

ロボット博士養成講座

ロボティクスプロフェッサーコース

不思議アイテムI-1③

第6回

シンセサイザー^{えん そう}演奏と人工生命

講師用

目 次

0. シンセサイザー演奏と人工生命

0.0. 「シンセサイザー演奏と人工生命」でやること

0.1. 必要なもの

1. シンセサイザーで遊ぼう

1.0. コントローラーの接続と設定

1.1. ボタンカスタマイズ1

1.2. ボタンカスタマイズ2

1.3. エフェクター

1.4. シンセサイザー演奏会

2. 人工生命をつくろう

2.0. ライフゲームとは

2.1. 代表的なパターンを試してみよう

3. まとめ

○ 授業開始にあたって

授業のはじめは、着席させ、大きな声であいさつしてから始めます。

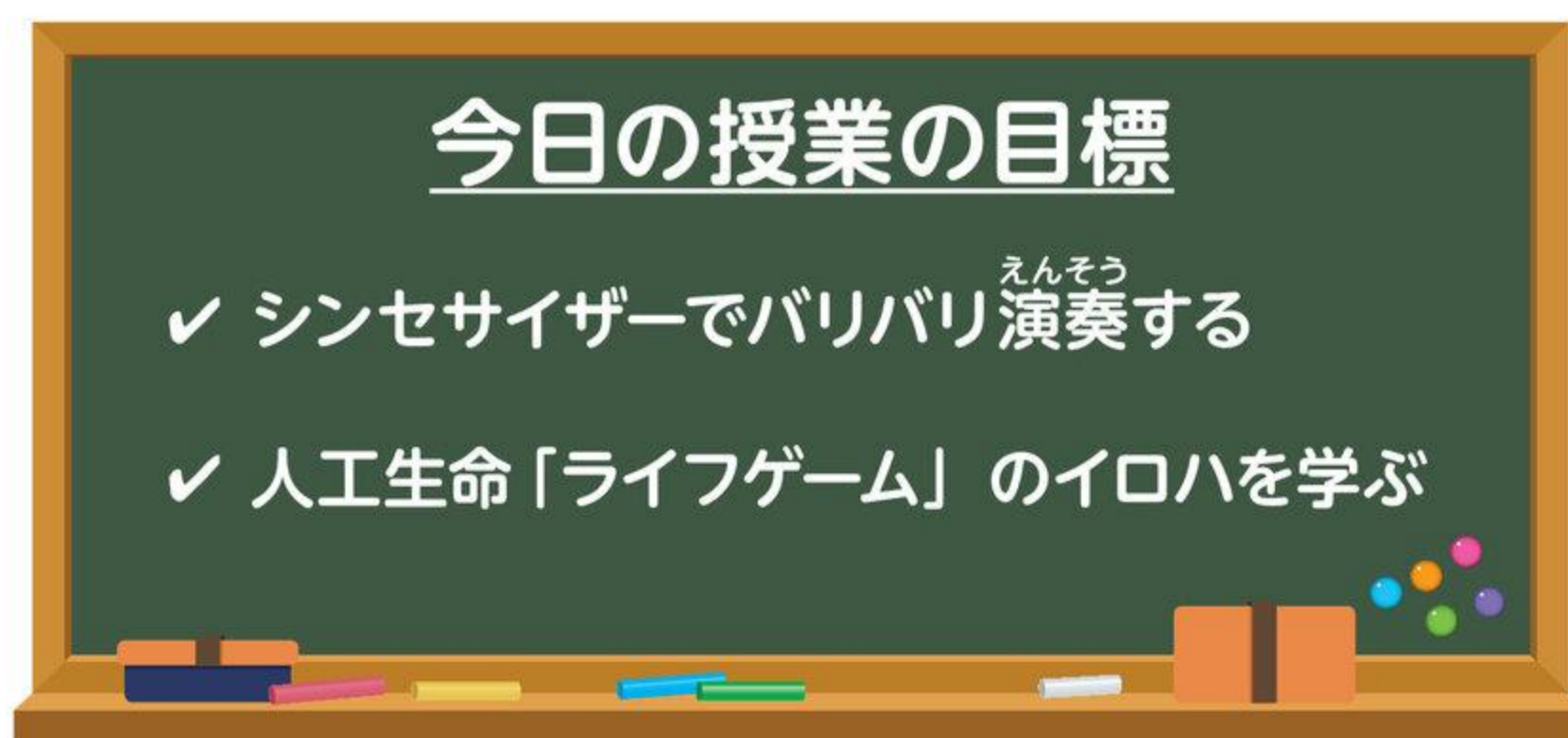
○ 今回の目標をパネルで用意するか、黒板に予め書いておきます。

(授業の目標を明確化することは大変重要なことですので、生徒によく理解させます)

目安時間は授業時間 120 分のうち、休憩 10 分程度を取ることを想定しています。
生徒の進捗状況により、休憩時間などを調整して授業を行ってください。

0. シンセサイザー^{えんそ}演奏^そと人工生命 (目安 10分)

0.0. 「シンセサイザー^{えんそ}演奏^そと人工生命」でやること

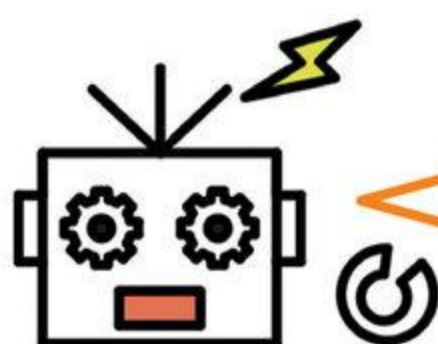


いよいよ、不思議アイテムI-1も最後となりました。プログラムロボットの脳みそとはどうやって組むものなのか、なんとなくわかってきたでしょうか？

不思議アイテムI-1では、ロボットの要素である、「感じて」「考えて」「動く」のうちの、「考えて」と「動く」の部分を重点的に学んできました。そして、「動く」では、モーターが動くだけでなく、マトリクスLEDやスピーカーといった、ロボットの意思表示部分の使い方にふれてきました。

今回は、「考えて」の部分です。人工生命をこのマイコンの中にもみ出します！人工生命というと大変すごいものを感じてしまいがちですが、そもそも生命とは何でしょうか？小さい微生物、細胞は、「もし、〇〇だったら、□□する」というレベルの単純なルールで生きています。人間は非常に複雑ですが、小さい部分で見れば、細胞がたくさん集まってできています。その小さい部分をつくってみようというのが今回の目的です。

また、シンセサイザーも、もっとカスタマイズしていきます。プログラムを使って、バリバリの楽器にして、演奏会をしましょう！



人工生命をつくっちゃうってカミサマみたいダネ！

0.1. 必要なもの

マイコンボード、ロボプロシールド、マトリクスLEDシールド、マトリクスLEDの順番に積み重ねていったユニットを、今回も使います。なお、タッチセンサーは、今回使わないため取り外しておきましょう。

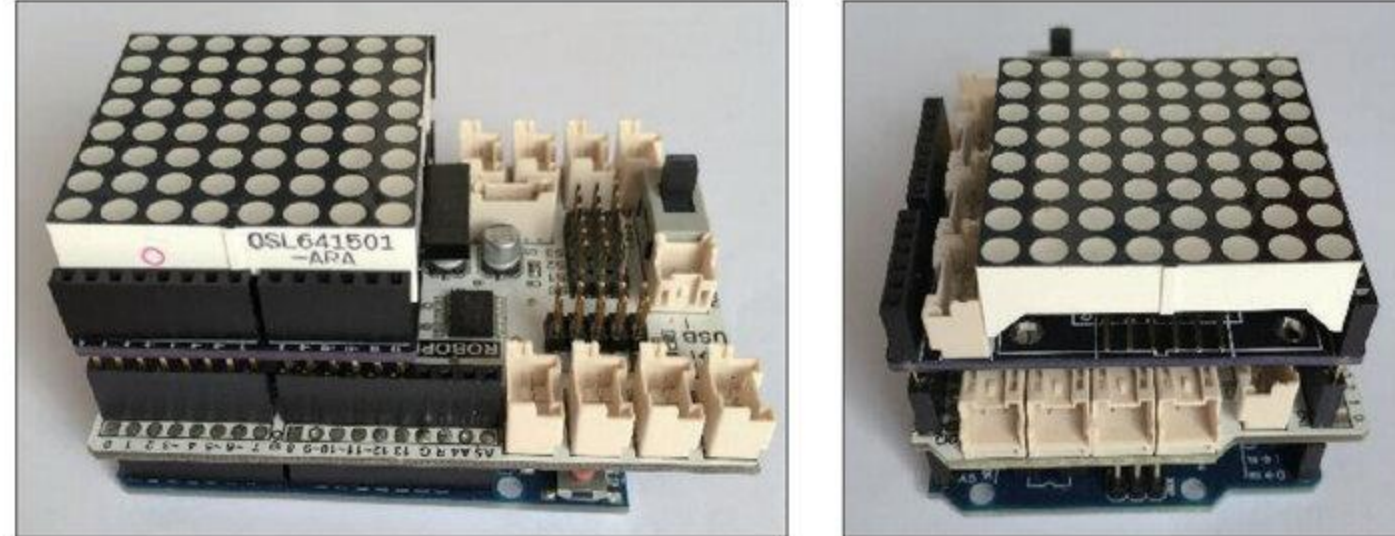


図0-0 前回までに組み立てたもの

| | | | | | | | |
|---|---|--|---|---|---|---|---|
| USBケーブル | 1 | マイコンボード | 1 | ロボプロシールド | 1 | リボンケーブル | 1 |
|  | |  | |  | |  | |
| コントローラー | 1 | 無線受信モジュール | 1 | マトリクスLEDシールド | 1 | マトリクスLED | 1 |
|  | |  | |  | |  | |
| スピーカー | 1 | | | | | | |
|  | | | | | | | |

図0-1 必要なもの

1. シンセサイザーで遊ぼう (目安 40 分)

1.0. コントローラーの接続と設定

コントローラーを接続して、シンセサイザーをつくります。

1) 無線受信モジュールの取り付け

ロボプロシールドからタッチセンサー2個を外します (スピーカーは残します)。そして、ロボプロシールドの [CN9] のコネクタにリボンケーブルを取り付けます。ケーブルの向きは、赤いラインがマトリクスLED側にくるようにしましょう。

リボンケーブルのもう一方の先には、無線受信モジュールを取り付けます。赤いラインは、アルファベットのロゴマークがついている側にくるようにしましょう。

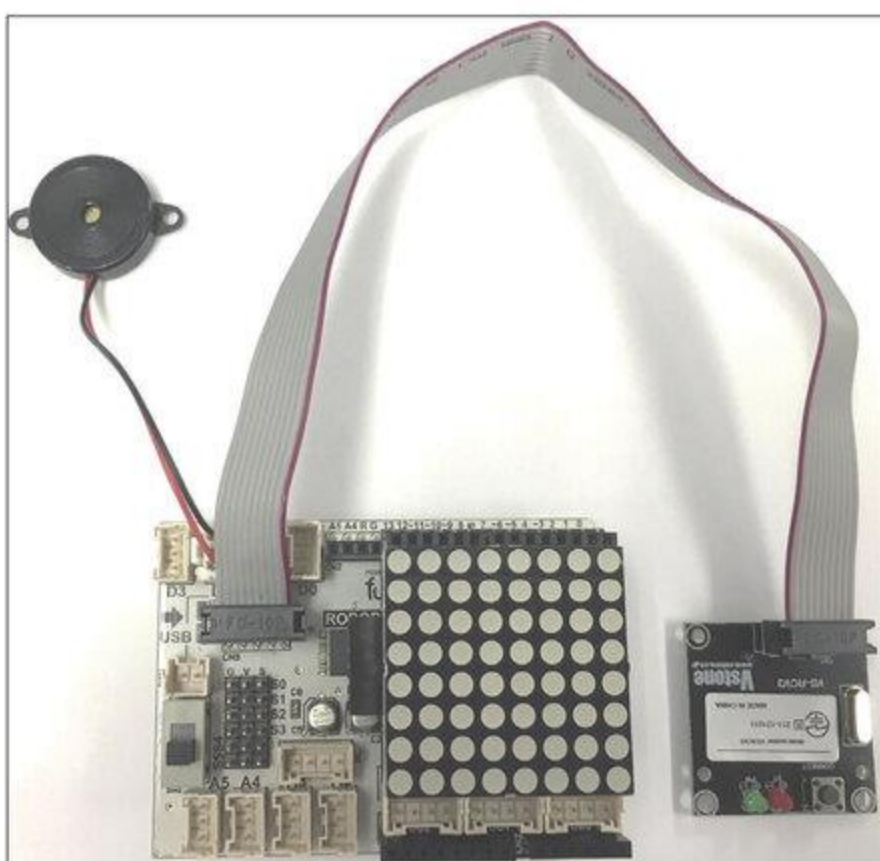


図1-0 ロボプロシールドと無線受信モジュールの接続完成図

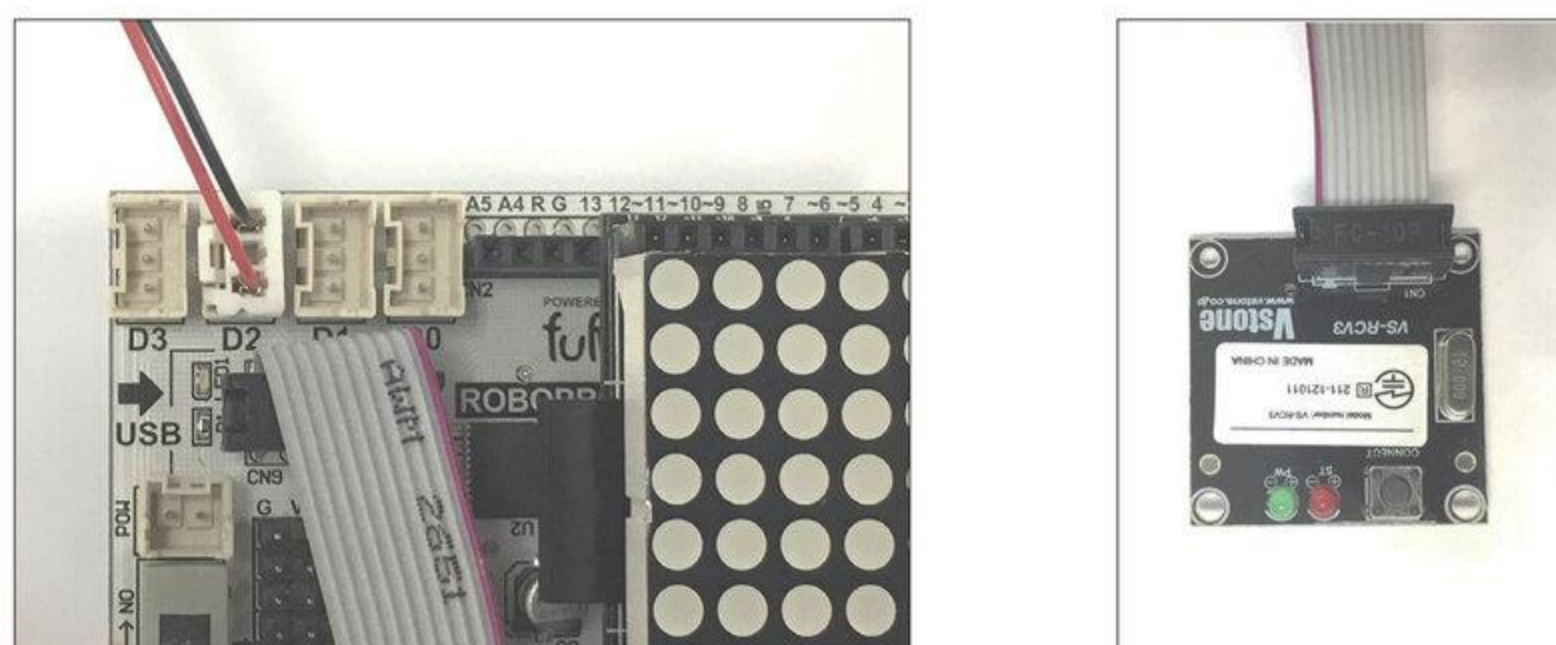


図1-1 リボンケーブルの向き

正しく接続されていれば、マイコンユニットをパソコンにつないだときに、無線受信モジュールの緑色LEDが点灯し、赤色LEDは点滅^{てんめつ}しているはずです。

講

タッチセンサーを必ず取り外させてください。タッチセンサーがロボプロシールドに接続されたままだと正しく動作しません。

2) ペアリング

続いて、コントローラーと無線受信モジュールをペアリングしましょう。すでに第3回の授業で設定してあるはずなので、コントローラーの電源を入れれば自動的に接続します。無事に接続されると、無線受信モジュールの赤色LEDが点滅を終えて点灯しているはずですが、わからなくなったら第3回のテキストにもどって接続の仕方を確認しましょう。

1.1. ボタンカスタマイズ 1

第3回でも演奏をしましたが、さらにカスタマイズをして、カッコよく演奏できるようにしましょう！ 以下のプログラムを実行してください。

プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemA6 > TonePS3

プログラムをのぞいてみると、以前のプログラム「TonePS2」と変更されている点に気づくかもしれません。自動演奏キーをつくり、[R3] ボタン、[L3] ボタン、スタートボタン、セレクトボタンに、自動演奏の演目を設定しました。どんな曲が流れるか、試してみましょう。そして、マトリクスLEDの表示も観察して、以下に結果を記入しましょう。

R3ボタンを押すと

 星が表示される。

L3ボタンを押すと

 三角が表示される。

スタートボタンを押すと

 円が表示される。

セレクトボタンを押すと

 四角が表示される。

講

R3・L3 ボタンとは、それぞれ右と左のアナログスティックです。
アナログスティックは押し込むことでボタンとしても使用できます。

やってみよう!

プログラム「TonePS3」の黄色の部分^{ぼっすい}をさしかえて、好きな曲を設定してみよう。

□ プログラム「TonePS3」より^{ぼっすい}抜粋

```
char *song1 =  
  "M2A:d=4,o=6,b=180:16g,16c7,16f7,16p,16c7,16d7,16g7,16p,16g,16c7,16d7,16f  
  7,16g7,16d7,16c7,16d7";
```

講

右のアナログスティック (R3) を押し込むと、設定した曲が流れます。

1.2. ボタンカスタマイズ2

次に、音の高低を自在にあやつれるようにしましょう。

チャレンジ課題

プログラム「TonePS3」を書きかえ、コントローラーのL2ボタンを押している間だけ、鳴る音が高くなるようにしてみよう!

💡 ヒント

「NOTE_C4」などのミュージカル・ノートは、周波数と同じように2をかけると音のオクターブが1上がるよ! L2ボタンを押しているときだけミュージカル・ノートの値が2倍になって、そうでないときはミュージカル・ノートの値が変化しない、つまり1倍のままになるようにできるかな?

講

解答例は以下のプログラムです。
RoboticsProfessorCourse1 > MagicItemA6 > TonePS4

1.3. エフェクター

どんどんいきますよ！ 以下のプログラムを実行してください。

∞ プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemA6 > TonePS5

実行結果：音を出している最中に、左右のアナログスティックを傾けると、音がゆがむ。
プログラムを見て、アナログスティックの上下左右の動きがどのように連動しているか、わかりますか？

やってみよう！

プログラム「TonePS5」の黄色の部分を、3や4に変更してみよう。どうなるかな？

□ プログラム「TonePS5」より抜粋

```
//-----  
// アナログスティックで音をゆがめる(エフェクト計算)  
//-----  
effect = ps2x.Analog(PSS_LY) + ps2x.Analog(PSS_RY) +  
2 * (ps2x.Analog(PSS_LX) + ps2x.Analog(PSS_RX));
```

講

音のゆがみ方が変わります。

1.4. シンセサイザー演奏会

パワーアップしたシンセサイザーで、いろいろなアドリブを入れながら演奏してみましょう！
好きではないキー配置や、#音の場所をカスタマイズして、オリジナルシーケンサーにもできますよ。

講

「TonePS5」で使われている「ミュージカル・ノート」を変更する時は、第3回のテキストを確認させてください。

2. 人工生命をつくろう (目安 50 分)

2.0. ライフゲームとは

マイコンが勝手に進めてくれる「ライフゲーム」という人工生命のゲームをつくります。マトリクスLEDの8×8の昇目ますめを利用して行うゲームです。ゲームのルールは自然界の生物たちのふるまいを真似してつくってあります。いわゆる、自然界のシミュレーションまねごと（真似事）です。まずは、以下のプログラムを実行してください。

∞ プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemA6 > LifeGame

実行結果：LEDに表示される二つの線がチカチカとうごめく。

今回のプログラムは少し長いので、理解しなくてもよいですが、わかる人は読み込んでみてください。プログラムは、ずっと勉強してきたforとifでできています。

やってみよう!

プログラム「LifeGame」の黄色の部分を変更へんこうしてみよう。

□ プログラム「LifeGame」 より抜粋

```
int cell[10][10]=
{
  { 0,0,0,0,0,0,0,0,0,0 },
  { 0,0,0,0,0,0,0,0,0,0 },
  { 0,0,1,1,1,0,0,0,0,0 },
  { 0,0,0,0,0,0,0,0,0,0 },
  { 0,0,0,0,0,0,1,0,0,0 },
  { 0,0,0,0,0,0,1,0,0,0 },
  { 0,0,0,0,0,0,1,0,0,0 },
  { 0,0,0,0,0,0,0,0,0,0 },
  { 0,0,0,0,0,0,0,0,0,0 },
  { 0,0,0,0,0,0,0,0,0,0 }
};
```



```
int cell[10][10]=
{
  { 0,0,0,0,0,0,0,0,0,0 },
  { 0,0,0,0,0,0,0,0,0,0 },
  { 0,0,1,1,1,0,0,0,0,0 },
  { 0,0,0,0,0,0,0,0,0,0 },
  { 0,0,0,0,0,1,1,0,0,0 },
  { 0,0,0,0,0,0,1,0,0,0 },
  { 0,0,0,0,0,0,1,0,0,0 },
  { 0,0,0,0,0,0,0,0,0,0 },
  { 0,0,0,0,0,0,0,0,0,0 },
  { 0,0,0,0,0,0,0,0,0,0 }
};
```

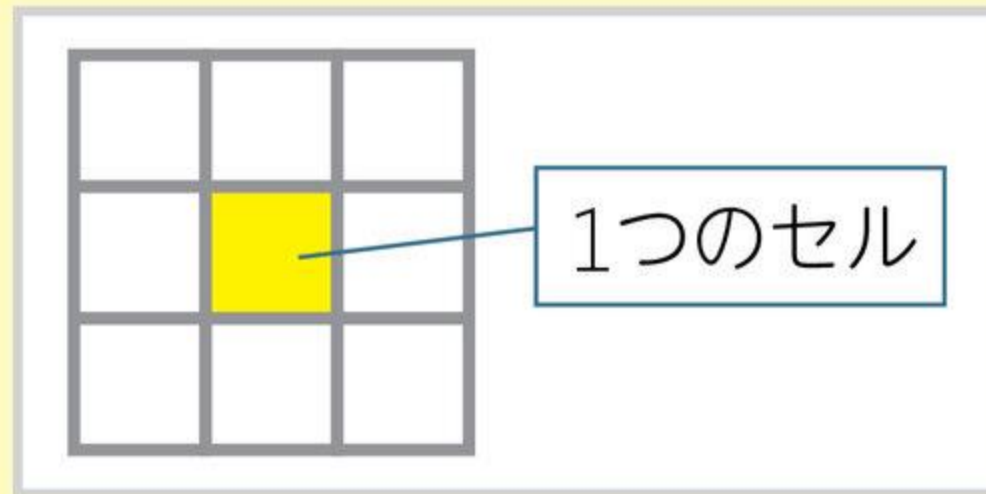
実行結果は不思議な結果になったかと思います。たった1文字の^{へんこう}変更だけで、これだけ変化するのも面白いですね。

実は、このプログラムは簡単なルールだけで動いています。1個1個のLEDを一つの生物（生命）だと思ってください。ここでは、それをセル（^{さいぼう}細胞）とよびます。

 POINT

ライフゲームのルール

ライフゲームの1つの^{ます}升はセルとよばれます。セルは周囲8つのセルと接しています。



各セルには「生」と「死」の2つの状態があり、セルの次の世代の状態は周囲の8つのセルの状態により決まります。

セルの生死は次のルールで決定されます。

1. 誕生 死んでいるセルに^{りんせつ}隣接する生きたセルがちょうど3つあれば、次の世代が誕生する。
2. 生存 生きているセルに^{りんせつ}隣接する生きたセルが2つか3つならば、次の世代でも生存する。
3. 過疎 ^{かそ}生きているセルに^{りんせつ}隣接する生きたセルが1つ以下ならば、^{かそ}過疎により死滅する。
4. 過密 生きているセルに^{りんせつ}隣接する生きたセルが4つ以上ならば、^{しめつ}過密により死滅する。

このルールは3 x 3の^{ます}升の中心に対して適用されます。例にすると、下図のようになります。生きているセルは■で、死んでいるセルは□で表現しています。

| 誕生 | 生存(維持) | 過疎(死) | 過密(死) |
|----|--------|-------|-------|
| | | | |

生命1個では子孫が^{ぜつめつ}残らず絶滅し、多すぎると食物を食べつくして^{ぜつめつ}絶滅することを画面上で再現しています。

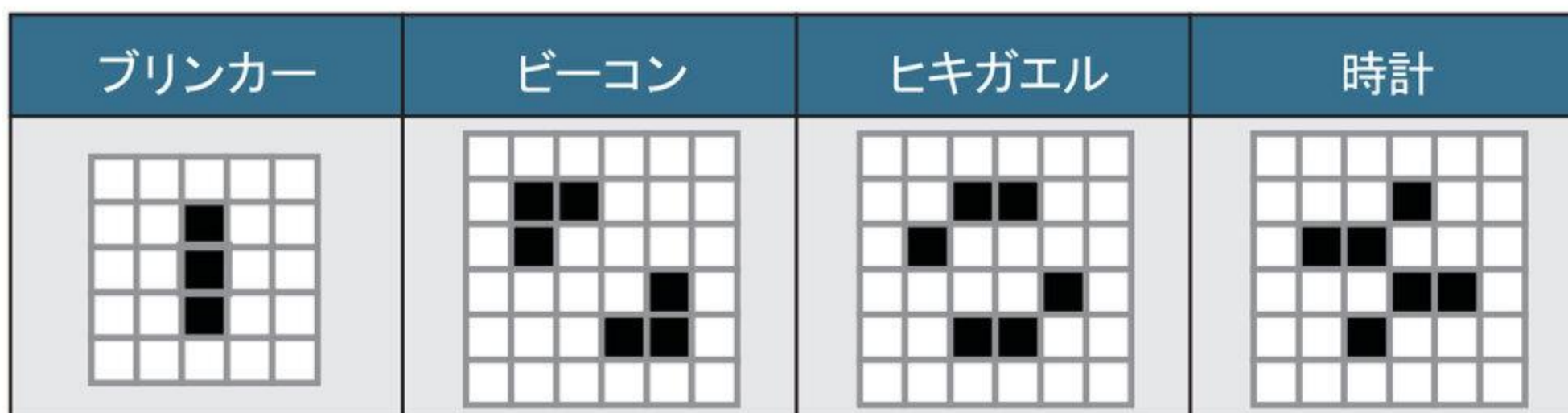
「やってみよう!」では、はじめに生物のいる場所を指定していたのです。最初にどこに生物がいるかで、その^{はんえい}繁栄の状況は大きく変わってしまうことがわかりますね。

画面が変わるたびに、その人工生命の一世代目、二世代目と世代交代をくり返し、最後にはなくなってしまうパターンもありますし、「やってみよう!」のように固まって安定するものもあります。逆に、^{へんこう}変更前のプログラムのように同じ動きをくり返し続けるパターンもあります。

ライフゲームの^{はんえい}繁栄の状況は、前のページに書かれているような「最後にはなくなってしまうパターン」「固まって安定するパターン（固定物体）」「同じ動きをくり返し続けるパターン（^{しんどうし}振動子）」のどれかであることがほとんどです。固定物体と^{しんどうし}振動子のうち代表的なものを、いくつか下で^{しょうかい}紹介します。



図2-0 固定物体のパターン

図2-1 ^{しんどうし}振動子のパターン

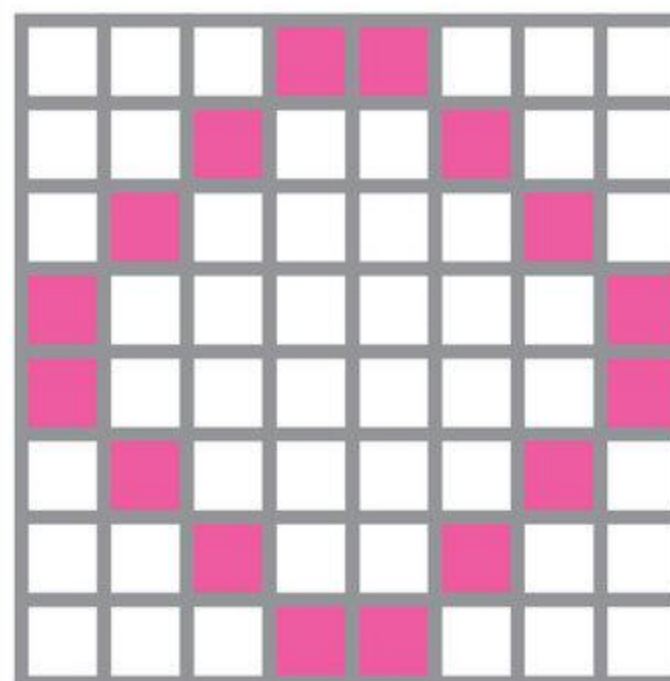
^{しんどうし}振動子は、くり返されるパターン数のことを「周期」といいます。図2-1に登場した^{しんどうし}振動子はすべて2パターンで1周なので、周期2といえます。もちろん、周期が2よりも大きい^{しんどうし}振動子も見つかっています。

やってみよう!

8×8マスでつくれる^{しんどうし}振動子の中に、「八角形」という名前の周期5のパターンがあるよ。名前と大きさをヒントにこの^{しんどうし}振動子をつくり、下のセルに書いてみよう！ただし、黄色の部分の中だけを変更してね。

□ プログラム「LifeGame」より^{ぼっすい}抜粋

```
int cell[10][10]=
{
  { 0,0,0,0,0,0,0,0,0,0 },
  { 0,0,0,0,0,0,0,0,0,0 },
  { 0,0,1,1,1,0,0,0,0,0 },
  { 0,0,0,0,0,0,0,0,0,0 },
  { 0,0,0,0,0,0,1,0,0,0 },
  { 0,0,0,0,0,0,1,0,0,0 },
  { 0,0,0,0,0,0,1,0,0,0 },
  { 0,0,0,0,0,0,0,0,0,0 },
  { 0,0,0,0,0,0,0,0,0,0 },
  { 0,0,0,0,0,0,0,0,0,0 }
};
```

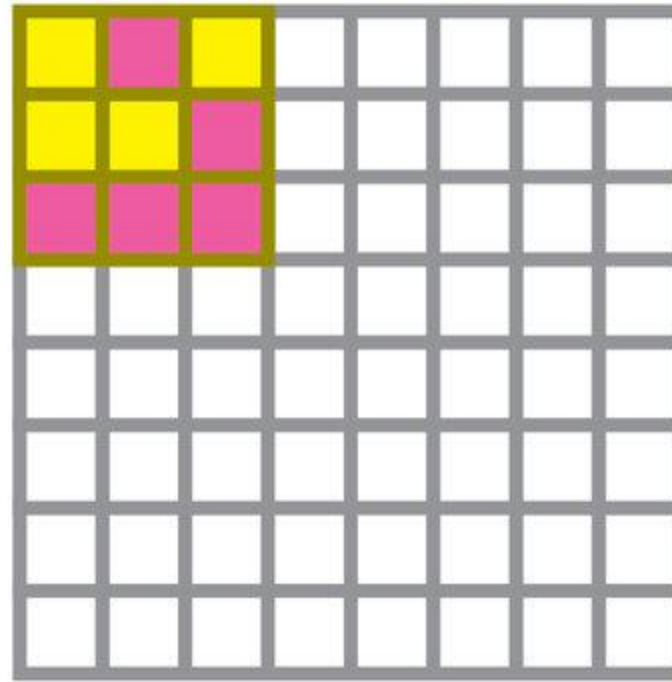


2.1. 代表的なパターンを試してみよう

また、ライフゲームには「固定物体」「振動子」のほかに、「移動物体」とよばれるパターンも存在します。振動子と同じようにいくつかのパターンをくり返して形を変化させながら、さらに場所も移動していく、というものです。代表的なものに、3×3マスの範囲でつくれる「グライダー」という名のパターンがあります。

やってみよう!

黄色のマスの状態をいろいろ変更して、グライダーのパターンを探してみよう!



非常に単純なパターンなのに、多彩な動きを見せて不思議ですよ。自然界には、いろいろな種類の生物がいても、環境に適応したものだけが生き残り、そうでないものは死滅するというシステム（淘汰といいます）があります。ライフゲームも淘汰のシステムを模したものといえますね。

淘汰のシステムをプログラムにいかしているものにはほかに「遺伝的アルゴリズム」などの手法があり、新幹線の先端の形を決定するのに活用されるなどして注目されています。

人工生命、自然界の不思議を少し垣間見ることができたでしょうか？ 単純なものが積み重なって、複雑に見えているのです。これは前回お伝えした、プログラムのつくり方とまったく同じですね。単純な機能のプログラムをたくさんつくってつなぎ合わせることで、ロボットも複雑な動きをします。本当にロボットは、自然の真似事をしているわけなんです！

ライフゲームは、「生」と「死」という2種類の升(セル)を使っていて、「生」と「死」の切りかわりにはある決められたルールがありました。このように、周りのセルと影響しあいながらセルの状態を切りかえていくシステムを「セル・オートマトン」といいます。ライフゲームもセル・オートマトンの一種ですが、ライフゲームとはまた違ったルールでセルを切りかえていくものもあります。たとえば、下のようなルールのセル・オートマトンを考えてみましょう。

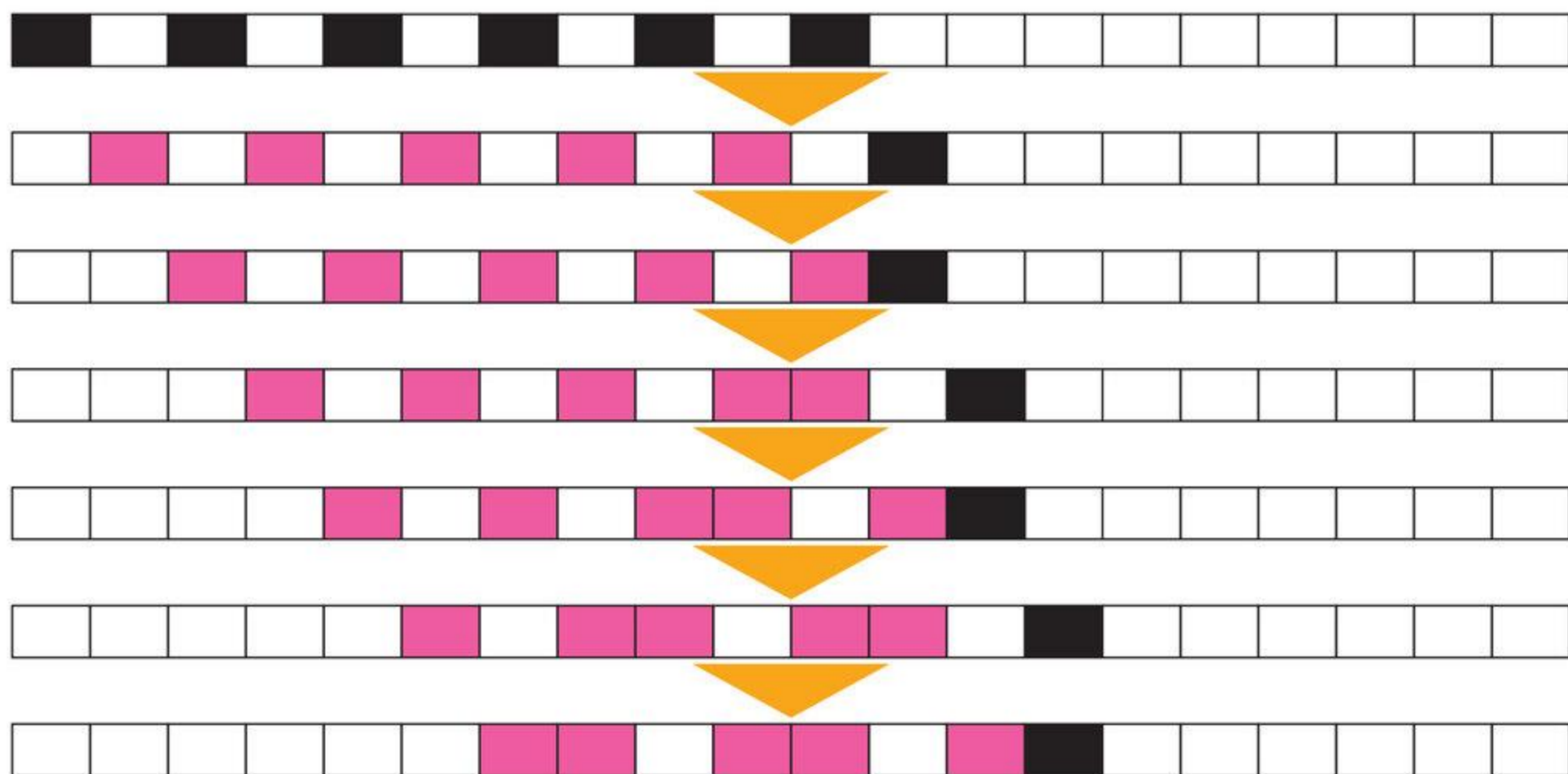


POINT

- ルール①：横1列にセルが並んでいる。セルには黒と白の2種類がある。
- ルール②：黒いセルは1マスずつ右に移動していく。
- ルール③：ただし、黒いセルの左隣にある黒いセルは、前がつまっているので移動しない。
- ルール④：集団の中で一番右にある黒いセルだけは、2回に1回しか移動しない。

チャレンジ課題

上のルールにしたがい、黒いセルがどのように移動していくか書いてみよう！
一番右にある黒いセルは図に示しておくから、他のセルを考えてみてね。



右のセルの遅れが左のセルに影響していきましたね。そして、黒いセル同士の隙間がどんどんなくなってきました。

実はこれは、交通渋滞のようすを簡単に表したシミュレーションなのです。黒いセル1つ1つが道を走る車を表しています。一番前の車が頻繁にブレーキをかけると、どんどん後ろが渋滞していつてしまうことがよくわかりますね。あわせて車同士の間隔がせまくなり、交通事故の危険も大きくなってしまいます。

このように、セル・オートマトンはたくさんの生き物の分布、人の動きなどの、観察できないくらい複雑な様子をシンプルに表せるという利点があり、災害時の人々の避難の様子を予想したりするのに使われています。



コラム ライフゲームのあれこれ

ライフゲームは、1970年ごろにイギリスの数学者コンウェイが考案したゲームです。ゲームといっても、ユーザにできることはライフの初期配置を決めることだけで、あとは画面上の変化を楽しむものです。

ライフゲームの規則はとても単純ですが、そこから生まれる「生命」のパターンは非常に多彩です。

単純な要素が集まっているだけなのに、全体的にはとても複雑な動きを見せていますね。人工知能の開発において、このような現象のことを「創発」といいます。

実際の生物も、はじめは非常に単純な作りだったものが、自己複製を経てどんどん複雑な作りへと成長していきます。

もし、コンピューターでもこれが再現できれば、「人工生命」をうみ出すことすらできるかもしれませんね。



豆知識

これまでに紹介したライフゲームのパターンはすべて、はじめにつくった物体がそのまま残ったり形を変えたりするだけで、生命が「ふえて」いくことはほとんどありませんでした。しかし実は、新たな生命を無限にうみ出せるパターンも見つかっているのです。はじめにつくった物体が無限に大きくなっていく「マックス」、前のページでつくった「グライダー」を次々とうみ出していく「グライダー銃」などです。

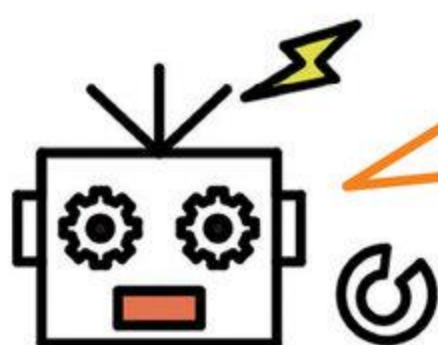
また、蒸気機関車のように煙を次々とうみ出しながら進む物体も見つかっており、「シュシュポッポ列車」と名づけられています。ライフゲームの物体の中には、「空母」「銀河」「重量級宇宙船」といった壮大なものもあれば、シュシュポッポ列車のようなコミカルな名前のももあります。興味のある人は調べてみると楽しめるでしょう。

3. まとめ (目安 10 分)

以上をもって「不思議アイテム I-1」は終わりです。ロボットらしくないと思ったかもしれませんが、ここで学んだことも立派なロボットの一部です。

「感じて」、「考えて」、「動く」、この基本ルールを実行するために、ロボットに「考え」させる「if文」、「for文」、「switch-case文」などを勉強しました。そして、ロボットに意思表示させるために、マトリクスLEDやスピーカーなどの使い方も勉強しましたね。

マイコンをさらに深く理解してあげて、もっとすごいプログラムを書けるようになっていきましょう！そして、カシコイロボットをつくりましょう！みなさんがカシコクなればなるほど、ロボットもカシコクなります。そして、当たり前ですがプログラムを書く人以上にロボットはカシコクなりません。数学、理科はもちろん、自然の摂理など、どんなこともロボットプログラミングには役立つので、いろいろなことに興味を持って勉強してってください！



これで、意思表示はバッチリできるようにナッタゾ！
内気なロボットは卒業ダゼ！ イエーイ

講

- 以下の理解度を確認します。
 - ・シンセサイザーでバリバリ演奏する
 - ・人工生命「ライフゲーム」のイロハを学ぶ
- 次回、1年目コースは「リンクロボット」、2年目コースに進級する場合は「センサーロボット」になります。

《次回必要なもの》

今回は、以下のパーツを持ってきてください。

| | | | |
|---|--|---|---|
| ラジオペンチ 1 | ドライバー 1 | レンチ 1 | USBケーブル 1 |
|  |  |  |  |
| マイコンボード 1 | ロボプロシールド 1 | 電池ボックス 1 | ギアドモーター 2 |
|  |  |  |  |
| リボンケーブル 1 | 無線受信モジュール 1 | 白円形ボード 1 | M3L5ネジ 2 |
|  |  |  |  |
| M3L25ネジ 4 | 8mm角スペーサー 4 | 30mm角スペーサー 2 | リンクアーム (大) 12 |
|  |  |  |  |
| リンクアーム (中) 12 | リンクアーム (小A) 12 | リンクアーム (小B) 12 | 脚リンク (上) 12 |
|  |  |  |  |
| 脚リンク (下) 12 | フレーム 8 | モーターホーン 12 | ユニバーサルボード 1 |
|  |  |  |  |

図3-0 次回必要なもの①








| | | | | | | | |
|---|----|--|----|---|---|---|----|
| M3L8ネジ | 10 | M3L6フラットヘッドビス | 48 | M2.3L6タッピングネジ(B) | 2 | M3L6ネジ | 22 |
|  | |  | |  | |  | |
| M3ナット | 29 | M3L10ネジ | 4 | 15mm丸スペーサー | 6 | 25mm角スペーサー | 12 |
|  | |  | |  | |  | |

図3-1 次回必要なもの②