見つけたデータで世界地図の上にグラフを描いてみましょう。

発展課題

- ・上記プログラムでは、2010 年のデータが表示されません。プログラムを変更して、2010 年のデータまで表示してみましょう。
- ・グラフの元になった地域名や数値をテキストで表示するようにしてみましょう。

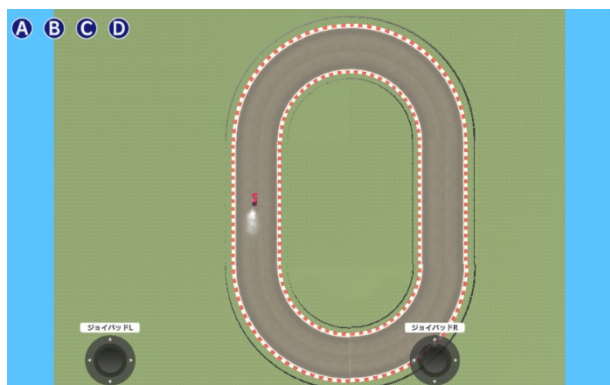
12. ちょっとお勉強 3：「サーキットを走ろう」

ラボ「サーキットを走ろう」ではレースカーを走らせます。ジョイパッドを使って操縦したり、レースカーの動きを自動記録したり、プログラムによって自動走行することにも挑戦してみます。



ラボメニューから「ちょっとお勉強」を、続いてラボ「サーキットを走ろう」の「車の操作」を選んでスタートしましょう。

1. 車の操作



ラボ「1.車の操作」をオープンしたら早速スタートボタンを押してプログラムを実行してみましょう。楕円形のサーキットコース左側に赤い車が1台見えます。ラボ「車の操作」ではこの車を画面下にある2つのジョイパッドを使ってコースを走行させます。

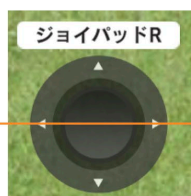
2つのジョイパッドの使い方は下図の通りです。

上向：前方向へ加速



下向：後方向へ加速

左向：左へカーブ



右向：右へカーブ

パソコンでは1度に2つのジョイパッドの操作は難しいので、可能であればタブレットやタッチ画面を備えたパソコンで試してください。

最初は操作が難しいかもしれませんが、何回かサーキットを周回すればすぐに慣れます。どれくらいのスピードでカーブに侵入して、どれくらい車の向きを変えたらよいかを試しながら車の操縦を練習しましょう。



操縦に慣れてきたら、ラボ「車の操作」のプログラムを調べてみましょう。ジョイパッドLのプログラムは左図のようになっています。

ジョイパッドLを上または下向きに操作したら、車をz方向（進行方向）へジョイパッドの上下の値（最大値：1、最小値：-1）で移動させます。

ジョイパッドLを上向きに操作した時は車は進行方向に進みます。下向きに操作した時は車を後向きに進ませます。



左図はジョイパッドRのプログラムです。

このプログラムではジョイパッドRで車の向きを変えています。ジョイパッドRを右向きに操作すると、車の前輪はy方向にジョイパッドを動かした値（最大値：1.0、最小値：-1.0）だけ右に向けます。左向きに操作すると（負の値が入ると）車の前輪は左に向きます。

ジョイパッドRの左右の値（-1.0～1.0）の値を使って車のタイヤの向きを変えます。その際、-1.0～1.0では反応が大きすぎるためこの命令にあるようにジョイパッド右コントローラの値を2で割ってその範囲を-0.5～0.5として使っています。



車のスタート位置や向きは、シーン編集画面で設定されています。

車(赤)			
座標	-23.33	0.16	0.00
角度	0.00	0.00	0.00
大きさ	1.00	1.00	1.00
グループ			

向き（角度）は x:0、y:0、z:0 で、画面の上方向にコースに沿った方向に車は向いています。y座標の値を変えることで車の向きを変えることができます。

発展課題



ジョイパッドは上下左右に $-1.0 \sim 1.0$ の範囲で動きますが、値を変更することで車の速度や操作性を変えることができます。下図はその一例で上下のジョイパッドの値を 2 で割ることで、スピードを遅くして落ち着いて操作できるようになっています。上下、左右のジョイパッドの値をいろいろ試して自分にとって使いやすい設定を見つけてみましょう。

2. 記録と再生



ラボ「2. 記録と再生」では視点を車の後ろに置いて車を追いかけるようにしてサーキット走行を楽しめます。下図はラボをオープンして、スタートボタンを押したところです。

ラボ「1. 車の操作」と同じように画面下の2つのジョイパッドで車を操縦してサーキットを走行します。



視点の変更はカメラオブジェクトの中の左図のプログラムで行っています。上図の視点のように車の後方、斜め上から車を見るには左図のようにカメラの座標と角度を設定します。座標の y 、 z や角度の x の値を変えて視点がどのように変わるかも試してみてください。車の前から見たり、下から見上げたりすることもできます。



左図はラボ「記録と再生」のサーキット全体を上から見たところ。走行する時の参考にしてください。中央で立体交差になっているのがわかります。

このラボでは車の走行を動画撮影してみましょう。



プログラム実行画面の左上に赤いボタンがあります。このボタンを押すと、実行中の様子が動画で撮影されます。



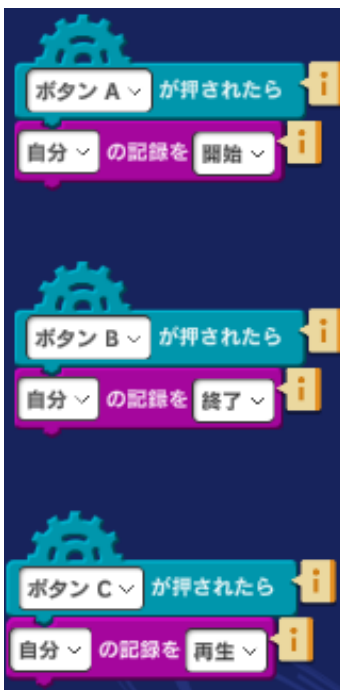
もう一度このボタンを押すと、撮影が終了します。

動画は、お使いの OS のデフォルトの動画保存フォルダに保存されます。

注意： Windows で録画機能を使用する場合

Microsoft ストアからダウンロードした Mind Render には録画機能がありません。Windows で実行画面を録画する場合は、以下の手順に従ってください。

1. ストアからダウンロードした Mind Render でプログラムを作成する
2. Mind Render ホームページから Mind Render をダウンロードする
3. 2 の Mind Render で実行画面を録画する



もしくは、車オブジェクトに入っている左図のプログラムで車の走行を自動記録して、再生することもできます。

ボタン A が押されたら自分（車）の動きを開始して、ボタン B が押されるまで記録を続けます。ボタン B が押されたら記録を完了して、ボタン C が押されたら記録した動きを再生します。

動きの記録ができるのはジョイパッドで制御できるものの動きだけです。記録はビデオ撮影のようなものではなく、実際に記録されるのはジョイパッドの操作データとなります。そのために再生は完全に元の動きとは一致しない場合もあります。

記録できる最長時間は 2 分間です。

発展課題



カメラの視点を変えて運転席から眺めを見ながら運転してみましょう。左図のような視点にするにはカメラをどのように設定すればよいでしょうか？

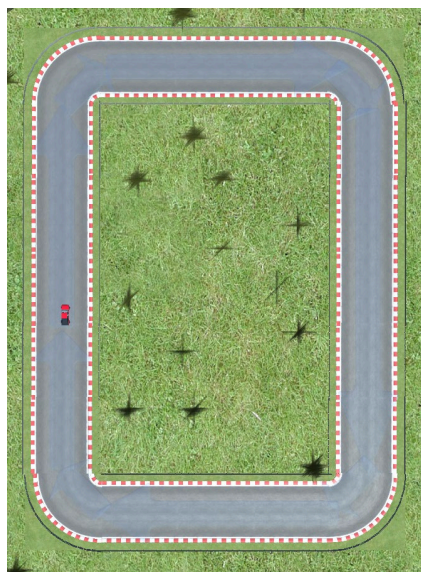
カメラの座標、角度をいろいろに変えて見つけてみましょう。

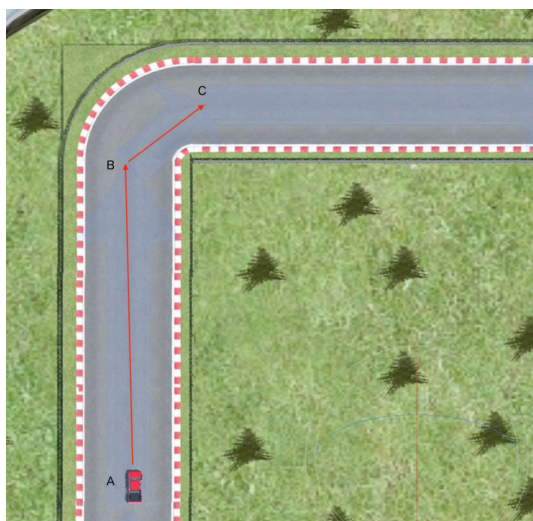
カメラ			
座標	-23.50	1.60	-26.30
角度	12.00	0.00	0.00
大きさ	1.00	1.00	1.00
グループ			

左図は一例です。座標や角度などを決める参考にしてください。運転席の他、車の上にカメラを乗せて走る、車のボンネットにカメラを乗せる、上空のカメラで車の走行を追いかけるなどいろいろなカメラアングルを試してみましょう。

3. 自動運転

ラボ「自動運転」ではサーキットコースを自動運転させます。下図のように四角形のコースです。このコースには自動運転のための矢印がコースに沿って置かれています。矢印は幅広で薄い色をつけてあります。車はこの矢印の指し示す方向に走ることでコースから外れることなく走行します。





左図はコースの一部です。

A から B (矢印の先端)、B から C (矢印の先端) のように車は移動します。赤い矢印は説明用です。

コースに沿って置かれた矢印には ① から ⑨ までの番号で名前がついています。

このラボでは、車の向きをコントロールするために三角関数を使います。三角関数を使う時は、三角関数用の角度が必要になります。「矢印」や「車」といったオブジェクトの向きとの違いは、以下のようにまとめることができます。

- ・ 3D 角度 (矢印や車の角度) : 地図の北を起点にして右回りで計測する角度
- ・ 三角関数の角度 : 地図の東を起点にして左回りで計測する角度



	三角関数の角度 $90^\circ - (A)$	3D 角度 (A)
①	90°	0°
②	45°	45°
③	0°	90°
④	-45°	135°
⑤	-90°	180°
⑥	-135°	225°
⑦	-180°	270°
⑧	-225°	315°
⑨	-270°	$360^\circ (0^\circ)$

このラボでは、こちらの角度が必要

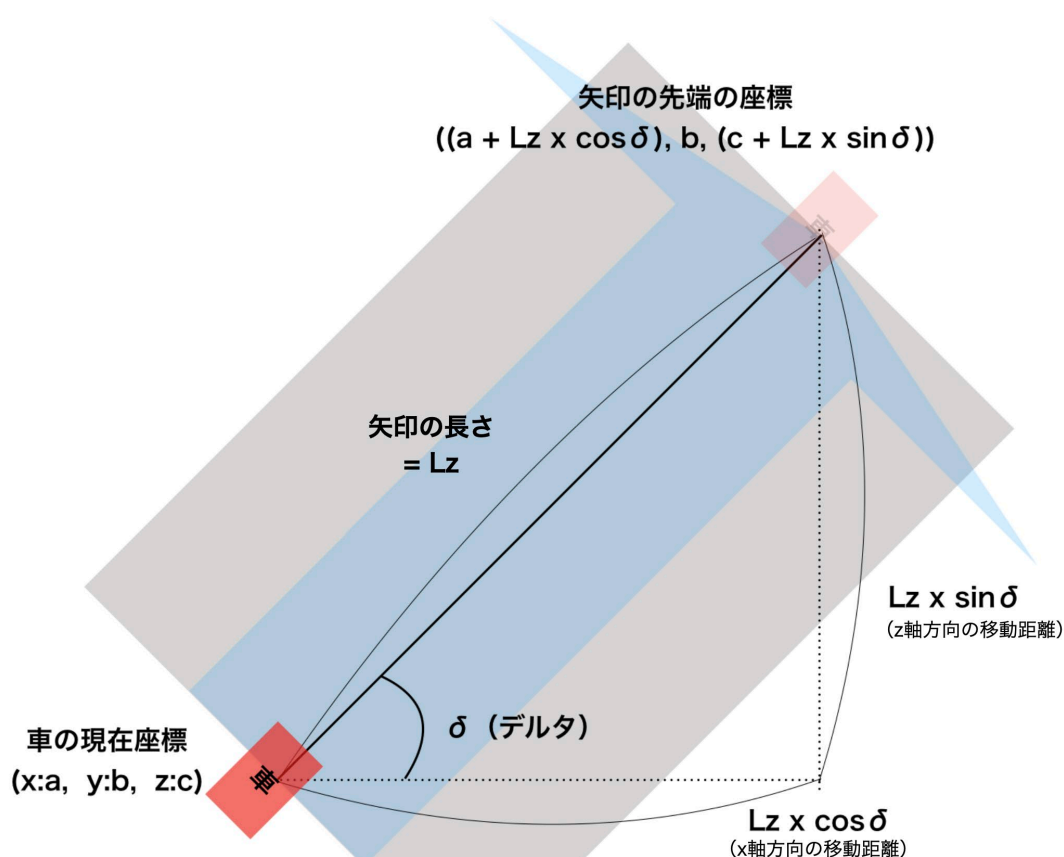
前ページの表にある ①～⑨ の矢印の「三角関数の角度」を δ (デルタ) とすると、自動運転を実現する方法は次のようになります。

車が矢印に触れた時に、

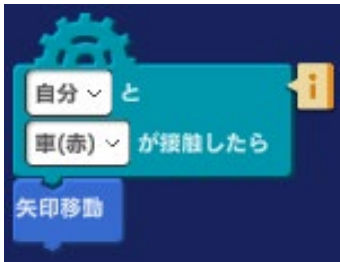
車の方向 (y 軸の角度) を δ にする (矢印と同じ方向を向く)

車を矢印の先端の座標に移動する

矢印の先端の座標は 三角関数 を使って下図のようにして求めます。



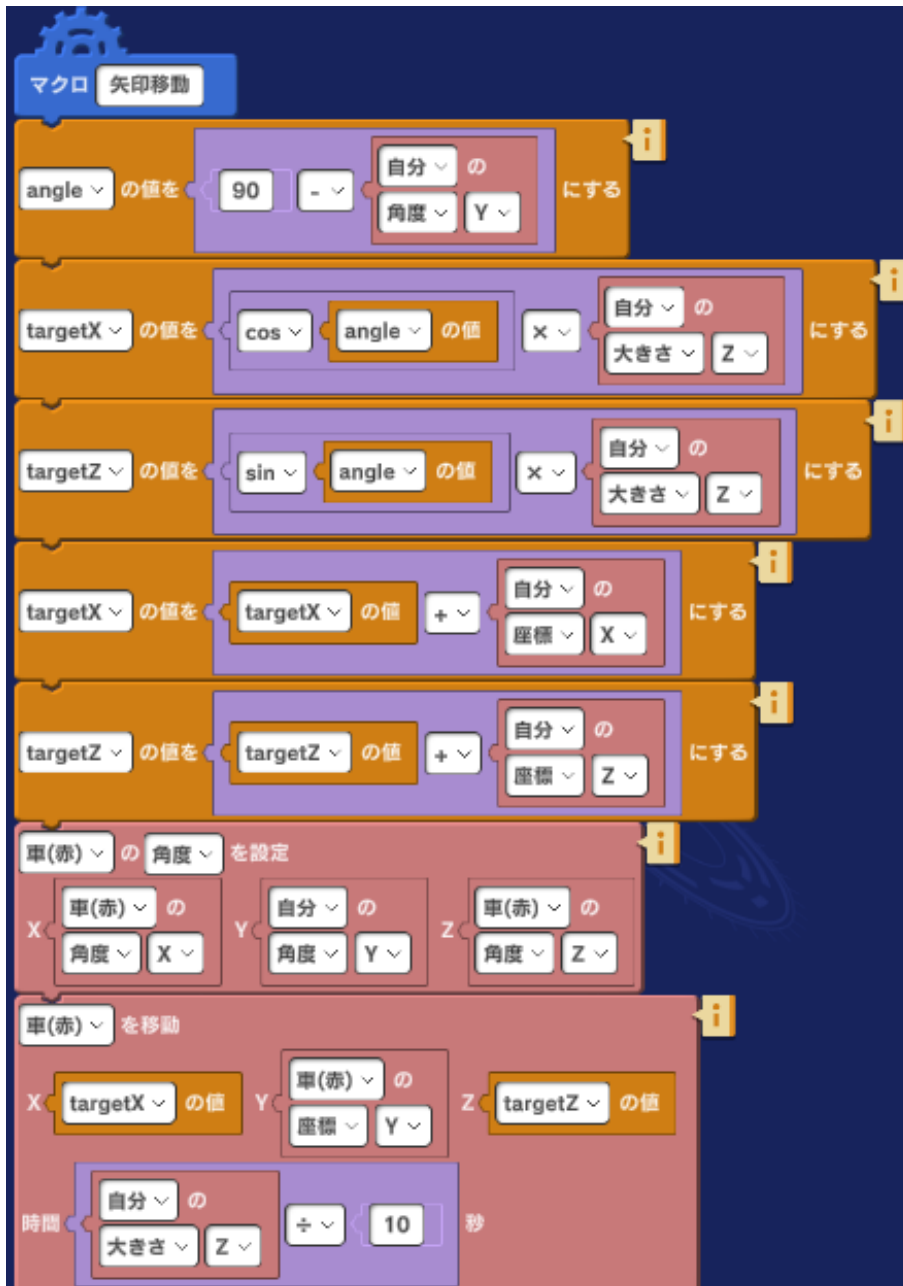
このラボでは上図のような計算をしてくれる「矢印」が用意されています。ここではすでにコース上に矢印が敷き詰められていますが、自分でこの矢印を使ってプログラムを作ることできます。ただし、同じ水平な平面（高低差のない面）でのみ有効であることに注意してください。



9 個の矢印オブジェクトの中には左図のプログラムが格納されています。自分（矢印）と車が接触したら、「矢印移動」を呼び出して実行します。「矢印移動」は自動運転をするためのプログラムで自動運転式というオブジェクトに格納されています。



矢印オブジェクトが呼び出す「矢印移動」というプログラムを見てみましょう。自動運転式オブジェクトの中にあります。



矢印移動プログラムでは「マクロ」という命令を使っています。

マクロは「複数の命令」を使って新しい命令を作るために使われます。マクロの中で実際に処理する内容を定めることを「定義する」といいます。

マクロを使っていったん定義した処理は、大きなプログラムであってもマクロ名を呼び出すだけで何度でもどこでも使うことができます。

マクロを使えば、大きなプログラムを繰り返し作る手間が省けるためたいへん便利です。

「矢印移動」という マクロ 命令の中身は下図のように定義されています。内容は上で説明した自動運転のやり方を具体的にプログラムで実現したものとなっています。

最初の命令は 矢印の「3D 角度」を「三角関数の角度」に修正するために、 90° から「現在の y 軸の角度」を引いています。**エラー! ブックマークが定義されていません。** ページの図に δ という角度がありました、ここで求めている angle は δ のことを指しています。

angle (三角関数の角度) = $90^\circ - 3D$ 角度



2 番目の命令、車を x 軸方向に移動させる距離となります。その計算式は以下のようになります。

targetX = cos (angle) x 矢印の長さ



3 番目の命令は、車を z 軸方向に移動させる距離となります。その計算は以下のようになります。

targetZ = sin (angle) x 矢印の長さ



4 番目の命令は、先ほど求めた target X に「現在の矢印の x 座標」を加えて、実際に車が目標とすべき x 座標を求める計算を行なっています。

target X = 先ほど求めた targetX の値 + 自分 (矢印) の x 座標



5 番目の命令は、さきほど求めた target Z に「現在の矢印の z 座標」を加えて、実際に車が目標とすべき z 座標を求める計算を行なっています。

target Z = 先ほど求めた target Z の現在の値 + 自分 (矢印) の z 座標



* 上記で x と z 軸方向だけを計算しているのは y 軸方向は天地（上下）となるためです。ここでは水平面の動きだけを考えています。

6 番目の命令は、「車の向き」を「矢印の向き」に揃えるためのものです。車の角度を自分（矢印）の角度にセットします。



7 番目となる最後の命令は、上で求めた x 座標 (targetX)、z 座標 (targetZ) に車を移動するためのものです。移動に要する時間には、

自分（矢印）の z 軸の大きさ ÷ 10（秒）



を指定しています。10 で割っているのはスピードを調整するためです。割る数値を大きくすれば速く、小さくすれば遅くなります。

さらにこの マクロ命令では矢印がコース上に配置された時の座標、大きさ、角度をプログラムの中で呼び出して使っているため、9 個の矢印を置くたびに座標を調べたり、大きさを調べたりする手間がありません。矢印はコースのどこにでも、直線でもカーブでも好きな場所に置くことができます。矢印は置く場所を変えても、長さを変えてもプログラミング無しで利用できます。

こうした工夫によって手間と時間を節約することができます。



矢印オブジェクトの
角度、大きさ、座標を
プログラムが読み出して使っている

自動運転の仕組みが理解できたら、実際に実行して車が自動走行する様子を見てみましょう。どの矢印の場合も矢印の先端に向かって車が走っていく様子が見えるでしょう。

発展課題

以下のようなプログラムに挑戦してみましょう。

- ・コースの高低差に対応するために、コースの x 座標の角度を加えたプログラムを作る
- ・A~D ボタンを押してカメラの視点を切り替えるようにする
- ・ラボ「5. コースづくり」では自分でコースを作ることができる。自分で作ったコースにラボ「3. 自動運転」の矢印を置いて自動運転にチャレンジしてみる

さらに、ラボ「AI と機械学習」では、車が障害物を自分で避けて走行する AI プログラムを実装した自動走行を見ることができます。A、B、C ボタンを押して視点を切り替えながら AI 自動走行を楽しんでください。



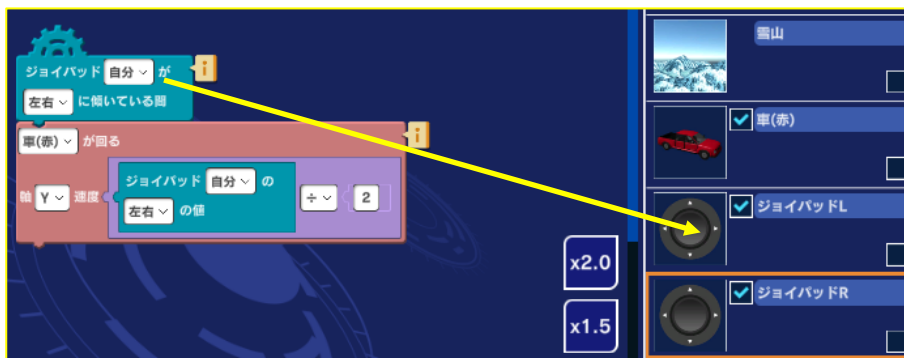
AI 自動走行のプログラムは Mind Render の関連商品「Mind Render/AI Drill」を使って作られています。「Mind Render/AI Drill」の詳しい説明は[ウェブサイト](#)をご覧ください。

4. 車の操作 (雪山)

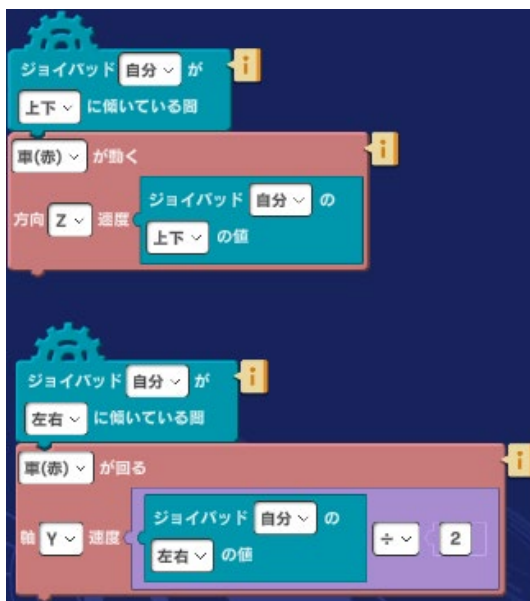


ラボ「4. 車の操作 (雪山)」では雪山を巡ります。雪山の景色を楽しみながら走行してください。ラボを開いて実行すると左図のようになります。2つのジョイパッドを使って車を操縦するのはラボ「車の操作」や「記録と再生」と同じです。

このラボをパソコンでも使いやすくするためにジョイパッド1個だけで操縦ができるようにプログラムを変更してみましょう。ジョイパッドLは上下の動きだけ、ジョイパッドRは左右の動きだけに使っているのでジョイパッド1個に上下左右の動きをまとめることができます。ここではジョイパッドLにジョイパッドRのプログラムをまとめる方法を説明します。



左図は「ジョイパッドR」オブジェクトを開いたところです。プログラムをドラッグドロップで「ジョイパッドL」にコピーします。



コピーが終わったら、左図のように2つのプログラムがジョイパッドLの中にあるか確認します。ジョイパッドL上下の動きで前進・後退、ジョイパッドL左右の動きで左右へ向きを変えることができるようになりました。

発展課題



雪山コースでの走行に慣れてきたら、コース上にいろいろな障害物を置いてスリルのある走行を楽しんでみましょう。例として下図ではスタート直後の位置に廃車を置いたところ です。



左図は雪山コースの全体となります。コース全体を見て走行がより面白くなるようにいろいろな障害物をコース上に配置してみましょう。

5. コースづくり



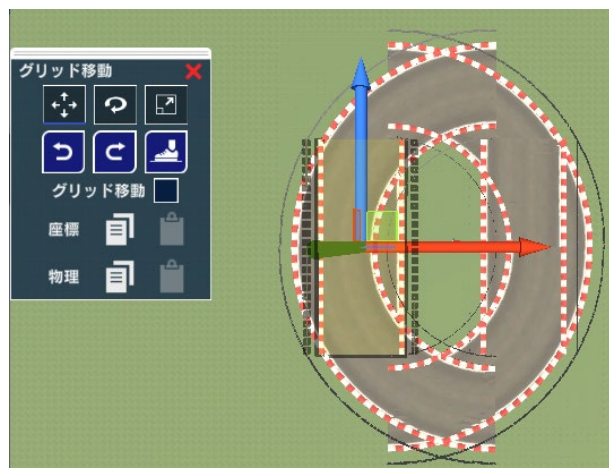
ラボ「5. コースづくり」では自分でサーキットコースを作ることができます。

左図はラボを開いて、シーン編集画面にしたところ です。

このラボでは自分でサーキットコースを作るためのコース部品がいくつか、あらかじめ入っています。コース部品が足りなければいくつでも追加することができます。

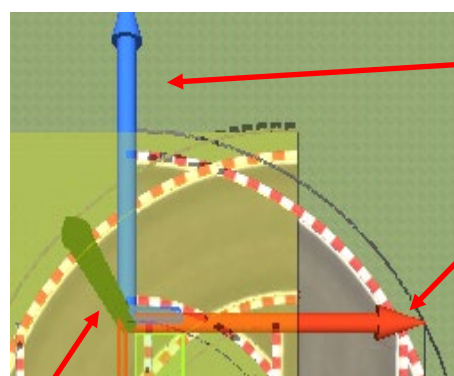
ここでは簡単な例として円形のサーキットコースを作ってみましょう。

全てのコース部品は芝生の上で動かすことができます。コース部品を並べてサーキットを作る手順を下に説明します。



1) シーン編集画面で、移動したいコース部品を選択します。オブジェクトリストで選択するか、直接コース部品をタップ(クリック)します。

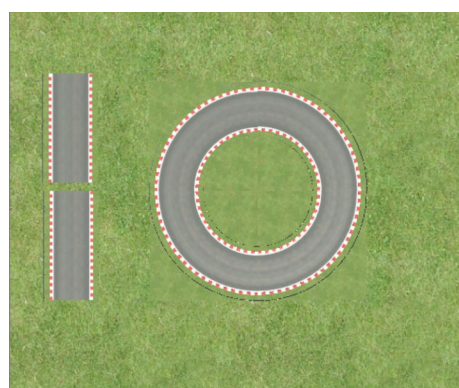
2) 選択したオブジェクトに左図のような矢印が表示されます。この矢印はオブジェクトを動かすためのものです。



青矢印はz方向（画面の手前から奥）へ動かすためのもの

赤矢印はx方向（画面の左右）へ動かすためのもの

緑矢印はy方向（上下方向）へ動かすためのもの（このラボでは使わない）



3) 矢印を使ってこのミッションに用意されている4つのカーブコース部品を組み合わせると円形コースができます。



4) 自分の好きな車をオブジェクト一覧から追加しましょう。

5) 車オブジェクトのプログラムはここまでのミッションを参考にして自分で作ってみましょう。



例えば左図のようにジョイパッドコントローラを追加してプログラムを書けばラボ「車の操作」「記録と再生」「車の操作(雪山)」でやったようにジョイパッドを動かして車を操縦することができます。その他にもラボ「自動運転」で使った矢印を置いて自動走行に挑戦することもできます。

発展課題



コンピュータによる自動運転の車とレースをしてみましょう。ラボ「自動運転」で使った自動運転矢印を使うこともできますが、ここではAIを使った自動運転車を使ってみましょう。

オブジェクト一覧のモデルタブで「その他」を選ぶと「AI 車」というオブジェクトが見つかります。このオブジェクトはあらかじめセンサーを使って自動運転するプログラムが組み込まれています。このAI車をコースに置いて走行させ、加えてコース上には障害物としてサッカーボールなどをいくつか置いておきます。



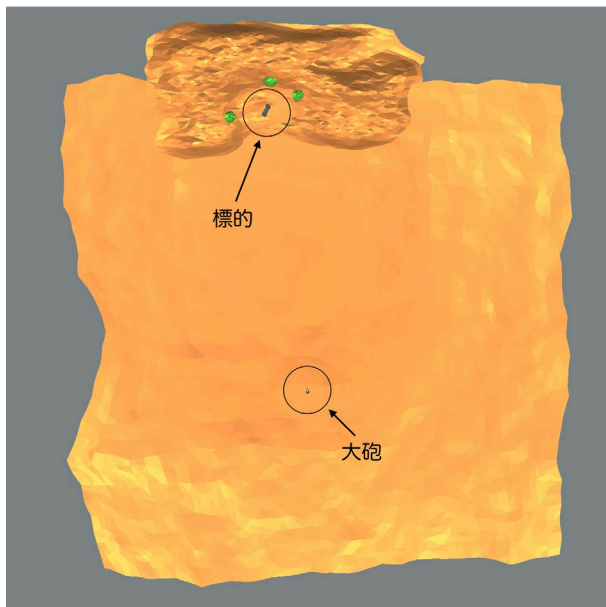
プログラムを完成させて実行した例が左図です。自分の車の操縦用にはジョイパッドコントローラを一つ使っています。

13. ちょっとお勉強 4：「大砲を撃ってみよう」

ラボ「大砲を撃ってみよう」では大砲を撃って標的を狙います。風が吹いたり、壁が出現したりとさまざまな条件で見事に標的に当てることができるでしょうか？



1. 砲弾を当てる 1



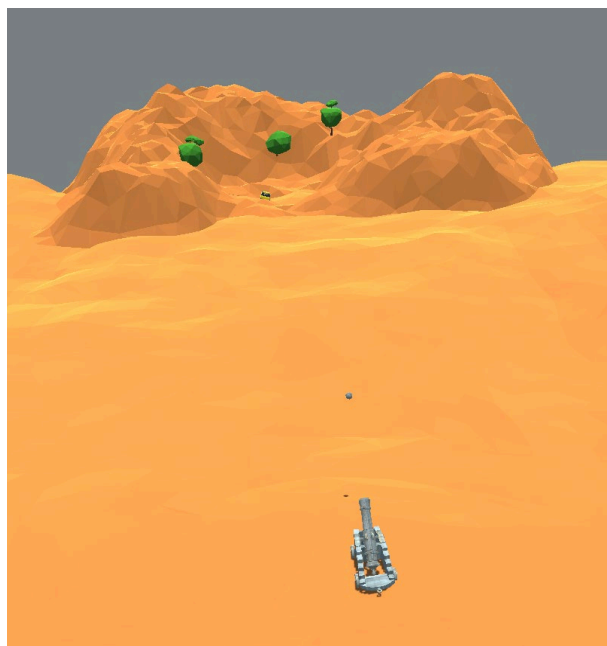
ラボ「1. 砲弾を当てる 1」は標的を狙って大砲を撃ちます。標的は廃車です。大砲から標的までのおよその距離を左図で確認しておきましょう。



このラボでは大砲の操作は A ボタン と B ボタンを使います。

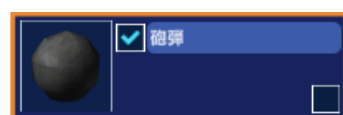
A ボタン：砲弾を大砲にセットする（大砲の砲身の中に砲弾を置きます）

B ボタン：砲弾を撃ち出す



左図は B ボタンを押して砲弾を撃ち出したところ
です。

B ボタンを押して砲弾を撃ち出すと、砲弾は標的
まで届きません。標的に命中するようにプログラム
を変更しましょう。



砲弾を撃ち出すプログラムは砲弾オブジェクトに格納されています。



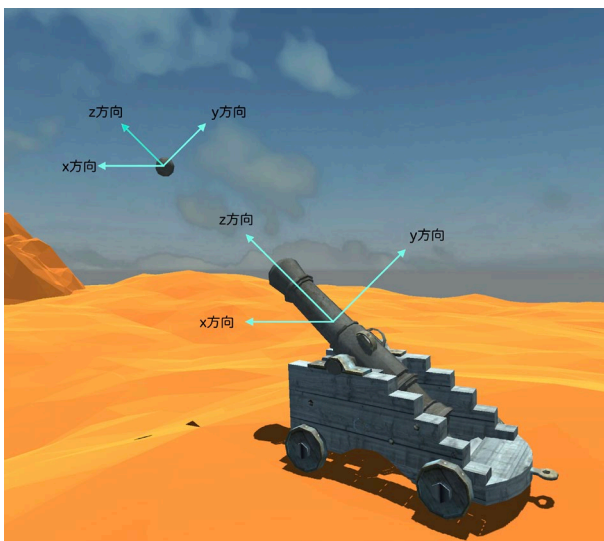
左図は A ボタンが押された時、砲弾を砲身の中に
置くプログラムです。砲身は本物のように見えます
が、中に本当に砲弾を入れておくことはできませ
ん。砲弾を砲身に置くとはいわば砲弾を空中に留め
ておくことです。砲弾が地面に落ちてしまわないよ
うに物理エンジンを off にします。砲弾の向きは
x、y、z すべてを砲身の向きに揃えます。



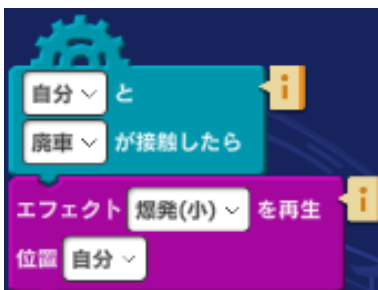
B ボタンを押して砲弾を撃ち出すプログラムが左図です。

本当の大砲であれば、火薬が爆発する力を使いますが、ここでは「力」を砲弾に加えて撃ち出します。力が有効になるよう物理エンジンを on にします。砲弾の大きさ、砲弾の重さを設定して、z 方向（砲身の方向）にどのくらいの力を加えるかを設定します。

砲弾が飛ぶ距離を変えるためには左図の矢印で示した箇所、押す力（z 方向）の数値を変更します。数値を大きくすればより大きな力で押すので遠くまで砲弾が飛びます。注意してほしいのは、ボタン B は砲弾が砲身の中に無くても押すことができるので、砲弾を正しい位置で発射するためには発射のたびに A ボタンを押して砲弾を砲身の中に置く必要があるということです。

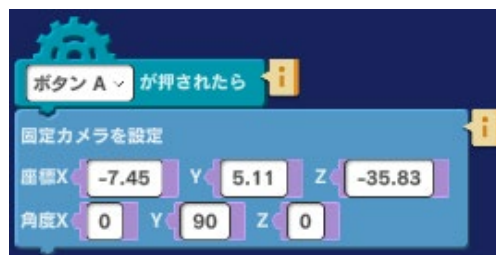


砲弾を飛ばす方向を x、y、z 座標でどのように指定するかは左図を参考にしてください。



砲弾にはもう 1 つ、左図のプログラムもあります。このプログラムは砲弾が標的の廃車に命中した時に爆発のエフェクトを再生するためのものです。左図のプログラムではエフェクトは「爆発（小）」を指定していますが、他のエフェクトに変更することもできます。

このラボではプログラムの途中でカメラの視点も変わっています。カメラのプログラムを調べてみましょう。



A ボタンが押されるとカメラの座標は

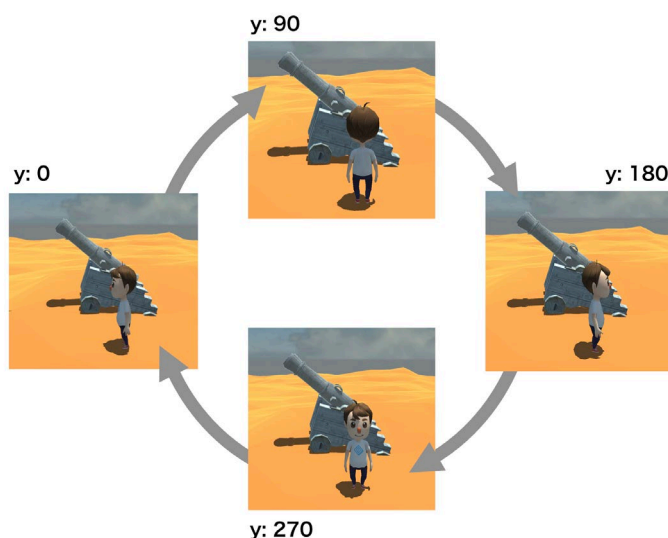
x: -7.45

y: 5.11

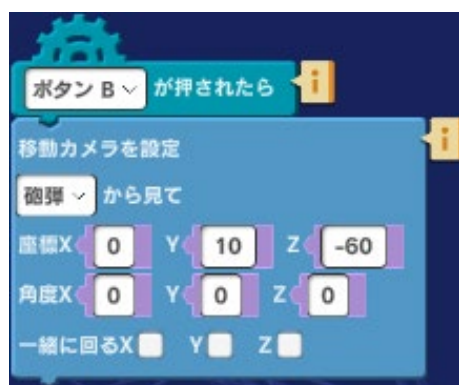
z: -35.83

の位置に設定されます。

固定カメラとは指定位置で固定されるカメラのことです。角度の設定で y: 90 となっているのはカメラを y 軸（地面から空の方に伸びる軸）を中心に回して、大砲の方を見るためのものです。



カメラと砲台の位置の関係については、左図の少年のところにカメラがあると思ってみてください。砲台の方を向くのはカメラの角度が y: 90 である時だということがわかります。



B ボタンが押されると、カメラは固定から移動に変更されて、取り付け位置が「砲弾」を基準に指定されます。座標や角度とは、砲弾に対してどのくらいの距離で、どの向きでカメラを取り付けるかを指定するものです。左図では y: 10 は砲弾の少し上、z: -60 は砲弾の後ろにカメラを取り付けています。カメラをもっと砲弾に近づけたり、砲弾を下から見上げるようにすることもできます。「一緒に回る x, y, z」にチェックを入れると、砲弾が回転するとカメラも同じように回転するようになります。

ます。

発展課題

押す力の値を変えると弾道が変化します。どのように弾道が変わるかを見るためにはカメラの位置を変えて弾道が最初から最後まで見えるような座標と角度を探してみてください。



さらに、弾道がよくわかるようにするために、弾道の軌跡を線を描いたどるることができます。このために使う命令は「効果」カテゴリーにある「ラインを描く」を使います。



2. 3つのターゲット



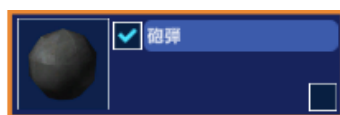
ラボ「2. 3つのターゲット」では標的の車が3台になっています。3つの標的を狙って次々に命中させるプログラムを作りましょう。

丘には左側、中央、右側にそれぞれ一台ずつ標的の廃車が置かれています。左図が3台の廃車が置かれた場所です。

プログラムは次のようになっています。

- A ボタンを押すと、砲弾を砲台へセットする（ラボ「1. 砲弾を当てる 1」と同じです）
- B ボタンを押すと、中央の標的へ向けて砲弾を発射
- C ボタンを押すと、左側の標的へ向けて砲弾を発射
- D ボタンを押すと、右側の標的へ向けて砲弾を発射

発射する度に A ボタンを押して砲弾をセットしてから次の標的を狙います。

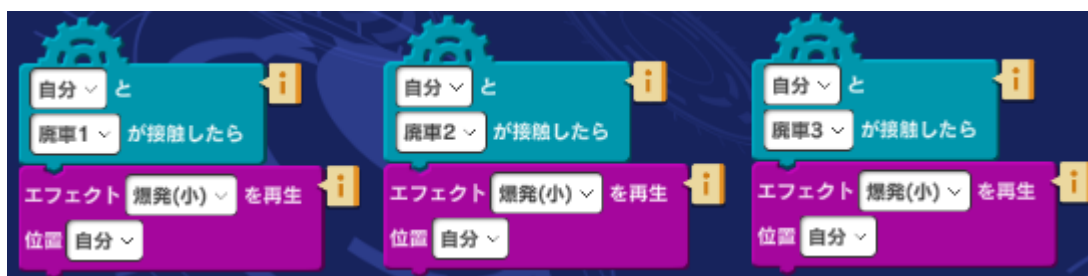


砲弾オブジェクトのプログラムで設定がどのようなになっているか見てみましょう。

3つの標的を撃つための設定は下図のようになっています。ボタン B、C、D のそれぞれで押す力の x、y、z 方向の設定に注目してください。これらの設定は一例です。別の設定でも命中させることができますので試してみましょう。



標的に命中した時のエフェクトもそれぞれの標的について設定が必要です。下図を参考にしてください。



発展課題

3つのターゲットを自動的に次々と狙って撃つプログラムに挑戦してみます。一例としてシグナルを使ったやり方を以下で説明します。実際に自分でもやってみましょう。

シグナルとは合図のことで、Mind Render の中でいくつかシグナルを先に決めておいて、必要に応じてシグナルを送信したり、受信したりして使います。

この発展課題の中では、砲弾を砲台に装填（指定位置に置くこと）するには、「装填」というシグナルを送り、最初のターゲットに向けて砲弾を発射する時は「1 番発射」というシグナルを送るなどします。



新しいシグナルを設定する時はブロックカテゴリーの下の方にある「拡張」をクリックします（左図）。

次に拡張設定画面（下図）で新しく設定したいシグナル名を入力して「追加」ボタンをクリックします。



- ・装填
- ・1 番発射
- ・2 番発射
- ・3 番発射



シグナルの送信、受信は「イベント」カテゴリーの中の2つの命令を使います。



左図のプログラムは全体の流れをコントロールするためのものです。このサンプルでは「発射ボタン」を追加して、このボタンをクリックすることで全体の処理がスタートするようになっています。

シグナルを順に送信することで、「装填」と「発射」を交互に実行しています。

それぞれのシグナルを送った後には8秒とか15秒の時間を取るようになっています。これは「秒待つ」命令を間に入れないとシグナルを次々に送ってしまい、例えば砲弾がセットされていないのに発射をしてしまうようなことが起きるためです。何秒待ったらよいかは自分で試して決めてください。



「装填」や「1番発射」などのシグナルが送信されると、受信側のプログラムが実行されます。左図は「装填」シグナルを受信した際の処理です。このラボのプログラムを使っています。



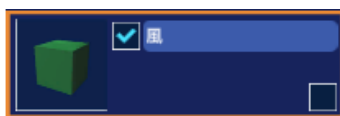
左図のプログラムもこのラボのものを使い、最上部の命令だけを「1 番発射シグナルを受信したら」に変更しています。2 番発射、3 番発射も同様にラボ「3 つのターゲット」のプログラムに変更を加えて作ります。

ラボ「2. 3 つのターゲット」の中には A ボタン や B ボタンが押されたことをきっかけに実行されるプログラムもありますので自分で作る時には変更が必要となります。注意してください。

3. 横風の影響



ラボ「3. 横風の影響」では、砲弾が飛んで行く間に横風が吹くようになっています。横風が吹く範囲はわかりやすいよう緑色の箱で示されています。箱の中が風が吹く範囲です。砲弾が発射されてこの中を通ると横風が砲弾を押して軌道に影響を与えます。どんな力が働いて、どのように調整しなければいけないかを調べないと標的に命中させることができません。



このラボでは「風」というオブジェクトが追加されています。このオブジェクトの中に風を吹かせるプログラムがあります。

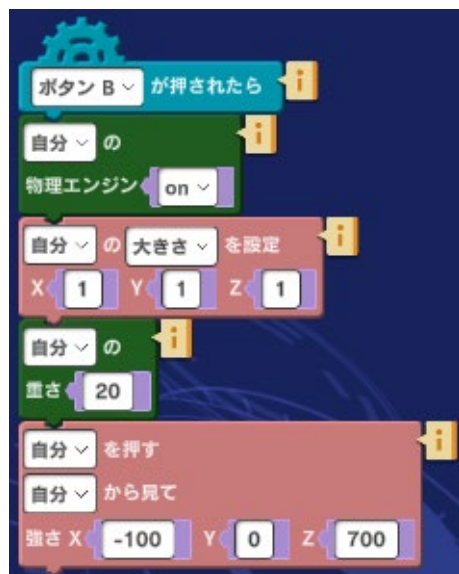


左図が風オブジェクトのプログラムです。1 番目のプログラムはスタートボタンが押された後の設定で、緑の箱の中を砲弾が通過できるよう「通過できる」を on にしています。

2 番目のプログラムは、砲弾が緑の箱に接触したら風の力を加えます。「砲弾を継続的な力で押す」ため、緑の箱の中に砲弾がある限りは x 方向（横方向）に 100 の大きさの力で砲弾に力が加えられます。x 方向に 100 とは砲身から標的を見た時、左から右の方向に力がかけられます。



砲弾オブジェクトのプログラムでは緑の箱の中で加えられるこの力を打ち消す設定をしないと標的を命中させることはできません。



サンプルプログラムでは左図のように、x 方向に -100 の力を加えています。しかし、この設定では砲弾は標的の右に大きくそれてしまいます。

どの程度の大きさ（例 -100）をかけたら標的に命中させることができるのかは試行錯誤で見つけます。

砲弾に加える力の向きは大きさは「風」オブジェクトのプログラムの変更で自由に変えることができます。向かい風、追い風、下から吹き上げる風、上から押さえつける風などいろいろな風を吹かせてみましょう。それができたら異なる方向から風が吹いた時にどうすれば標的に命中させることができるかやっ

てみましょう。

発展課題



砲弾以外のものを風の中で動かしてどのように影響を受けるか試してみましょう。下図の例はオブジェクトメニューから「ならず者」を追加して、風の中を歩かせている様子です。他にもボールや車などいろいろなもので試してみましょう。

4. 砲弾を当てる 2

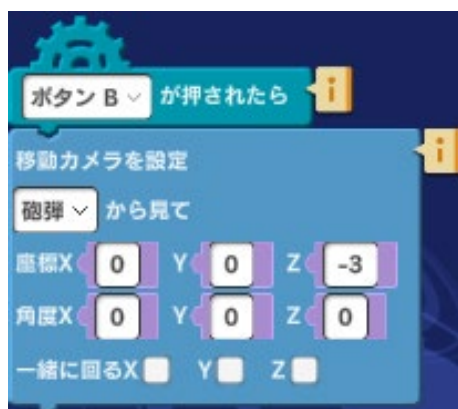


ラボ「4. 砲弾を当てる 2」ではカメラを砲弾のすぐ近くに置いて、砲弾が飛ぶのを追いかけます。まるで自分がドローンに乗って砲弾を追いかけて大砲から標的まで飛んでいくように感じられます。

Mind Render の VR 機能を使えばさらに迫力ある体験をすることができます。



このラボではカメラの設定がポイントとなります。カメラオブジェクトのプログラムを見てみましょう。



ボタン B が押されて砲弾が発射されると同時にカメラを砲弾に取り付けています。取り付け位置は z 方向に (-3)、砲弾の少し後ろにしています。



このラボのサンプルプログラムは VR で楽しむ時のために工夫がしてあります。砲弾オブジェクトのプログラムを見るとそれがわかります。



砲弾を撃つ命令ブロックはラボ「1. 砲弾を当てる 1」と同じですが、ボタン B が押された後で砲弾が発射されるまで 5 秒間待つようになっています。これは VR メガネを装着するなど VR の準備のための時間です。VR をしない場合にはこの命令を削除してもかまいません。



カメラの取り付け位置を工夫して迫力ある 3D 画像や VR 体験を楽しんでください。

発展課題



オブジェクトメニュー -> モデル -> プログラムの中にある「サブカメラ」と「サブモニター」を使うと遠くの様子を離れた場所で観察することができます。このしくみを使って、砲台の場所にサブモニターを置いて、ターゲットに命中する様子を見てみましょう。以下に参考となる情報を紹介します。



離れた場所の様子を見るにはサブカメラとサブモニターが必要となります。オブジェクト一覧から探して追加しましょう。「プログラム」カテゴリーにすると探しやすくなります。

サブカメラ1			
座標	-14.00	100.00	82.00
角度	90.00	0.00	0.00
大きさ	1.00	1.00	1.00
グループ			

サブカメラの位置は自由に設定できます。左図の例はターゲットとなる廃車の上で、真下を向くように角度が設定してあります。



また、「詳細」からオブジェクト詳細画面を開き、「物理」を OFF に設定しておきます。

サブモニター1			
座標	0.00	0.00	-39.00
角度	0.00	270.00	0.00
大きさ	3.00	3.00	3.00
グループ			

左図はサブモニターを砲台の横において、画面を横から見られるように角度を設定してあります。サブモニターも「物理」を OFF に設定しておきます。

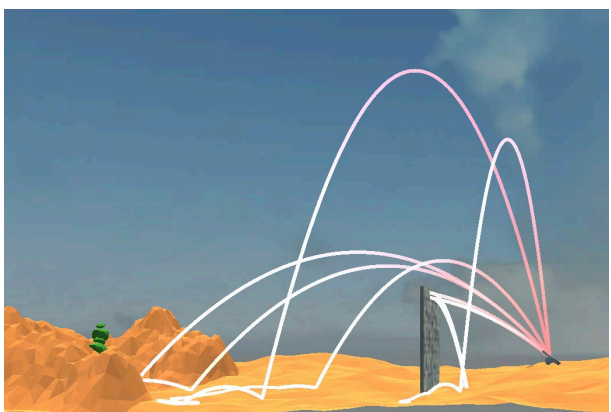


ラボのプログラムは、ボタン B が押されたら、カメラが砲弾を後ろから映すように切り替わるようになっています。そのままではサブモニターを見ることができなくなるので、ボタン B を押してもカメラが移動しないようにしておきましょう。

5. 高い壁

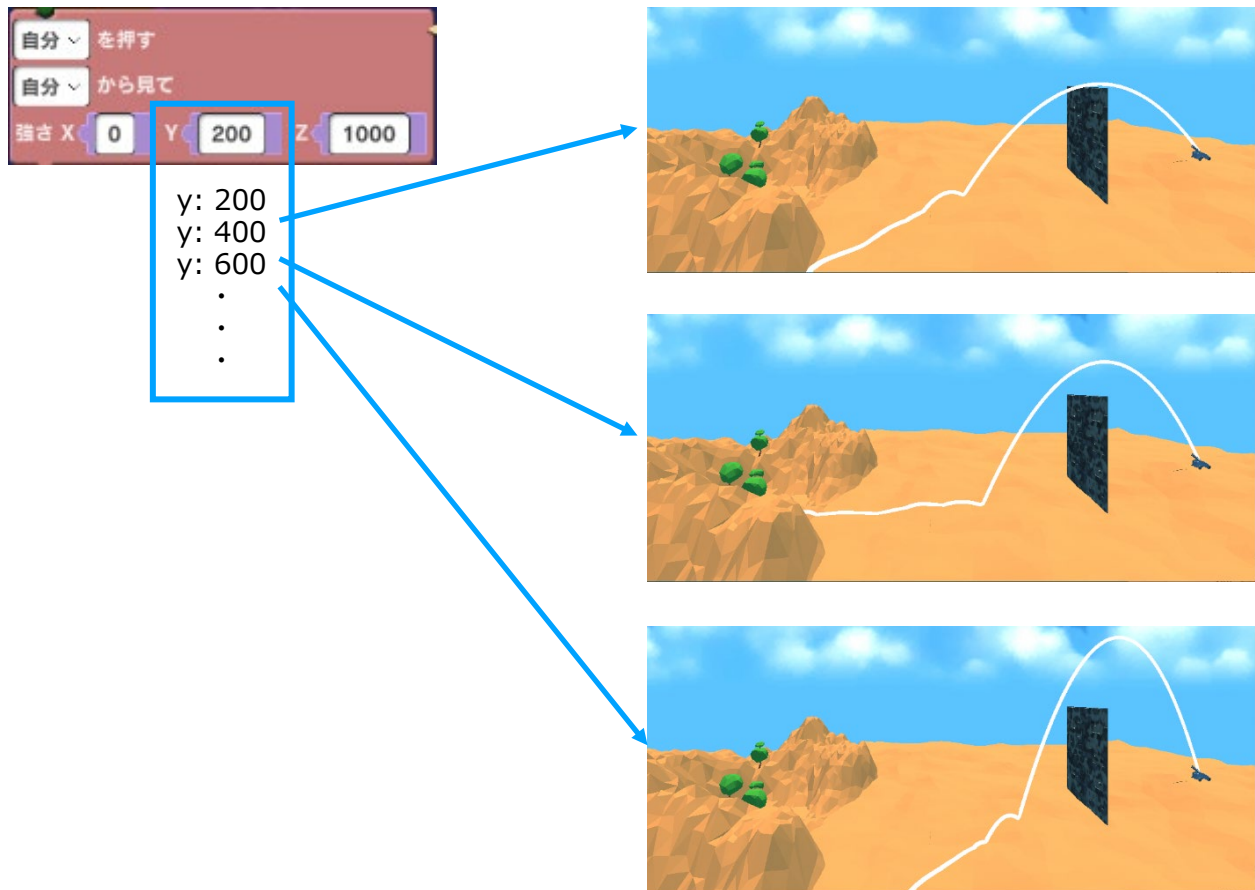


ラボ「高い壁」では砲台と標的の間に大きな壁が登場します。この壁を超えて標的を命中させます。壁を超えて標的に命中させるには砲弾の軌道をどうすればよいかわかりません。試してください。

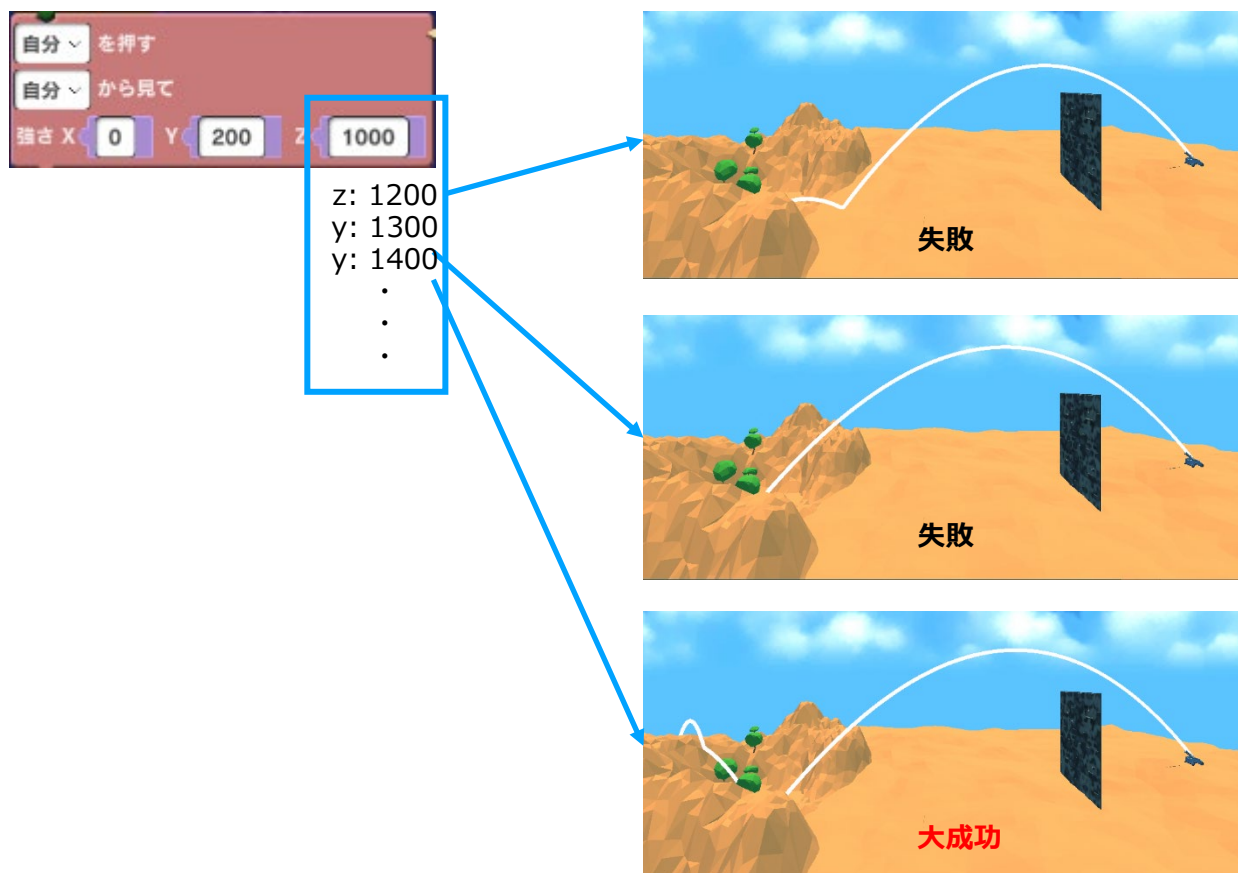


サンプルプログラムのままだと砲弾軌道の高さが足りないため壁に当たってしまいます。発射の際に瞬間に押す力で高さ方向の力をいろいろ変えて試してみましょう。

高さ方向に加える力を大きくしていくと砲弾を打ち上げる高さはどんどん高くなりますが、逆に到達距離は短くなってしまいます。高く打ち上げると同時に進行方向 (z 軸方向) に与える力も増やしていく必要があります。



高さ方向 (y) に加える力を大きくすると合わせて、進行方向 (z) に加える力も調整して標的の命中させるための設定を見つけてください。



発展課題



さらに面白くするためにどんな工夫ができるか考えてプログラムを加えてみましょう。参考例として以下に壁を前後に動かすプログラムを紹介します。

壁が砲台に近づいたり、遠ざかったりします。動く速度を変えるなどしてゲームの難しさを調整しましょう。

他にも壁が自動的に高くなったり、低くなったりするなどのアイデアを試してみましょう。

14. 外部とつなごう

「外部とつなごう」には、micro:bit やレゴ SPIKE など外部の機器と Mind Render を Bluetooth 接続していろいろな実験を行ったり、インターネットで対戦したりするためのサンプルプログラムが入っています。サンプルプログラムで遊ぶには、外部機器や事前準備が必要です。

micro:bit

micro:bit と Mind Render で双方向の通信を行い、ボウリングをしたり、2 つの micro:bit の距離を調べたりできます。

必要なもの | 事前準備

micro:bit2

HEX ファイル（Mind Render の Web サイト([こちら](#))からダウンロードし、micro:bit にインストールしてください。）

1.micro:bit で遊んでみよう



micro:bit の加速度センサーを使って、Mind Render 上でボウリングをすることができます。

2.micro:bit をつないでみよう



micro:bit の無線通信の信号強度を利用して 2 つの micro:bit の距離を調べ、Mind Render 上で文字（近い、遠い、・・・）、ゲージ、音で表現するプログラムが入っています。

注意： micro:bit2 が 2 つ必要です。



レゴ SPIKE



レゴ SPIKE のハンドルで Mind Render の車を動かすことができます。

2 つのラボのうち、一つは Mind Render のブロックで、もう一つは Python プログラムで SPIKE を制御します。

1.車を運転しよう (Block)

Mind Render のブロックで SPIKE を制御できます。



必要なもの | 事前準備

- ・レゴエデュケーション SPIKE プライムセット
 - ・ハンドルの組み立て
 - ・Python プログラム（Mind Render の Web サイトからダウンロードし、SPIKE ハブにインストールしてください。）
- * 詳細は、[こちら](#)をご参照ください。

2.車を運転しよう (Python)

Python プログラムで SPIKE を制御します。（Python の知識が必要です。）



必要なもの | 事前準備

- ・レゴエデュケーション SPIKE プライムセット
 - ・ハンドルの組み立て
 - ・Python プログラム（Mind Render の Web サイトからダウンロードし、micro:bit にインストールしてください。）
- * 詳細は、[こちら](#)をご参照ください。

3.SPIKE を操作しよう

SPIKE と接続して、いろいろな実験ができるラボを追加しました。センサー値を取得したり、SPIKE ハブの LED に文字や図形を表示したりするためのサンプルプログラムが入っています。



必要なもの | 事前準備

- ・レゴエデュケーション SPIKE プライムセット

M5StickC

M5StickC のマイクを通して音声でクイズに答える音声認識実験ができます。

出題されるクイズに、Yes か No で答えてください。



必要なもの | 事前準備

- ・ PC
- ・ M5StickC Plus
- ・ M5StickC 用プログラム（Mind Render の Web サイトからダウンロードし、M5StickC にインストールしてください。）

* 詳細は、[こちら](#)をご参照ください。

Raspberry Pi



Raspberry Pi と Mind Render を接続して車を運転したり、ロボットアームを動かすことができます。

1.車を運転しよう

Mind Render から PiCar-X を操作して走らせます。



必要なもの | 事前準備

- ・ Raspberry Pi Model B
- ・ Sun Founder 社 Raspberry Pi AI Car Kit PiCar-X
- ・ microSD カード
- ・ Python プログラム（Mind Render の Web サイトからダウンロードし、Raspberry Pi にインストールしてください。）

* 詳細は、[こちら](#)をご参照ください。

2.ロボットアームを動かそう

Mind Render からロボットアームを操作します。



必要なもの | 事前準備

- Raspberry Pi Model B
- Sun Founder 社 Raspberry Pi Robot Arm Kit
- microSD カード
- Python プログラム（Mind Render の Web サイトからダウンロードし、Raspberry Pi にインストールしてください。）

* 詳細は、[こちら](#)をご参照ください。

インターネット対戦



インターネット上の Mind Render ユーザーと対戦ゲームができます。

あらかじめ対戦相手(ロボット)が 1 人設定されているので、実際に相手がいないでもゲームを楽しむことができます。

1. ターンバトルをしよう



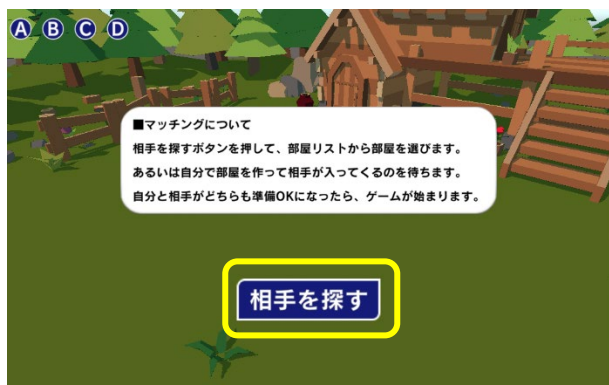
じゃんけんを応用したターン制バトルができます。3つの武器で対戦し、先に相手のHPを0にした方が勝ちです。

2. スピードバトルをしよう



スピードバトルができます。敵陣まで進むか、敵を全滅させた方が勝ちです。

対戦相手の探し方（マッチング）



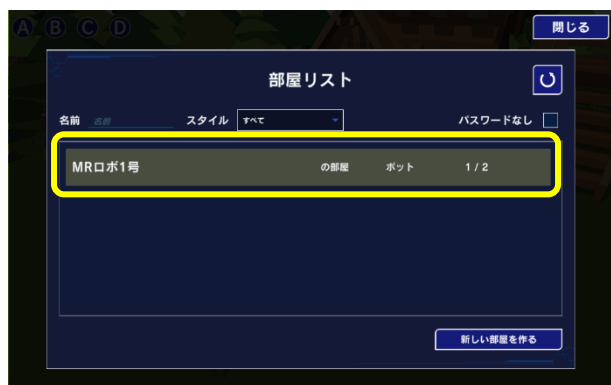
プログラムを実行し、「相手を探す」ボタンをクリックします。



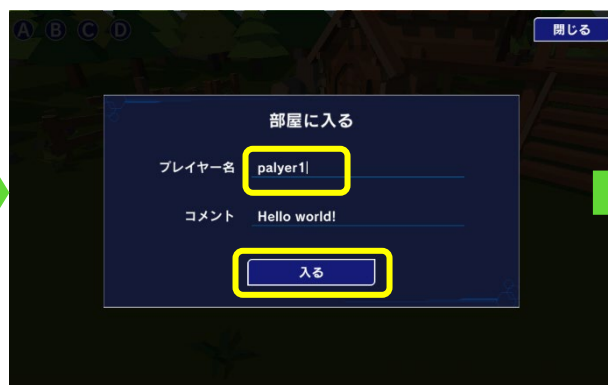
相手がいない場合は、「MR ロボ 1 号」(ロボット)を選択します。

相手がいる場合は、自分と相手が入る部屋を作ります。

MR ロボ 1 号と対戦する場合



「MR ロボ 1 号」(ロボット)を選択します。



自分のプレイヤー名をつけ、「入る」ボタンをクリックします。



「準備 OK」ボタンをクリックします。



自分も「準備 OK」に変わり、自動的にゲームがスタートします。

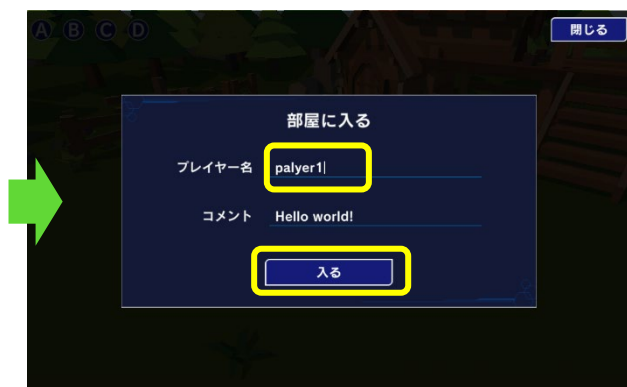
新しい部屋を作って他の人と対戦する場合



「新しい部屋を作る」を選択します。



「決定」ボタンを押します。



自分のプレイヤー名をつけ、「入る」ボタンをクリックします。



相手の入室を待ちます。双方が「準備 OK」ボタンをクリックして「準備 OK」になると、自動的にゲームがスタートします。

15. 遊び場 を使って自由に作品を作ろう

ラボメニューから「遊び場」を選ぶと、自由に自分の作品を作ることができます。遊び場は 1 から 9 までであるので最大で 9 個の作品を作ることができます。さらに多くの作品を作りたい場合はロッカーに保存しておくこともできます。ロッカーの使い方は巻末の「Mind Render 2 を便利に使うヒント」を参照してください。

16. さらに Mind Render を学ぶための方法

Mind Render の新しいリリース、最新の関連情報などは Mind Render オフィシャルウェブサイトなどをご参照ください。

Mind Render 公式ウェブサイト

<https://mindrender.jp/>

YouTube チャンネル

Mind Render の使い方について知りたい場合は、以下で関連動画を探してみてください。

オフィシャル YouTube チャンネル (@mindrender5672)

<https://www.youtube.com/@mindrender5672/videos>

Mind Render 情報チャンネル (@mindrender8955)

<https://www.youtube.com/@mindrender8955/videos>

こーじチャンネル (YouTube) (@ko-ji)

Mind Render を使った面白い実験動画があります。

<https://www.youtube.com/@ko-ji/videos>

X (Twitter) (@MindRender2)

バージョンアップやコンテストのお知らせ、その他プログラミング教育に関する情報をつぶやいています。

詳しいプログラミングの解説は下の書籍を参照してください。

「天才を育てるプログラミングドリルーMind Render で楽しく学ぶ」、白土良一他著、カットシステム刊 (ISBN-10 : 4877834362、ISBN-13 : 978-4877834364)

Amazon : https://www.amazon.co.jp/dp/4877834362/ref=cm_sw_em_r_mt_dp_-O0ZFbMB7TN0F

注意 : この書籍は Mind Render の初期バージョンをもとに書かれています。Mind Render 2 では画面や命令が変更されているため書籍の解説とは異なる場合があります。

17. Mind Render 2 を便利に使うヒント

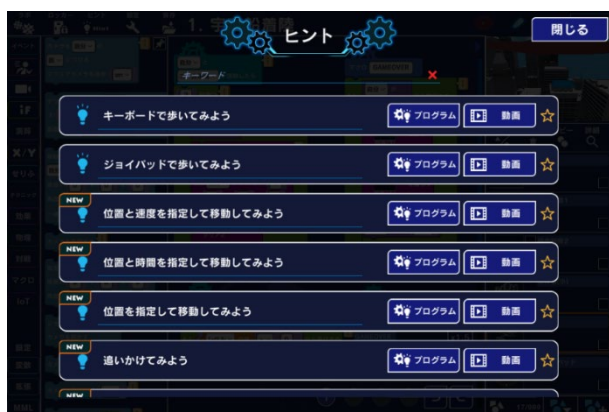
1) ヒント動画

こんなことをしたいけれど、どのようなプログラムを作ったらいいのだろう？という時、とても便利なのがサンプルプログラムとヒント動画です。ヒント動画は、プログラムの作り方や Mind Render の操作方法を動画で説明します。また、そのプログラムをダウンロードして、自分のプログラムで使うことができます。



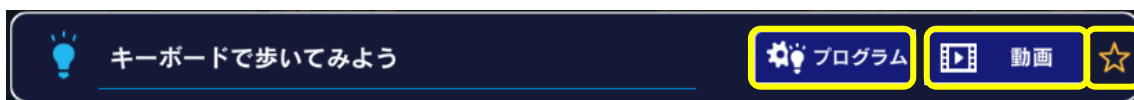
プログラム編集画面の左上に「ヒント」ボタンがあるので、これをクリックするとヒントの一覧が開きます。

す。



左図はヒントが開いたところです。スクロールすると多くのトピックが表示されますのでやってみたいことを探します。簡単なキーワード検索も可能です。

各トピックには下図のように2つのボタンと☆が表示されています。（プログラム無しで、ヒント動画のみのトピックもあります。）



「プログラム」ボタンをクリックすると、現在開いているラボで選択中のオブジェクトの中にサンプルプログラムが追加されます。



「動画」ボタンをクリックすると、ヒント動画で見ることができます。

「☆」を押すと、その項目が、お気に入りとして次回以降、項目の最初の方に表示されるようになります。よく使う項目は登録しておく探す手間が省けます。

トピック名は変更することができます。

2) プログラムの保存

ラボの保存

Mind Render のラボを変更したり、「遊び場」で新しく作成したら、プログラムを保存しましょう。



プログラムを保存するには、画面の左上にある「保存」ボタンをクリックします。これにより、ラボのプログラムを上書き保存することができます。



次回ラボを開くときには、「続きから」ボタンを押せば前回保存したところから再開することができます。

「最初から」ボタンを押すと、初期状態のラボが開きます。

注意： 変更内容はお使いのパソコンやタブレットに保存されます。別のパソコンやタブレットでも使いたい場合は、下記「ロッカー保存」をご利用ください。

ロッカーへの保存(クラウド保存)

ロッカーはクラウド上の保存スペースで、あなただけのプログラム保存場所です。ロッカーに保存したプログラムは、別のパソコンやタブレットからも呼び出すことができます。ロッカーを利用するには登録が必要です。まだロッカー登録をしていない場合は、後述を参照してください。



ロッカーにプログラムを保存したい場合は、画面左上の「ロッカー」をクリックします。



プログラムを初めて保存する場合は、タイトルをつけて「新規保存」をクリックします。

すでにロッカーに保存されているプログラムを保存したい場合は、同じタイトルで「上書き保存」するか、タイトルを変更して「新規保存」します。

注意： ロッカーに保存できるプログラムは最大 10 個です。



保存したプログラムを呼び出して楽しんだり、続きを作りたい時には「マイロッカー」を開きます。プログラムの右にある「読み込み」ボタンをクリックすれば手元のパソコンやタブレットにプログラムを読み込ませることができます。

不要になったプログラムは「削除」ボタンをクリックして削除します。

自動保存設定

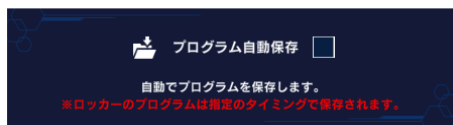
ラボ、ロッカーともに、自動保存の設定をすることができます。



プログラム編集画面の左上に「設定」ボタンがあるので、これをクリックすると設定画面が開きます。

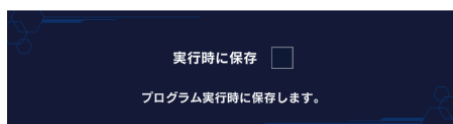


「ロッカー」タブを選択します。

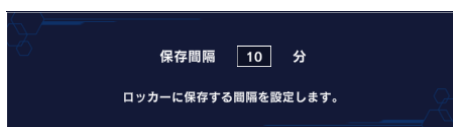


「プログラム自動保存」にチェックを入れます。

ラボのプログラムは、ラボを閉じた時にプログラムが保存されます。



「実行時に保存」にチェックを入れると、プログラム実行時にも保存されます。



ロッカーのプログラムの場合は、「保存間隔」で指定した時間ごとに保存します。

3) ロッカー登録



ロッカー登録をするには、画面左上のメニューボタンの真ん中のボタンをクリックします。

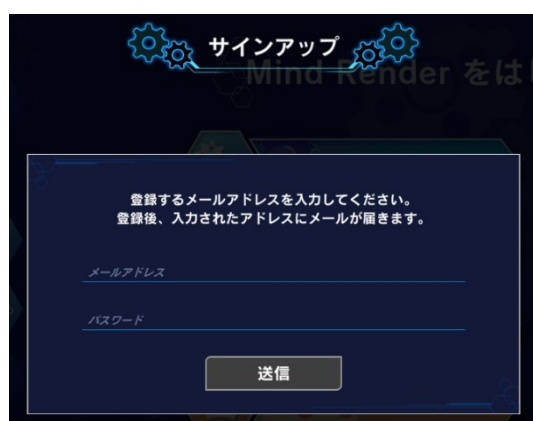


もしくは、起動後最初に開く画面で「ロッカー」を選択します。

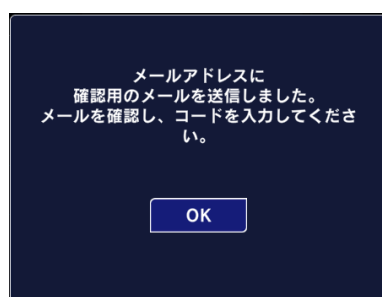
ロッカー画面が表示されます。初めて使う時には「新規登録」ボタンをクリックします。



「新規登録」ボタンを押して、登録に進みます。



メールアドレスを入力し、パスワードを設定して「送信」ボタンをクリックします。



確認メッセージが表示されるので、「OK」ボタンをクリックします。



登録したメールアドレス宛に確認用のメールが届きます。



コード入力画面にメールに記載されたコードを入力します。

「送信」ボタンをクリックしたら登録が完了です。

パスワードは忘れないようにしておきましょう。

登録したメールアドレスとパスワードを入力して、「ログイン」ボタンをクリックします。

4) プログラムの共有

共有キーの設定

マイロッカーに保存したプログラムは、共有キーを使って他の人と共有することができます。

「共有キー」はプログラムを共有したい人に教えて、その人があなたのプログラムを見たり、使ったりすることができるようにするためのものです。



ロッカー内の共有したいプログラムの「共有」欄に任意の文字を入力し、Enter キーを押します。

かな・カナ・漢字・英数字が使用できます。

複数のプログラムをまとめて共有したいときは、同じ共有キーを使用できます。

共有キーでプログラムを開く



共有キーを教えてもらったら、「共有ロッカー」タブ「共有キー」欄に入力します。プログラムのタイトルが表示されるので、プログラムを読み込むか、自分のロッカーに保存します。

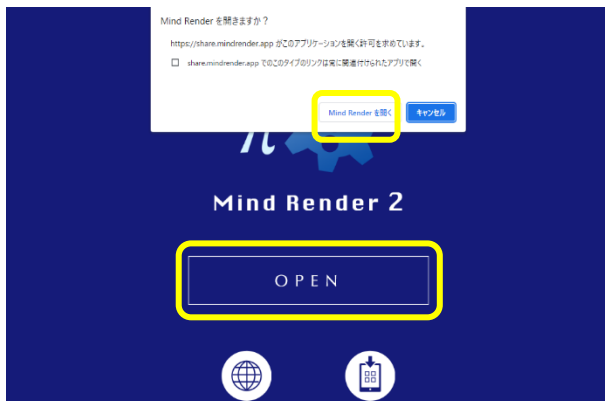
共有リンクを使ったプログラムの共有

共有キーを設定後、以下のボタンをクリックすると、プログラムの共有リンクが表示されます。



このリンクでプログラムを共有することもできます。

リンクをコピーして、メールやメッセージ等に貼り付けて相手に送ります。



受け取った人がリンクをクリックするか、ブラウザのアドレスバーにコピーして貼り付けると、左図のような画面が表示されます。「OPEN」ボタンを押すと、「Mind Render を開きますか？」という確認画面が開くので、「Mind Render を開く」ボタンをクリックします。

注意：事前に Mind Render がインストールされている必要があります。

5) プログラムの公開

作ったプログラムを Mind Render の他のユーザー全員に公開することができます。

プログラムを公開すると、他の人からコメントや「いいね」をもらうことができます。

共有と異なり、プログラムの中身は公開されず、プログラム実行のみ可能です。



「公開」するには、チェックボックスにチェックを入れます。

非公開に戻したい場合は、チェックを外します。

公開するときには、どんなプログラムなのか、遊び方やルール、操作方法などを説明欄に記載しておくことが親切です。この内容も一緒に公開されます。



公開されたプログラムは、「検索」タブで見ることができます。

クリックすると、プログラムが実行されます。



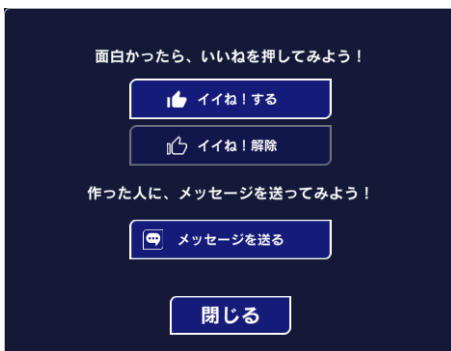
ゲーム内でスコアをカウントするようにプログラムされている場合は、この作品のスコアランキングを見ることができます。



作者に送られたメッセージを見ることができます。

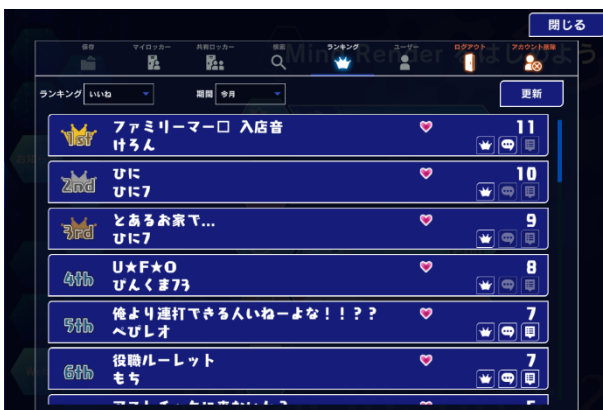


作者からのメッセージ、遊び方やルール、操作方法などがある場合は見ることができます。



作品をプレイして終了すると、左図のような画面が開きます。面白かったらコメントや「いいね」やメッセージを送りましょう。

6) ランキング



公開したプログラムにつけられた「いいね」の数によるランキングを見ることができます。

7) プログラムのコピー



作ったプログラムは簡単にコピーできます。

コピーしたい部分の一番上のブロックをマウスの左ボタンを長めに押します（または長めにタップします）。

ブロックの左肩に「Copy」という文字が現われて消えると、コピー完了です。

一番上のブロックを長押しした場合は、それ以降のブロック全部が、途中のブロックを長押しした場合は、そのブロックから下がコピーされます。

プログラミングエリアの空いている場所をマウスの左ボタンを長めに押すと（または長めにタップすると）、ペーストされます。

異なるラボ間でもコピーできます。

8) オブジェクトの追加

オブジェクトは、オブジェクトリストの右下にある「オブジェクト追加」ボタンから追加できます。

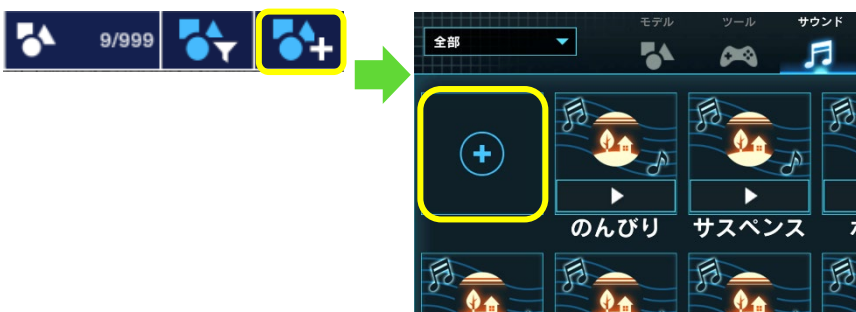


エフェクトやサウンドは、「オブジェクト追加」ボタンで追加しなくても、「効果」カテゴリーの命令から直接選択することもできます。

注意：エフェクトをキャラクターなどと同じように移動させたり、分身を作ったりしたい場合は、「オブジェクト追加」ボタンで追加してください。

9) サウンドの追加

パソコンやタブレットのマイクを使って声や音を録音してプログラムで使うことができます。



オブジェクト一覧で「サウンド」を選ぶと左上に「+」サインのボタンが見えます。このボタンをクリックします。



左図のような録音画面が開きます。

「名前」欄に後で呼び出す時に使う名前を入力します。

録音の用意ができたなら、赤丸のボタンを押して録音を開始します。

うまくいかなかったら何度でもやり直すことができます。

録音したサウンドを他の人と共有したい場合は、「共有する」にチェックを入れて、共有コード(任意の文字列)を入力します。

録音が完了したら「保存」ボタンをクリックして保存します。



保存したサウンドは左図のように「カスタム」の中に見つけることができます。



共有した音は左図のように「共有」タブで、共有キーを入力して呼び出すことができます。

Mind Render 2 簡単使い方マニュアル

2023 年 8 月 22 日 ver1.8 発行

著者 株式会社モバイルインターネットテクノロジー

発行 株式会社モバイルインターネットテクノロジー

〒102-0094 東京都千代田区紀尾井町 3-6 紀尾井町パークビル 1F

本書の内容の一部あるいは全部を無断で複写複製することは、法律で定められた場合を除き禁じられています。

本文中に記載のある社名/製品名は各社の登録商標または商標です。