

ユーザインタフェース レポート

氏名：吉谷 拓真

所属：工学部電子情報工学科

学籍番号：03-133007

URL： <http://blog.takuma7.com/2d-lego-builder/>

GitHub： <https://github.com/takuma7/2d-lego-builder>

1. ユーザインタフェースのデザインと実装

1.1 概要

本課題のシステムはPaper.jsを用いて実装を行った。利用環境としてはPCを想定し、ブラウザで実行するシステムである。

ブロックをドラッグ&ドロップで持っていくのはマウスを動かす距離の分だけ効率が落ちると考え、使いたいブロックの長さ分だけドラッグすることでブロックが出現するインタフェースを考案した。イメージとしてはドット絵方式に近いと考えられる。一度配置したブロックをドラッグ&ドロップで動かすことや右クリックで消すこともできるようにした。また全消去ができるクリアボタンも実装した。

特にデザインと操作性にこだわり、単なる長方形ではなくブロックのポッチも表現した。操作の際に音が発生するようにして本物のブロックで遊んでいるような感覚を演出した。ブロックの残数を分かりやすく表示するようにデザインした。

1.2 細かい使用方法と動作説明

1.2.1 画面の各部分の名称と役割

図1に本システムの画面の概要を示す。

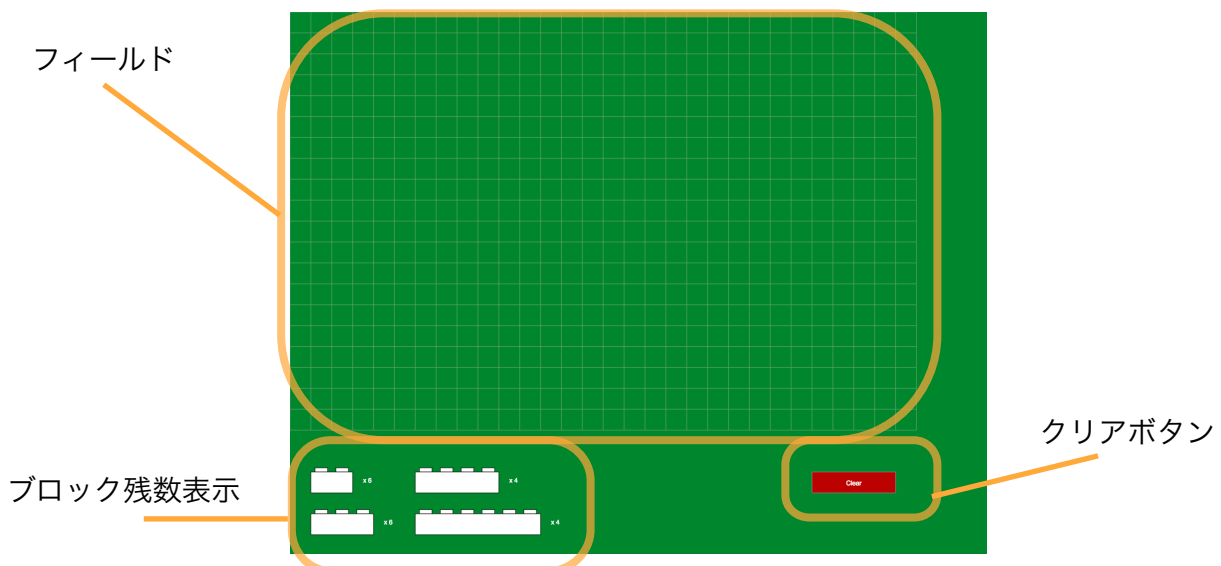


図1. 本システムの画面の概要

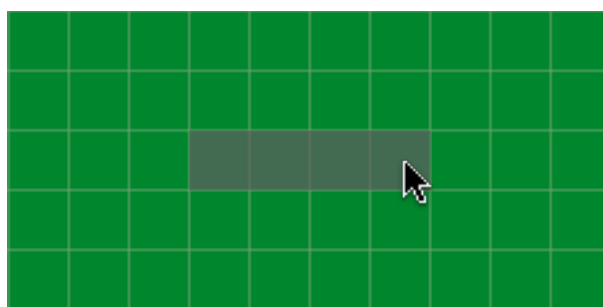
フィールド：フィールドはブロックを組み立てる場所である。

ブロック残数表示：どのブロックがあとどれだけ残っているかを表示する。

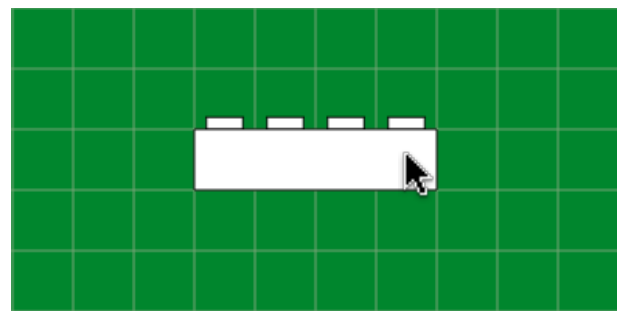
クリアボタン：フィールド上のブロックを全て消すボタン。

1.2.2 ブロックを作る

ブロックを作るにはフィールド上でドラッグを行う。フィールド上でドラッグを行うとドラッグを開始した点から現在のマウスポインタの点までの間に図2 (a)のように黒い影が出現する。この黒い影の長さは2, 3, 4, 6に限定されており、ドラッグ開始点から6マス以上離れた点にマウスポインタがあっても長さは6にとどまる。この黒い影が出ている状態でマウスのドラッグをやめると図2 (b)のように確定されブロックが出現する。



(a)



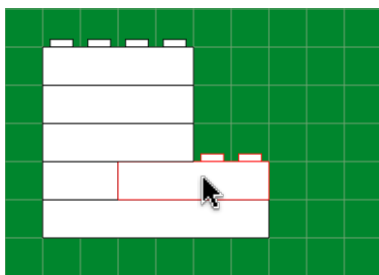
(b)

図2. フィールドでドラッグした状態(a) ドロップしブロックが生成された状態(b)

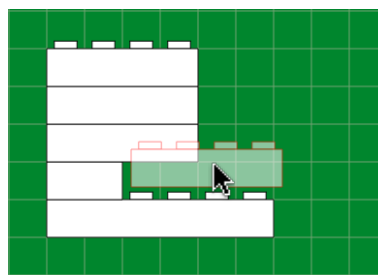
1.2.3 ブロックを動かす

ブロックの上にマウスカーソルを持っていくと図3 (a)のように枠線が赤くなる。この状態でドラッグをすると図3 (b)のようにブロックが半透明になり、自由に動かすことができる。所望の場所まで動かしドロップすると図3 (c)のようにもとの状態にもどる。

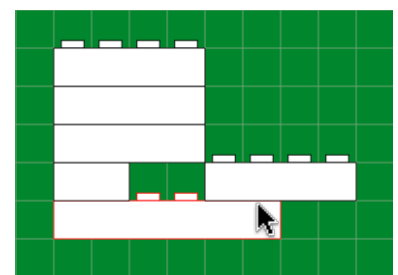
ブロック単体を動かすにはこのようにドラッグ&ドロップを用いるが、矢印キーを押すとブロック全体（逆に言えばフィールド）が矢印キーに合った方向に動く。



(a)



(b)



(c)

図3. ブロックを動かす. ブロック上にマウスを持って行き(a) ドラッグ(b) ドロップし確定(c)

1.2.4 ブロックを消す

生成したブロックを消すにはブロックの上にマウスカースルを持っていき右クリックをすれば良い。そうすると、選択されたブロックが消える。また全てのブロックを消したい際には画面右下のクリアボタンをクリックする。

1.2.5 ブロックの接続について

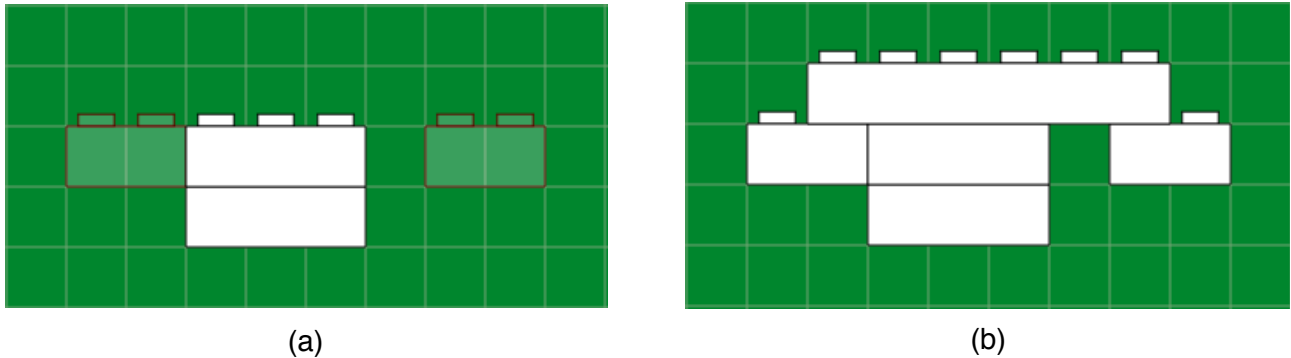


図4. 適切に接続されていないブロック(a) と適切に接続されたブロック(b)

ブロックは現実には可能な状態で接続されていなければならない。適切に接続されていないブロックは図4 (a)のように半透明で表示される。右側のブロックのように明らかに宙に浮いている場合はもちろん適切でないと判断されるが、左側のブロックのように接してはいるが現実には宙に浮いている場合も適切でないと判断される。これらを接続すると図4 (b)のようにすべて白く色が付く。

次に、どのグループを白くし、どのグループを半透明にするかについてどのように決めているかということについて述べる。ここでグループとはブロックが接続されたもので成る集合体のことである。白のグループを主グループ、半透明のグループを従グループと呼ぶことにする。アルゴリズムとしては、まず全てのグループを探し出しグループごとに違うラベル付けをする(図5)。次に、ラベルごとに何マス専有したかということを集計し、一番マス目の数が多かったグループを主グループとし白く描く。図5の例では'1'とラベル付けされたグループが最大であるので'1'のグループが主グループとして白く描かれている。'2'と'3'のグループは従グループということで半透明で描画を行っている。

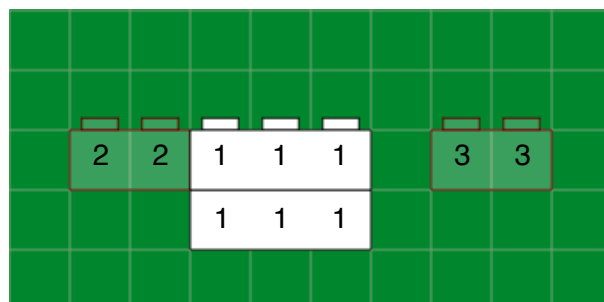


図5. グループごとにラベル付けされた様子

1.3 実装にかかった時間

本システムの実装には約30時間かかった。

2. ユーザテスト

2.1 被験者のプロフィール

被験者は2人用意した。以下ではそれぞれ匿名で被験者A, 被験者Bとすることにする。各被験者のプロフィールを表1に示す。

表 1. 被験者のプロフィール

	年齢	性別	所属	利き腕	機材
被験者A	20歳	女性	お茶の水女子大学 理学部情報科学科	右	MacBook Pro 15” Early 2013
被験者B	22歳	男性	東京大学 工学部電子情報工学科	右	MacBook Pro 15” Early 2013

2.2 観察結果

操作についての説明をしない状態でシステムを触ってもらった時の様子について述べる。

どちらの被験者も説明なしの状態ではどのようにブロックを配置するのかが分からず、左下のブロック残数のブロックをドラッグしフィールドにドロップするような行動が見られた。被験者Aについては説明するまでブロックの配置方法が分からなかった。被験者Bについてはフィールドをクリックしたりしているうちにブロックの配置方法を発見した。

どちらの被験者も説明なしの状態では矢印キーを使ってフィールドを動かすことができる機能を発見できなかった。説明後には当然使えるようになったが、課題をやっている間に使うことはなかった。課題が完成した後に作成物を動かして遊ぶという使い方が見られた。

2.3 被験者からのコメント

被験者からのコメントをそれぞれ以下にまとめる。

被験者A：

- ・楽しかった。
- ・音が出るのは実際に遊んでいる感じが出て良いと思った。
- ・最初ドラッグして（ブロックを）持っていくのかなと思った。
- ・3本指でドラッグするのが難しい。（開発者のMacでは3本指でドラッグできる設定になっており、ユーザテストの際にその方法を推奨した）
- ・ブロックの色を変えたい。
- ・なぜブロックの数に制限があるのか。もっとたくさんブロックが欲しい。

被験者B：

- ・使いやすいと思った。

- ・慣れると早く作れる。
- ・消える音などは消えた感じがして良いと思った。
- ・最初どのように操作すれば良いのか分からなかったので、チュートリアルのようなものがあると良いと思った。

2.4 実験結果

各被験者と開発者の各課題についてかかった時間（作業時間）を表2に示す。

表 2. 各課題完成までの作業時間

	練習用(説明前) [s]	練習用(説明後) [s]	課題1 [s]	課題2 [s]	課題3 [s]
被験者A	106	33	135	102	90
被験者B	59	18	43	46	47
開発者	-	18	21	26	27

2.5 ユーザの作成したモデルのスナップショット

それぞれの被験者に3つのモデルを作成してもらい、そのスナップショットを取った。図6-8に被験者Aによる作品、図9-11に被験者Bによる作品を示す。

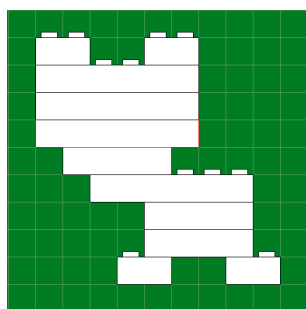


図6. 被験者A『犬』

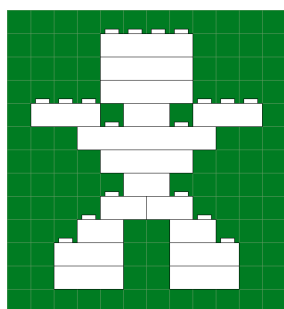


図7. 被験者A『松岡修造』

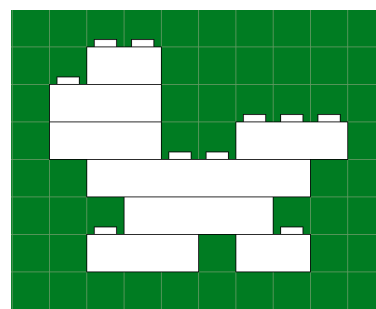


図8. 被験者A『ニワトリ』

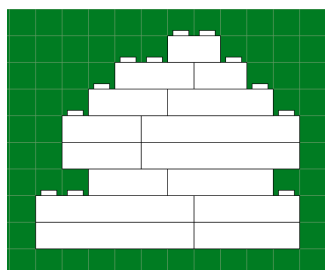


図9. 被験者B『ソフトクリームの上の部分』

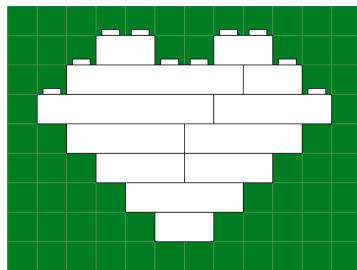


図10. 被験者B『ハート』

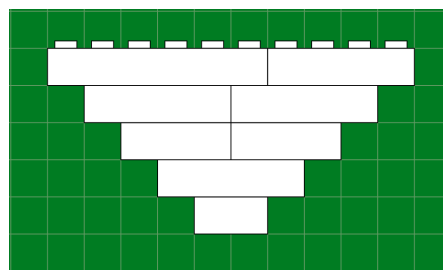


図11. 被験者B『パンツ』

3. まとめ

3.1 実験結果についての考察

まず、練習用課題については操作方法の説明によって作業時間が大幅に減少した。説明前は操作方法を模索するために時間がかかるのだと考えられる。

次に特筆すべき点は練習用課題から課題1に移った際に大幅に作業時間が増えるということである。課題1, 2, 3すべてにおいて作業時間にそれほど大差は無いが説明後の練習用課題と他の課題の作業時間には大きな差がある。これはユーザが課題を遂行する際に課題で示されたブロックの配置を観察するという作業に時間がかかるからであると考えられる。練習用課題は説明前に一度観察しているため、説明後の際にはブロックの配置についての観察が一旦は済んでいる状態である。そのため配置の把握にかかる時間が短縮され説明後の練習用課題だけどの被験者も短い時間で完成したのだと考えられる。開発者についてそのような特徴が見られないことは、開発者は開発を通して操作や位置把握について習熟してしまったためだと考えられる。

3.2 デザインについての考察

何も説明がない状態だと被験者が全員ブロック残数表示からフィールドにドラッグ&ドロップをしようとしたため、この点に関してチュートリアルを提供するなど何らかの方法でシステム側から説明をする仕組みが必要だと思った。もしくは、ブロック残数表示のデザインを工夫し、ここからドラッグ&ドロップするのではないと直感的に思えるデザインにするべきであった。

ブロックを生成する際、移動した際、消した際、できない操作をした際に音がなるようにしていたのだが、この点に関しては肯定的な評価が得られた。操作に合わせて操作に合った音がすることでユーザに操作をしたのだというフィードバックを与えよりシステムとのインタラクションがスムーズに行われたように思う。

ブロックをひとつひとつ生成していくスタイルのインタフェースの場合、ブロックの編集をどうするかというところがひとつのポイントとなると考えられる。例えばブロックのサイズを変えたいと思った場合ブロックを伸ばせるようなインタフェースにするとといったことも考えられる。今回実装したシステムではブロックを伸ばすことができる仕組みをあえて導入したが、それは今回のブロックがどれも比較的短く種類も少ないことからブロックのサイズを自由に変えたいと思うことは無いだろうという狙いからであった。実際に被験者に操作方法を説明せずに触らせてみたところ、ブロックのサイズを間違えた場合はブロックを伸ばしたり縮めたりしようとは一切せず、どのように消すことができるかという方法を探していた。ブロックを簡単に消せ、フィールドをドラッグしてブロックを生成、位置をドラッグ&ドロップで修正できるという本システムは被験者の直感に合っていたと言える。

好きなものを作ってもらおうというタスクを実行した後、ブロック数になぜ制限があるのかということや、他に色は無いのかといった疑問を抱いていた。この疑問は表現力をあげようと思った際に当然の疑問であると考えられるが、特に制限数に関してなぜ疑問が沸いたのかということデザイン面から考察してみることにする。本システムではフィールドを直接ドラッグ&ドロップするとブロックが生成されるというデザインにしたが、これは無からブロックを生成している感じがする。この場合、制限されたブロックの中から1つ持ってきたというような感覚は得にくくなっていると考えられる。つまり、たとえばブロック残数表示の部分からドラッグ&ドロップするというデザインにした場合であればそこにスタックがあり、そこから1つブロックを取ってドラッグする

のでそのスタック部分からはブロックが1つ減るという感覚が得られるが、無から湧いて出てきた場合はこの感覚を得難いのではないだろうか。リソースに制限のあるものをデザインする際にはこのようなことも考慮してデザインする必要があるのかもしれないという知見を得た。

3.3 まとめ

本課題ではLEGOブロックで二次元の作品を作成するためのユーザインタフェースを作成した。升目状のフィールドを用意し、その上でマウスでドラッグ&ドロップすることでブロックを直接生成するという方法を採用し、ブロックの編集機能やデザイン、音などのインタラクション面に特に注力して制作を行った。ユーザテストの結果、操作方法をはじめに教えないとユーザはどのようにブロックをフィールドに出すのかを理解できず困惑することが分かった。これにはデザイン上の工夫を行ったりチュートリアルを用意するといった対策が考えられる。操作方法を説明した後はユーザは快適に操作できたと述べ、特に苦勞する様子もなく課題を遂行することができた。練習用課題の説明前、説明後の作業時間変化、またその他の課題との時間の差から、今回の課題でユーザテストの課題として出されたものは一度目にやるのと二度目にやるのとでは作業時間に違いが出るのではないかという疑問が生じた。インタラクションデザインに関しては音を付け加えたことは良い評価を得ることができた。しかし、今回のデザインは課題の仕様であるリソース数制限への疑問が生じるようなデザインであることが分かり、リソース数制限を直感的に納得させるようなデザインとはどのようなものであるかと考えるきっかけとなった。