

# Let's えいごパズル !

誰でも楽しく英単語



平野正太郎 shotaro27.github.io

# きっかけ

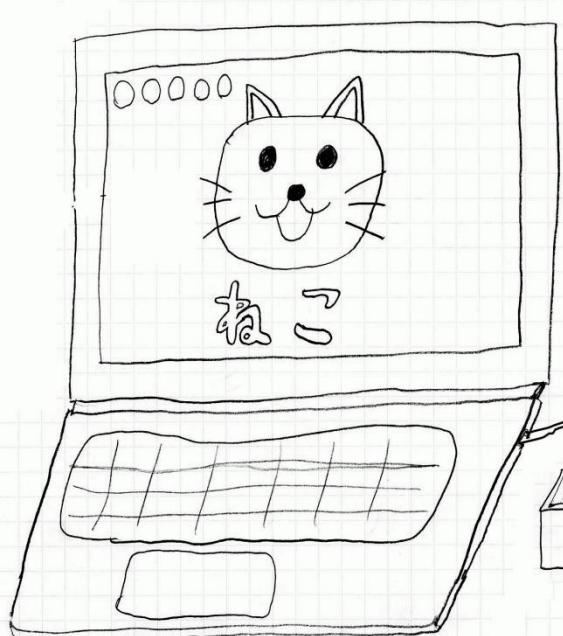
僕が作品展で作品を発表していると、小さい子がたくさん遊びにきます。小さい子は大きいスイッチを押したり、箱を動かしたりキラキラするものがとても好きです。そんな小さい子たちが楽しみながら、その上ちょっと勉強にもなるゲームがあるといいなと思い、この作品を考えました。

## 概要

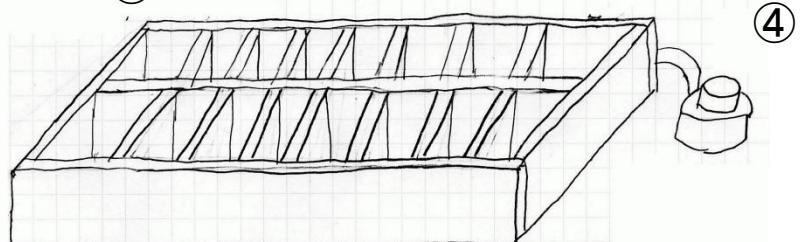
初期構成はこの様な形でした。それぞれいろんな形で繋がっています。

①のPCに問題が出ます。②パズル台に置く③キューブは、最初は上にマトリクスLEDをつける予定でしたが、液晶ディスプレイに変更しました。パズル台も横に8個までキューブを入れる予定でしたが、少し大きめの液晶ディスプレイに変更したために5個までに変更しました。④のボタンは問題を操作するために増やし、最終的に3個にしました。

①

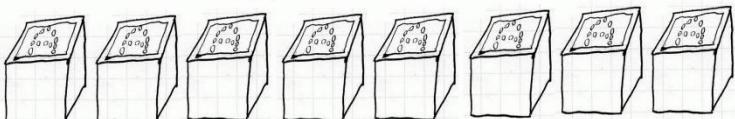


②



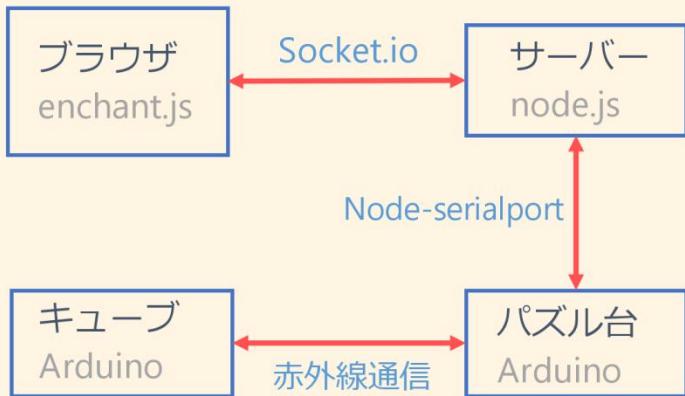
④

③



# 内部構造

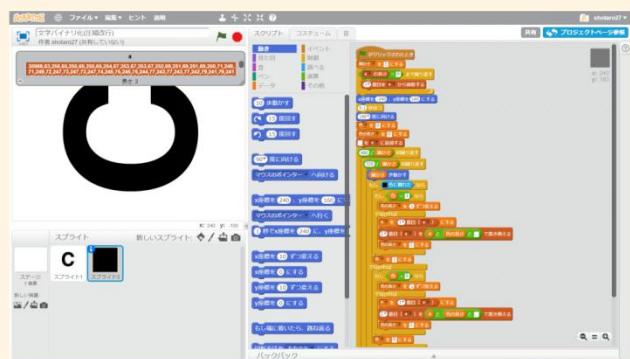
## 内部構造



キューブ編

## キューブについて ディスプレイデータの作り方

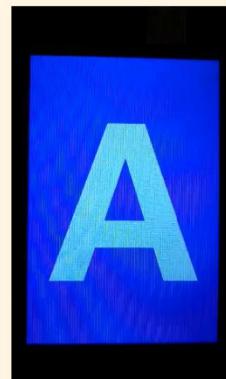
- Scratchで文字を出して、それを読み取るプログラムを作り、白黒を0,1のリスト化してソースに入れる仕組みを作った
  - <https://scratch.mit.edu/projects/231022726/>  
→ディスプレイを  
1pxの解像度で表示



## キューブへのデータの書き込み PROGMEM→ランレングス→メモリ

- ランレングス圧縮してPROGMEMに置く
- ランレングス法

**0000000000  
0000110000  
0001111000 → 14,2,7,4,13  
0000000000**



- 1pxの制度だとデータが入らないので、ランレングス圧縮してPROGMEMに置きました。ランレングス法は、データの0と1のつながっている長さだけを配列化したものです。

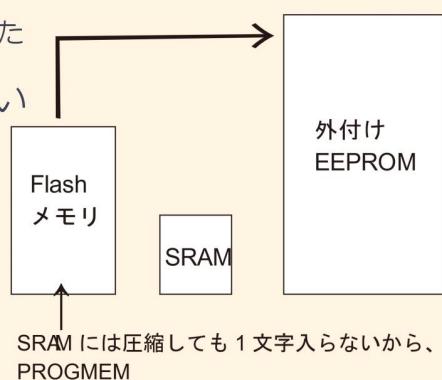
## キューブへのデータの書き込み PROGMEM→ランレングス→メモリ

- 26文字だとそれでも入らなくなつた

→外付けのメモリ (EEPROM) をつけた

- Arduino自体にデータが全部入りきらない  
→1文字ずつ文字データをいれる  
プログラムを作つて書き込んだ

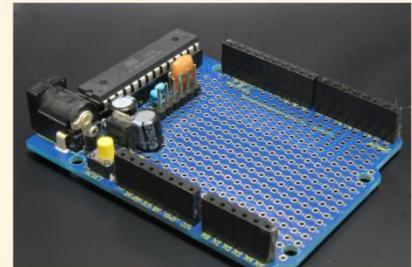
キューブのプログラム



<https://github.com/shotaro27/LetsEigoPuzzle/tree/master/cube>

## キューブver.1 互換基板

- Arduino互換機  
「Universalno」使用  
→Arduinoとユニバーサル基板が  
一体化した形



[http://ghz2000.com/wordpress/?page\\_id=2](http://ghz2000.com/wordpress/?page_id=2)

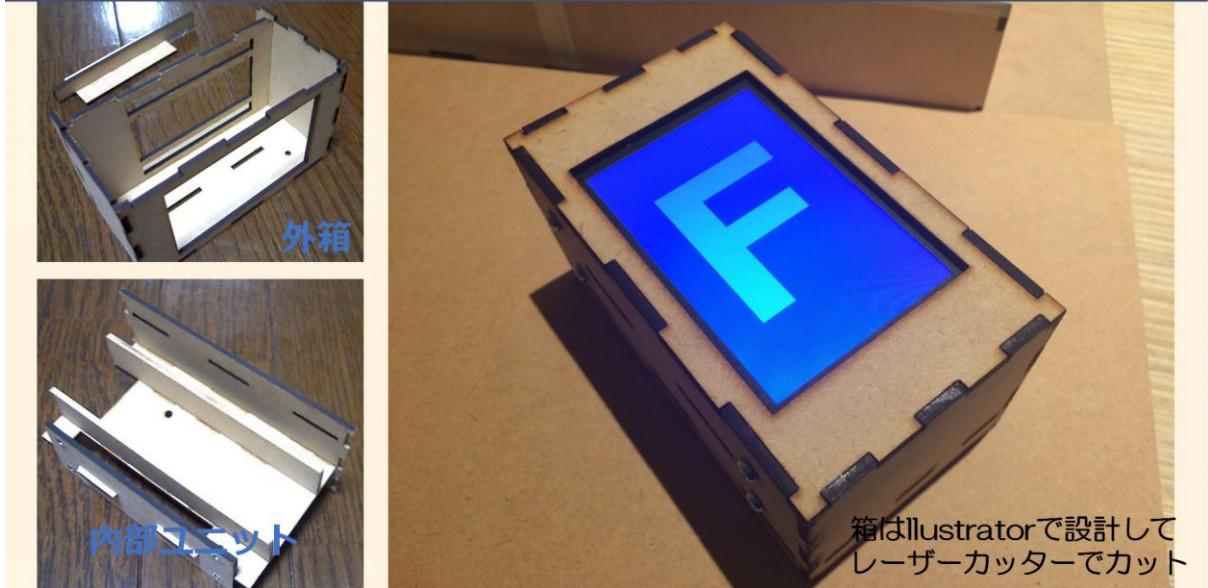
- Arduinoとユニバーサル基板が一体化した形で、自由にパーツが配置できるのでメモリや赤外線LEDなどをつけることができる。

## キューブver.1 内部構造

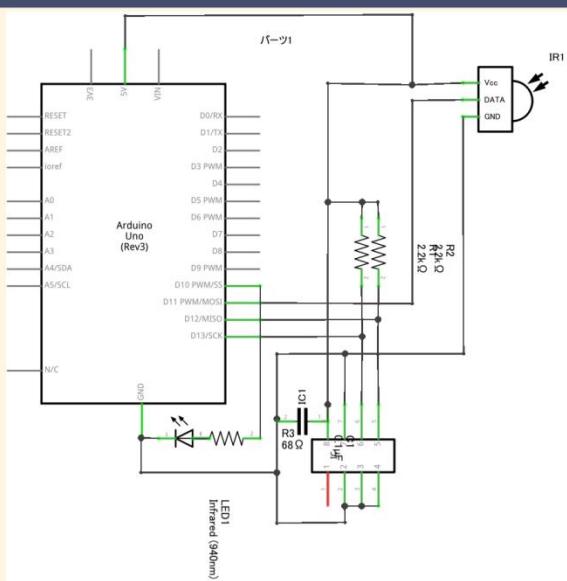
Universalnoにメモリを付け、裏側に受信機とLEDをつけた。



## キューブver.1 完成



## 回路図（キューブ）



## キューブver.3

- ver.1のキューブは大きくて持ちにくかった為、

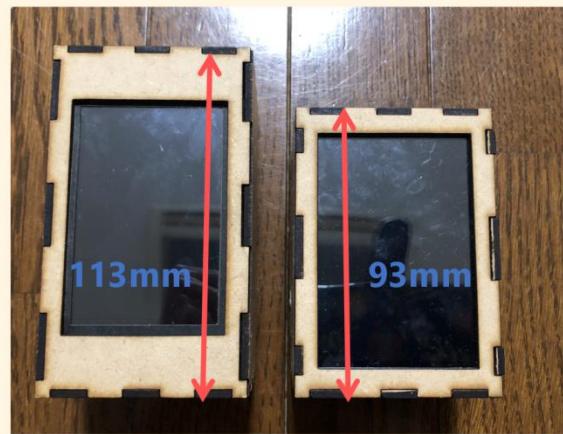
自作プリント基板を作成し約2cm縮小

中のプリント基板をeagleで設計して中国に発注し、  
ディスプレイから基板が出ないように再設計

Universalino使用



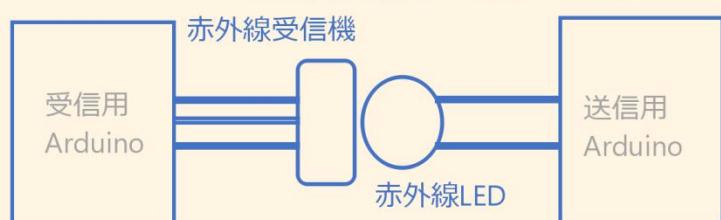
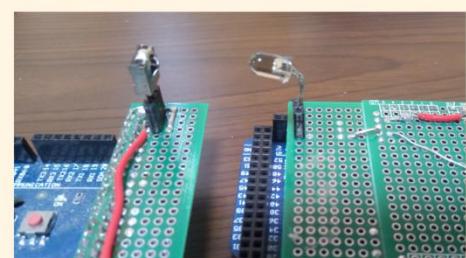
自作基板



# 通信編

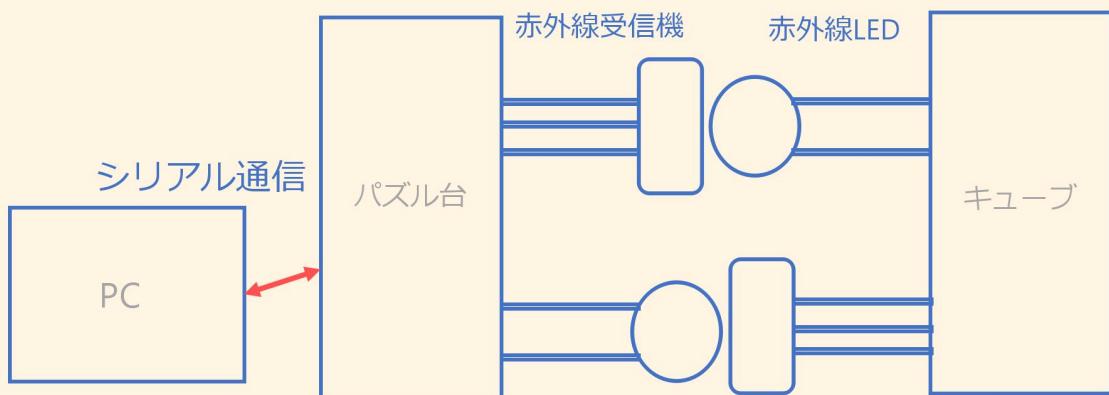
## 通信編 赤外線通信の方法

- ・ 文字をASCIIコードに変換 例) A→0x41 B→0x42 C→0x43
- ・ ASCIIコード通りにLEDをひからせる
- ・ 10~20回試してみて、  
動かさなければ100%成功していた  
送信側や受信側が動くと誤受信する  
→ノイズと振動に弱い  
→試しのはんだ付け
- ・ 38kHzの周波数でLEDを  
光らせる



## 通信編 赤外線通信の方法

- Arduino対Arduinoではうまく送れるようになったので、ゲームの試作としてPCにつなげて、パズル台対キューブで赤外線通信

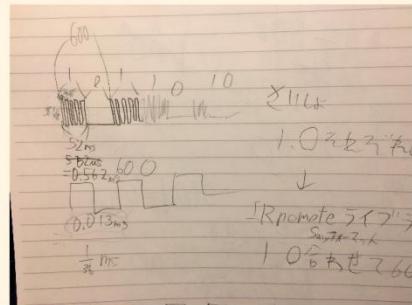


## 通信編 赤外線通信の方法 ライブラリを改造

- シリアル通信を介すると、誤受信が発生
- 別のデータが入ってくるのと区別するため、\$を一回送信してから文字のデータを送って、もう一回\$を送信する方法を試す  
例 (\$A\$)  
→変わらなかった
- タイミングを細かく調整してAだけはうまくいくようになった  
→Aで揃えるとCはうまくいかない

## 通信編 赤外線通信の方法 ライブライリを改造

- ・赤外線について改めて勉強
- ・赤外線は同期が大事
- ・テレビのリモコンは、どうやって通信しているのか考えた
- ・調べてみたところ、NECフォーマットやSONYフォーマットなどの通信方法が決められていた
- ・同期ができるで通信が速くなる



## 通信編 赤外線通信の方法 ライブライリを改造

- ・Arduinoでフォーマットを使った送信ができるライブラリを発見  
IRremote <https://github.com/z3t0/Arduino-IRremote>
- ・早速使ってみようとしたら...
- ・赤外線の送信をするピンが3番か9番ピンに決まっていて、両方ディスプレイに使うピンだった（タイマーのPWMの関係）  
  
ということで諦めかけたが
  - ・「ライブラリの改造はできないだろうか」  
9番ピンを使っている部分を10番に変更... 動かない

## ライブラリを改造

- PWMの設定レジスタについて一行ずつ調べて解決 ライブラリが使えるように改造

```
#define TIMER_ENABLE_PWM (TCCR1A |= _BV(COM1B1)) // ここを1Aから1Bに変えた
#define TIMER_DISABLE_PWM (TCCR1A &= ~(_BV(COM1B1))) // ここを1Aから1Bに変えた
#define TIMER_ENABLE_INTR (TIMSK1 = _BV(OCIE1A))
#define TIMER_DISABLE_INTR (TIMSK1 = 0)
#define TIMER_INTR_NAME    TIMER1_COMPA_vect
#define TIMER_CONFIG_KHZ(val) ({ \
    const uint16_t pwmval = SYSCLK / 2000 / (val); \
    TCCR1A      = _BV(WGM11); \
    TCCR1B      = _BV(WGM13) | _BV(CS10); \
    ICR1        = pwmval; \
    OCR1B       = pwmval / 3; // ここを1Aから1Bに変えた \
}) \
中略 \
#define TIMER_PWM_PIN 10 // ここを9から10に変えた
```

<https://github.com/shotaro27/LetsEigoPuzzle/blob/master/boarddefs.h>

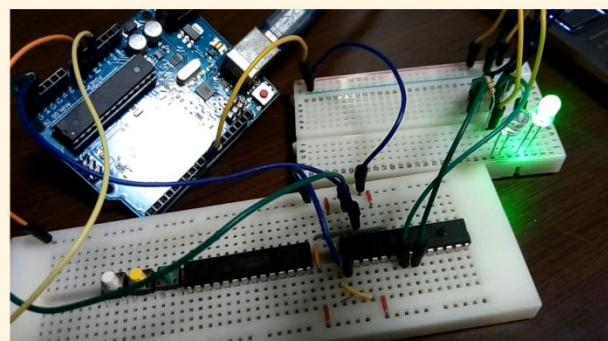
## パズル台編

### パズル台編 IOエキスパンダとトランジスタと電流

- ・パズル台は初期設定では16個赤外線LEDをつける予定だった
- ・**IOエキスパンダ**を使用してつなげる
- ・ライブラリが使えるように、  
トランジスタを使ってピンを  
切り替える

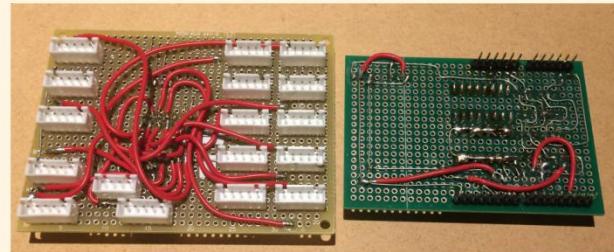


B: ベース  
C: コレクタ  
E: エミッタ



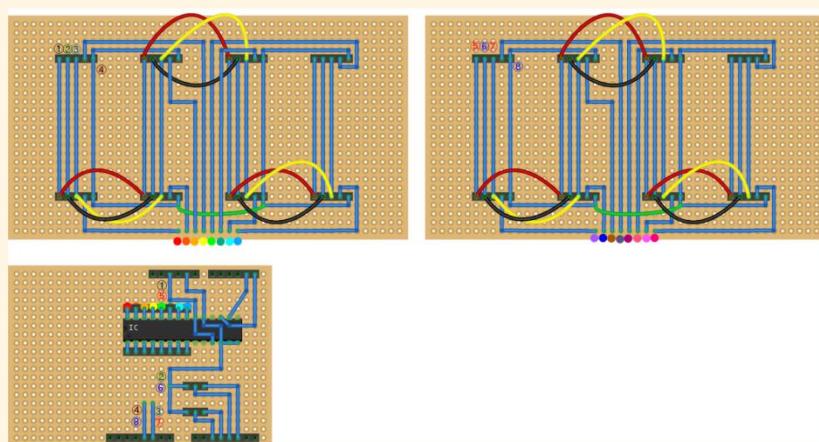
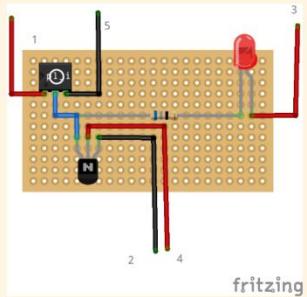
## パズル台編 IOエキスパンダとトランジスタと電流

- ・シールドを作った
- ・一枚に収めるのが大変  
→うまくはんだ付けが  
できていない部分が多くて、  
動くときと動かないときが  
あった



## パズル台編 IOエキスパンダとトランジスタと電流

- ・シールドを再設計



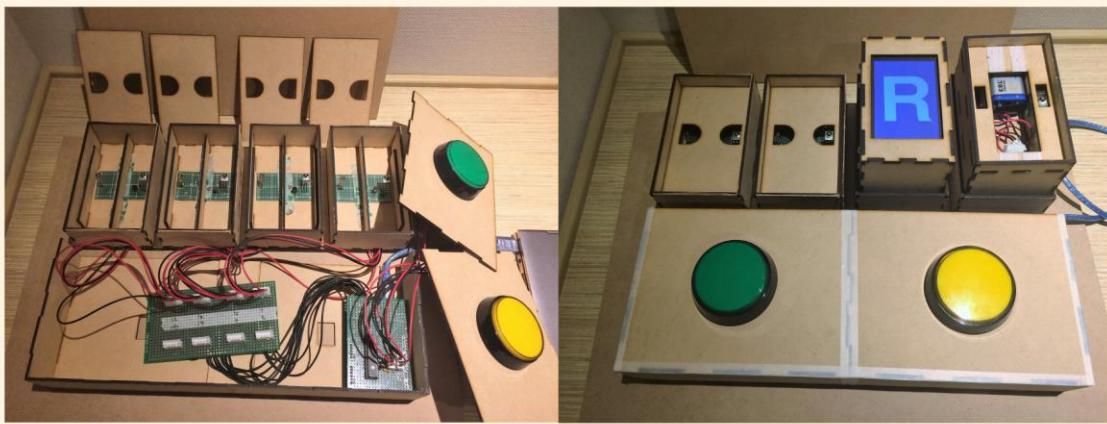
- ・プログラム

<https://github.com/shotaro27/LetsEigoPuzzle/tree/master/puzzle>

## ゲーム編

### ゲームver.1 内容とボタン

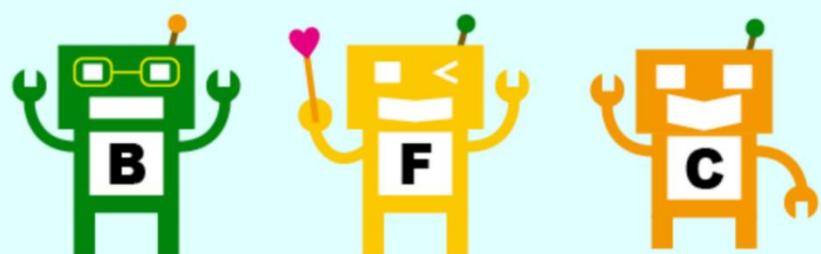
- ver.1は4個のパズル台と4個のキューブ
- ボタンは2つ



# Let'sえいごパズル！



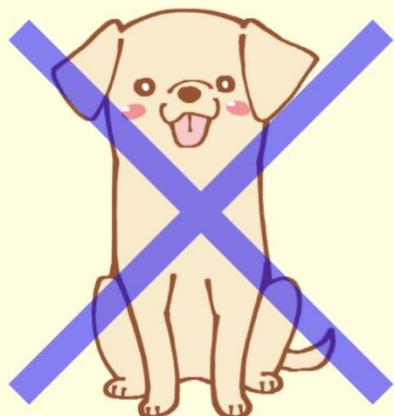
ボタンをおしてスタート



ver.1 トップ画面

もんだい3

いぬ



- はつおんを  
きく
- やりなおし
- こたえ  
あわせ



ver.1 ゲーム画面

もんだい1

● はつおんを  
きく

○ もういちど

★ つぎに  
すすむ

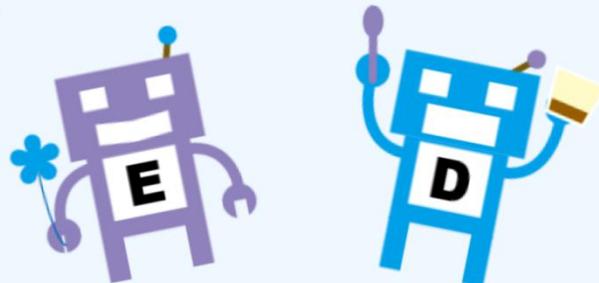


オレンジ

ORANGE

ver.2 ゲーム画面

1. × ORANGE
2. × CASTLE
3. × DOG
4. ○ BOOK
5. ○ APPLE



5もんちゅう2もんせいかいだよ！  
ぜんもんせいかいめざして、がんばろう！

ver.1 結果画面

## ゲーム本体

ver.2

ver.1では4文字までしか対応できなかったのが、穴埋め問題にする事によって、5文字以上の単語にも対応できるようになった。

ver.1



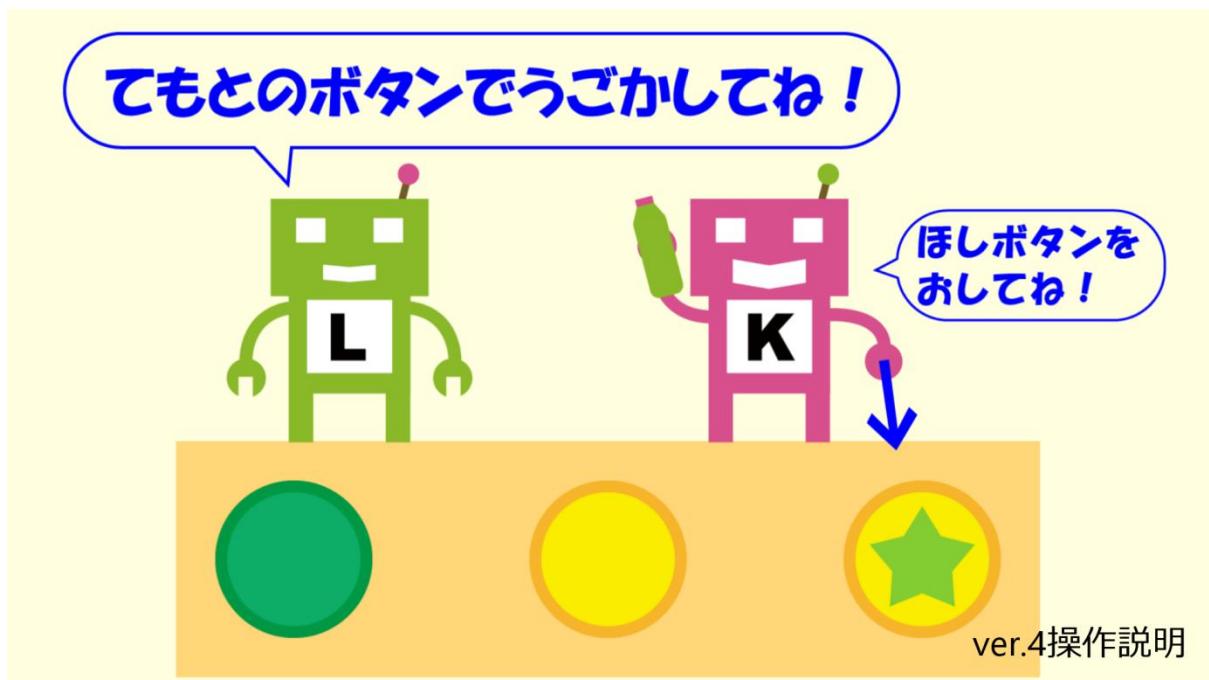
# Let's えいごパズル！ver.4

## ゲームver.4 改良点

作品展で発表をして、たくんさの人に体験してもらい改良を重ねた

### ver.4までの改良点

- ・キューブの小型化
- ・キューブの追加 4個→5個
- ・キューブの文字表示の速度アップ、色で誤受信を判別
- ・ゲーム中に発音を聴く機能、間違えたら正解までやり直す機能の追加
- ・難易度の追加、ステージの追加
- ・UIの改良（小さい子に親しみやすいかわいらしいデザイン）



# Let's えいごパズル！遊び方

手元の3つのボタンすべて操作します。

5つのキューブにはそれぞれ一文字づつアルファベットが表示されます。

画面の指示に従ってそれをなぞれます。



①ゲームの進行はすべてゲーム本体の3つのボタンを使用します。  
星ボタン を押してスタートします。



②ゲームのオープニング画面です。  
星ボタン で進みます。



③ゲームの遊び方の説明です。

1. 画面の絵をみて、キューブに表示されているアルファベット  
を正しいつづりに入れ替える



2. キューブの色が揃っていない時は、黄色ボタン を押して  
揃うまで再読み込みする。

星ボタン で進みます。



④ステージを選択します。  
黄色ボタン で右のステージに移動し、  
緑ボタン で左に移動します。  
STAGE 1は動物、STAGE 2は食べ物、  
STAGE 3はそれ以外です。  
星ボタン で決定します。



⑤緑ボタンでかんたんとふつうのモードを切り替えられます。  
かんたんは画面に絵と一緒に英単語が表示されるので、  
確認しながらアルファベットを揃える事ができます。  
黄色ボタンでステージ選択に戻る事ができます。  
星ボタンでゲームスタートです。



⑥かんたんモードでは画面に絵と日本語訳と英単語が表示されます。  
ふつうモードでは英単語は表示されません。  
問題が始まると同時に、それぞれのキューブに文字が表示されます。  
キューブの色がそろってない時は黄色ボタンを押して、  
色が揃うまでやり直します。  
色が揃ったらパズル台の上のキューブを並び替えて、  
正しいつづりにします。



⑦単語が5文字以上になる場合は、穴埋め問題になります。  
画面に表示されてない文字を当てはめます。

⑧緑ボタンで先に発音を聞いてみることもできます。

⑨星ボタンで答え合わせをします。

正解なら○が、間違いなら×が表示されます。

○の場合、画面に正しい文字が表示されます。

×の場合、間違っている所は表示されません。

もう一度正しく並び替えましょう。

並び替えたら、再度星ボタンで答え合わせをします。

⑩覚えられたかどうかの確認の為に、黄色ボタン

で再度同じ問題に挑戦することができます。

⑪緑ボタンを押すと発音を何度も聞くことができます。

⑫星ボタンで次の問題に進みます。問題は全部で3問です。



⑬3問終わったら、結果が表示されます。

⑭星ボタンで次に進みます。



⑮9個の枠に正解した単語が埋まっていきます。

ここに表示される9個は間違えた場合は表示されず、正解するまで

埋める事ができません。9個の単語は全て埋まるまで変わりません。

全て埋まった時点でデータがリセットされ、ランダムで

違う9個の単語になります。

緑ボタンで同じステージの同じモードを繰り返す事ができます。

星ボタンでトップ画面に戻ります。





1

はつおんを  
きく

やりなおし

こたえ  
あわせ



かえる  
**FROG**



ver.4ゲーム画面

2

はつおんを  
きく

もういちど

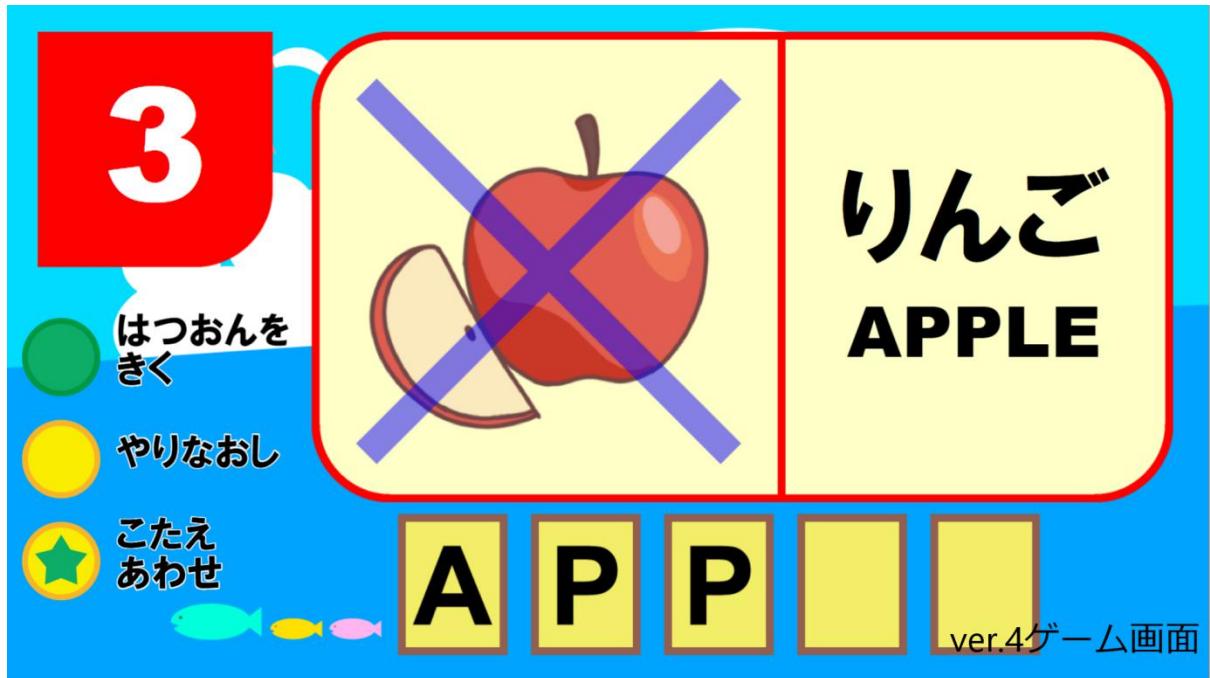
つぎに  
すすむ



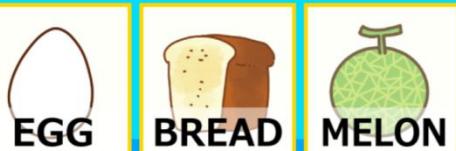
ばなな  
**BANANA**

**B A N A N A**

ver.4ゲーム画面



# おぼえたえいご



LEMON CORN MEAT



もういちど

トップに  
もどる

ver.4単語の表示

## まとめ

---

この作品は、8/3, 4日に東京ビッグサイトで行われた Maker Faire Tokyo 2019に出展してたくさんの人々に遊んでもらいました。小さい子が多く、英語の全くわからない子も楽しんでくれました。

その時は Ver.3 だったのですが、その時初めて小型化したキューブを会場の子供たちに触ってもらいました。今まで、大きくて持ちにくそうだったキューブがとても持ちやすくなっていました。

また画面の UI がまだ初期状態のままで、英語のかんたんモードもなかったので、英語がわからない子たちは適当に並び替えて遊んでくれていたのですが、やはり少しでもわかったほうが楽しめそうだったので、Ver.4 では単語のスペルを見ながら並べ替えができる簡単バージョンを追加しました。また、明るく楽しい感じのイラストをいれたデザインにしました。

今回、一番苦労したのは赤外線通信です。とにかくうまくいかなくて何度も失敗しての繰り返しでした。結果として、赤外線での細かい通信はテレビでもたまに失敗してしまう事でもわかるように、どうやっても完璧に通信させる事はできないと気づきました。

そこで今回は、うまく行かない赤外線通信でもこのゲームが成立するように、プログラムで受信ができない時とうまく受信した時を見分けるために、1問毎に色を変えるという工夫をしました。色が揃わない時は、上手く通信ができていない時なので、色が合うまで再読み込みをさせる事によって解消しました。

また今回のキューブの設計もとても大変でした。小さな箱の形の中に必要なパーツをすべて入れて、またパズル台に合う位置に赤外線送信 LED と受信器をつけて、電池を入れるためのスペースを開け、故障したときの事を考えてディスプレイなどを取り出せるように設計するのが大変でした。

そして、今までユニークな基板を使っていたところを、初めてプリント基板に挑戦して、設計して中国に発注した時は、戻ってくるまでドキドキしました。実は、1回失敗して2回目でなんとか成功しました。

とてもむずかしい挑戦で、途中で嫌になってしまった事もありましたが、いろんな工夫をしてなんとか完成了しました。これからもたくさんの人たちに楽しんでもらえる作品がつくりたいです。