

3次元表示を実現させる Scratch プログラミング
－ 3次元迷路と準天頂衛星を題材にして－

山 守 一 徳

キャリア研究センター紀要・年報 第8号 抜刷
高田短期大学
令和4年3月

3次元表示を実現させる Scratch プログラミング

— 3次元迷路と準天頂衛星を題材にして —

山守 一徳

三重大学 教育学部

Scratch は、X-Y 平面上で動かして表示させるのが基本機能である。そのため、3次元での動きを見せたい場合には、2次元平面上へ投影させるようなプログラムを作り込みする必要がある。一方、学校現場では、レゴロボットのような移動式ロボットを動かすプログラミング実践を行っている場面が見受けられるようになってきており、そのプログラミングも Scratch を用いている。その移動式ロボットの動きも2次元であり、学校現場でさらに興味を持たせるには、3次元の動きのものを取り入れたい。3次元の動きを伴う課題を与えて、プログラミングをさせたいものである。そこで、2次元平面での表示を基本とする Scratch において、3次元の動きを見せる課題に取り組みせる方法を示す。

ここで示す方法の一つは、3次元迷路を入口から出口まで通り抜けさせる課題であり、もう一つは、準天頂衛星の動きを理解させる課題である。どちらも3次元での動きを必要とするものであり、それを Scratch の2次元平面上で表示させながら、プログラミングをさせることになる。迷路を抜けさせる課題は2次元迷路ならば、動きが上下左右の4方向なので、簡単な課題となるが、3次元迷路にすると、移動した経路に沿って現状位置を把握しないといけない点と移動する方向が立体的になる点から、課題の難しさが大きく上がってくるものである。準天頂衛星の動きは、地球の丸い球体を表示させる方法から工夫する必要があり、Scratch のブロックを使ったプログラミングの計算精度にも限度があり、できるだけ現実の動きに近づけようと工夫を行っている。

1. はじめに

ブロックを並べることで動作するツールである Scratch は、2019年1月2日に Scratch2.0 から Scratch3.0 へバージョンアップされ、Internet Explorer で動かなくなった。Scratch2.0 では FlashPlayer を利用していたため、FlashPlayer に対応していない iPad のタブレット機やスマホのブラウザで動かないということが起きていたが、Scratch3.0 では利用できるブラウザを広げるために FlashPlayer を使わず、HTML5 と JavaScript で開発された。一方、Internet Explorer は、Internet Explorer しか使えないシステムもあるほどの、少し古い独自仕様のブラウザであったため、Scratch3.0 の開発に使われたプログラムを翻訳できないことが起きた。その結果、Scratch3.0 では、Internet Explorer で動かなくなった。

Scratch2.0 から Scratch3.0 に変わり、移動式ロボットを Scratch で動かすプログラムも、Scratch2.0 に対応していたものから、Scratch3.0 対応に変える必要あり、レゴロボットは、すぐに Scratch3.0 対応の

ものがリリースされた。しかし、近年流行してきているドローンを動かすプログラムは、Scratch3.0 対応が遅れている。ドローンは、3次元の動きをさせることができ、これまでの移動式ロボットの前進、後進、右回転、左回転という2次元の動きだけで飽きている点を改善させることができる。

現状では、Scratch でドローンを制御するには、ライブラリを読み込んだ後、Scratch2.0 オフラインエディターを利用して、実機のドローンを動かす。それには、ライブラリのインストールやScratch2.0 オフラインエディターのインストールの手間がかかるのが難点である。また、Scratch3.0 を利用して、実機のドローンを動かすには、Scratch3.0 拡張機能の作り込みをする必要がありかなりの技術と労力を必要とする。そこで、実機のドローンを使わず、ドローンシミュレーターを使ってプログラミング演習を行っているのが実情である。

そこで、3次元の動きを見せるもので、学校現場で容易にプログラミング教育として使えるものを提供しようと考えた。Scratch3.0 の2次元画面上で、3次元表示させることを行う。

2. 3次元迷路

入口から出口まで通り抜けさせる課題で2次元の場合は、上下左右の動きだけである。迷路の表示も壁の有無を決めるアルゴリズムが難しいだけで、表示に関しては、難しい点はない。図1に2次元迷路の表示例を示す。児童は、左上から右下の出口まで進ませるのに、「前に進む」「右向け右」「左向け左」のブロックを並べていくだけである。ブロックの定義は、壁の黒色に接触したら最初の位置に戻る

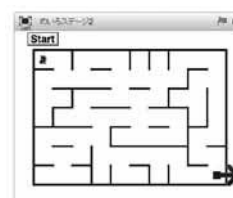
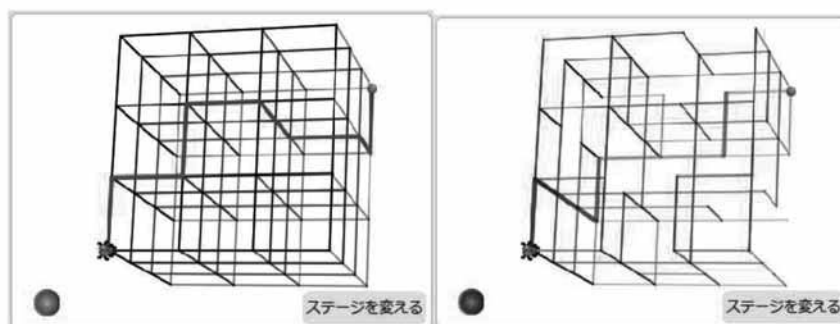


図1 2次元迷路

のと、ゴールの青色に接触したら成功の文字を出させるようにしている。迷路の作成は、棒倒し法と呼ばれるアルゴリズムで作成し、正解の経路が1つだけとは限らない、単純なアルゴリズムである。



(a) ステージ1

(b) ステージ2

図2 3次元迷路

3次元迷路の場合は、立体的に表示させる必要があり、表示のさせ方に工夫がいる。まず、2次元迷路の壁を表示させる代わりに、通過可能な通路を表示させ、通路上を移動させる。通路から外れたら最初の位置へ戻ることとする。そして、通路を示す線が重ならないように透視投影法を使って立体的に表示させる。図2に3次元迷路の表示例を示す。左下手前がスタート地点で、右上奥がゴール地点である。ステージ1は、通過できる通路に制約は無く、赤い線は通過目標として表示させているだけで、赤い線から外れた通路を通過しても失敗には成らない。ステージ2では、表示されていない辺は、通過できないことを示

し、この表示
されていない辺
を通過しようと
すると失敗
として先頭に
戻るようになって
いる。ス
テージ2で太
い線に表示さ
れている辺
は、通過目標
として表示さ
せている。

2. 1 実現 方法

図3が透視
投影による座

標変換を行うプログラムである。 x,y,z の3次元座標値が、
投影面 x ,投影面 y の2次元座標へ変換される。右方向が x
軸、上方向が y 軸、奥行き方向が z 軸の座標軸を持ち、3次
元座標の原点が左下手前に位置する。迷路の立方体は、軸
回転させる前の初期位置において0から600までの座標
値に位置し、1つの辺の長さは200である。カメラの位置
が、 $(0,0,-2000)$ の座標値にあり、投影面が、 $(0,0,-300)$ を通
過する xy 平面に位置する。迷路の立方体を x 軸回りに10
度回転し、 y 軸回りに19度回転させることで、立方体の各
辺が重ならない位置に表示できている。図2の図で、立方
体の底の面が下から見えた状態になっているのは、 x 軸回
りに10度回転している効果であり、立方体の右側の面が
見えた状態になっているのは、 y 軸回りに19度回転して
いる効果である。

図4にScratchの緑色の旗を押して開始させる時のプロ
グラムを示す。「経路作成」のユーザ定義ブロックでは、ス
テップ2の時に使う立方体の各辺を表示させるのか表示
させないのかを決める各辺の重みを設定する。「 x 方向」の

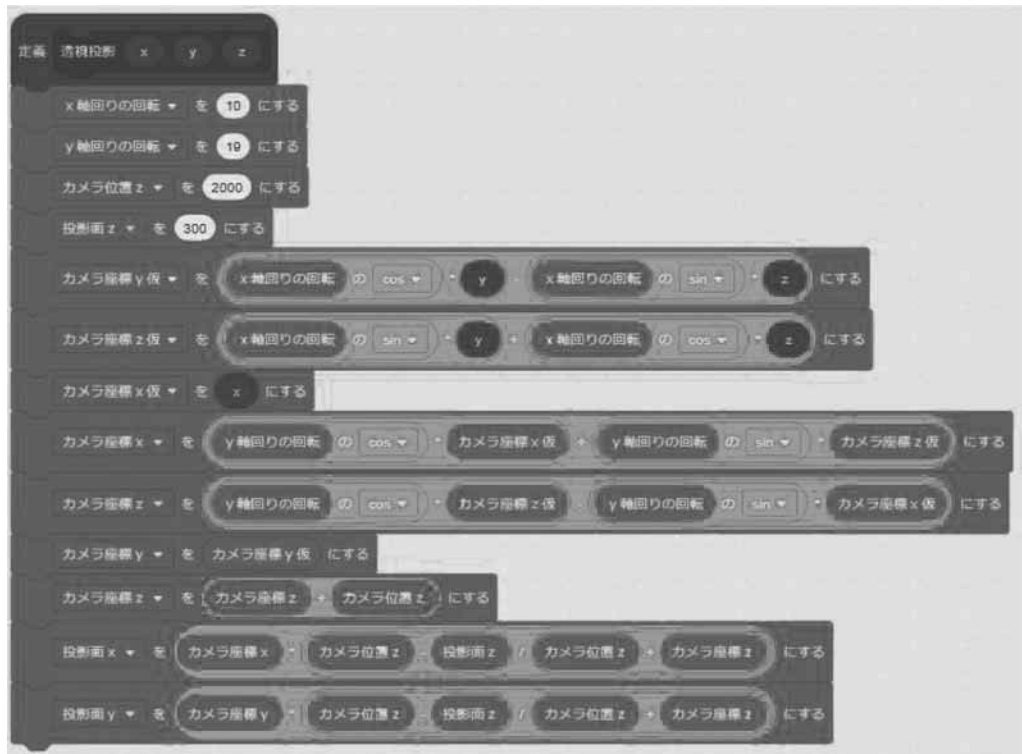


図3 透視投影の座標変換プログラム



図4 開始プログラム

ユーザ定義ブロックを図5に示す。
 これは、x方向に向かう辺の描画を行う。「正解経路選択」のユーザ定義ブロックは、図2の(a)では赤色の辺、図2の(b)では太い辺をどこにするかを定める。「正解経路」のユーザ定義ブロックで、その描画を行っている。「透視投影」のユーザ定義ブロックを0,0,0で呼び出しているのは、スタート地点の投影面x、投影面yを求めるためであり、600,600,600で呼び出しているのは、ゴール地点の投影面x、投影面yを求めるためである。原点移動x、原点移動yの値で、描画位置を平行移動させるのは、図2の「ステージを変える」ボタンや左下の緑色や青色の丸ボタンと重ならないようにするためである。図4の最後にスタート&ゴール配置のメッセージを送ると、スタート位置とゴール位置の描画が行われ、経路移動開始の準備が完成する。

図5にx方向に向かう辺の描画プログラムを示す。y方向、z方向もこれに似たプログラムになっている。中ほどに、x方向リストのn番目の値が8未満ならペンを下ろし描画させるようになっているが、x方向リストの値は、0から10までの乱数の値が、「経路作成」のユーザ定義ブロックの中で設定されており、8以上はその辺を表示しない動きになっている。8を調整すると、表示する辺の量が変動するが、スタート地点からゴール地点まで辿り着ける経路が存在しないことが起きるため、「正解経路」のユーザ定義ブロックで上書きすることにより、最低でも一つの正解経路が存在するようにしている。



図5 x方向の辺の描画プログラム

「正解経路」のユーザ定義ブロックで上書きすることにより、最低でも一つの正解経路が存在するようにしている。

図6に「右に行く」のユーザ定義ブロックを示す。この「右に行く」以外に、「左に行く」、「上に行く」、「下に行く」、「奥に行く」、「手前に来る」のユーザ定義ブロックが用意されており、このブロックを並べることで、スタート地点からゴール地点まで虫の絵のスプライトを移動させる。正解経路は、x方向に行くのが3回、y方向に行くのが3回、z方向に行くのが3回で、辿り着けるため、「左に行く」、「手前に来る」と「下に行く」のユーザ定義ブロックは使う必要はない。「正解経路選択」のユーザ定義ブロックでは、x方向に行くのが3回、y方向に行くのが3回、z方向に行くのが3回をランダムにシャッフルさせるだけである。x方向に行くのが



図6 右に行くプログラム

「右に行く」のユーザ定義ブロック、y方向に行くのが「上に行く」のユーザ定義ブロック、z方向に行くのが「奥に行く」のユーザ定義ブロックを使うことで実現できる。図6のプログラムで100ずつ2回に分けて移動させているのは、表示されていない辺の位置へ移動した時に、通路から外れたことを検出させるためである。「移動」のユーザ定義ブロックを図7に示す。ゴール地点に到着したかを調べるメッセージ送信をした後で、赤色が白色に触れたかで判定し



図7 移動のプログラム

ているのは、スプライトの虫の絵の中の中央表示されていない白い背景の白色部分に触れたかという判定の仕方を利用している。

図2の左下の緑色や青色の丸ボタンは、経路移動を開始させるためのボタンである。図8にそのボタンを押した時に動き出す移動開始のプログラムを示す。「この下にゴールまでの動きを並べよう」のユーザ定義ブロックが呼び出されるようになっており、そのユーザ定義ブロックに図6の「右に行く」ブロック等を並べていくことになる。ブロックを並べては、緑色や青色の丸ボタンを押してゴール地点ま

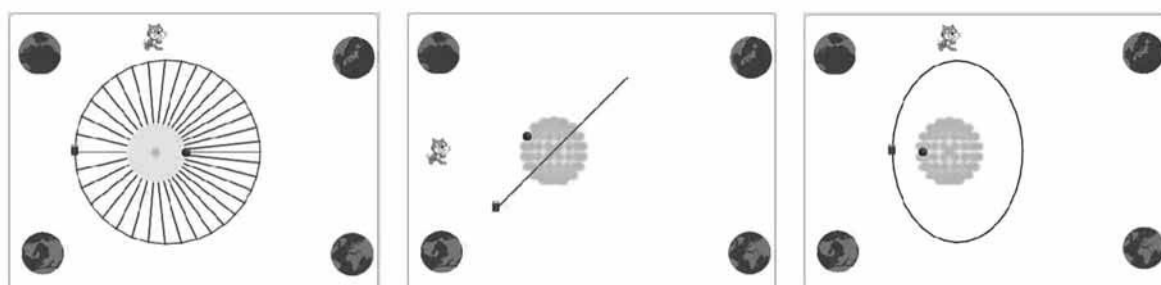


図8 移動開始のプログラム

付近の赤色が、辺の表



図9 順天頂衛星の軌道 (Wikipedia より)

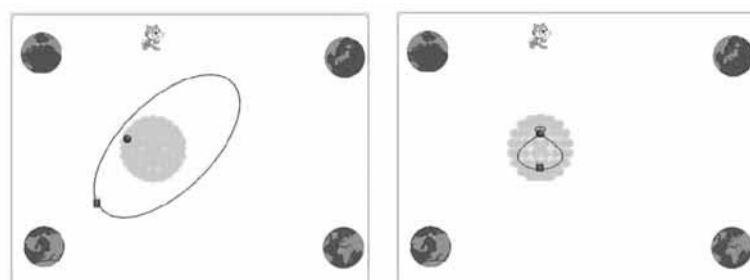


(a)開始ボタンを押した時

(b)右下ボタンを押した時

(c)左下ボタンを押した時

で辿り着けるかを試行錯誤することができる。正解経路を変更したい場合は、Scratch の緑色の旗をクリックすれば良い。



(d)左上ボタンを押した時

(e)右上ボタンを押した時

図 10 準天頂衛星の動き画面

3. 準天頂衛星の動き表示

人工衛星の動きは、模型で示しながら見せるのがわかりやすいが、3次元空間を使う模型も用意するの

が簡単ではない。そのため、シミュレーション映像を作って見せることが考えられる。静止衛星が自転と同じ速度で赤道上で周回させれば良いため、作成は簡単であるが、準天頂衛星となると、動きが複雑となり、シミュレーションも困難となる。それを Scratch 上で再現させることを目指した。準天頂衛星は、日本の上空に長く留まるようにさせるため、楕円軌道で周回している。準天頂衛星から地球上の真下に降ろした位置を追い掛けると、8の字の軌道(図9参照)を示すものである。何故8の字の軌道になるのか、シミュレーションで示すことは衛星の動き理解の助けとなる。基本的な動きとしては、日本の真上の位置からスタートし、地球の自転に合わせて円軌道を描かせると、地球が半回転した時に、衛星の位置は、オーストラリア上空に位置し、再び半回転すると、日本の真上の位置へ戻ってくる動きとなる。日本とオーストラリアの上空を直線的に行ったり来たりすると思われるが、そうではなく、赤道上に交点を持つきれいな8の字になる。軌道が傾いているため、赤道を横切る頃に、衛星の東西方向の速度は自転速度より遅くなる。逆に、日本を通過する頃には、緯度 35 度付近の地表自身の東西方向の速度は赤道付近より遅くなるが、衛星は赤道と並行の向きになるため、衛星の東西方向の速度は地表より速くなる。オーストラリアを通過する頃にも同様である。よって、日本とオーストラリアの上空では、東西方向の衛星の速度は速く動いて行き、赤道を横切る時は遅く動いて行き、8の字を描くことになる。この動きが基本であるが、日本の上空にいる時間を長くするために、衛星の円軌道を楕円軌道にしている。そのため、8の字の交点位置が日本寄りになった歪んだ8の字になっている。楕円軌道にすると、面積速度一定の法則により、焦点に近い時は、衛星の速度が速まり、焦点から遠い時は、速度が遅くなる。オーストラリアに近い時は、衛星の速度が速く、日本に近い時は、速度が遅くなる。赤道上と比べて緯度 35 度の地表の東西方向の速

度が小さくなるのだが、衛星との差が、オーストラリア付近では差が広まり、日本付近では差が縮まるそのため、8の字の横方向の大きさが、オーストラリア側では大きくなり、日本側では小さくなる。

3. 1 動作画面

以上の動きを Scratch で見せるために作成した画面が図 10 である。図 10(a)は、Scratch の緑の旗のアイコンをクリックした時に表示される画面であり、楕円軌道の面積速度一定の法則により、焦点に近い側が速く動くことを示している。赤い円筒形の絵が衛星を示し、紫の丸が、日本の位置を示している。図 10(b)は、赤道から見た時の動きを示し、衛星がオーストラリア上空にいる場面から、右上方向へ動き、日本の位置が右側に来た時には、衛星が右上に位置し、衛星が地球の裏側を取って左下に戻る動きをする。日本を示す紫の丸も地球の前面を通った後に、地球の背面に回った時には表示されず、最後に地球から顔を出した位置で止まる。青い線は、衛星の軌跡を示し、この図の場面は、衛星が一周回った後で停止した時の場面である。図 10(c)は、北極から見た時の動きを示し、衛星がオーストラリア上空にいる場面から、反時計回りへ動き、一周回る。日本を示す紫の丸も反時計方向へ動いて、一周回る。衛星の軌跡が縦方向の楕円に見えるのは、衛星が図 10(d)のように傾いている軌道を北極の上から見た場面だからである。図 10(d)は、斜めから見た時の動きを示している。地球の球体が立体的に見えるように黄緑の小丸も経度に沿って円状に描き、日本を示す紫の丸も、その円に沿って動いていく。衛星の動きも、左下から地球の手前を回り、右上に位置した後に、地球の裏側を回って降りていくように動く。図 10(e)は、日本を固定にした時

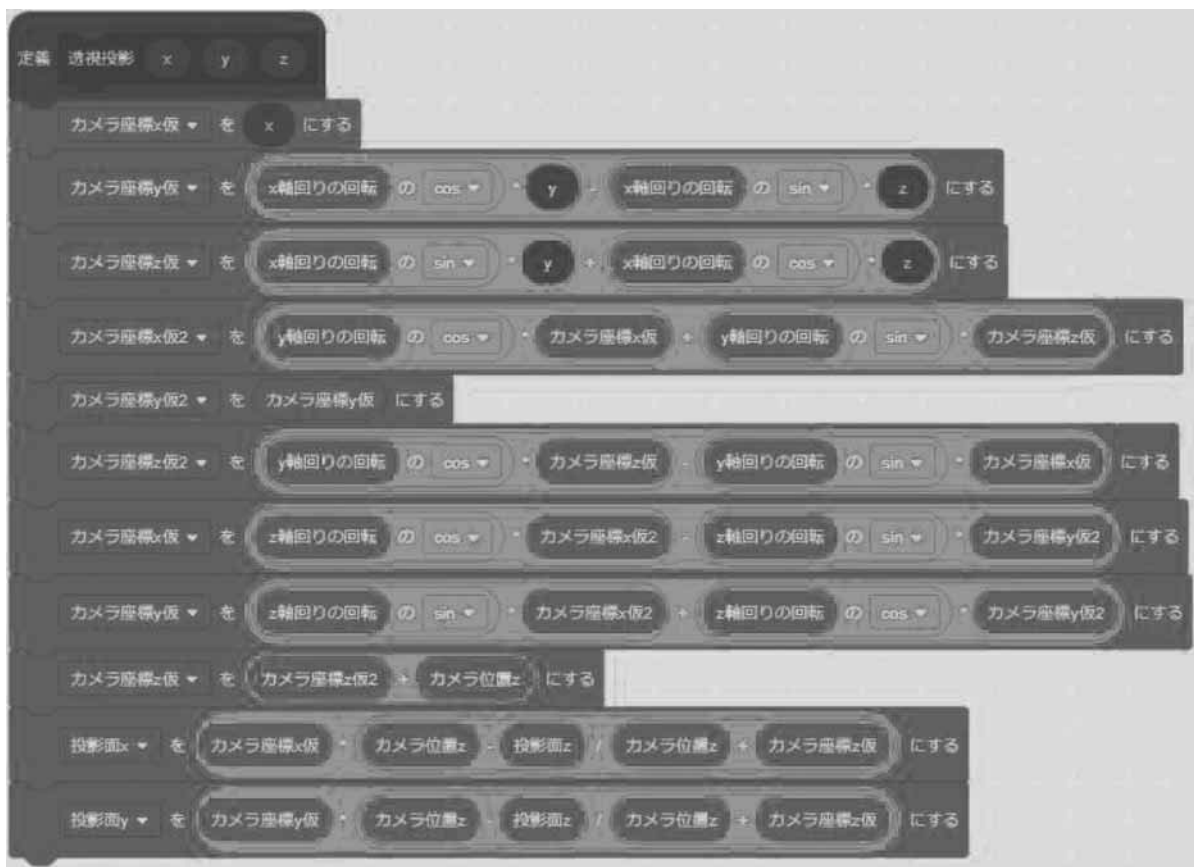
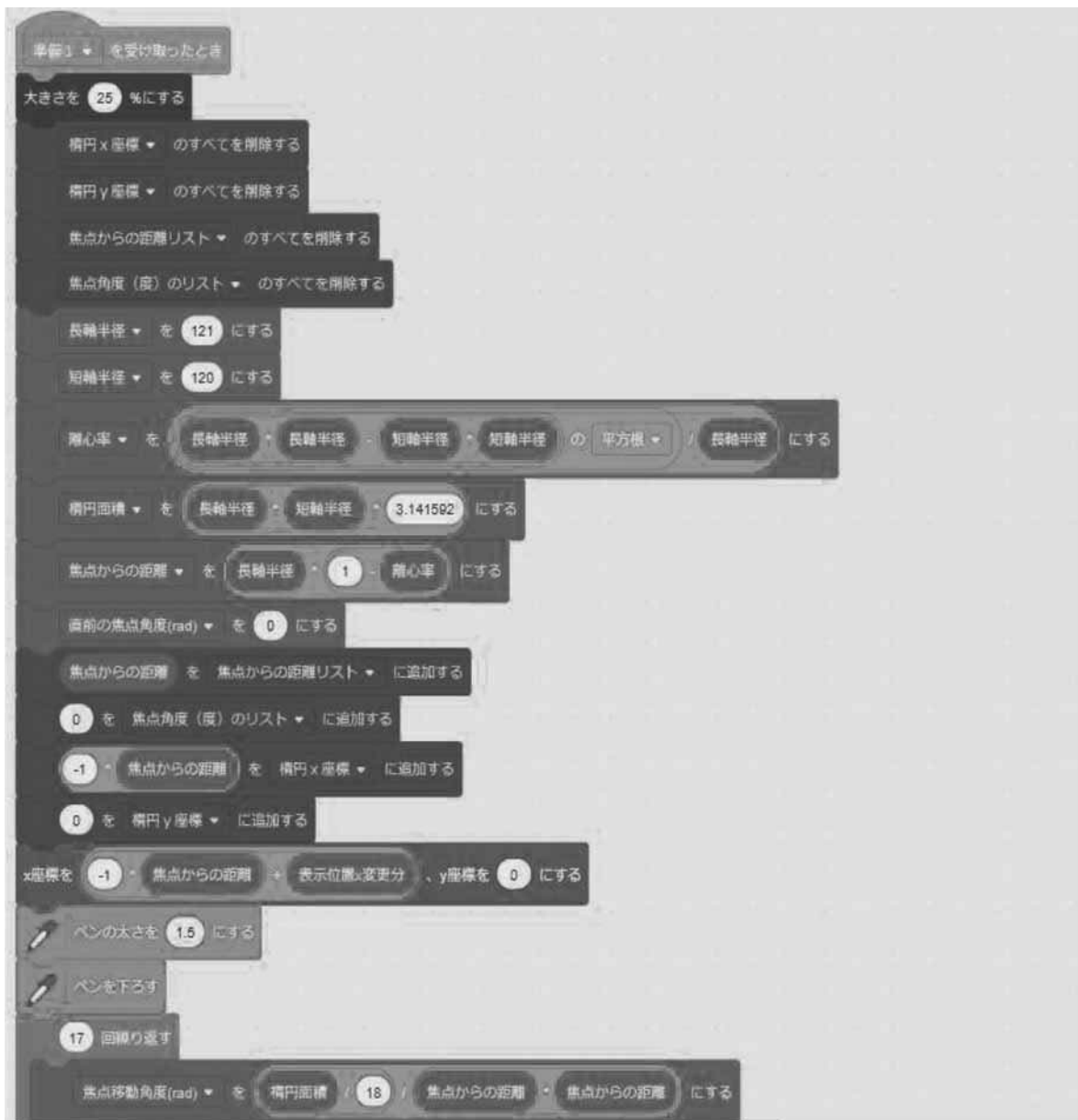


図 11 透視投影のプログラムその2

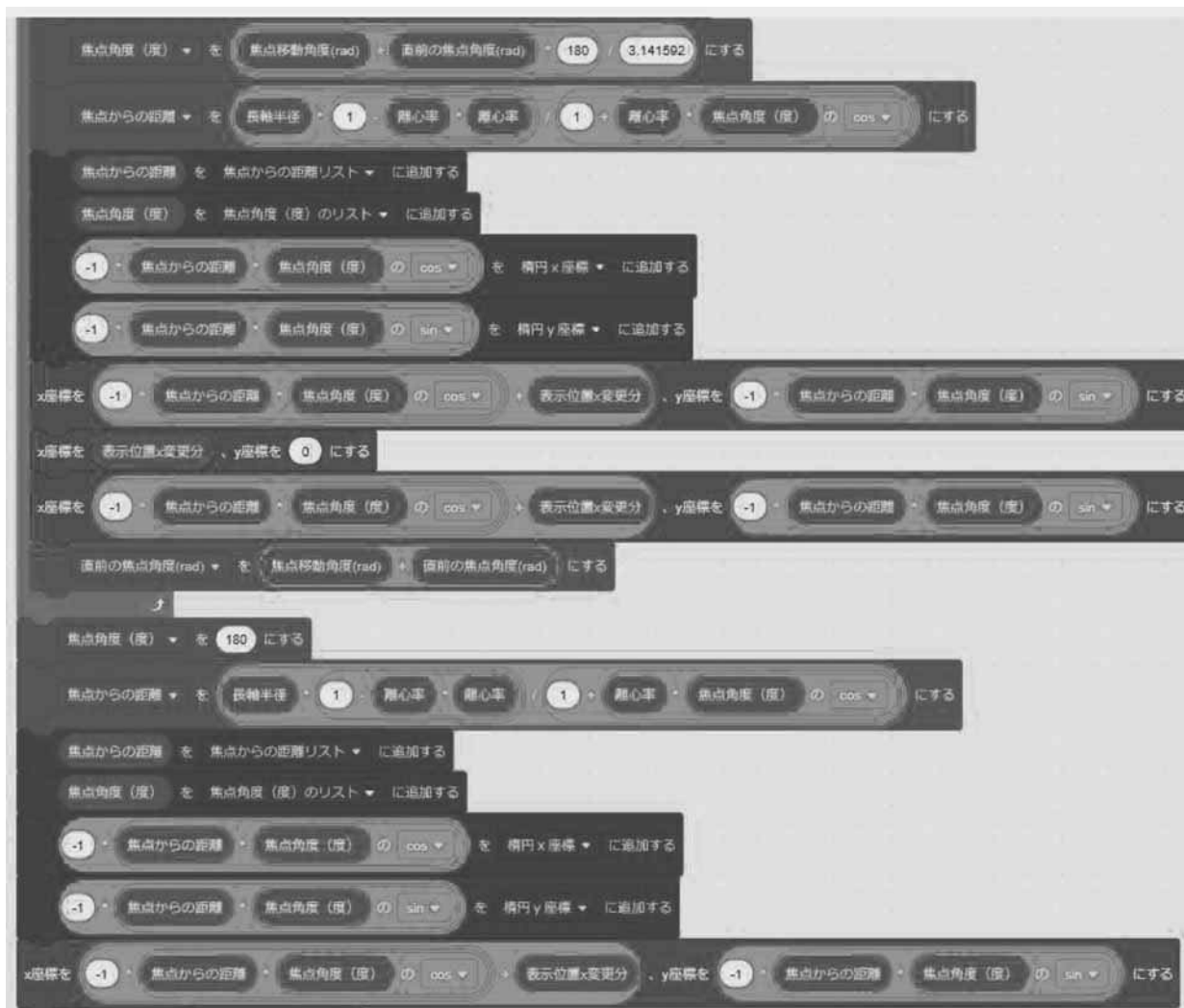
の、衛星の動きを示す。衛星は8の字を描くように回り、オーストラリア上空に居る時で最終的に止まった場面を示している。

3. 2 実現方法

図 11 が透視投影による座標変換を行うプログラムである。図 3 と比べて、z 軸回りの回転が入っているため、図 3 よりも式が長くなっている。図 10(b)の右側に向かうのが x 軸方向、北極の上に向かうのが y 軸方向、紙面奥に向かうのが z 軸方向、地球の半径を 40dot で表示、カメラの位置が、(0,0,-300)の座標値にあり、投影面が、(0,0,300)を通過する xy 平面に位置する。図 10(d)では、x 軸回りに 45 度回転させた場面での表示になっている。図 10(d)で地球表面に黄緑の小丸を表示させているが、この黄緑の小丸は、xz 平面上に 10 度刻みで 36 地点を円周上に位置計算しており、その円周は、北極から南極まで 9 層の高



(a) 前半部分



(b) 後半部分

図 12 楕円軌道の計算部分 (途中まで)

さから成っている。円周の y 座標の値は、 y 軸から 20 度刻みに開脚していく時の高さを用いており、赤道近くは差が大きく、極近くは差が小さい。表示に当たっては、黄緑の小丸の位置の 3 次元座標値を緑の旗アイコンを押された時に一度計算して配列に保存させ、地球の絵の選択ボタンを押した時に、透視投影変換した座標に変換している。日本を示す紫の丸の位置も、黄緑の小丸の位置の配列を利用している。地球の自転を示すために、紫の丸の位置は、 37 回移動して、最後には最初の位置に戻る動きをしている。衛星の動きは、楕円軌道の面積速度一定の法則に沿った位置を計算するのであるが、Scratch のプログラムで正確な計算までできないため、概算計算を行っている。図 12 にその計算の一部を示す。図 10(a)の衛星の位置の下半分を図 12 中の 17 回繰り返すのブロックで描いている。図 10(a)の青線の放射状の線と楕円との交点位置を求める時、下半分の焦点からの距離リスト、焦点角度 (度) のリストの値を用いて、上半分の交点位置を求めている。17 回繰り返すを 36 回繰り返すにして一周分全て求めてしまうと、累積誤差により上下対称にならないためである。図 12 で一周分の交点位置を配列に保存した後に、図 10(d)の斜めから見た位置に見えるように、図 11 の透視投影変換を使って、表示位置を計算している。図 10(e)の 8 の

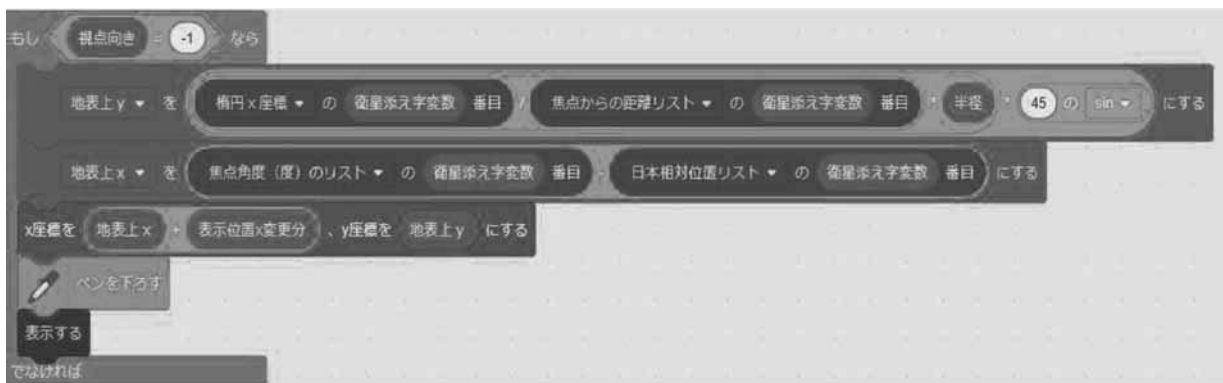


図 13 8の字を描く部分

字を描く部分のプログラムを図 13 に示す。8の字の縦方向の動きは、楕円の交点位置の座標から計算しているが、8の字の横方向の動き、すなわち、衛星と地球自転との速度の差については、正確に計算させることが難しく、日本相対位置リストの配列の中に、コサイン関数で生成した変動値を作成しておき、楕円運動との差から横方向の動きを再現させている。その変動値の作成のプログラムを図 14 に示す。10度刻みで36



図 14 自転速度との差の計算

回すところで、10度の値をコサイン関数で変動させている。この変動が、衛星軌道が赤道と並行でなく、傾いていることにより発生している東西方向の速度の差の部分である。

表示上のテクニックとして、衛星の軌跡を示す青い線が、地球表面を示す黄緑の小丸の上に表示させるには、黄緑の小丸は、スタンプを使って表示した後に、そのスプライトを隠した状態にしないと、その上に線を描くことができない。また、黄緑の小丸は、クローンを作成して、表示位置を変えて表示していくが、作成できるクローンの数には上限がある。上限数は300個と言われており、現状で18×9個表示している。黄緑の小丸の位置の計算は、36×9個求めているが、全て表示してしまうと上限を超えてしまうため、1個置きに表示を行っている。1周に36地点計算しているのは、日本を示す紫の丸が、その36個の計算地点を使って自転の動きを表示させているからである。

3. 3 授業実践

特殊な軌道を描いている「みちびき」という準天頂衛星について知ってもらおうと、津市立栗真小学校5年生3クラスを対象に、2021年3月11日2,3,4限目に各クラス45分間の授業を行い、さらに、附属中

学校1年生4クラスを対象に、2021年7月7日と14日2,3限目に各クラス45分間の授業を行った。準天頂衛星の動きをするプログラムは完成品を渡し、静止衛星の動きをするプログラムを追加してもらうことを行った。完成すると図15の赤い軌道の線が表示される。

授業後のアンケート結果によると、静止衛星がなぜ止まって見えるか理解できましたかという質問に対し、表1の結果となり、準天頂衛星がなぜ8の字に見えるか理解できましたかという質問に対し、表2の結果となった。その理解度の分布を帯グラフで示したものが図16と図17である。静止衛星がなぜ止まって見えるかの理解度は、中学1年生と小学5年生でそれほど差がないが、小学5年生では、全く理解できないレベルの児童がやはり多くいる。準天頂衛星がなぜ

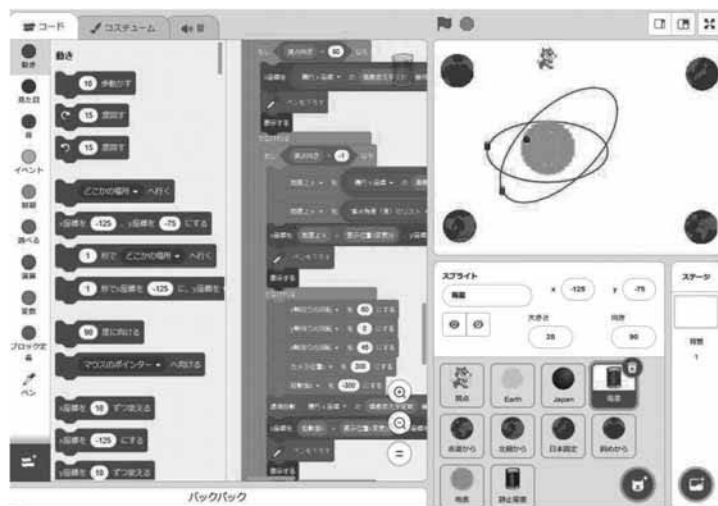


図15 Scratchの動作画面

表1 静止衛星がなぜ止まって見えるか理解できましたか

	理解できた	ちょっと理解できた	あまり理解できなかった	全く理解できなかった
中学1年生	42人(31.6%)	49人(36.8%)	25人(18.8%)	17人(12.8%)
小学5年生	30人(31.3%)	30人(31.3%)	19人(19.8%)	17人(17.7%)

8の字に見えるかの理解度については、静止衛星が理解できる側に回答した小学5年生はちょっと理解できるというレベルに落ちてきている。一方、理解できる側に回答した中学1年生はあまり理解できないというレベルに落ちてきている。また、小学5年生で、

表2 準天頂衛星がなぜ8の字に見えるか理解できましたか

	理解できた	ちょっと理解できた	あまり理解できなかった	全く理解できなかった
中学1年生	21人(16.0%)	47人(35.9%)	42人(32.1%)	21人(16.0%)
小学5年生	23人(24.0%)	42人(43.8%)	22人(22.9%)	9人(9.4%)

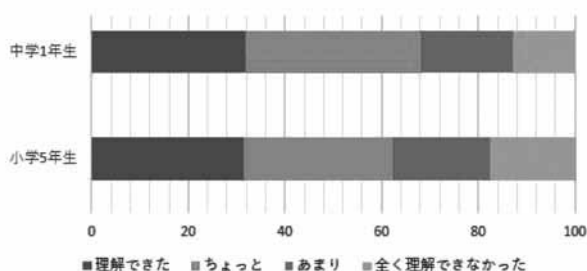


図16 静止衛星がなぜ止まって見えるかの分布

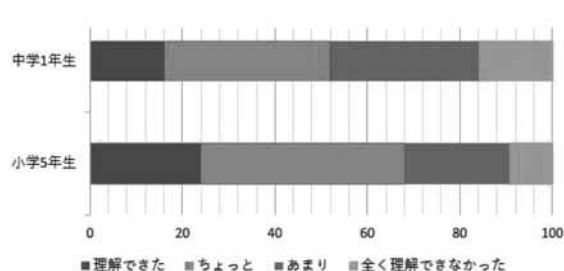


図17 準天頂衛星がなぜ8の字に見えるかの分布

全く理解できないレベルの児童が少なくなっている。静止衛星も準天頂衛星も知らない小学5年生は、静止衛星の動きよりも8の字の動きになる方の解説を授業で長く聞いたため、この評価になったと思われる。

4. まとめ

小学生や中学生にプログラミングをさせる題材としては、3次元表示させるものを最初から作らせるのは難しいが、3次元表示させたものを見せると、興味を惹くことができる。Scratch は、2次元の動きが基本であるが、3次元の動きのように見せることはできた。3次元迷路のプログラムは、授業実践をしていないが、児童に試したいところである。地図を見ながら移動する時に、迷子になってしまう子は、自分の居る位置と向きが、地図の中で読み取れないことが起きている。移動中の曲がった道で方角の把握がずれてしまうという要因があるが、辿ってきた道を覚えていないという要因もある。3次元迷路のプログラムはその訓練にもなるのではないかと思う。人工衛星の動きを示すプログラムは、三角関数が出て来るため、小学生・中学生では、難しすぎるという感があるが、人工衛星の動きを知ってもらうことはできる。小中学校の理科の授業で、準天頂衛星まで解説は出て来ないが、「はやぶさ2」の探査機の活躍があり、宇宙への関心は広まっている。「みちびき」の準天頂衛星も2017年10月10日に4号機の打ち上げが成功し、2018年11月1日からサービス提供しており、児童・生徒へ知らせることは重要であると思う。

今後もますます新しいものを提示していくので、教育に使っていただけたらと思う。

参考文献

- (1) 山守一徳：Scratch を用いた小学生向けプログラミング教育、高田短期大学キャリア研究センター紀要・年報第3号 pp.63-72 (2017年3月)
- (2) 山守一徳：Scratch を用いる学年別プログラミング教育、高田短期大学キャリア研究センター紀要・年報第6号 pp.40-51 (2020年3月)