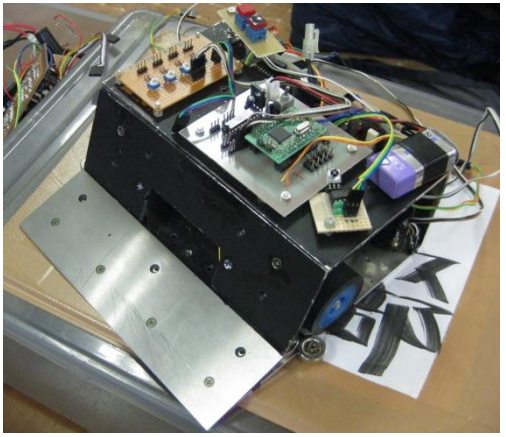
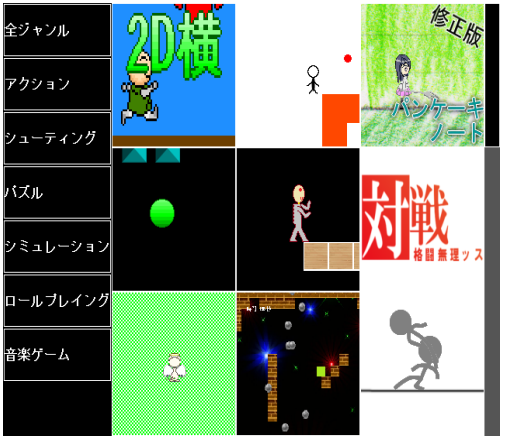
2010年度版

明治大学
エレクトロニクス研究部





機関誌

目次

P2 ： 幹事長挨拶
P3 ： ハード班
…P4 ： 相撲ロボットと全日本相撲大会について
…P5 ： 相撲ロボット『天先』
…P14 ： 個人製作物
P31 ： ソフト班
…P32 ： 一年生製作物
…P74 ： 二年生製作物
…P95 ： 三年生製作物
P108 ： 雑記

幹事長挨拶

本日は明治大学エレクトロニクス研究部の展示ブースにご来場いただきまして、誠にありがとうございます。

現在、我々エレクトロニクス研究部は 1,2 年生合わせて 35 名の部員がおり、全員がプログラミングによるソフトウェア製作を行うソフト班と、ロボット製作を行うハード班のいずれか、もしくは両方に所属する形で研究製作を行っております。

今回の学術研究発表会における展示には、ソフト班は各班員がプログラミングして製作したゲームの展示を、ハード班は今年 10 月 9 日に開催された全日本ロボット相撲大会の出場機体や、各班員の個人製作物等の展示を行っております。

各展示物の詳細におきましてはこの機関誌の後頁においてそれぞれの製作物の紹介をしております。また、展示物それぞれに製作者が説明員としておりますので、お気軽にお声をおかけください。

2010 年 12 月 19 日

明治大学エレクトロニクス研究部

平成二十二年度幹事長 塚本 耕司

ハード班について

私たち、明治大学エレクトロニクス研究部ハード班といたしましては、半田ごてを右手に電子工作を行ったり、ドライバーを右手にロボット製作を行ったりしています。その活動の一環として、毎年 10 月に行われる全日本ロボット相撲大会に出場するための相撲ロボットを製作しています。また、毎年 11 月に行われる生明祭や 12 月の学術研究発表会に向けて各個人がテーマを決め製作を行っています。

ハード班の活動は自作する事を主体としています。電子工作としては、秋葉原で基板や部品を購入してきて使います。基板はユニバーサル基板やプリント基板を使います。プリント基板を使う場合は、EAGLE というソフトウェアを使って配線図を作り、それを基板にエッチングして使います。ロボットのボディの加工についても、基本的にアルミ板の切断や穴開けなどの加工は自分たちで行っています(一部業者に発注したものもあります)。

また、製作に必要な知識と技術を後輩に伝えるために、毎年ハードゼミを行います。ここでは上級生から新入生に対して電子工作についての基礎知識や基礎技術を教えます。ゼミが終わった後も普段の製作の中で少しずつ IC や各種センサの応用的な事や相撲ロボットについても教えていき、知識の向上を図っています。

本年度も昨年度と同様、相撲ロボットを製作し、全日本ロボット相撲大会関東大会へ出場し、ベスト 16 進出を果たしました。

平成二十二年度ハード班長　中村 太郎

相撲ロボットと全日本相撲大会について

全日本ロボット相撲大会とは

全日本相撲ロボット大会とは、富士ソフト株式会社と社団法人全国工業高等学校長協会が主催しているロボット大会です。この大会は、個人またはグループで製作した相撲ロボットを土俵の上で戦わせて勝敗を競うものです。

相撲ロボットとは

相撲ロボットにはラジコン型と自立型があります。ラジコン型はリモコンでロボットの動きを制御するものです。また、自立型は一度スタートスイッチを押したあとはプログラムで動きを制御するものを指します。故に、自立型のロボットには周りの状況を調べるために赤外線センサや超音波センサなどのセンサ類がついています。

～ルール～

- 直径 154cm の円形鉄土俵で戦う。
- ロボットの大きさは縦 20cm×横 20cm 重量 3kg までとする。
- ロボットの高さに関しては制限なし。
- スタートスイッチを押してから 5 秒後に発進するようにする。
- 相手を場外へ押し出すと一本。二本先取で勝ち進むことができる。
- 出場選手は安全のためにグローブ、ゴーグルを着用する。
- 暴走防止のための緊急停止回路を取り付ける。
- 敵に向かっていかない場合戦意喪失とみなされ、一本とられる。

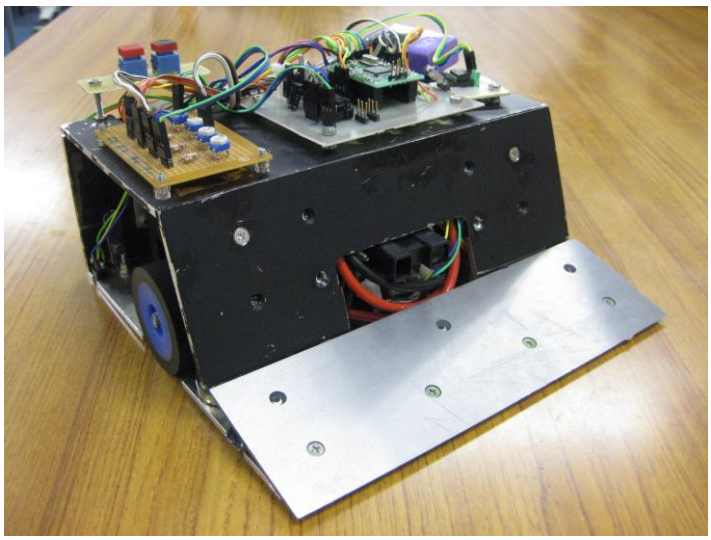
参考

全日本相撲ロボット大会（富士ソフト ABC） http://www.fsi.co.jp/sumo/

相撲ロボット『天先』について

はじめに

今年の相撲ロボットは昨年の『タック』を参考にして製作しました。今年は新入部員が多く入ってくれたおかげで部費が潤沢にあったため、例年より多くの資金を相撲ロボットに投入する事が出来ました。去年度までは部室にあるモータを使いまわしていましたが、今年度は新しいモータを新規購入しました。詳しくは以下のそれぞれの項目をご覧ください。



外装

相撲ロボットの外装の材料にはジュラルミン材(A2017)を使用しました。ジュラルミンとはアルミニウム合金の一種で強度が高いという特徴を持っています。昨年の相撲ロボットの外装には純アルミニウムを使用していました。純アルミニウムと比べジュラルミンは比重が2.7から2.8と少々重くなっていますが高い切削性を維持したまま強度の向上を計ることができるためジュラルミンを採用しました。加工には主にNCフライス盤を使用しました。

今回はブレードの設置角度を鋭角に取るためその分のスペースに他のパーツを配置することができず設計がとても難しかったです。その結果モータドライブ回路は内部に納めることができたもののメインボード回路等を機体上部に設置

せざるを得なくなり、見た目も安全性もよろしくないものになってしまいました。また設計のやり直しや、部品の作り直しをする時間が無く大会にもその状態で迎えてしまったためこれを改善するために機体フレームの見直しや部品の更なる小型化が必要と思われます。

モータ

モータはMaxon Motor社製DCコアレスモータのRE35というシリーズのものを使用しています。

以下が今年使用したモータの概要です。

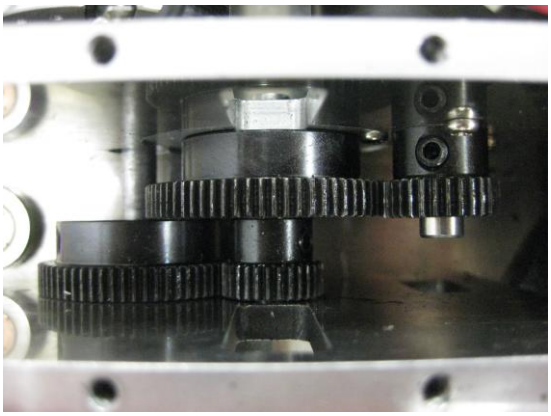
Maxon Motor RE35の仕様

型番	消費電力 (W)	公称電圧 (V)	定格回転数 (rpm)	定格トルク (mN・m)	停動トルク (mN・m)
273752	90	15.0	6270	73.2	874

今年はギアヘッドによる減速は行わず歯車のみによって減速させています。ギアヘッドが無くなった為モータ自体の長さは短くなりましたがそれでもモータ2本を直列に配置するのは難しかったため前後並列に逆向きに配置し、ギアトレインにより左右のタイヤへ伝導するようにしました。

＊停動トルク：モータが出しうる最大のトルクです。これ以上の負荷が掛かるとモータは回転しません。

ギアトレイン



今回はスピード型の相撲ロボットにしようということになったためギアトレインの設計に入る前に速度計算をしました。

土俵の直径が 154cm であるので往復で 1 秒、秒速 770mm/s となるような減速比で歯車を検討しました。今回の相撲ロボットではモータの下部にバッテリを二つ格納するための高さを確保しなければならずモータからタイヤまでの軸の位置の設計が大変でした。またモータのギアヘッドを使わなくなったため減速を全て歯車により行わなければならないのですが 1 段階の減速では充分な減速が得られず、また歯車の直径も大きくなってしまうため 2 段階での減速をすることにしました。1 段階目では歯数 30→76 の減速を行い 2 段階目で 30→66 の減速を行いました。

最終減速比は(76÷30)×(66÷30)≒5.57 です。

走行速度は 6270÷5.57÷60×45≒844mm/s となりました。

走行速度は想定していた速度より速いものとなってしまいましたがこれは理論値であり実際は様々な抵抗や負荷によりこれより遅くなるものと思われたためこの速度設計にしました。

歯車の仕様

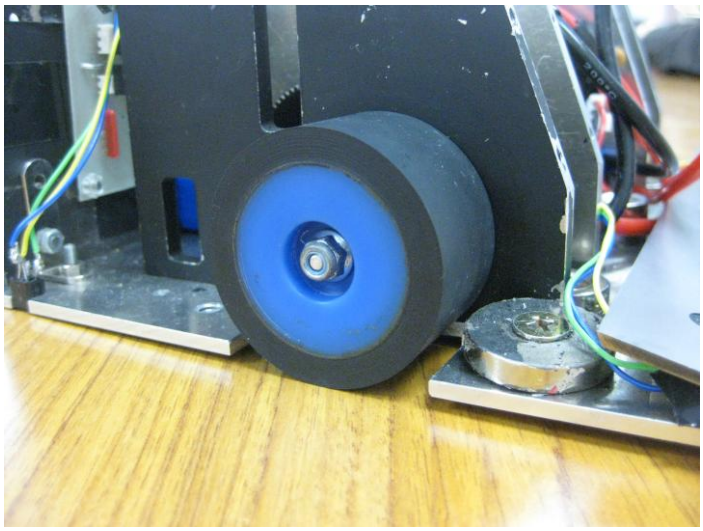
メーカー	小原歯車工業		
型番	SS0.5-30C	SS0.5-66A	SS0.5-76A
モジュール	0.5		
材質	S45C		
基準円直径	15mm	33mm	38mm
歯数	30	66	76

タイヤ・ホイール

タイヤの材質はニトリルゴムという合成ゴムで硬度は 70 度の硬めのタイヤを使用しています。硬度は低くなるほど材質が柔らかくなりグリップ力が増し、硬度が高くなるほど材質が硬くなりグリップ力が低下しスリップしやすくなります。

土俵上の急発進、急停車、急旋回を繰り返す相撲ロボットにはグリップ力の高い低硬度なタイヤが良いのですが試合中、相手ロボットとの押し合いの膠着状態になってしまった場合タイヤがロックされモータに大きな負荷が掛かりモータが焼けてしまう場合があるためそのような状況であっても力を逃がすことのできる高硬度のタイヤを採用しました。

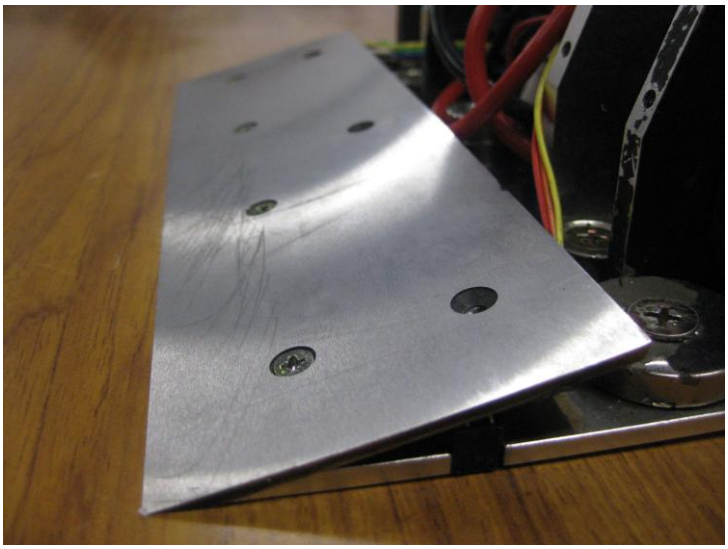
また今年はできるだけ相撲ロボットの車高を下げることを考えていたので低硬度のタイヤを使用していた場合重量によりタイヤが潰れ、車高が変わってしまい設計が難しくなってしまうということもありニトリルゴムタイヤを採用するに至りました。



タイヤ・ホイールはロボット環境整備工房製のものを使用しています。

ブレード

ブレードはハイス鋼という材質のものを使用しました。ハイス鋼とは高速度鋼(high speed steel)のことで一般に金属加工等の高速切削にも耐えうる材質としてドリルやエンドミル等の刃物に使用されているものです。



相撲ロボットは一般的にロボットの底面に強力な磁石を配置し、その磁石を土俵である鉄板に吸着させ相手から押し出されないようにしています。その吸着力は数 10kg～100kg 以上にもなりモータの力だけで押し出すのはとても難しいものとなっています。そのため、押し出すのではなく相手をすくい上げるためにブレードを相手の下に潜り込ませ磁石ごと引き剥がして押し出すという作戦が採られています。そのために先端が鋭利になっている刃物をブレードとして使うのが主流となっています。

相撲ロボット用のブレードには主にハイス鋼やフェザー、バネ鋼といったものが使われています。フェザーとはカミソリ等に使われている薄い刃物で、バネ鋼とは文字通りバネのような高い弾力性を持った鋼板のことです。

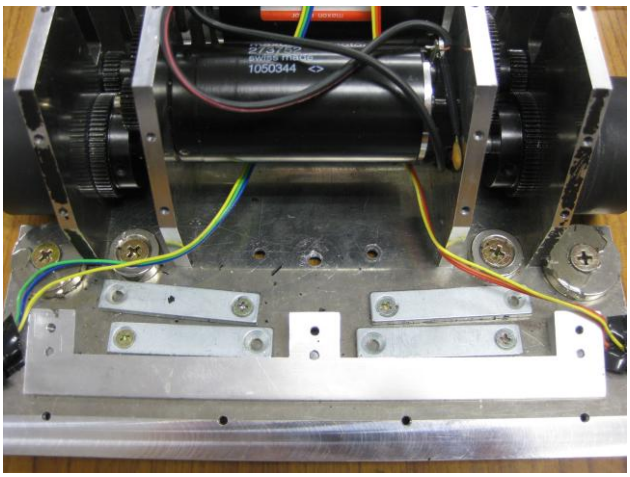
フェザーとバネ鋼の二つは軽量ですが薄い材料を使っているので高速でぶつかり合う相撲ロボットでは破損してしまう危険性がありました。

ロボット相撲大会のルールでは試合中にブレードが破損し飛散した場合にはいかなる状況であっても失格となってしまうため今回はこの二つは避け、強度に優れるハイス鋼を使用しました。

使用したハイス鋼は福田刃物工業株式会社に依頼し、製作して頂きました。

磁石

磁石はネオジウム磁石というとても強力な磁石を使用しています。磁石の配置はブレードがある前面に集中させブレードが浮かないようにし、またタイヤの前後にも配置することによってタイヤのグリップ力を高めるようにしました。しかし今回の設計では底板であるジュラルミン板の上部に磁石を設置したため底板と鉄板の間に板厚 3mm 分の間隙ができてしまいその分磁力が弱まってしまいました。結果、ギアトレインの減速比を低くした高速型の相撲ロボットの制動や相手からの突進を受け止めることができなくなってしまいました。



ネオジウム磁石の効力を最大限に発揮するために鉄板と磁石との距離をより近くすることが今後の課題だと思われます。

ネオジウム磁石は株式会社二六製作所製のものを使用しています。

バッテリ

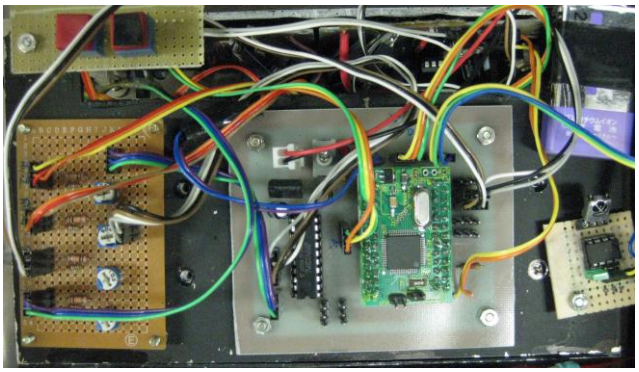
使用したバッテリはHyperion社のLiPo電池です。仕様を以下にまとめました。

Hyperion CX G3 Power Pack

パーツ番号	表示電圧 (V)	セル数 (個)	公称容量 (mAh)	放電率 (C)
LG325-2500-2S	7.4	2	2500	25/45

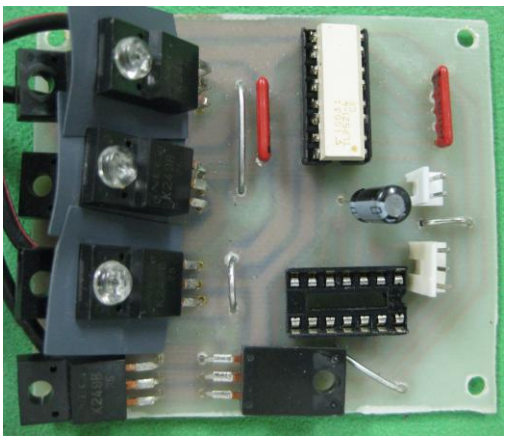
例年はバッテリが炎上するのを恐れてLiPo電池を使用しませんでしたが、今年は勇気を出してLiPo電池に手を出してみました。昨年は19.2V、一昨年は24Vのバッテリを使用していましたが、今回の相撲ロボットで使用するモータの公称電圧が15Vであることを受け、14.8Vにしました。バッテリをロボットの底に配置するに当たって、4セルでは高さが高すぎたため、上記のバッテリを2つ直列に繋いで使用しました。

メインボード



相撲ロボットの脳にあたる部分です。H8/3664F というマイコン（CPU）を一つ搭載し、センサやスイッチの状態を読み取ってモータドライブ回路に信号を送ります。電源は 006P 型 9V 充電池を使用し、三端子レギュレータ(TA7805S)を介して 5V の電圧をメインボードに送っています。またマイコンの周りにはノイズを取るコンデンサ、動作確認用の LED、スイッチ類や論理 IC（74HC645AP）が乗っています。白線検知センサ周りは別の回路として横に配置し、得られた信号をまとめてメインボード側の論理 IC に送っています。この論理 IC はセンサから送られてくるアナログ信号を高いものと低いものに分け、安定した信号をマイコンに送る役割があります。

モータドライブ回路



メインボードからの信号を受け取り、モータへと電流を流す役割を果たすのがこのモータドライブ回路です。相撲ロボットのモータを動かすには大電圧、大電流を流す必要があるため、モータドライブ IC ではなく自作のモータドライブ回路を作りました。私たちのモータドライブ回路では、回路一枚につき一つのモータを動かすので二枚用意しました。使用したパーツは論理 IC、フォトカプラ、FET、抵抗、コンデンサです。

モータドライブ回路は H ブリッジ回路の形をとっています。メインボードからそれぞれのモータドライブ回路につき三本の信号線を出し、その組み合わせによってモータを動かしています。

メインボードからの信号はまず論理 IC に入ります。今回用いた論理 IC は昨年と同じ 74HC125 です。この IC を使用する事によって H ブリッジ回路がショートしないようになっています。

IC から出た信号はフォトカプラに行きます。フォトカプラは昨年と同じ TLP521-4 を使用しています。メインボード側とモータ側では使う電圧が違うため、このフォトカプラを使って回路を分けています。

H ブリッジ回路のスイッチの役割をするのが FET です。フォトカプラから FET のゲート端子に電圧をかけることによって、ドレイン端子からソース端子に電流が流れるようになります。これを利用して on/off を切り替えています。FET は昨年と同じ 2SK2498 を使いました。昨年までは一つの FET で一つのスイッチを担当していましたが、モータの起動電流が FET の定格ギリギリだったので二つに増やしました。

今年は本体の小型化に伴い、モータドライブ回路にも小型化が求められたため、各種素子の配置や配線にも気を配りました。その際、昨年まで整流用に入っていたダイオードを外しました。これは本来 FET の保護のために入っていたものですが、逆電流が流れるという状況が考えづらく、また何度か試しても FET が壊れることはなかったため問題ないと判断しました。

また毎年モータ側の回路の銅箔が焼け切れるのですが、流す電流からあるべき銅箔の太さを計算すると回路があまりに巨大になってしまうので、半田を盛ることで対処しました。

白線検知センサ

白線検知センサにはローム社の RPR-220 を使用しています。赤外線の反射率を利用して土俵端の白い枠を読み取り、相撲ロボットが土俵から落ちないようにするための重要なセンサです。反射してきた赤外線を受光すると回路に電流が流れ、それによって抵抗に生じた電圧降下を信号としてメインボードに送っています。

敵検知センサ

今回、敵検知センサとして使用したのは SHARP 社製の GP2Y0D02YK0F です。デジタル信号を出力する測距センサです。検知範囲内の物体を検知したら一定の電圧が出ます。

緊急停止スイッチ

今回使用したものはPIC12F675とTL-IRM0101です。相撲ロボットを遠隔操作で止めるためのものです。リモコンからの赤外線を受信し、PICマイコンが決められた信号（High→Lowの繰り返し）をマイコンに送ることによってマイコンが相撲ロボットを停止させます。

相撲ロボットは大きく重くそして速いので素手で止めるのは危険です。そのため、大会規定により必ずつけなければなりません。

相撲ロボットのプログラム

相撲ロボットは土俵から落ちないようにしたり、敵を発見して動くためにマイコンを使っています。PCでプログラムを作り、それを書き込むことでマイコンはプログラムの通りに動きます相撲ロボットのプログラムはC言語で書かれていて、コンパイラにはベストテクノロジー社のGCC Developer Liteを使用し、h8writeで書き込みました。相撲ロボットのプログラムは基本的に、スイッチが押されてからの5秒間の待機、PWMの設定、モータへの命令、センサ状態の読み取りから成り立ちます。緊急停止センサからの信号が入ったら即座にモータを停止します。マイコン機能は割り込みとタイマーを使い、緊急停止には割り込みを使います。また5秒間の待機やPWMにはタイマーを使いました。

DanceDanceRevolution 専用コントローラ

理工学部 2 年電気電子生命学科
赤井 優也

はじめに

KONAMI の音楽ゲーム「DanceDanceRevolution(以下 DDR)」及び、それの派生作品である、「stepmania」、「In The Groove」、「PUMP IT UP(以下 Pump)」をゲーセン以外でも楽しもうということで、製作に至りました。家庭用のマットが既にありますが、よりアーケードに近い形を目指しました。

動作原理及び構造

1.ボタンの構造

一般に販売されているゲームコントローラは、指でスイッチを押下し、それによりマイクロスイッチやカーボン板などで基盤を導通させ、電流の返ってくるポートからそのボタンの ON/OFF を判別しているものがほとんどです。今回実現しなければいけないボタンは「足で踏んで電気が通るボタン」なので、以下のような構造を考えました。

厚さ 2㎜の MFD にアルミホイルを貼り付け、それを 10㎜のベニヤ板に貼り付け、隅に小さく切った厚さ約 5㎜のクッションを取り付けたものです。MDF にアルミホイルを貼り付けることにより、擬似的な導通可能な金属板を作り、それをクッションが隔てて向かいあわせることで、力が加わらないときは互いは離れ、力が加わるとクッションが潰れ、接触及び導通し、また、力が加わらなく

なると、そのクッションの塑性によりまた離れます。こうすることにより、踏んでいる間は電気が通り、踏んでいないときは電気が通らないスイッチを作りました。これを実現するだけであれば、市販のマイクロスイッチを用いれば済むのですが、よりアーケード筐体に近づけるため、踏んだときのクリック感があるマイクロスイッチよりこちらを採用しました。

2.使用した基盤とゲーム機器/PC との接続機構について

The King of Fighters XII 3in1 USB Pro Stick (White)というコントローラを分解して、中にある IC のついてる基盤だけ使用し、自作した足で踏んで反応させるボタンに接続しました。詳細は上記の家庭用アーケードスティック Wiki 及び、公式 HP 参照のこと。このコントローラは口が PS2/USB と 2 つあり、PS2/PS3/PC で使えます。

3.基盤への接続

今回製作したコントローラは、上下左右を使う DDR と、斜めと真ん中を使う Pump とを両方対応させ、かつ二人プレイを実現する為、3×3×2=18 個のボタンを用意し、先述の基盤を 2 つ用意しました。

用意したボタンを基盤に接続するためにプリント基板を使用し、コントローラの基盤の各ポートに接続しました。

接続したボタンの対応は以下の通りです。

L1 △ R2		7 1 6
□ St ○	数値でいうと	4 9 2
R1 × L2		8 3 5

上下及び左右を同時に押した場合、それぞれの方向のどちらでもなく「ニュートラル」として信号が返ってしまうので、それを回避すべく、あらゆる同時押しでも反応する R/L ボタンを採用しました。

反省・考察

ほとんどの箇所、特に拘っていたパネルの踏み具合は、概ね満足のいく出来に仕上げることができました。それぞれの部品の耐久性も十分で、11 月 19～21 日に行われた生明祭では、パネルはキズ一つつきませんでした。

ですが、大きな欠陥が二つありました。一つは、擬似的な金属板を作るために用いたアルミホイルが、踏んだ状態で摩擦されることにより剥がれてしまい、誤反応が出てしまうことです。もう一つは、クッションの塑性が思ったよりも早くなくなってしまい、数曲プレイしただけでクッションが戻らなくなり、反応し続けてしまうことです。

この二つに関しては、ボルトとバネによる改善策を考えています。

最後に

2010 年の 4 月頃に突発的に思いついたネタでですが、思った以上にマトモにプレイできるものに仕上げられて嬉しく思います、というか自分で驚いてます。

寸法は実機をメジャーで計測しましたが、用意したアクリル板との寸法があわなくて苦労しました。また、左右同時押し問題や、個々のパネルの固定法にも苦労しました。

思い立った 4 月から作ってきたこの作品は、自分にとって今年度の象徴ともいえるものです。このような機会を与えてくれたエレクトロニクス研究部、付き合ってくれた親や友人、そして、私の大好きな DDR 全てにありがとうを言いたいと思います。

ここまでページ数を無視して長々と語ってきましたが、DDR、もとい「音ゲー」に対し、より多くの人が興味を持っていただけると嬉しい限りです。

Others

SPECIAL THANKS

ぺーたん、Y･けてぃ･D、たぬ、灼熱-Beach Side Bunny-

PANEL DESIGN

ごまちゃん/SESAME

COLLABORATION

NAOKI/芦沢 直樹

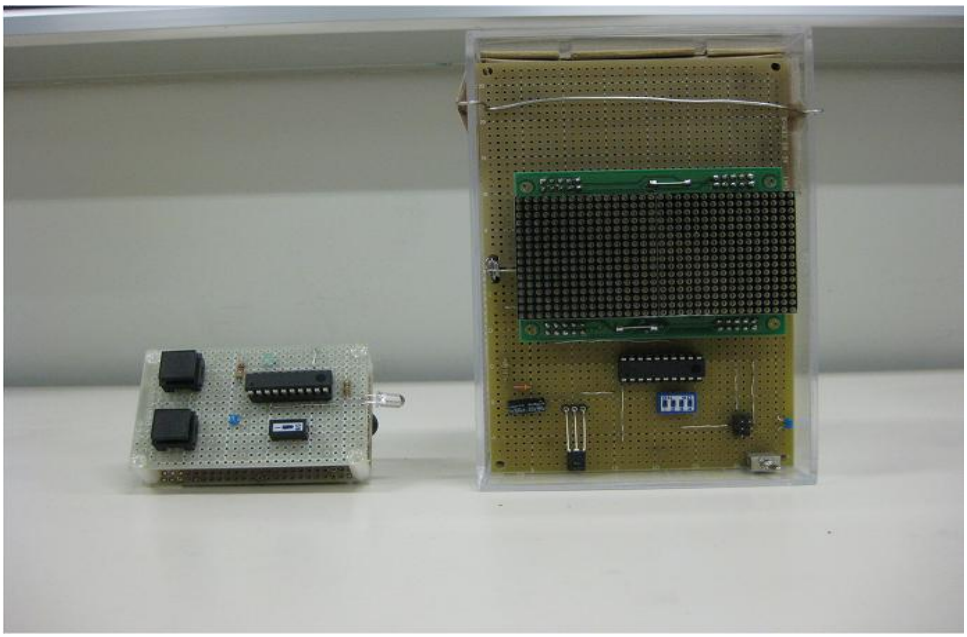
ソフト班製作ページ「Step R」へ。氏のゲームもこれで動きます。

アンケート集計装置

理工学部 1 年電気電子生命学科
坂梨 弘幸

はじめに

これは、おもにAVRマイコンやドットマトリックスLEDを使って作ったアンケート集計装置です。これは、電子工作マガジン SUMMER2010 に掲載されていた回路・プログラムを用いて製作しました。



操作方法

まず、受信機・送信機のスイッチを入れます。すると、受信機側に YES NO の文字と 00 のカウントが表示されます。送信機の YES NO のボタンを押すと、カウントされます。ID を確認しているので、二重に投票はできません。

原理

ドットマトリックス LED はモジュール内で縦 1 列の LED のアノード、横 1 列の

カソードがつながっているため、同時にすべての LED を点灯できません。このモジュールでは一度に横1列のなかの必要な LED を光らせ、次の横列を光らせています。

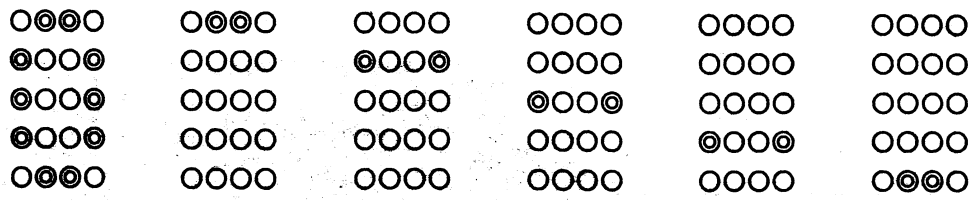


図　ドットマトリックス LED の表示方法

さいごに

参考文献の回路でそのまま作ると、リボン線だらけになってしまうので、なるべくリボン線を使わないように配置を工夫しました。本当は送信機をもう少し作ろうかと思いましたが、時間があまりなくて作れませんでした。また、周波数が遅いためか、受信機がちかちかしてしまったので、次回作る機会があったら、そのような点を改善したいと思いました。

参考文献・資料

「電子工作マガジン SUMMER2010」　電波新聞社

「32×16 ドット LED　マトリックス表示装置パーツセット　解説書」
秋月電子通商

あみだくじ生成機

理工学部 1 年機械工学科
小林 真
理工学部 1 年電気電子生命学科
宮崎 泰行

はじめに

あみだくじ生成機とは、その名の通り縦 4 列のあみだくじを自動で生成してくれる機械です。地味に便利かなと思い、製作しました。

二枚の基盤を加工して繋いでおり、外装は簡単に取り外しが可能です。

操作説明

右側についている主電源を入れると、あみだくじの縦線に当たる LED が点灯します。左側のスイッチを押せば横線が点滅し、離すと横線の点滅が止まり、あみだくじが出来上がります。

動作原理

このあみだくじ生成機は、74HC4060AP という IC を使用して横線を点滅させています。この IC は発振回路を内蔵したバイナリカウンタです。バイナリカウン

タというのは、2 進法に従って 1 または 0 でカウントアップしていくもので、例えば、10 進法で 0、1、2、3、4 とカウントアップしていくところを、4 桁の 2 進法であれば 0000、0001、0010、0011、0100 というふうにカウントアップしていきます。

74HC4060AP では 2 進数で 14 桁までカウントできますが、そのうちの 10 桁のみが出力されます。この出力された値にトランジスタを通し LED を光らせるようにしています。どういうことかというと、横線の一つ一つがそれぞれの桁に対応しており、対応する桁が 1 なら点く、0 なら点かないとなっています。

このあみだくじ生成機は 19 の横線を使用しているため、74HC4060AP を 2 つ使用しています。10 桁の 2 進数の出力をあみだくじの横線として表示させるようにしたので、横線が全くでない場合もあります。

問題点

まず、このあみだくじ生成機は長い間電源をつけ続けることができません。

あみだくじ生成機は LED を大量に使うので多くの電力を必要とし、電池を電源に使うとすぐ消費してしまいます。そこで家庭用電源から AC-DC コンバータを使って直流電源にしており、コンバータの出力電圧（9V）が高いので、電圧を下げるために 5V 出力の三端子レギュレータを使用しています。問題はこの三端子レギュレータで、3 個を並列に繋ぎヒートシンクをつけていますが、消費する電力が膨大すぎて、熱くなってしまうのです。それが危険なため、長い間つけ続けることができず、使用した後は主電源を切る必要があります。

また 74HC4060AP は一瞬で 2 進数を生成しているわけではなく、高速でカウントアップしているだけ（最大値になると 0 に戻る）なので、桁数の小さい部分は高速で点滅し、あたかも常時点灯しているように見えますし、桁数の大きい部分は点滅が遅く、点滅していないようにも見えてしまいます。

参考文献

「実践！LED 電子工作マスターブック」伊藤尚未 著　誠文堂新光社

elebeat コントローラー

理工学部１年機械工学科
川上 魁

目的

ソフト班で作成した elebeat を操作するため、そしてこれを遊んだ人が原作 jubeat に興味を持ってもらえるように作成しました。

使用方法

15 インチ 1024*768 の解像度を持つ液晶の上にこのコントローラを設置するとちょうどよくゲームとして動作します。

使用部品

アクリル板
タクトスイッチ
ゲームパッド
その他配線

回路

回路自体はシンプルなもので、各アクリルパネルの下にスイッチが 4 個設置されています。

そのスイッチが押されたとき、配線を通じて配線先に接続したゲームパッドに信号を伝えその伝えられた信号を PC が認識します。

反省点・今後の改善点

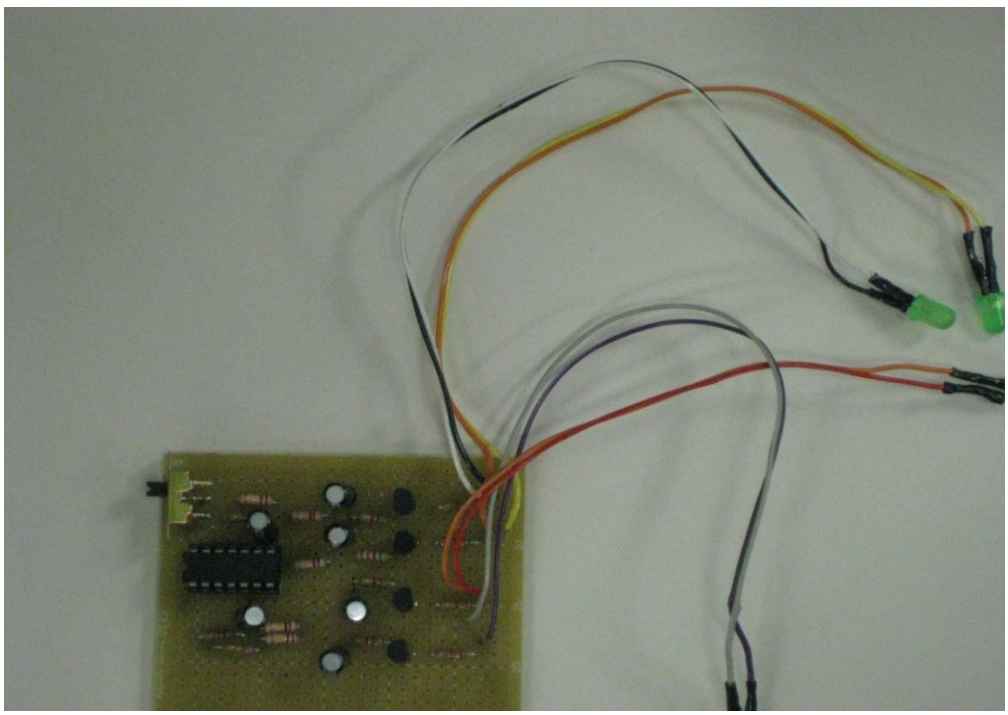
配線が目隠しされておらず汚かったという点が今回最大の反省点です。学発までには汚くなっていた配線をアルミチャンネルで目隠ししして本家「jubeat」に近いものを完成させたいです。

アンホタル

理工学部１年機械工学科
中里 祐斗

はじめに

アンホタルとは、74HC04AP という IC を使い、コンデンサと抵抗器とトランジスタを組み合わせて LED をゆっくり点滅させる回路です。LED を使った電子工作を紹介している本の中で興味を持ったため作成しました。



原理

IC からの出力が 1 だった場合、コンデンサに電気が少しずつ溜められ、溜まるに従ってトランジスタのベースに電流が流れるので、LED が少しずつ明るくなっていきます。IC からの出力が 0 になるときは反対にコンデンサに溜まった電気が少しずつ減るので、減るに従って暗くなっていきます。

工夫点

LED とビニルコードを半田付けして接続する際に、熱収縮チューブを使用して絶縁したことです。これによって他の LED と接触したとしてもショートしないようにしました。

課題

IC の調子によって LED が点滅しなくなります。また、外装がぱっとしないのでホタルらしく人工樹を周りに設置したり、ホタルの模型と合わせるなどして、改善していきたいと思います。

参考文献

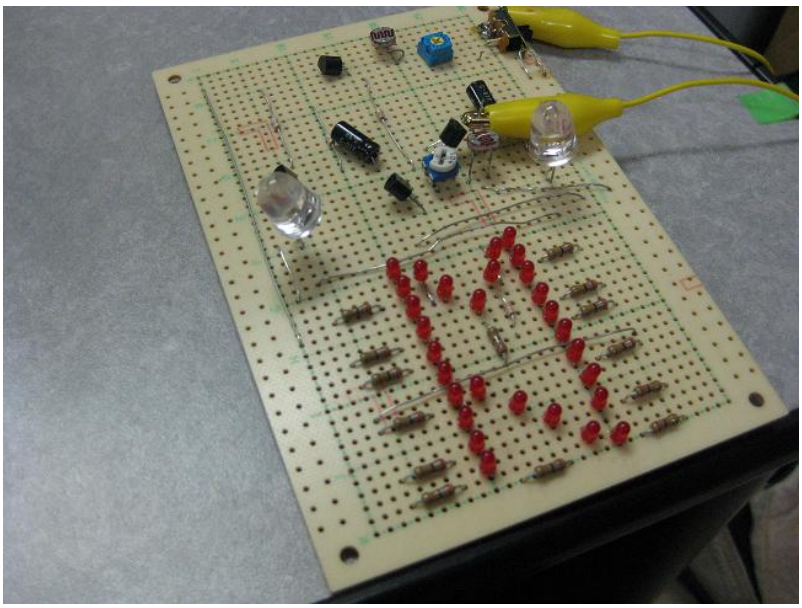
「実践！LED 電子工作マスターブック」伊藤尚未　著　　誠文堂新光社

光感知 LED 回路

理工学部 1 年機械工学科
渡辺 宏光

初めに

この回路は暗くなると自動で点灯し、明るくなると消える LED の基本的な回路の一つです。街灯や広告灯が人の操作なしに点いたり消えたりするしくみに興味があったので作ってみました。



動作

スイッチを入れると部屋が明るいときは「M」字型に配置した LED が点灯します。逆に、暗いと「N」字型に配置した LED と２つの大きめの LED が点滅します。

センサ

部屋の光に反応するように CdS とよばれる受光センサを使用してます。しかし、このままでは個々の部屋での明るさが違ったりする場合にうまく動作しないため 100kΩの半固定抵抗器を使って感度の調整ができるようにしました。

回路

明るいときは半固定抵抗器を通り、CdSを通りトランジスタのベースに電気が流れないため「M」字型のLEDが点灯しますが、暗くなるとCdSの抵抗が大きくなりNPN型トランジスタ2SC2120のベースに電気が流れ込みこのコレクターエミット間をスイッチングさせ「N」字型LEDと大きい２つのLEDが点灯します。また、この部分はPNP型トランジスタ2SA1015を2個とコンデンサー2個を使った非安定マルチバイブレータ回路になっておりLEDを交互点滅させています。

最後に

「M」字型と「N」字型で暗い時と明るい時とで切り替わるようにしたのは自分なりにオリジナリティを出したかったからなのですが、「M」字型の方で1箇所だけ不備がありました。「M」字型はLEDを2個と抵抗１つを５セットとLED1個と抵抗1個とで組み合わせて作ったもので、そのLED1個のところだけが常時点灯している状態となってしまいました。確認してみると抵抗の量が数Ωほど足らなかったようなのですが、それを調節する部品がなく、また入れるスペースもありませんでした。次からはこういったミスを防げるよう部品の繋げ方などに気をつけていきたいと思います。

参考文献

「LED工作テクニック」　誠文堂新光社

反省

今年は昨年の活動状況を踏まえ、「製作の仕方がわからない」「進行状況がわからない」といった状況を解決するため、製作状況などの情報の共有、技術・知識の共有を目指しました。

しかし実際に製作が佳境に入ると、自分の作業に集中してしまい、他の班員を慮る余裕がなくなってしまいました。そのせいで誰が何をやっているのかわからないという昨年と同じ状況に陥り、また一年生への相撲ロボットの技術・知識の引き継ぎも遅く曖昧になってしまいました。

また毎年の反省として、製作を開始するのが遅いせいで完成するのがギリギリになってしまいテスト走行が充分に出来ないというものがありました。

これらの問題を解決するため学術研究発表会後も活動を継続し、技術・知識、今年の反省点や気付いた事などを後輩に伝えていきたいと考えています。

ハード班員

理工学部２年電気電子生命学科　中村太郎
理工学部２年電気電子生命学科　赤井優也
理工学部２年電気電子生命学科　岩田純一
理工学部２年機械情報工学科　須田和樹
理工学部２年機械情報工学科　田村淳一
理工学部２年応用化学科　小沼宏彰
理工学部１年電気電子生命学科　坂梨弘幸
理工学部１年電気電子生命学科　宮崎泰行
理工学部１年機械工学科　川上魁
理工学部１年機械工学科　小林真
理工学部１年機械工学科　中里祐斗
理工学部１年機械工学科　渡辺宏光

協力

緊急停止回路の製作
八木澤圭

ソフト班について

ソフト班では、プログラミング言語を用いてゲームなどのソフトウェアを製作しています。C、C++を中心に自由に言語を選択して製作を進めています。

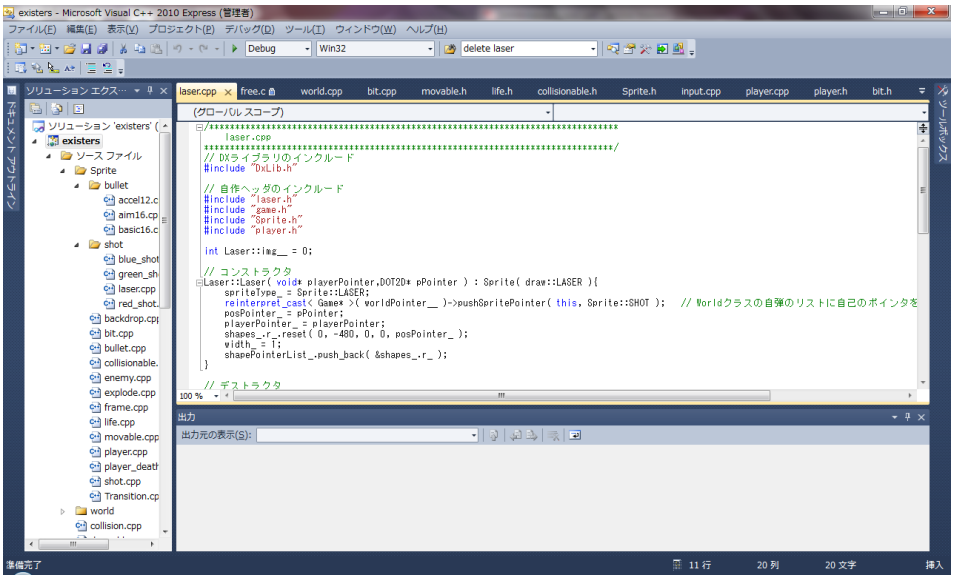
ハード班と同様、新入部員を対象としたソフトゼミを行っています。ソフトゼミはゼミAとゼミBに分かれていて、ゼミAではC言語の基本的な勉強を、ゼミBではゲーム製作の勉強を行っています。

DirectXやOpenGL等を使いこなすのは難しいのでゼミBでは簡単なライブラリを利用してのプログラミングを学び、アルゴリズム部分の習得を目的としています。2010年度のゼミBではDXライブラリを利用しました。

これらの他、定期的に集まってプログラミングをしたり、情報交換をしつつ、夏休み中には発展編となるゼミCで勉強をしたり、外部の勉強会に参加するなどの活動を行っています。

今年度は1年生の新入部員が例年と比較して非常に多く、様々なゲームが揃いました。楽しんでいただければ幸いです。

平成二十二年度ソフト班長　坂井　誠也



StepR -仮 Edition-

理工学部 1 年電気電子生命学科
芦沢 直樹

はじめに

某音ゲーを作りました。こういった音楽ゲームの譜面を作ったことがあったので、エレ研に入ってプログラムから作ろうと思って製作しました。
仮判定が出来てからは実際に楽しく遊びながら作っています。∩(ˆ ω ˆ*)∩ﾜｰ

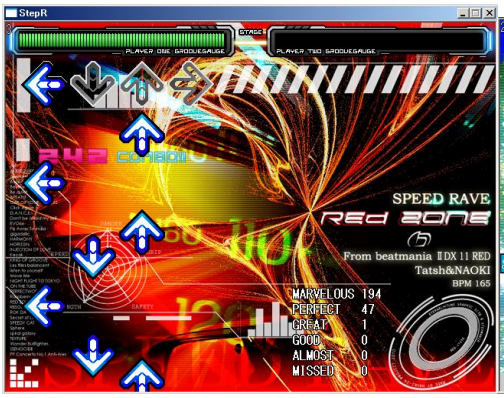
開発環境

- プログラミング
  Microsoft Visual C++ 2010
  DX ライブラリ
- 画像作成とか
  Adobe Photoshop 7.0
- 音源編集とか
  SoundEngine
- 先輩方

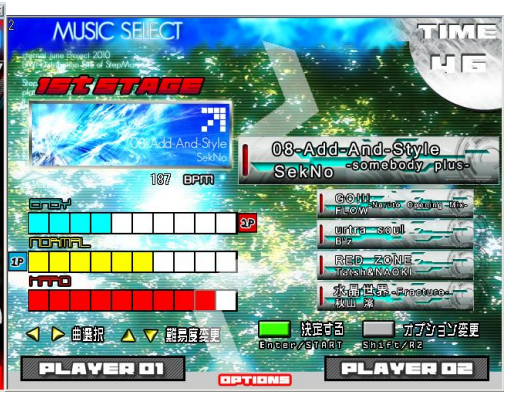
僕自身プログラミングの経験が無かったのですが、
先輩方の指導のおかげでゲームができました。
ありがとうございました。

ゲームの説明

矢印が判定エリアに重なるﾀｲﾐﾝｸﾞでﾊﾟｧｰﾉｳ(ボタン)を踏みます。
その時のタイミングが判定され、判定によりライフゲージが増減します。
曲終了までに少しでもライフゲージが残っていたらクリアです。
ジョイパッド設定を ON にすることで、専用コントローラーを使用することができます。



↑ゲーム中画面



↑曲選択画面

操作方法

各画面ごとに使用するキーが異なります。

各キーの説明

	ゲーム中	曲選択中
←→	矢印を踏む（押す）	曲選択
↑↓		難易度変更
Enter	なし	決定
Shift	なし	オプション変更

工夫した点

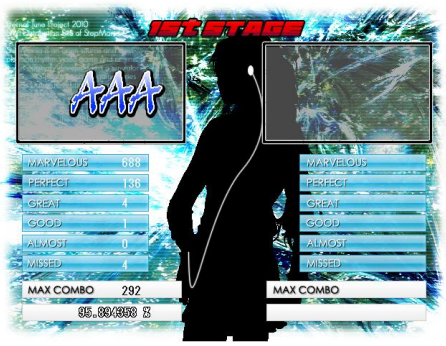
なるべく本家に近いものを作ろうという気持ちがありました。

譜面の間隔を綺麗にするために等速時の１拍分のピクセル数はノートの大きさで決めました。そこから BPM とオフセット値を変数として各曲共通で譜面をスクロールさせる関数を作りました。オフセットとは基準となるある点からの相対的な位置のことであり、StepR では曲の再生時を基準としてオフセット合わせを行なっています。

あと、GOOD 以下の判定でノートの色が黒くなるようにしました。本家には無い描画です。ゲーム中残っているノートが分かりやすくなるので気に入っています。

反省点

某音楽ゲームの譜面形式である.sm ファイルを読み込ませたかったのですが、どのように読み込ませるかやり方が分からず独自の譜面形式をとりました。その結果、テキストデータを手で打ちこむ作業に時間が掛かってしまいました。また、スキンやシステムが未完成での公開になってしまったことが悔まれます。来年に完成版を作るかまだ分かりませんが、今回は仮-Edition-として提出しています。

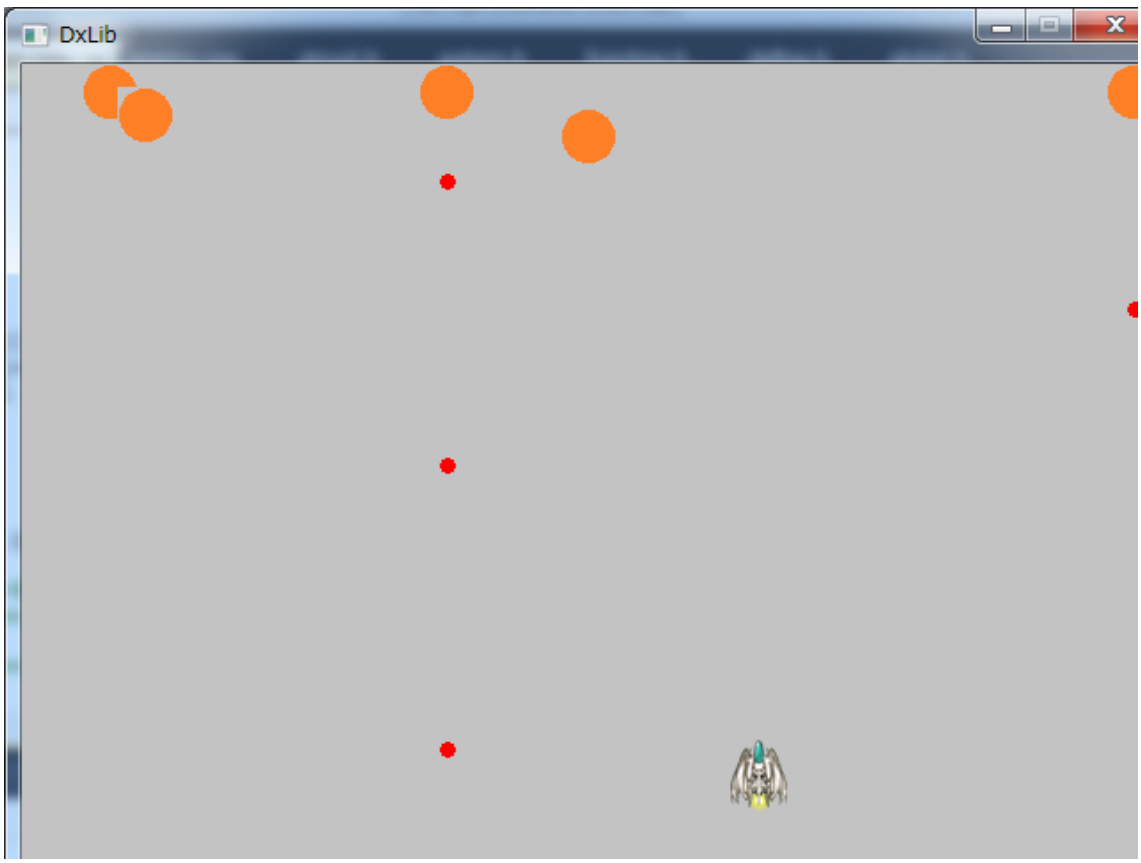


THE・シューティング

理工学部１年応用化学科
市野　恭弘

はじめに

エレクトロニクス研究部に入部して、初めてＣ言語を先輩方に学びそのことを活かして、シューティングゲームのプログラミングを始め、ゲーム製作を行ってきました。



開発環境

・C言語

・DXライブラリ

・Microsoft Visual Studio 2008

ゲーム操作・内容

１、　操作方法

Ｚキー　　自弾発射

方向キー　自機移動

２、　内容

基本的には普通のシューティングです。弾を撃って丸い敵を倒して下さい。一定時間生き残ればクリアになります。逆に、敵とぶつかったり、敵が発射した弾に一回でもあたるとゲームオーバーになります。

工夫した点

さしあたって、工夫できた所はありませんが、強いて挙げるとしたら敵の動きをいくつかのパターンをつくったところです。

反省点・感想

まず、ゲーム製作を始めるのが遅すぎたことと、Ｃ言語を理解しきってなかった事です。

敵の画像を透過させることもできませんでした。本当なら、最後にボスをつくったり、自機が発射する弾にも、いくつか弾幕を作りたかったです。あと、違う動きをするのに敵の画像が一種類だったのがちょっと残念でした。

今後の課題

今回できなかったことを次までにできるようにして、より面白いものを作っていきたいです。

棒人間

理工学部 1 年機械情報工学科
太田雄貴

はじめに

明治大学エレクトロニクス研究部に入り、初めて c 言語に触れて、ゼミで先輩の方々にゲームの作り方を教えていただいてこのゲームを作成しました。

開発環境

Microsoft Visual C++ 2010 Express Edition
DX ライブラリ

ゲーム説明

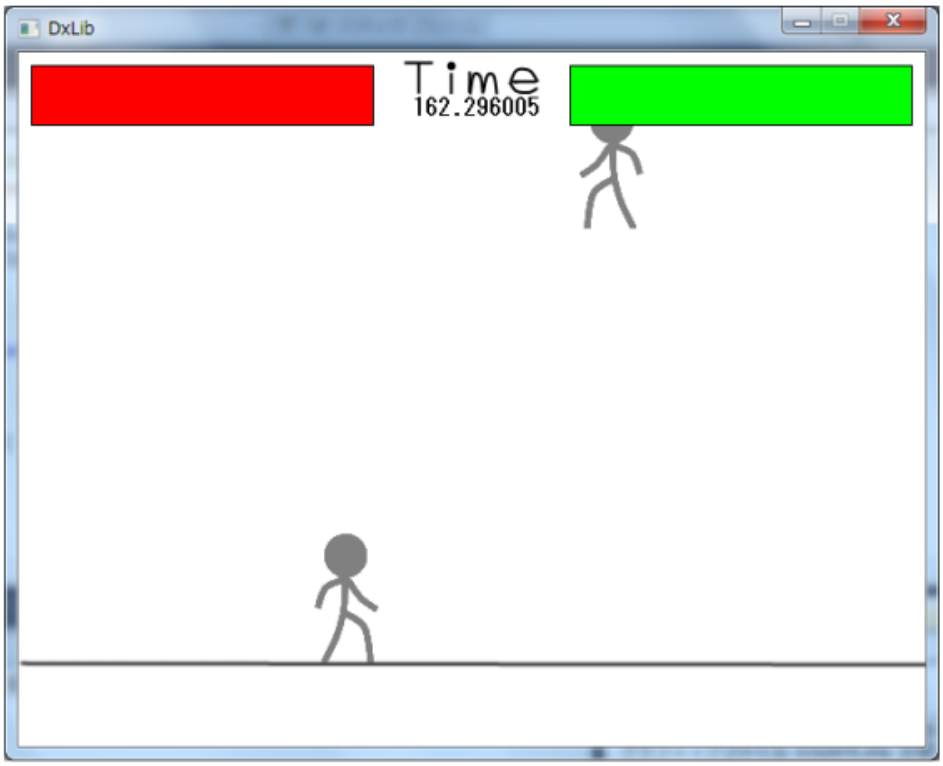
プレイヤー１，２を動かし敵の HP をなくしたら勝ちという、格闘ゲー風なゲームです。

操作方法

１Ｐ
←→　：移動
↑　　：ジャンプ
Ｌ　　：パンチ
Ｏ　　：蹴り
↑＋Ｐ：滑り込み
２Ｐ
ＡＤ　：移動
Ｗ　　：ジャンプ
Ｈ　　：蹴り
Ｂ　　：パンチ
Ｚ＋Ｇ：滑り込み

工夫した点

なるべく動きをスムーズにしたかったので一つの行動に４枚の画像が連続して入れ替わるようにしました。



反省点

まず、攻撃を受けたときの反応が滑り込み以外ではつけていないので、攻撃があたってるかわからなくなってしまっている点。２つ目には、攻撃があたっていてもいなくても攻撃が続行できてしまってただ体力の削りあいになってしまうことです。

ほかにも、ダッシュが片方しかできないなど中途半端になったしまったところが多かったです。やはり原因としては、夏休みにあまり時間をとらずに後期になってから本格的にはじめたのが原因だと思います。来年からはもっと時間をかけて作ってより完成度の高いものを作りたいと思っています。

elebeat

理工学部 1 年機械工学科
川上 魁

はじめに

ゲームセンターなどで有名(?)な「jubeat」という音楽ゲームを作ってみたかったのとハードとソフトが融合した作品を目指したらこうなりました。

開発環境

Microsoft Visual Studio 2008
DX ライブラリ
Azpainter

ゲーム説明

音楽に合わせて画面の光ったところをタッチするゲームです。

操作方法

ハード班で作成した「elebeat コントローラー」を使って操作します。
特に決まった操作方法はありません。

反省点

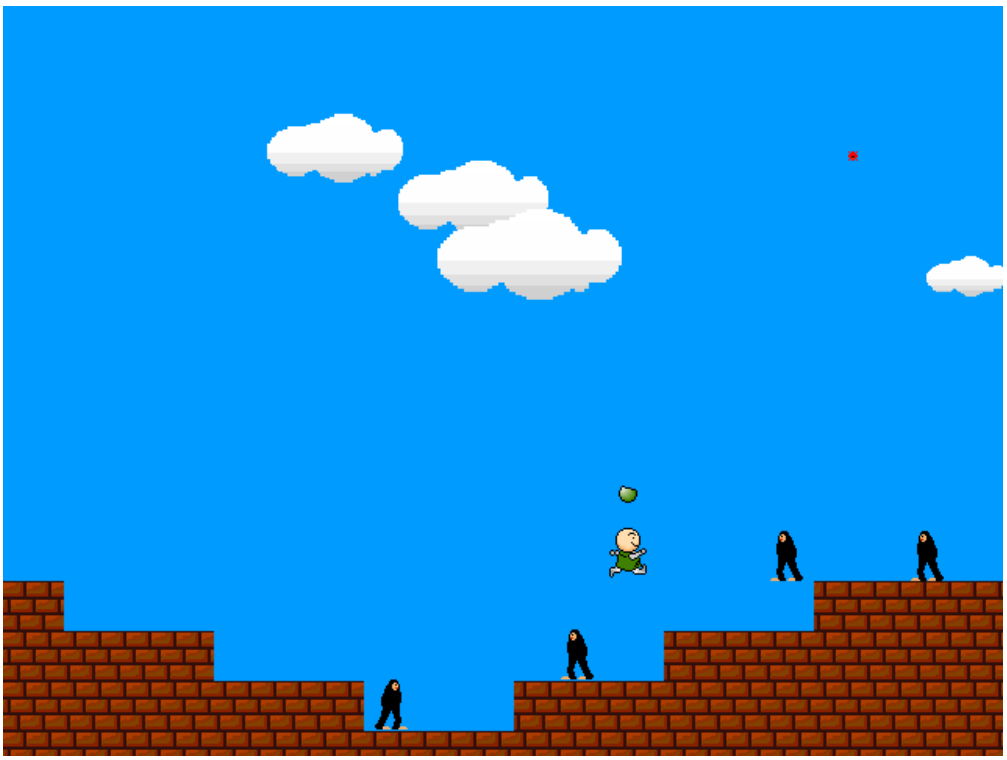
原作を元として、ゲームとしてあってはならない点を反省点とするとまず一つ目に楽曲の少なさが上げられる。
生明祭時点での楽曲総数はたったの一曲であり、さらにその難易度も経験者にはぬるすぎる、非経験者には戸惑ってしまう判定設置であった点がまず一つの反省点である。
二つ目にスコアの非設置である。
本来音楽ゲームとはスコアを競うのが主であり、それはゲーム内で唯一データとして残る本人の勲章である。
そのようなゲームでの達成感を感じさせる物を設置しなかったのは反省点である。
三つ目は画像表示の不安定さである。
これはいまだ原因が不明であり次の学発までには修正したい点であるのだが、問題の例として上げられる例は「判定において用意していた物と違う物が出現する」という例である。
配列に入れているものがこの様な挙動をするのでそこを修正したい。
他の反省点もあるのだがほとんどハードの反省点となるのでハードのほうで反省していきたい。

2D 横

理工学部 1 年機械工学科
小林 真

はじめに

今回私は、C 言語を使って 2D 横スクロールアクションゲームを作りました。



開発環境

Microsoft Visual C++ 2008
C 言語
Dx ライブラリ
edge

ゲーム説明

敵を踏み潰しつつ、右へ進む横スクロールアクションゲームです。ある地点まで到達すれば、ゴールです。プレイヤーのライフは、頭の上にある丸い小物体で表現しています。敵に横か下から接触しダメージを受けると、丸い小物体が欠けていきます。小物体が無くなっているときにダメージを受けると、昇天してゲームオーバーになります。また、ステージの最中に所々ある穴に落ちてしまうと、即ゲームオーバーです。

敵は、

① 転がって迫ってくるボール
② 歩いて迫ってくる何か
③ 常に跳ね続けるボールのようなもの
④ 跳ねながらプレイヤーを追ってくる法の執行者

の 4 種類となっています。動きが違うので、①や②の敵は弱いですが、④の敵はかなり強いようになっています。

ステージは一つだけですが、はじめの方は簡単で、最後の方は敵の速さを大きくしたり、敵の配置を工夫したりして、難しくなるようにしました。

主人公の名前は A 太郎です。服の A という文字が特徴のいい子です。

操作説明

移動　　：　十字キー
ジャンプ　：　Z キー
ダッシュ　：　X キー

工夫した点

長い時間遊んでくれた人のために、太陽を用意しました。また、クリアタイムのハイスコアのセーブ機能を実装し、タイムアタックが出来るようにしました。地味にうざいエンドロールもつけました。

反省点

自分の技術不足と時間的制約で、アイテムや壊せるブロック、オプションなどの実装したかった機能を実装出来ず、ステージを 1 つしか作れませんでした。また製作する機会があれば、それらの機能を実装したものを作りたいです。

対戦しよっ！！

理工学部 1 年機械情報工学科
今野 裕太

はじめに

エレクトロニクス研究部に入って学んできたことをいかして新しい知識を得ながらゲームをつくりました。

開発環境

C 言語

Microsoft Visual C++ 2010

DX ライブラリ

ゲームの説明

このゲームは、ふたりのプレイヤーが魔法を撃って相手を倒し、二回の勝利を勝ち取ることを目指します。格闘がない格闘ゲームです。

操作方法

プレイヤー１（スタート時画面左のキャラ）

ジャンプ：W

右：D

左：A

弱攻撃：S

中攻撃：Z

ガード：Q

プレイヤー２（スタート時画面右のキャラ）

ジャンプ：I

右：L

左：J

弱攻撃：K

中攻撃：M

ガード：O

工夫した点

弱攻撃が自機とＹ軸が同じ座標で動くようにしたことです。これを導入したことで戦術や作戦の幅を増やすことができました。

反省点

プレイヤーが二人必要なこと戦闘が単調すぎるなど反省点はいろいろとあります。

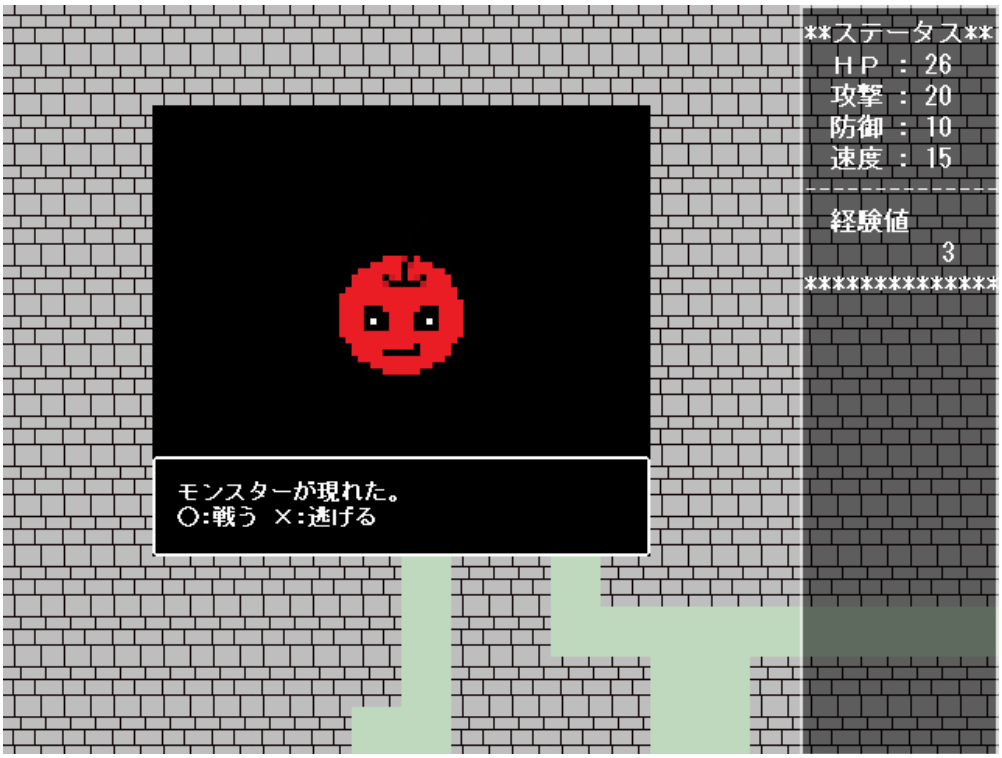
来年はこの反省点を改善したゲームを作っていきたいです。

RPG という名の迷路

理工学部 1 年電気電子生命学科
坂梨　弘幸

はじめに

明治大学エレクトロニクス研究部に入って初めて C 言語を学びました。ソフトゼミを通して、ある程度ゲームを作る知識が身についたのでこのようなゲームを作りました。



開発環境

C 言語

Microsoft Visual Stadio 2010

DX ライブラリ

ゲーム説明

・十字キー　　：　移動

・○ボタン（N キー）　：　戦う（決定キー）

・×ボタン（M キー）　：　逃げる

これは RPG のように見えますが、実は迷路のゴールを目指すだけの簡単なゲームです。ゲームを始めると主人公は家の中にいるので外に出て、洞窟（迷路）に向かってください。外や迷路に出ると敵が出てくるので、倒したり逃げたりして HP が 0 にならないようにしてください。

反省点

本当はもっと RPG のようなゲームを製作したかったのですが、夏休みに思ったより作業が進まなかったことや自分のプログラミングの技術不足などにより、このような簡単なゲームになってしまいました。また、敵も 1 体しか出ないことや、アイテムがないこともありゲームとしての完成度が低いので、来年は早い時期から開発をして、今年のゲームよりも面白いゲームを作りたいと思います。

ブロック崩し

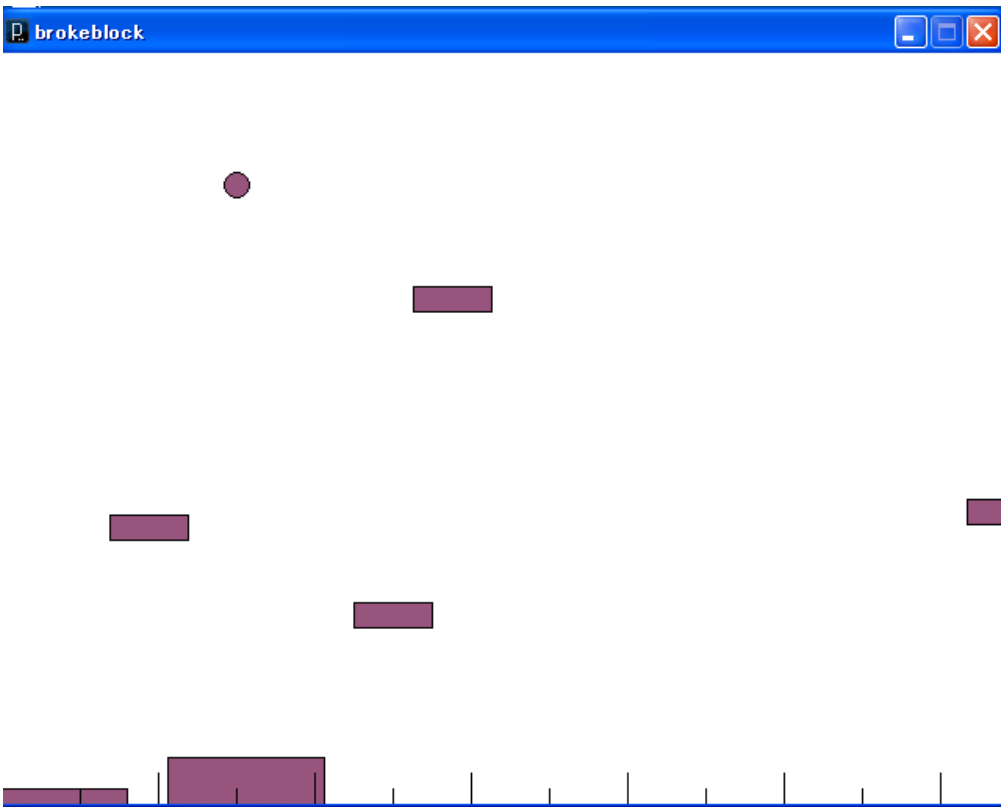
理工学部 1 年数学科
武内　誠

はじめに

　エレ研に入部したものの、ゲーム制作に自分のレベルが間に合わず友人のアドバイスにより簡単なものを選びました

開発環境

　prosessing



操作方法

　左右キー　移動

ゲーム説明

　飛んでる球をはじきブロックを崩しスコアを稼ぐ

工夫点

　ブロックを崩すと残りのブロックの色が変わる

反省点&改良点

　今回は形式が異なるので次回作はC言語を使いたい
ブロック崩しというシンプルな物に仕上がったので次回作は
もう少し思考を凝らしたい
(パズルゲームは変えない方向で)

イライラとダメ人間と宝探し

理工学部１年応用化学科
西川　太千

はじめに

ただひたすら歩き当たりの宝箱を見つける。途中のハズレの宝箱のコメントにイラッとするゲーム（？）です。自分でもどうしてこんなコメントになったのか分かりません

開発環境

ＶｉｓｕａｌＳｔｕｄｉｏ２０１０
ＤＸライブラリ

操作方法

十字キーのみ：移動のため

ゲーム概要

スタートしてからひたすら歩くだけの何もないゲームです。途中で飽きることはよくあることなので、多めに見てあげてください。

おそらく３個くらい当たりを引くとゲーム終了になります。ゲームオーバーわないのでゆっくりダラダラとプレイしていただきクスリとでも笑ってくれれば幸いです。

反省点

自分のＣ言語についての勉強不足により満足いくようなデキにはなりませんでした。もう少しミニゲームやボタン機能などを付けてみたかったのですが、アイディア不足などもたたってなんとも中途半端なかんじが否めない結果となってしまいました。

スペース・シューター

理工学部１年情報科学科
広田 達哉

はじめに

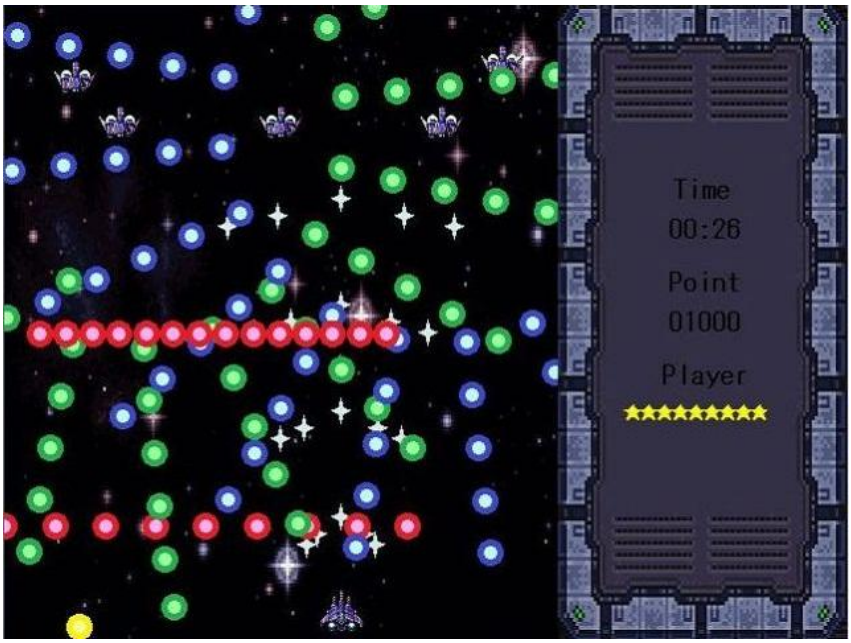
大学に入学するまでプログラムに関して無知だった私は今回のゲーム製作で、シンプルなシューティングゲームを作りました。

開発環境

C 言語
DX ライブラリ
Microsoft Visual Studio 2008

ゲームの説明

このゲームは縦型シューティングゲームです。自機が破壊される、もしくはステージの最後まで到達するとゲームが終わります。ゲームが終了するとスコアーが表示されるので、その数値を競います。スコアーは生存時間、敵機撃墜によるポイント、自機の耐久度（画面右の星の数）から計算されます。



操作説明

方向キー：移動

Ｚ：ショット１

Ｘ：ショット２

反省点

夏休み中にあまりゲーム製作が進められず、時間に余裕を持たせることが出来ませんでした。その結果、アイテム・ボスキャラなどの作成が出来ず、シンプルなシューティングゲームとなってしまいました。しかし、今回自分でゲームを作成出来たというのは、大きな経験になりました。

Simple Shoot

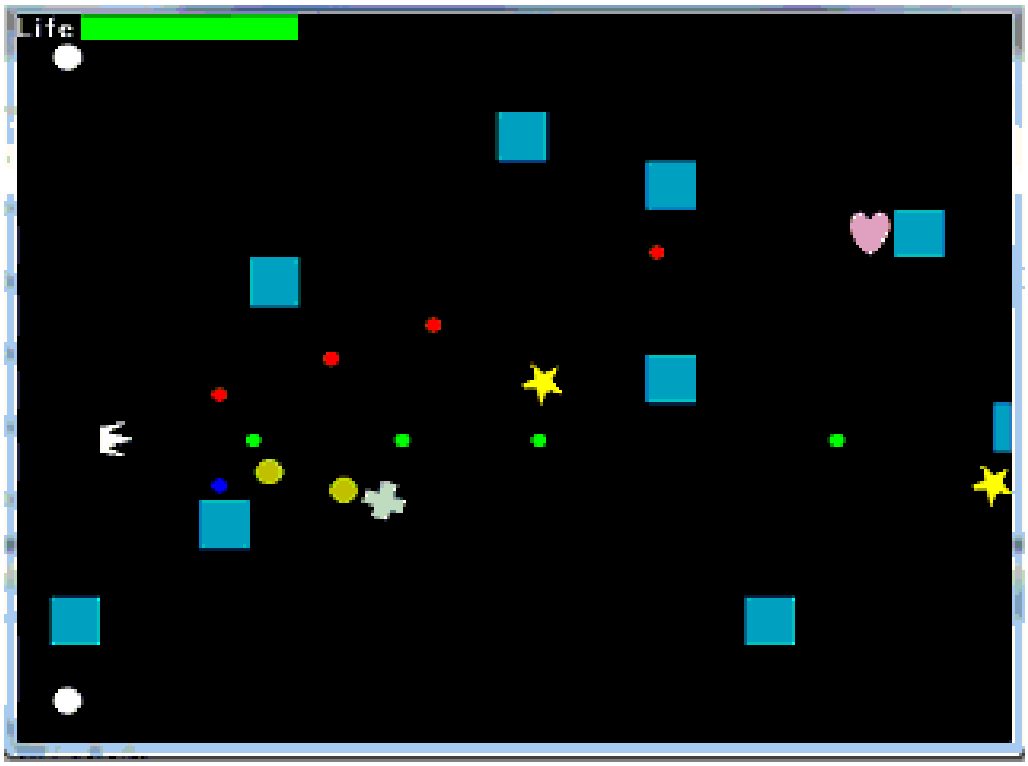
理工学部 1 年電気電子生命学科
藤尾 岳人

はじめに

明治大学エレクトロニクス研究部に入部し、先輩方が教えてくださったプログラミング言語を用いて、このゲームを製作しました。

開発環境

C 言語
DX ライブラリ
Microsoft Visual C++ 2010



操作説明

十字キー ： 移動

Z キー ： 弾を撃つ

ESC キー ： ゲーム終了

ゲーム説明

画面の右から出てくる敵を倒しながら進む横スクロールシューティングゲームです。途中にある壁をうまく使って敵弾を避け、アイテムを取りながら，奥へと進んで行きましょう。敵や敵弾に当たると体力が減り、壁と画面左端に挟まれると体力が０になります。ステージ最後にいるボスを倒すことでゲームクリアとなります。

工夫した点

一つは敵の動きを数種類用意し、敵弾も動き方の違うものを複数作って、避けるのを難しくするように配置しました。また敵の配置に合わせてマップを作り、敵の攻撃を壁で回避しながら相手を攻撃する、というスタンスのゲームに出来たと思います。

もう一つはボスにバリアを装備させて倒しにくくし、数種類の攻撃をランダムに出させるようにして、単調な攻撃や動きにならないよう工夫しました。

反省点

あまり計画的にゲーム製作を行っておらず、そもそもゲーム経験があまり無いのもあって、難易度設定や何を実装するかという部分で詰まってしまいました。特に、実装の点については行き当たりばったりで行っていたためゲームバランスの調整がうまく出来ず、製作に時間が掛かってしまいました。

また、デザイン面に関しては画像が非常に寂しいものとなっており、背景やスコアといった要素も無いため、シューティングゲームとしては完成度の低いものになってしまった感がありました。

次のゲーム製作では、どんなゲームを作りたいのか、どう作れば楽しめるゲームになるか、ということを考えて、ただプレイできるだけのゲームから、人を楽しませるゲームを作れるように計画を立ててゲーム製作を行っていきたいと思います。

流星のナナシマン X

理工学部 1 年機械情報工学科
古川 耕大

はじめに

エレクトロニクス研究部に入って学んできたことをいかしてゲームをつくりました。

開発環境

C 言語

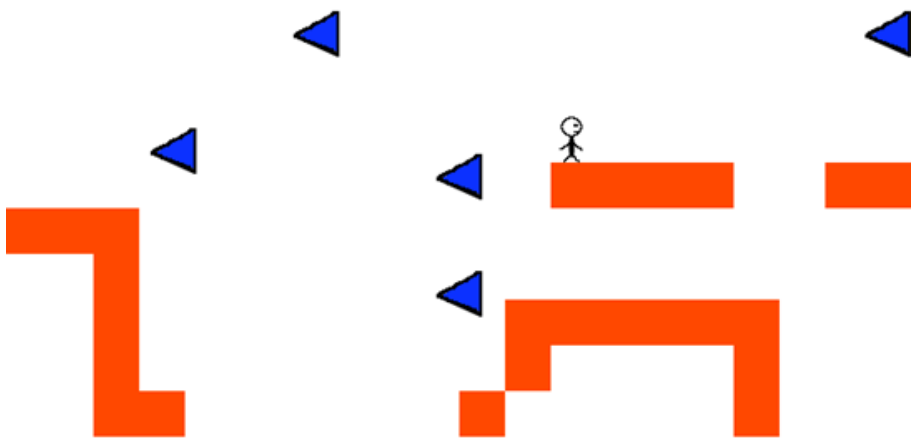
Microsoft Visual C++ 2010

DX ライブラリ

ゲームの説明

このゲームは、敵を倒したり、トラップを避けたりしながらゴールを目指すアクションゲームです。

敵を倒すとたまに回復アイテムが出ることがあります。



操作方法

方向キー右左　：　移動

Z 又は〇　　：　弾を撃つ

X 又は×　　：　ジャンプ

工夫した点

動く床や敵を倒すと回復アイテムがたまに出てくるようにしました。

反省点

音を入れることができていなかったことと絵がひどかったこと敵の動きが単調すぎる

など反省点はいろいろとあります。

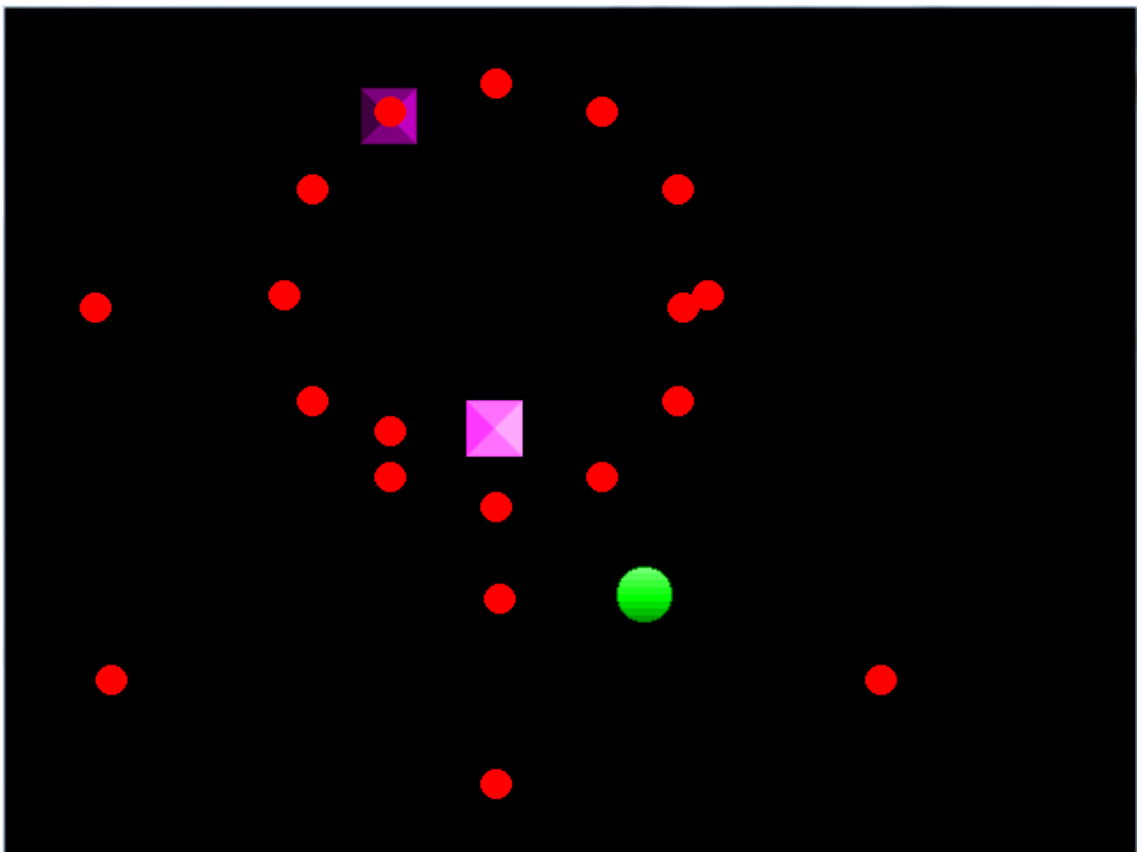
来年はこの反省点を改善したゲームを作っていきたいです。

シューティング

理工学部 1 年機械情報工学科
松田元希

はじめに

エレクトロニクス研究部で習ったプログラミングを用いてシューティングゲームを作成してみました。



開発環境

Ｃ言語

ＤＸライブラリ

Microsoft Ｖisual Ｓtudio

ゲーム説明

① 操作方法

移動・・・矢印キー

ショット・・・Z キー

変速・・・Shift キー

② ゲームの目的

上から４種類の敵が来て違う種類の弾を撃ってくるのでそれをその弾を避けつつ自分も弾を撃ち敵を倒していきます。敵を倒すと得点が加算されていくので高い点数を目指しましょう。得点をきそうものなのでゲームクリアはありません。

反省点

こんなゲームを作ろうというイメージができていなかったので、絵やゲーム自体が微妙な物になってしまいました。プログラミングの技術不足のため他人のかき方をたくさん聞いてしまった。来年以降は早い段階からの計画・製作に着手し、更なるゲームクオリティの向上などを目指して行こうと考えています

THE DURA

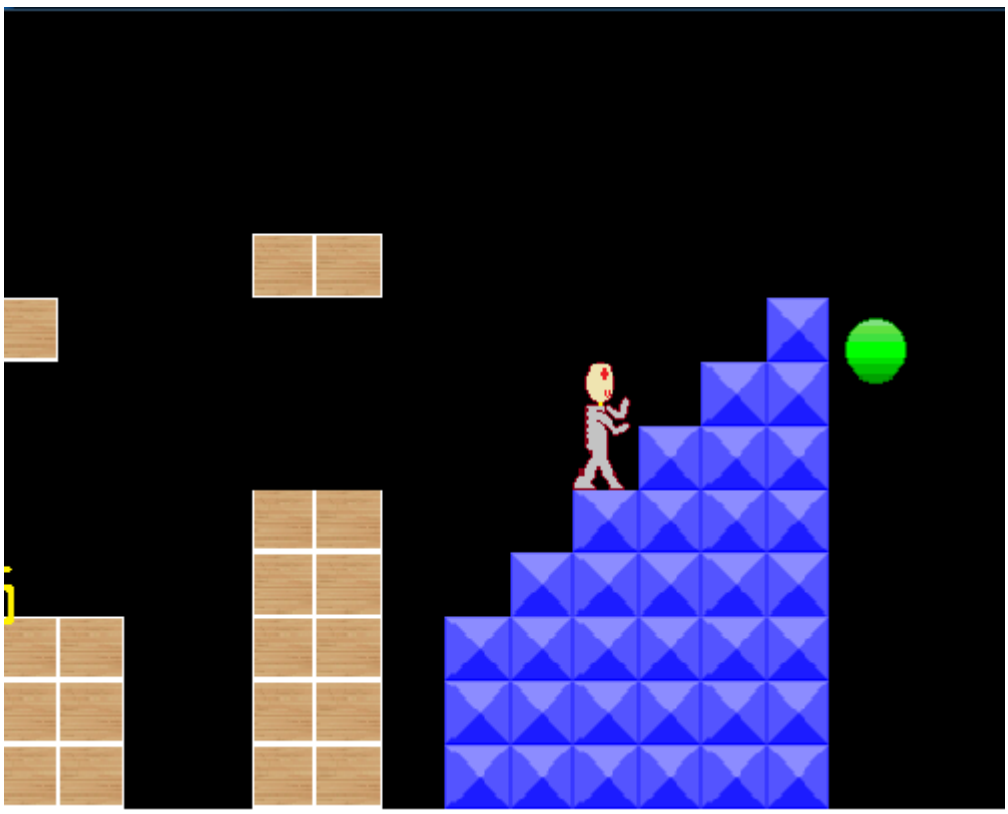
理工学部 1 年機械工学科
光村　聡麻

はじめに

このゲームは敵キャラクターに接触したり、穴に落ちたりしないように主人公「DURA」を操作してゴールのお城を目指す、劣化マ〇オです。

開発環境

DX ライブラリ
C 言語



ゲームの説明

矢印キー（十字ボタン）：移動・カーソル移動

Z キー（２ボタン）：ジャンプ

※（）内のボタンはゲームパッドの初期配置です。

工夫した点

敵の動きのパターンを数多く作り、単調な動きをできるだけ少なくしました。

反省点

・ライフをつけなかったのですぐにゲームオーバーになってしまったこと。

・1 ステージしか作れなかったので、すぐにクリアできてしまうこと。

・スコアや、時間制限、アイテムなどを実装できなかったこと。

・自機のアニメーションが無く、動いても足が動かないこと。

しろくろっ！

理工学部１年電気電子生命学科
宮崎　泰行

はじめに

このゲーム「しろくろっ！」は新感覚のパズルゲームです。マップ下側のキャラクターと結界（六芒星と円になっている部分）を動かして、結界内にある白石・黒石をすべて消したらゲームクリアとなります。

ゼミＢのシューティングゲームがプログラムのもとになっています。

開発環境

・Visual Studio 2010
・DX ライブラリ
・DigiOnSound5 Express
・ペイントツール SAI Ver.1

工夫した点

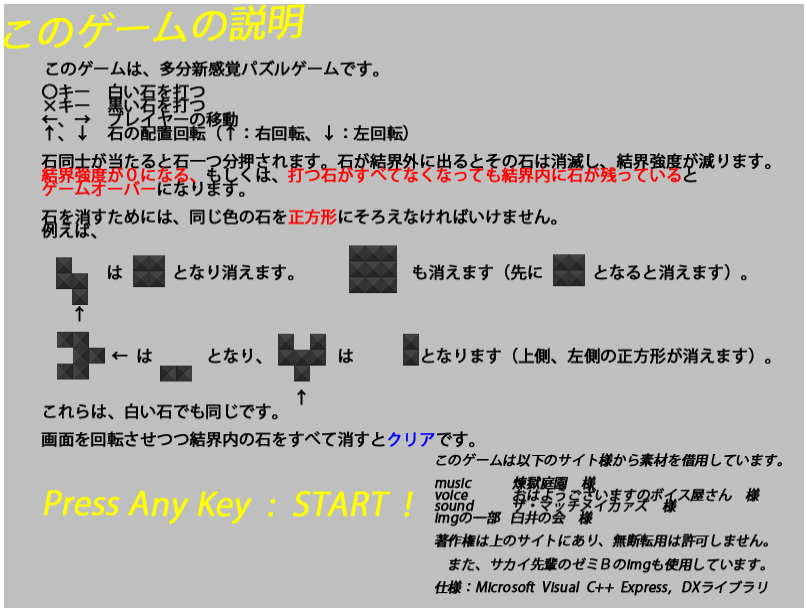
「石は当たると一つ分前に進む」という当たり判定を製作し、ゲームの難易度と考えさせる部分を多くしました。

連鎖、もしくは九個の石の正方形で消した場合、カットインが入ります。

ゲームプレイ中、ゲームクリア、ゲームオーバー時にタイトルに戻らずそのままやり直すことが可能です。

キー入力、石が他の石に当たるなどに対してサウンドが、石を打つ、石が消えるなどにはボイスが設定されています。また、タイトル、ゲームプレイ中、ゲームクリア、ゲームオーバーそれぞれに音楽があります。

ゲーム説明を起動時に見られます。さらにプレイ時にも見ることが可能です。



問題点・これからの課題

これは元から作ったものではなく、あくまでもゼミＢを改造してここまで作ったものなので、あまり構造体などを理解しないまま作られています。そのため、プログラム内であまり分からない部分が結構あります。

またこのゲームはゲームマップが一つしかなく、いきなり難易度が高いのでクリアできずにそのまま、という状況になった人が多くいました（生明祭公開時）。そのため、簡単、普通程度のマップの構成をして、もっと敷居の低いゲームとして製作を進めていきたいと思います。

FIRST QUEST

理工学部１年機械工学科
室岡龍介

はじめに

このゲームはエレクトロニクス研究部で教わったＣ言語の知識をもとに、ゲームとして基本的な処理をどうするか考えながら、プログラミング技術を向上させるために製作しました。

開発環境

Visual Studio C++ 2008
Dx ライブラリ

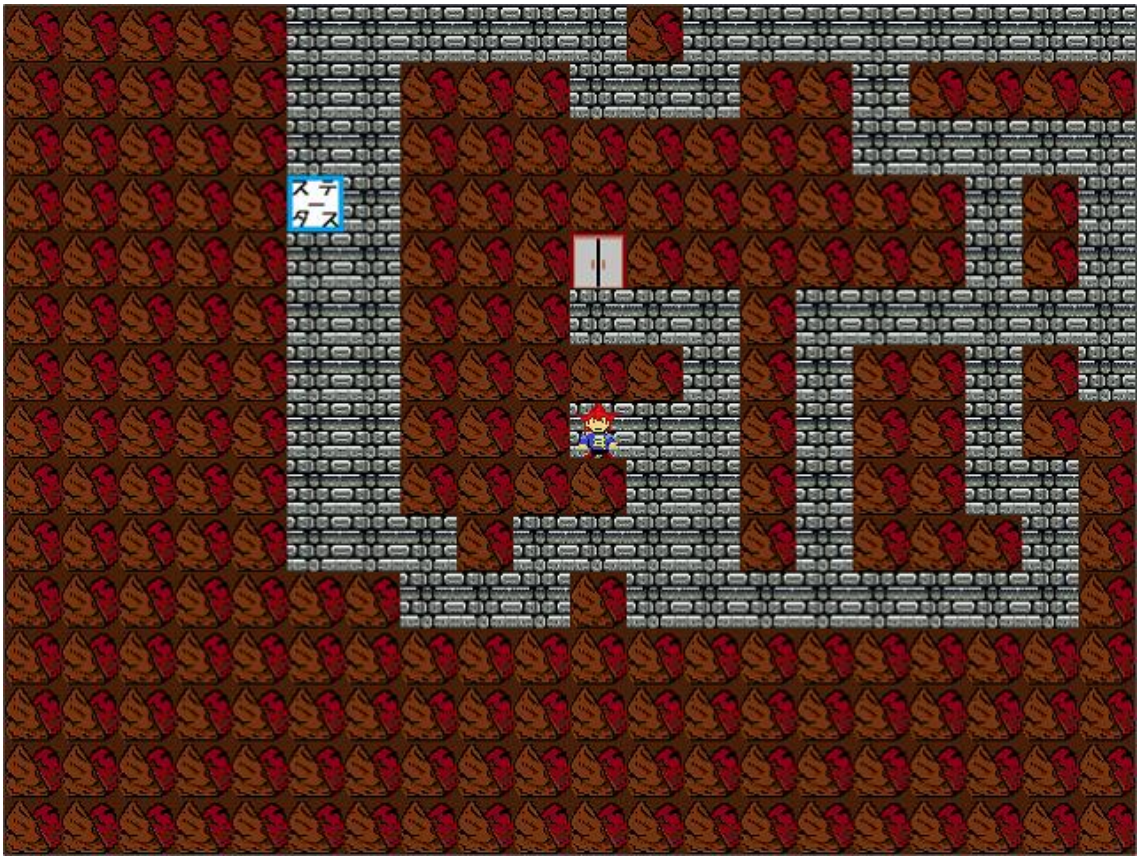
ゲームの説明

方向キー：移動・コマンド入力
文字キー：名前入力
Enter キー：名前入力の決定

初代ドラクエ風ＲＰＧです。
敵を倒すとお金と経験値が手に入ります。店でこまめに回復したり、レベルアップしてください。プレイヤーのレベルが上がると、敵も強くなるので注意してください。最奥にいるボスを倒すとクリアです。

工夫したところ

まずマップの作り方を覚えてから、マップ上のイベントを色々作っていきました。
メッセージの表示は、マップの外にメッセージウィンドウを表示させる事で、画面上に見えるようにしました。戦闘に関しては、敵の画像を分割して配列から読み込ませる方法で、マップ上に複数種類の敵を毎回ランダムで出現させる事に成功しました。



反省点

ソースの大部分をマップ表示関連の処理とデータが占めており、マップが広がるほどソースが長く見づらくなってしまいました。マップを 140×50 程の配列ですべて処理しているため、イベントで使うマスを１マスずらす度にマップをかなり書き換えなければならなかった事が非効率的でした。また、プレイヤーキャラの画像が固定だったり、戦闘エフェクトが効果音だけだったり、決定・キャンセルキー等を使用していなかったりと、表現がやや乏しかったです。また、製作の半分以上の時間が戦闘にかかっていたため、他の部分をじっくり修正できなかった事も原因だと思います。

Proto3

理工学部 1 年機械情報工学科
山本　祥太郎

はじめに

　このゲームは、画面左方にいる敵からの攻撃をよけつつ、画面上方にいるボスを倒すことを目的としたアクションゲームです。

開発環境

C 言語

Microsoft Visual Studio 2010

DX ライブラリ

操作方法

左右移動 ： ← →

ショット ： Z

ジャンプ ： X

がんばった点

あまり必要性はなかったけれど、プログラミングの練習だと思い、自機に合わせて画面がスクロールするようにしてみました。

また、敵がいたところに星が残るなどの演出部分もがんばった部分です。

反省点

まず、自分が当初考えていたゲームよりだいぶ下回る出来となってしまったことです。これは、計画をもたずにゲーム製作に取り組んだためです。

主に不完全だった点は、自機と敵ショットのあたり判定がうまくいってない、グラフィックが適当、バグがのこってしまった、などです。

来年度はもう少し計画的にゲーム製作に取り組み、今年度の不満点はすべて解消したゲームをつくっていきたいです。

感想

初めてのゲーム製作は、想像以上に大変であり、満足するものはできませんでしたが、自分でゲームを作ることはとてもおもしろく、いい経験になったと思います。

来年以降より完成度の高いゲームを製作していくためがんばっていきたいと思います。

action

理工学部 1 年情報科学科
梁 耀仁

はじめに

大学で習った C 言語を利用して初めて製作したゲームプログラムです。明治大学エレクトロニクス研究部で学んだことや DX ライブラリのヘルプを参照しつつ作りました。

開発環境

・C 言語
・Microsoft Visual C++ 2008
・DX ライブラリ

操作説明

・キーボード　　↑　　:ジャンプ
　　←→　　:移動
　　Z　　:決定

・アナログキー　　↑　　:ジャンプ
　　←→　　:移動
　　○　　:決定

ゲーム説明

横スクロールアクションゲームです。プレイヤーはゴール地点である、白マスに弾をよけつつめざしてください。落ちたり、弾にあたってしまったら GAME OVER です。

※弾の y 座標はランダムになっています。
※プレイヤーは壁に反発します。

工夫した点

・タイトルの実装。

・クリア時間をわかるように、ゲーム画面左上に時間をつけました。

・プレイヤーには動きやすいように慣性をつけ、なめらかにしました。

・弾の位置には乱数を使用してランダムにしました。

反省点

プレイヤーと壁との反発が自然ではない感じに仕上がってしまいました。アクションゲームということなのでもっとグラフィックにこだわるべきでした。（背景のみフリー素材を使用）

ただゴールを目指すというだけなのでゲーム性にかけてしまいました。

Stump the beatS

理工学部 2 年電気電子生命学科
赤井 優也

はじめに

十数年前に大流行した、KONAMI の音楽ゲーム、DanceDanceRevolution(以下 DDR)と、2008 年より大好評稼働中の jubeat をあわせたようなゲームを作りました。

DDR に関しては、ハード班製作ページ「DanceDanceRevolution 専用コントローラー」に詳しいので参照のこと。というかそっちも自分ですが。

その DDR 専用コントローラーが、上下左右の 4 方向ではなく、斜めと真ん中を含めた 9 つのパネルが反応するようになっているので、それをフルに使うべく「3*3 パネル」を「足で踏む」ゲームです。

開発環境

総合開発環境:

Microsoft Visual C++ 2008 Express Edition with SP1 - JPN

使用ライブラリ:

DX ライブラリ

グラフィック関連:

Microsoft Paint

Greenfish Icon Editor Pro 1.6

Adobe Illuslator CS2

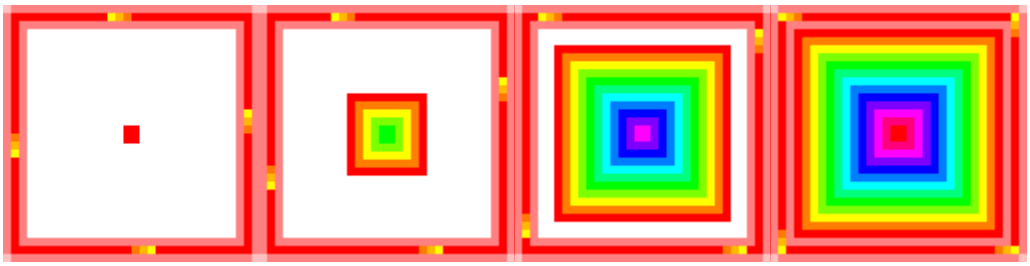
ゲーム説明

タイトル画面で何かキーを押すとスタートです。ARE YOU READY?の音声のあと、音楽が流れはじめます。

画面に表示された 9 パネルに、それぞれ踏むタイミングを示すマーカーが表示されるので、音楽のタイミングにあわせてパネルを踏むか、キーボードの場合、テンキーを押しましょう。

曲が終わった時点で、スコアが 7 割を超えていればクリアです。また、得点

に応じてランクと評価が与えられます。



MISS　　GOOD　　GREAT　　PERFECT

反省点

　このゲームは、ハードウェアの製作が先にあり、それをうまく活用すべく製作した作品であります。

　正直言ってアレを発表するだけであれば、Pump機能もついているstepmaniaて事足りるわけですが、せっかくプログラミングを習得したのだからゲームも自作しようということになったわけです。

　その際、従来のように縦方向にオブジェが流れるゲームであれば、確実に既存のゲーム未満にしかならないと思ったのでこういう形式のゲームにしたのですが、結局既存のルールの組み合わせになってしまいました。(縦方向スクロールの音ゲーに関しては、ソフト班製作ページ、芦沢直樹氏製作「StepR」参照)

　また、去年度製作したゲームがシューティングだったので、1から音ゲーを、ハードウェアと平行して製作したので、まだまだいい加減だったり荒削りな箇所が多くあります。

　しかし、ハード、ソフト共にある程度は思っていたものを実現できてよかったと思います。

SPECIAL THANKS

-SOUND INFO and COMPOSER-

MUSIC　　:　sky perfume

GENRE　:　TRANCE

COMPOSED by　　：　せくの/sechno(オーディオエンジニアリングクラブ)

-HARDWARE-

ハード班製作ページ「DanceDanceRevolution 専用コントローラー」へ

大丈夫なゲーム

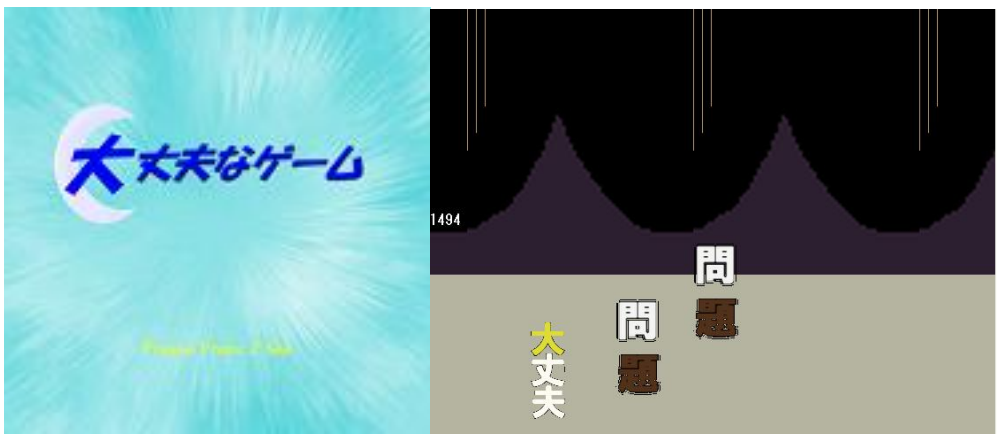
理工学部 2 年電気電子生命学科
岩田　純一

はじめに

去年製作したゲームはプログラム内部にツギハギが多く、細かい部分の修正に多くの時間をとられ、仕上がったゲームのバランスはあまり芳しくありませんでした。そして、ゲームの開発データが大学入学時に購入したネットブックごとぶっ飛んでしまう事故もありました。

そこで、今年は去年のゲーム開発で得た知識と技術を活かし、より効率的に手際よく、特にバックアップに問題なくデータの消えない大丈夫な、すばらしいゲームを作りたいと考えておりました。

↓タイトル　　↓ゲーム画面



開発環境

使用言語：　C 言語
使用ソフト：　Microsoft Visual C++ 2008
使用ライブラリ：　DX ライブラリ

操作方法

十字キー：　移動

Ｚキー：　決定、当たり判定表示

ゲームの説明

ゲームが始まると大丈夫が走っているのが見えます。これは、プレイヤーです。彼には３２通りの名前があるらしいです。確か最初の名前は・・・jiki()。操作が効かない奴でした。貴方は彼を操作して、突進してくる問題を避けなければなりません。しばらく避けていると問題の数が増えます。がんばって避けましょう。

また、道を走る時はいくら必死でも、良識ある市民である彼は通行人の迷惑にはならないよう、気をつけて走るべきだと意識しています。特にお年寄りにぶつかった時には激しい罪悪感に心が押しつぶされてしまうことでしょう。

さて、最終的な目標は神を倒すことです。神を倒すことは、可能でしょうか？神を倒すと、それまでにかかった時間が表示され、ゲームは終了です。
いかに大丈夫な結果が残せるか、それはプレイヤーに任されています。

工夫したところ

キャラの色配分。爺やの出現。タイトルのインパクトを強くしたこと。

反省点

このゲームは本当に短時間で出来上がりました。すばらしいゲームを作りたいという気持ちは持っているだけでは意味がありません。その気持ちは日々の製作をコツコツと積み重ね、その努力が思い描くゲーム像に近づいていく喜び、それを分かち合える仲間、勇気、友情、勝利とかが大事だと思います。このゲームを作り始めた時期は確か、君たちにとっては半年前だが、私にとってはつい最近(ry
えー、何が言いたいかというと・・・・・まあ、・・・今回も駄目だったよ。
次はこれを遊んでもらった君にも協力してもらうとしよう。

FEUERN

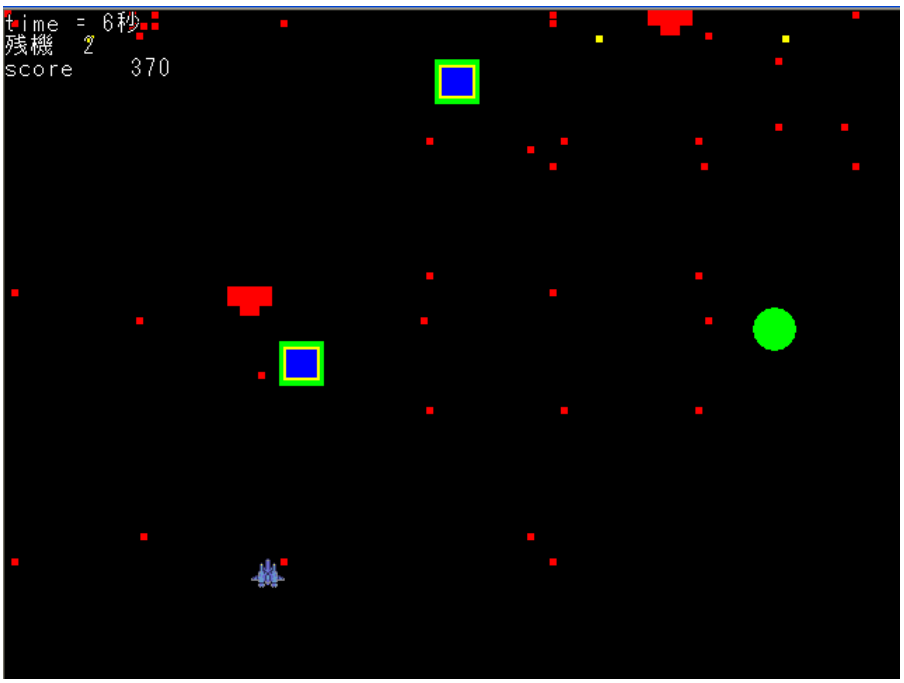
理工学部 2 年応用化学科
小沼 宏彰

開発環境

C 言語
Microsoft Visual C++ 2008
DX ライブラリ

ゲームの説明

画面の上から敵が出てきます。敵の攻撃を避けながら弾を撃って倒してください。弾に当たると残機が減ります。残機が 0 の時にあたるとゲームーオーバーです。



操作方法

移動	→←↑↓
ショット	Z

反省点

開発を本格的に始めるのが遅くなってしまい、やろうと思っていたことが十分に出来なかった。そのため、キーコンフィグやハイスコアを記録する機能をつけることが出来なかった。今後は計画性を持って行おうと思う。

Ｆｉｎａｌ　Ｓｕｒｖａｉｖａｌ

理工学部２年情報科学科
齊木　雅博

はじめに

このゲームは、画面下の要塞を守りながら、要塞と協力し、敵を倒していくシューティングゲームです。

開発環境

C 言語

DX ライブラリ

Microsoft Visual Studio 2008

ゲームの内容

① 操作方法

移動・・・矢印キー又は方向ボタン

ショット・・・Z キー又は 1 ボタン

② ゲームの目的

要塞に耐久値が設けられており、敵が画面下部に入り込むと要塞の耐久力が減るようになっていきます。プレイヤーはそれを自機を使って阻止します。

また、要塞からも一定時間ごとに援護砲撃をしてくれます。

要塞とともに敵を撃破し、ハイスコアを目指しましょう。

また、ゲームクリアは仕様上存在しません。

反省点

まず、今回のゲームは前年度に製作したシューティングゲームで出来なかった事をするという事を目的に、前年度の反省を踏まえて、早めの段階から計画・構想を練り、製作していました。しかし、製作後期にデータの紛失があったために、それまでの今回は細かいファイル分けや、プログラミングのスキル向上にともなった内部処理の変更等はおこないましたが、前作の再現をするので手一杯となり、外見においても前作と比べて変更点が見い出せなかったため、とても残念でした。来年もまた機会があれば、こういったミスのないように、今度は新たなジャンルへのゲーム製作に着手したいなと思いました。

Existers

理工学部 2 年情報科学科
坂井 誠也

はじめに

このゲームは、オブジェクト指向言語を学習し、それを生かしたプログラミングの経験を積むために製作しました。オブジェクト指向言語としての部分だけでなく、リスト構造などのデータ構造を利用した動的メモリの確保を用いたプログラムも使って、自分の技術力を惜しみなく投入しています。

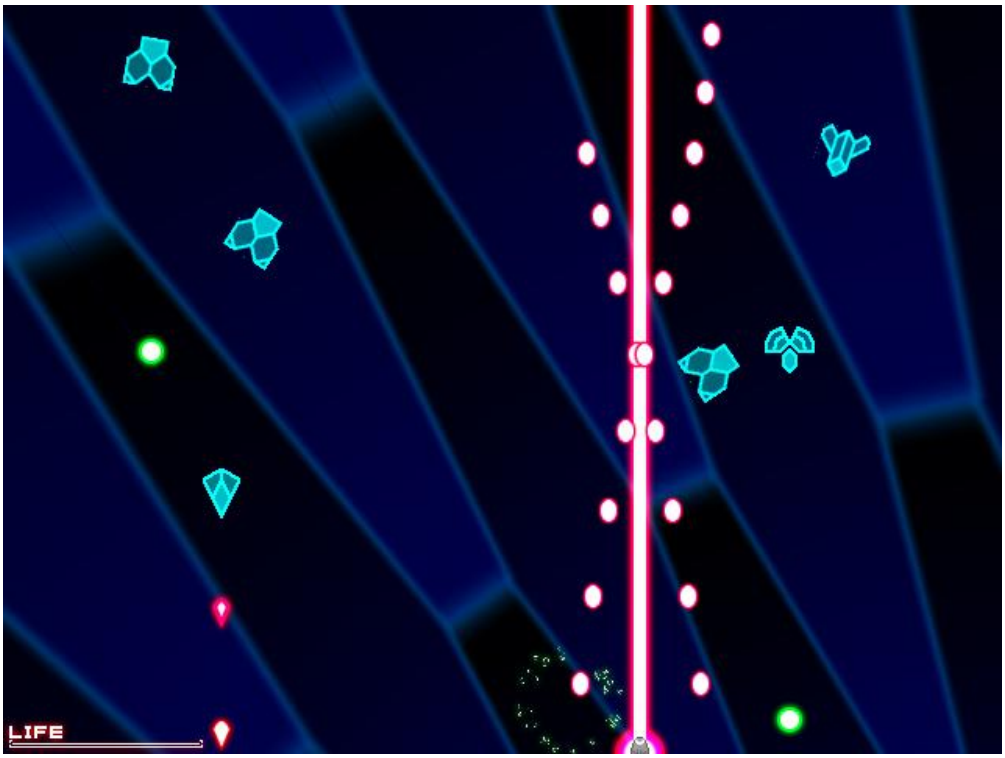
製作環境

・プログラム
  ・Microsoft Visual C++ 2010
  ・DX ライブラリ 3.04b
・グラフィックス
  ・Photoshop CS5 体験版
  ・edge
  ・Paint.NET

ゲームの説明

	ゲーム画面	タイトル画面
上下左右キー	自機移動	カーソル移動
N キー	赤ショット(Radiant Ray)	決定
M キー	緑ショット(Glow Gnomon)	キャンセル
<キー	青ショット(Blazing Blade)	使用しません

キーコンフィグを搭載しているので、ジョイパッドにも対応します。ゲーム自体は、出現し続ける雑魚敵を倒し続けるだけの無限ループするゲームです(プログラミングに時間をかけすぎました)。



工夫したところ

画面上の全てのオブジェクトは、Worldクラス内のリストにポインタを登録されていて、このポインタから関数を間接参照することで、多様性によりオブジェクトにより様々な異なる挙動を実現しています。また、これらのオブジェクトのポインタの登録は、コンストラクタ内で自動的に行われ、消去時は自己をデリートし、デストラクタ内から登録されているポインタを自動的に削除しています。

反省点

ここまで製作して思ったことは、サブクラスを大量につくることを前提にしたプログラムになっているせいで、プログラムがごちゃごちゃしてしまい、大変だ、ということです。

出来ることなら Xtal や、Lua 等のスクリプト言語を習得し、下位のサブクラス内でのメンバ関数のオーバーライドによって書き換えている部分をスクリプト言語で置き換えたほうが良いだろうと思いました。

ハイスピードスタイリッシュ（笑）避けゲー

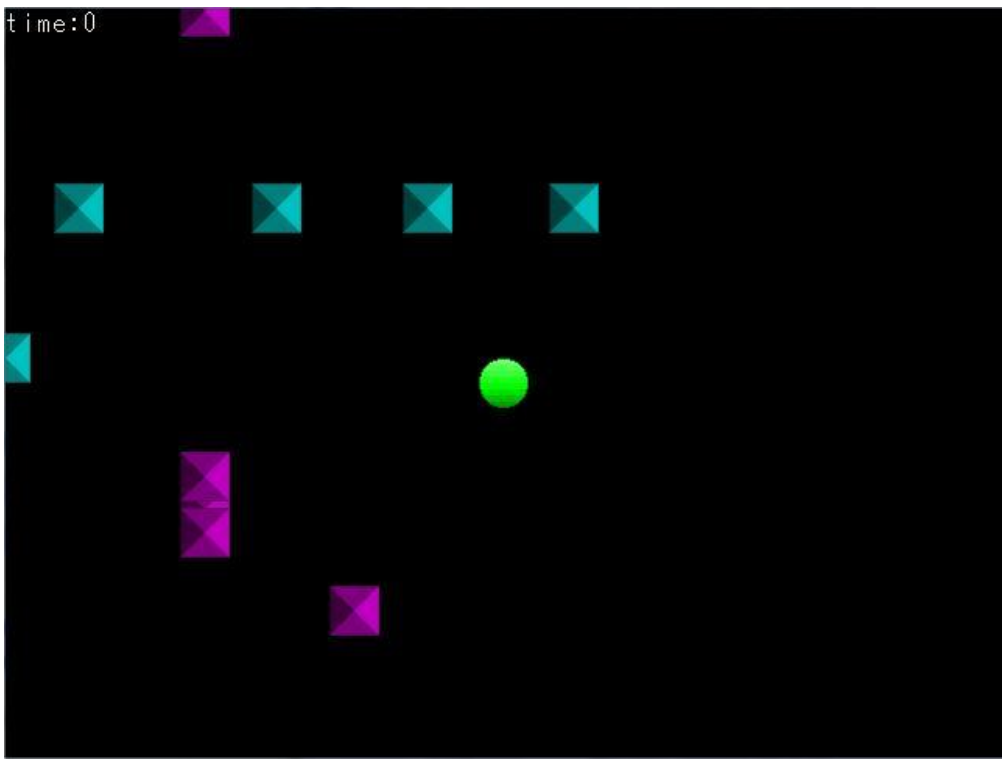
理工学部 2 年機械情報工学科
須田　和樹

はじめに

前回製作したのはアクションゲームだったので、今回は別のジャンルのゲームを作ろうと思い、シューティングゲームを作ることにしました。結果的にただの避けゲーになってしまいましたが。

開発環境

使用言語：　C 言語
使用ソフト：　Microsoft Visual C++ 2010
使用ライブラリ：　DX ライブラリ



操作方法

十字キー：　移動

ゲームの説明

このゲームはただひたすら敵から避けるだけのとてもシンプルなゲームとなっています。操作コマンドも移動キーだけです。四方から敵が出現するのでそれを避けてください。一回でも敵に当たってしまうとゲームオーバーとなります。

工夫したところ

ランダム関数を積極的に取り入れました。

感想（ほぼ反省点）

今回はゲーム製作に入ったのがとても遅れてしまったので、製作期間がかなり短くなってしまいました。その為凝ったゲームが作れず、何か比較的簡単に作れて面白いゲームは作れないかと模索しました。シューティングにしようと思っていたのですが、弾の導入や軌道など考えないといけないのでこれは断念しました。ですが、逆に弾を完全に無くして「避けゲー」にしてしまえば良いんじゃないかと思い当たり、製作を始めました。

作っている途中でも色々と面白そうなアイデアが出てきたのですが、時間的な余裕が無かったり技術的に足りない部分があったりしたので導入することはできませんでした。

完全体ではありませんが、ランダム関数を上手く使えば凝らなくてもそこそこに面白いゲームが作れるんだと分かりました。

戦略

理工学部 2 年応用化学科
田辺 優

はじめに

去年はアクションゲームを製作したので今回は全く違うジャンルのゲームを製作したいと思い簡単なシミュレーションゲームを製作しました。

開発環境

C 言語
DX ライブラリ
Microsoft Visual Studio 2008

操作説明

矢印キー	：カーソルの移動
N キー	：決定
M キー	：キャンセル
	：ターン終了
R キー	：タイトルに戻る
ESC キー	：ゲーム終了

ゲーム説明

黄色い味方にカーソルを合わせて決定キーを押すと 1 ターンに一回だけその味方を動かすことができるので味方を動かして赤い敵を倒してください。動かしたい味方がいなかったらターン終了をしてください。ターンを終了するとこちらと同じように敵が動いて味方を攻撃してきます。敵のターンが終わると自分のターンになりすべての味方を再び動かすことができます。これを繰り返して敵を全員倒すと勝利です。逆に味方が全員倒されると敗北です。スキルは SP を消費して強力な攻撃を行うことができます。「戦」というキャラは隣のマスにしか攻撃出来ませんが、「弓」というキャラは一つ離れたマスにも攻撃できます。味方を動かすときは、「岩」のマスと敵のいるマスは移動できません。

工夫した点

敵の動きのパターンを作るのに苦労しました。最初のうちは敵が全く動かなかったり、自分のほうに向かってこなかったりしてしまいましたが、最終的には味方にひたすら向かってくる敵と、攻撃範囲に味方がいたら向かってくる敵の 2 パターンの敵を作ることができました。

反省点

本当はキャラの種類を増やしたり、レベルの概念を入れたりしたかったのですが、時間が足りずにそれらの要素を組み込むことができずあまり戦略性がないゲームになってしまいました。また画像が用意できず見た目がとてもシンプルになってしまいました。

真・一機倒千 ～四角い奴らを丸くするっ！～

理工学部 2 年情報科学科
塚本 耕司

はじめに

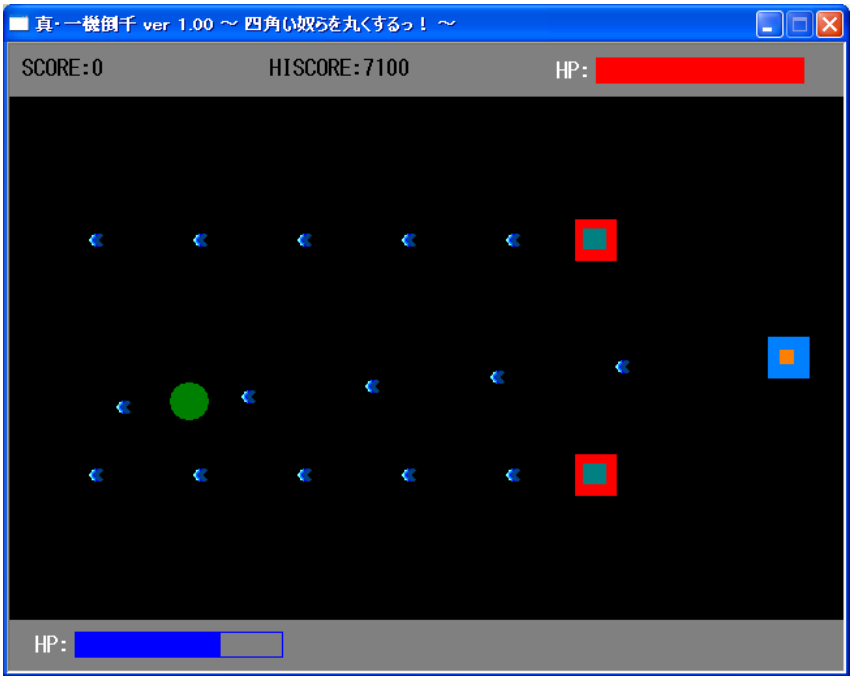
明治大学情報科、及びエレクトロニクス研究部に入って 1 年が経ち、去年の経験や反省を活かしてもう一度シューティングゲームを作ろうと思い、製作しました。

開発環境

使用言語　　：C++
統合開発環境　：Microsoft Visual C++ 2008 Express Edition
使用ライブラリ　：DX ライブラリ
グラフィック関連：Microsoft Paint

ゲーム説明

敵や敵の弾幕を上手くよけながら自分の機体の弾を当てて敵を倒す、シンプルなシューティングゲームとなっています。敵を倒したりアイテムをとったりする事でスコアが増え、最後に出てくるボスを倒せばゲームクリアです。



操作方法

今回の操作方法。カッコ内はパッドでの割り当てです

十字キー　　：自機の移動

Z キー(A ボタン)　：ショット

工夫した点・反省点

去年製作したゲームでは、当たり判定が上手く動作せず、敵に当たっているのに敵を倒せないという酷い仕様となっていました。それを受けて今回は最初に当たり判定の関数を実装し、その結果去年は if 文内の条件式の処理順序のミスという事がわかったので、その後は気持ちを新たに製作をつづける事が出来ました。

また、去年ソースの肥大化を招いた主な原因である敵の初期データの代入をテキストファイルからの読み込みに変更した事で、敵の配置や出現タイミングなどをしっかりと把握しながら編集できたのは非常に良かったと思います。

その一方で、企画段階で実装しようと考えていた複数ステージやアイテムによる自機のパワーアップ等ができなかったのが心残りです。

Commander

理工学部 2 年情報科学科
松下 竜真

開発環境

C++

Win32 API

Borland C++ Compiler 5.5

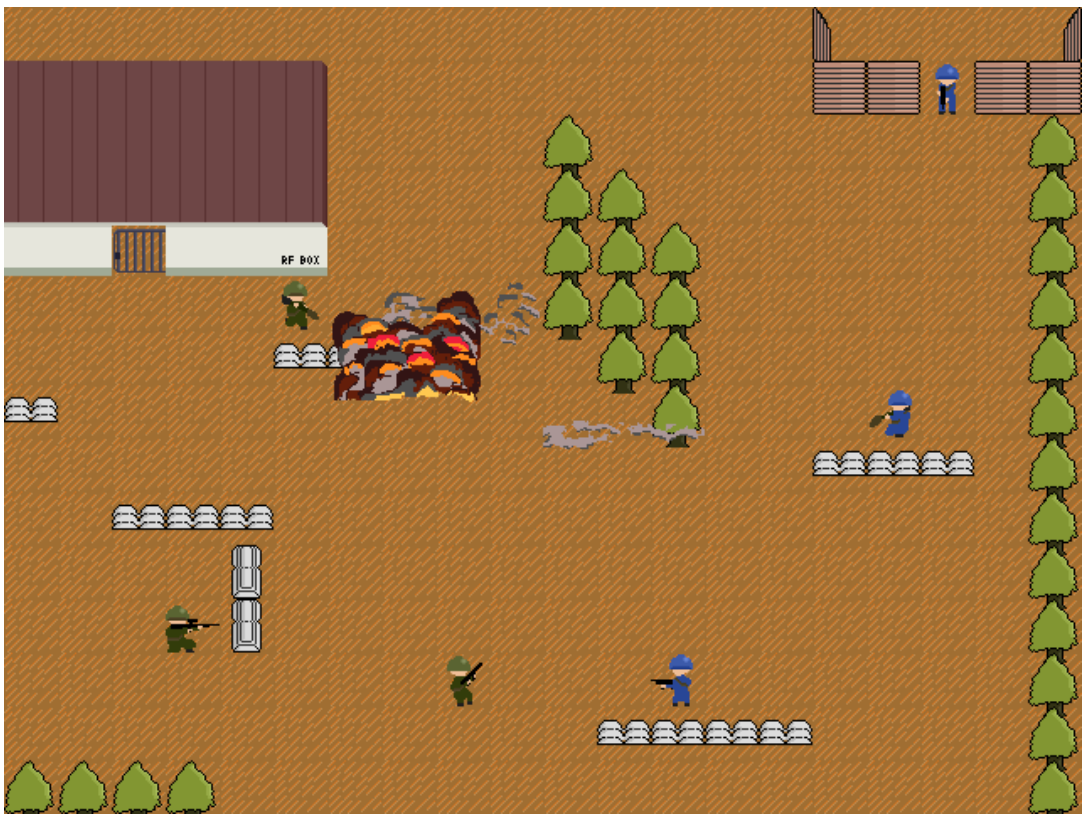
サクラエディタ 1.6.5.0

EDGE 1.29b

ゲームの説明

指揮官となって自軍に命令を出し、任務を遂行する戦略シミュレーションゲームです。

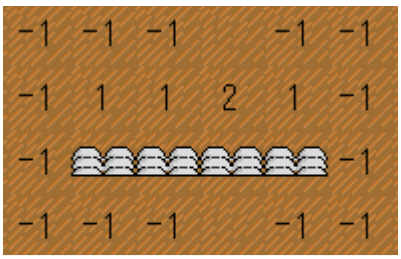
工夫した点



敵(CPU)に移動先を決定させる方法として、各マスに評価値を振る方法を取りました。様々な評価基準を設け、そのそれぞれの基準により評価された値の合計を、最終的なそのマスの評価値としました。

・そのマスがプレイヤーの視界に入っているならば -1
・土のうを挟んでプレイヤーを狙えるマスならば +1

といった具合です。最終的な評価値が最も大きく、現在位置から最も距離が近いマスに移動させるようにしました。



各マスの評価値を可視化した図

反省点

去年の反省点から、今年はメモリ使用量に気を配りながら製作しました。その甲斐もあってか、全体的に動作は軽くなったのですが、やはり今年も敵の思考中に処理が重くなってしまいました。その原因として、処理を１フレーム中に詰め込み過ぎていることが考えられます。これは、処理を小分けにして分散させることで解決するかもしれません。

そらりん

理工学部 3 年電気電子生命学科
日下部 隼也

はじめに

前作に普通の音ゲーを作ったので今作は普通じゃない音ゲーを作るというコンセプトのもとゲームを製作しました。結局既存に近い感じに落ち着いてしまいましたが、オリジナルな部分を見ていただけると幸いです。

開発環境

- Microsoft Visual Studio 2007
- DX ライブラリ
- EDGE

ゲーム説明

奥から手間に迫る！新感覚スティック操作音ゲー

操作方法

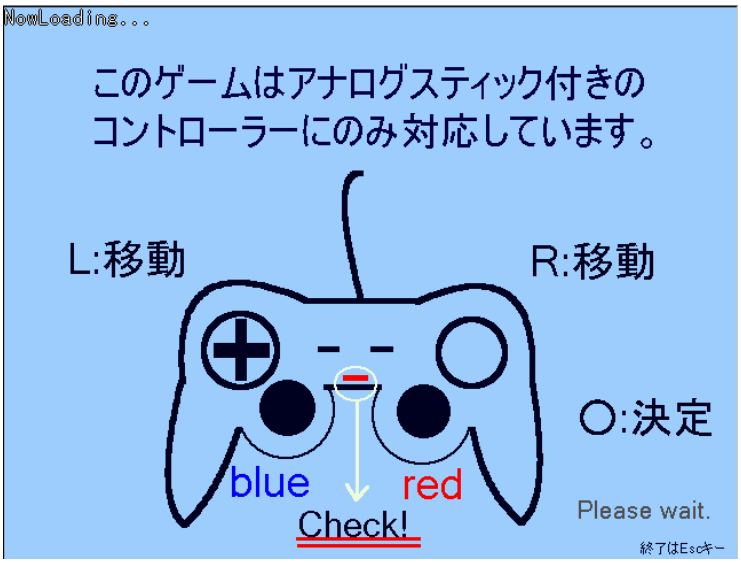


図１コントローラーの基本操作

デュアルショックにのみ対応

L と R によるカーソルの移動

○ボタンで決定

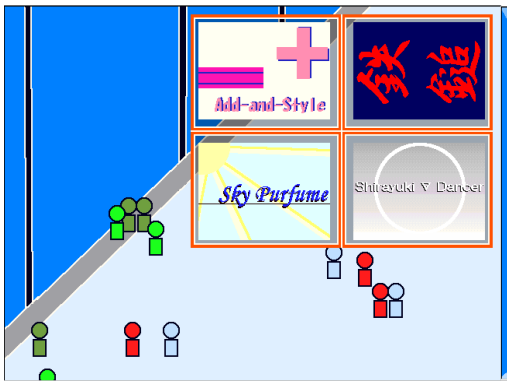


図２. 選曲画面

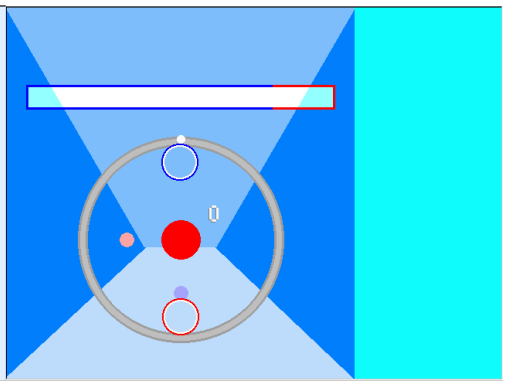


図３. ゲーム画面

スティック左：青

スティック右：赤

ゲームのルール説明　（図３参照）

後ろから迫ってくる赤や青の丸（ノート）に対応する方向にスティックを倒し続ける（１小節）ことにより上部のゲージが溜まっていく。
赤や青の丸い輪っかは次に来るノートのヒントを示している。これを手がかりにプレイするとうまく出来るかもしれない。

感想

折角楽曲提供してもらったり絵を描いてもらったりしているのにプログラムのほうが中途半端で大変申し訳ない気持ちでいっぱいです。
判定に問題があるために実装できなかったシステムがいくつかあるので気力があればもう少し手直ししたいです。
初見で全然やり方がわからないというクレームが多数ありますがなんとか理解してください。

パンケーキノート

理工学部 3 年機械情報工学科
真島 俊樹

はじめに

三年生なので順当に三作目になりました。相変わらずなアクションゲームを製作しました。昨年、一昨年の作品についてはエレ研のサイトに以前の機関誌があるのでご覧ください。今作はノートの中の色鉛筆で描かれた世界のお話です。

Fig 1 いつもの感じのタイトル画面

開発環境

統合開発環境　　：VisualC++ 2008
使用ライブラリ　：DX ライブラリ、WindowsAPI、Lua
グラフィック関連：Microsoft Paint ver5.1　……主にドット絵
　　：ウェブアートデザイナ ver7.0　……タイトル画面とか
　　：Paintgraphic ver3.0.4　……イラスト
　　：Microsoft Picture Manager　……スキャンデータの処理

開発環境も昨年、一昨年と変わらずです。ドット絵を描くときなどは Edge などのドット絵専用ソフトを用いると便利という話を聞きますが、MS-Paint が使い慣れているので MS-Paint を用いました。またウェブアートデザイナというソ

フトはホームページビルダーの付属のおまけソフトですが、ベクトル系ソフトで文字関連のエフェクトが扱いやすいため愛用しています。

ゲームの説明

普通のアクションゲームとなっています。右に行くとクリアです。今作の特徴は背景画面がすべて色鉛筆で描かれた紙をスキャンしてそのまま使っている点です。あとは前作に比べてキャラ絵が小さくなって可愛くなりました。

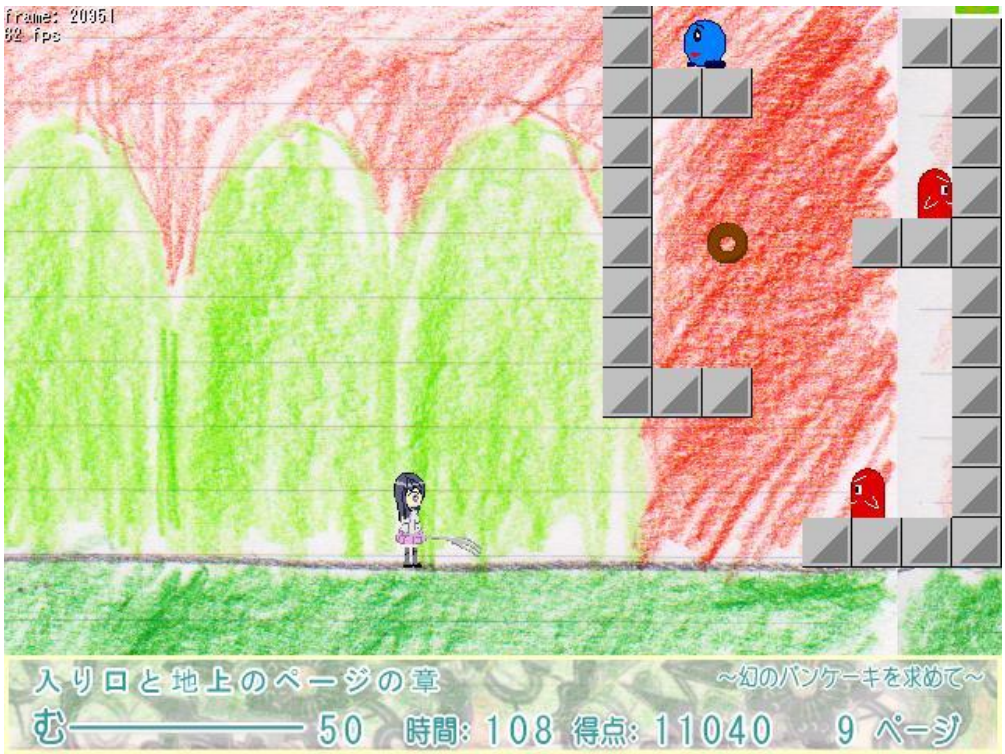


Fig 2 いつもの感じのゲーム画面

攻撃方法は敵を上から踏むのとフォークを使っての攻撃です。前作でフォークを使った攻撃が心残りだったのと、そのシチュエーションが(製作者が)気に入ったので今回もフォークをどっかから持ってきました。キャラクタは前作同様、樹沙ちゃんです。これで 3 作全部でましたね。

ドット絵について

毎年ですがドット絵についてはわりと気合いを入れて描いています。ツールは

前述の通り MS-Paint です。今作は主人公の絵で 15 枚です。歩行モーション 5 枚、攻撃モーション 9 枚、ズコーが 1 枚です。

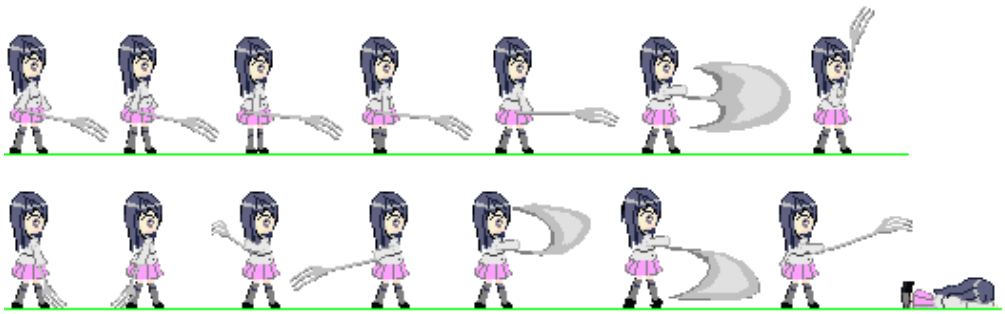


Fig 3 ドット絵

波動砲が出ているのがありますがアニメーションにすると案外普通に見えます。この波動砲は攻撃モーションの攻撃判定が出ている時の絵で 2、3 フレーム(約 0.3 秒)しか表示されないので人間の目からみるとフォークを振りかざしているように見えます。またこうやって大げさにすることで当たり判定の範囲をプレーヤに明確にするという意図もあります。通常のような細いフォークだと一瞬で通りすぎてしまうので攻撃の判定範囲がわかりづらくなってしまいます。

基本の立ち絵が一番左上のもので、幅 57pixel 高さ 61pixel でこのドット絵からフォークの角度や足の動きをつけていきました。角度をつける作業は通常のイラストだとツールの傾ける機能を用いれば簡単に実現できますが、ドット絵なので基本的に 1 ドットずつ打っています。ツールの傾斜機能を用いてしまうとディザが働いてしまいドット絵独特の明瞭感が失われキャラクタ絵としても見づらいものとなってしまいます。ドット絵は 1 ドット違うだけでも印象が変わってしまうので自動化はかなり難しいところです。

配色は極力視認しやすい色を採用しました。また縁の部分は中で塗っている色とは別の色を用いて輪郭がはっきりするように気をつけました。地味にアンチエイリアスのテクニックも用いています。今作は背景がアナログなので大丈夫だったのですが、通常のドット絵主体の背景だと背景色とキャラクタの輪郭色や塗り色がかぶってしまうことがあるため注意が必要です。今作も初めは通常のドット絵主体の背景を想定していたため、背景としてよく使われる青色や黒色は避け、白とピンクの服にしました。髪の毛に光沢反射があるのもそのためです。

ズコー絵(倒れている絵)は結構気に入っています。やられたときはこのドット絵に「むにゅ〜」というのでかなり可愛いです。

敵のドット絵についてはその時のフィーリングで適当に描いたものなので敵だとわかればいいやくらいの気持ちで描きました。その分カラフルにしておきましたが一応敵っぽくはなったみたいです。

オブジェクトの取り扱いについて

主人公で 15 枚、また敵もドット絵が 3、4 枚とキャラクタによってドット絵の数が不定です。また主人公のドット絵の場合は高さや横幅も変わるた基準となる座標も指定しなければなりません。これら一つひとつにプログラム上で固有値を与えていると非常に面倒なので全て外部テキストファイル(独自形式)に書き出しました。せっかくの C++なので画面に表示しているもはほぼ全てが共通の基底クラスから派生させています。この基底クラスからそのオブジェクトの読み込むべき画像ファイルとその数、基準となる座標や透過の有無、点滅の処理までを一括で管理しています。

このクラスによってキャラクタ絵を追加したり、新しい敵キャラを作ったときの拡張性が非常に上がりました。またそれだけではなくメモリ管理も一元化されました。このクラスは昨年作ったもので、かなり使い勝手がよかったので今年もそのまま魔改造を経て使われることになりました。

Lua について

基本的に C++で開発しましたが、敵の動きなどに Lua を用いました。Lua とはスクリプト系言語で C 言語と連携ができる便利な言語です。最近では商用ゲームでも採用例が多く、今作では試験導入という形で部分的に試してみました。

Lua を用いることでゲーム本体と敵の動きを分けることができ、コンパイルし直すことなく様々な敵の動きを試すことができます。Lua は C 言語から Lua の関数を読んだり、Lua から C の関数を読んだりすることができます。また同様に変数も双方で扱えます。よって敵の動きを Lua で記述し、ゲーム内部では Lua の関数を呼び出すことだけを記述すればいいです。また動作速度も他のスクリプト系言語に比べて非常に早く、ゲームの動作には全く支障のないレベルでした。文法などは Pascal に近いそうですが、C 言語の知識で十分に対応できました。

ステージについて

背景絵でステージの地面が描かれているため、通常のマップチップを使ったマップ配置ができないのでベクトルを用いました。ベクトルだけでは役不足があったのでマップチップとの併用です。

昨年、一昨年はマップチップのみを使ったステージでした。これは処理が簡単なのですが斜面への応用(一昨年の作品は無理な処理を加えて実現した)が効かず、色鉛筆で描くなめらかな曲線をマップチップだけを用いて表現するのは不可能でした。

ここでなめらかな曲線をいくつかの細かい直線に区切り、その直線の区切りった部分の点の座標と次の点の座標(実際には次の点の番号)を記録する方法を考えました。この方法だと色鉛筆で描く曲線をトレースすれば地面ができあがります。昨年はここまではいったのですが、肝心の地面との接触判定に苦しみ開発を断念しましたが、今作ではなんとか実現しました。

またマップチップのほうも、なめらかな曲線を補完するために最小で高さ16、幅 8pixel の小さなマップチップを用意しました。ステージによって使うマップチップの大きさを可変させています。マップチップの大きさをステージによって臨機応変に変えることで極力ファイルサイズを小さくしています。

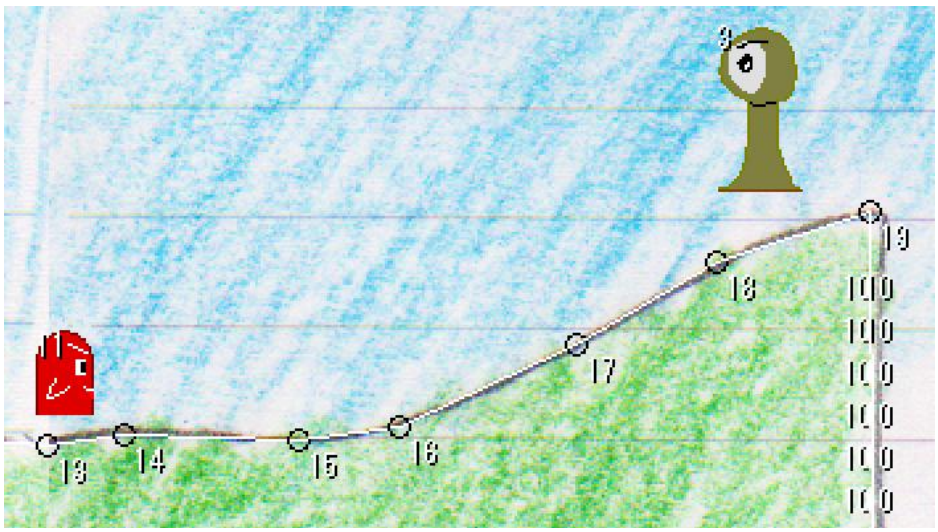


Fig 4 背景絵に沿ったラインをトレースする

マップエディタについて

Fig 5 今年度のマップエディタ

今年は作りました。初めは C#を利用して作っていたのですが、初めての C#だったのである程度形になったところでマップエディタの開発をやめてしまいました。その後ステージをベクトルで用いることになったのを機に一昨年同様、C++と DX ライブラリを用いて作り直すことにしました。今作完全対応のマップエディタになりました。

ステージは複数のファイルで構成されています。それらを自由に編集でき、背景絵の座標設定、ベクトルの地面生成、マップチップの配置、敵の配置の各レイヤーの管理がこのマップエディタでできます。このマップエディタはそれなりに汎用性があるので、後日エレ研 wiki にアップロードする予定です。

背景画像について

すべて手書きで色鉛筆によって描かれています。B5 のレポート用紙に思うがままに描き、その後スキャナで 150dpi にてスキャン、軽く色調補正とトリミングをして完成です。完成した画像データをマップエディタでつなぎ合わせて実際のゲームのステージとなります。マップエディタ側で背景画像の高さを調整できるので手書きしているときはステージの地面の高さを気にする必要はありません。しかし、背景がそのまま地面となるため、あまり無理な地形にするとジャン

プが届かなかったりするので注意が必要です。なので補助として通常のマップチップを配置し、ジャンプが届くようにしたり助走をしやすくしたりしています。

また手書きという制約上、繰り返しのものはかけないのでその辺もマップチップを利用しています。

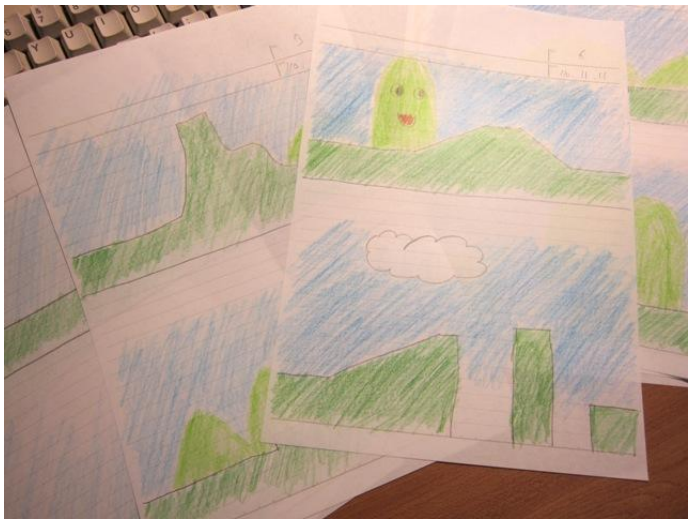


Fig 6 手書きしたマップ

おわりに

とりあえずは作りましたが、出来はどうなのでしょうか？　気分よく「今年はゲーム作るんだ」ということで始めましたが気づいたらプログラミングしていました。プログラミングをすることとゲームを作ることは似ているようで違います。フラグの管理やキャラクタの座標計算などはプログラミングです。絵を描いたりステージを作っているときがゲーム製作です。個人で作っている以上、一人でプログラミングとゲーム製作の二つをやらなければなりません。あまりにプログラミングに力を入れてもゲームは面白くなりませんし、ゲームに製作を入れてもプログラミングを疎かにしてはゲームは完成しません。ですが個人で作っているからこそ無理をしたり、途中で仕様を変えたりと自分の気分でよりよい作品を作ることができます。個人ゲーム製作の難しさと面白さがここにあると思います。

来年度は作るかどうかは未定ですが、もし作るのであればもう少し小規模なものに小綺麗にまとめたいと思います。

ランチャ

理工学部 2 年情報科学科
坂井 誠也

はじめに

ソフト班で製作したゲームを選択、起動するためのプログラムです。部員の数が増えたので、ジャンル分け、タイル表示による一覧性の向上を計っています。

製作環境

・プログラム

・Microsoft Visual C++ 2010

・DX ライブラリ 3.04b

操作説明

左クリック	ボタンを押す

工夫したところ

ジャンルを変更したり、スクロールバーをクリックした時に、全体がなめらかにスクロールするようにしました。待ち時間が長くならないように、調節するの

が大変でした。

反省点

環境によって、ゲームを遊び終わった後に、ランチャが一緒に終了してしまうようです。たぶんゲームとランチャの両方でDXライブラリを使用しているため、初期化処理で何らかの競合が発生しているものと思われます。しかし、環境によって発生するときとしない時があるので、よくわかりませんでした。

また、起動するプログラムのファイル名がgame.exeで固定であること等、制約の多いプログラムをつくってしまった事も残念です。

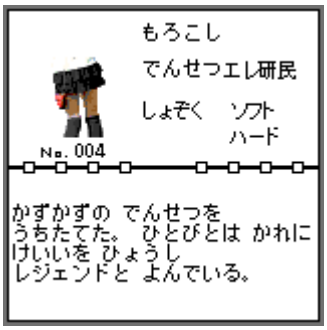
特定ディレクトリ下にある、ファイル名のリストを取得する方法がわからなかったので、list.txtにゲームの入っているフォルダ名のリストを書きこむことで対応しています。できれば、ここの部分の処理は自動化したかったです。

雑記

一年生図鑑 in エレ研

はじめに

今年はエレ研に数多くの１年が入部し我々同期の１年同士はおろか、先輩たちですら把握しきれていないと思われたため、簡単な紹介と所属班をエレ研民+その他にポ〇モン図鑑風に紹介しようぜ！というのが始まりです。



みさかな
つよき　エレ研民
しょぞく　ソフト
ハード
No. 007

どんなに ひへい していても
インなんとかさんと ビリビリの
ためなら なんどでも
よみがえるさ。

セミヌード
はんら　エレ研民
しょぞく　ソフト
No. 008

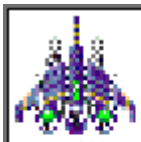
いつも はんらで であるいて
いる。そのため よく
たいちょうを くずして
どうぶつびょういんに かよう。

とうゆ
いぶくろエレ研民
しょぞく　ハード
No. 009

ハードはん として ひび
たんれんを つんでいる。
はすみの スープを のみほす
ていどの いぶくろを もつ。

ミースケ
ようせいエレ研民
しょぞく　ソフト
No. 010

あさによわいといい いちげんの
じゅぎょうを けっせき しつづけ
たんいを おとしたが はんせいは
していない。



ﾊﾟﾙﾀﾞﾛﾊﾝ
きしょうエレ研民
しょぞく　ソフト
No. 011

あまり みかけない まぼろしの
ソフトはんいん。みかけた
ときは すぐに かくほ しないと
いろいろ こまってしまう。

フジテイル
ノーマルエレ研民
しょぞく　ソフト
No. 012

どうきの なかで いっぱんてきな
じょうしきを もつひとの ひとり。
この とりでを くずされると
1ねんは ほうかいする。

ふじもん
あーじこエレ研民
しょぞく　ソフト
No. 013

ガ○ダムの はなしに なると
よく しゃべる。あいよう
きたいは しなんじゅ らしい。

クディ
サムライエレ研民
しょぞく　ソフト
No. 014

しろいおめんの サムライ。
たにんの ぶっぱを
カウンターする。あいての
ふみこみが あまいと おこる。

ウェンズ
すいようエレ研民
しょぞく　ソフト

No. 015

じぶんの いしに かんけいなく
そうじはんちょうに
にんめいされ とまどっている。
てんしちゃん まじ てんし。

ヅラ
センパイエレ研民
しょぞく　ソフト

No. 016

どんなひとに たいしても
せんぱいづらで せっしてくる。
しかし そのことにたいしては
こうかいは していない。

はやお
さんぼうエレ研民
しょぞく　ソフト
ハード

No. 017

じしょう さんぼう。しょうじき、
こじんてきには さんぼうでは
ない。ほんみょうを しっている
ものは かず すくない。

ルームヒル
となりのエレ研民
しょぞく　ソフト

No. 018

たいいくけいの サークルにも
しょぞくする スポーツマン。
いそがしくても やるべきことは
こなす。10えんは だいじ。

かんきち
カード　エレ研民
しょぞく　ソフト

No. 019

ナージャ という カードがあると
つい かってしまう。
デュエリスト。
とくぎは チョップ。

しろるり
さえないエレ研民
しょぞく　ソフト

No. 020

じしょう いけてない 1ねん。
たしかに いけてない 1ねん。
あさシャンは しない。
ほんとに いけてない。

じゅうやく
えらぶるエレ研民
しょぞく　ハード

No. 021

だいがくせいにして すでに
じゅうやくの ふうかくがある。
せいそくちは いくたから
とほ 3ふん らしい。

おわりに

　えー・・・予想以上にカオスな図鑑になってしまってエレ研民でも全部分かる人はそうそういないのでは・・・？「作った意味なくね？」「逆にわかりづらい！」との声が多々ありましたが・・・こまけぇこたぁいいんだよ！
個人的には「こんな１年がエレ研にはいるよ！」ってことがわかればいいんじゃないかなーと思いましたｗｗｗまぁ、楽しんでいただければ幸いです。

謝辞

　この図鑑を作るにあたり同期の一年達の協力なしには完成しませんでした。この場を借りて謝辞を申します。ありがとうございました。

料理から考えるプログラムのアルゴリズム及びデータ構造の設計

黒兼 百合

はじめに

タイトルを見て「料理とプログラムって何の関係があるのだ？」と思った読者は幾人かいるだろう。無論確かにそれほど近い存在ではないのだが、一点において共通するところがある。『上等なモノを作るならば経験則とセンスが必要である』。つまり日頃から食事を作らなければ手際よく、またその経験と味覚に従いアレンジを加えることができてこそ上手く料理を作ることが出来るのと同様、何千何万行ものコードを書いた経験に対し閃きを加えることで始めてスマートなプログラムが出来るのであり、一朝一夕に上等なコードなど書けたりはしないのである。これはプログラムの規模が大きくなれば大きくなるほど顕著である。なら我々プログラマースキルの向上を目指す者がすべきはまずは経験を詰み、正しい知識を身につけることであると私は考える。
とどのつまり料理にもイロハがあるようにコーディングにもイロハがあるということであり、私が書きたいのは私自身の考えるそのイロハである。
……はずなのだが、それを書くというのはどうにも基本的に無意識で行っていることを意識レベルまで引き出す果てしなく困難な作業に思える。事実私はどうにもそれをうまく文章にすることができなかった。結果これよりつづる文章にはまとまりという概念が欠落している。どうかそれを踏まえた上で読んでいただきたい。

理解・分解・再構築

どこかで見たような単語群だが別にここで先日完結した錬金術の漫画の話をしたいわけではない。プログラムの組み方の基本に対する表現方法について、キャッチーでイメージしやすいものを考えたらこれが一番しっくり来るというだけの話だ。

さて理解・分解・再構築がプログラムの組み方の基本だと前述したがそれはつまりこういうことである。

一つ、自分がコンピューターに行わせたい処理を理解する。

一つ、行わせたい処理がどういう処理の積み重ねで成っているか分解する。

一つ、その処理を自分の頭の中の概念的なものからプログラム言語へと再構築

する。

注意するがこれはそれぞれ順に行うわけではない、このプロセスを何回も繰り返していくようなものだ。ではこのプロセスをカレーの作り方を元に実際に行ってみようと思う。

カレーの作り方の理解・分解

ここでは普通のレシピの書き方のようにその作り方を処理の最初から列挙するのではなく、まずカレーという完成品から材料の用意の段階まで逆に辿ることで理解と分解について考える。

では最初に『カレーという料理はなにから出来ているか』、『煮込んだ具材に対しカレールウを加えてとろみがつくまでまた煮込んだもの』。ならば『煮込んだ具材とはどのようなものか』、『切り分けた材料を 800ml のお湯で火が通るまで煮込んだもの』。次に『切り分けた材料とはなにか』、これに関しては色々あるだろうがポピュラーなものを上げれば『微塵切りにした玉葱』、『乱切りにした人参』、『一口大に切ったジャガイモ』、こんなところだろう。更にここからそれぞれの具材の切り方や下ごしらえなどに処理は細分化されるだろうが、今回はここまでにしておく。

さてではなぜこのように処理を分解していくのか。答えは明白である。教える相手が分かるようにするためである。もしも教える相手がシチューを作ることができる人だとするならば、いささか横暴ではあるがカレーの作り方の説明など「シチューのルウをカレーのルウに変えるだけだよ」とでも言ってしまえばことが済む、なぜなら煮込んだ具材を教えられずとも用意することができるからだ。だがもし料理を始めたばかりの人に教えるならば、それこそ材料の切り方から教えなければなるまい。

コンピューターのプログラミングも同様である。コンピューターは単純な計算しか出来ないが、その代わり計算が圧倒的に速いだけなのだ。だから我々はプログラムをするとき、その処理をコンピューターが分かるレベルまでにまで細分化しなければならないわけだ。そしてまた処理を分解するためにはその処理に対する理解が必要であることも明白だ。

そしてその頭の中で理解、分解した処理をコンピューターが分かる形へと書き起こす、いわば再構築することこそがプログラミングなのである。

データ構造

ここまででプログラムのアルゴリズムの設計について語ってきたが、ならばそれに伴うデータ構造の用意はどうするのか。これも料理と同じであろう。その処理に必要だから用意するのだ。

例えばカレーを作るとき自分が材料をきる担当でもう一人材料を煮込む担当がいたとしよう。材料を切り終わったとき貴方はもう一人にどのようにしてその切った材料を渡すだろう。無論何らかの器に入れて渡すのが賢いだろう。だから器が必要となるので用意する。

プログラムでも同じである。例えばユーザーから何らかの値を入れてもらい、それを関数に渡したいとする。だから変数が必要となってくるので用意する。つまりまず行わせたい処理ありきなのであり、スマートに処理を進めるためにデータ構造を用意するのである。変数はいい加減に用意するものではない、行わせたい処理に応じて用意するものである。

ここまで語れば変数の値の変更のタイミングも分かってくるだろう。値を変えなければならないタイミングで変えるのだ。

例えば材料を包丁で切るとき、同じ材料を切り分けるときに一切り毎に包丁を洗ったりするだろうか。では別々の料理のための材料をそれぞれ切る際はどうだろう。切る材料が使われる料理が違ってくるたびに包丁を洗うだろう。これは包丁に他の料理の味がつき、一方の料理に対し別の料理の味が加わるのを防ぐ必要があるためだ。

プログラムも同様だ。例えば連続で入力された値の最大値を求めたいとき、最初に最大値を保存しておく変数は初期化をするが、入力毎にその最大値を保存している変数を初期化などしないだろう。そんなことをしたら今まで保存していた最大値が消えてしまう。だが別のデータの最大値を求める時となれば、また最大値を保存しておく領域を初期化する必要があるだろう。

データ構造はアルゴリズムありきなのである。最適なデータ構造があるからスマートなアルゴリズムが書けるのではなく、スマートなアルゴリズムを書くために最適なデータ構造を用意していくのである。つまり変数宣言など、C 言語では関数の最初に書かなければならないのだが、処理を書いてからあとで関数の頭に戻って宣言を行えばいいくらいなのである。

総論

ここまで書いてきた上で我々がプログラミングスキルを上げるためにはどうすればいいのか、その結論は明快である。コンピューターに対し行わせたい処理を自分が手計算、ただし単純な計算の積み重ねで、同様にできるようにすることである。自分が行えない処理をプログラムに起こすことなど出来うるはずはないのだ。もし自分が用意に行えない処理があるならば、それは他人の力を借りればいい、ライブラリを使えばよい。ライブラリがなければどうするか、分かる人間に助言を請えば良い。プログラミングで問題なのはカンニングすることではない、間違った処理をさせることだ。それでもダメならば、とにかく考え抜け、いずれ閃くだろう。

だから私はプログラムスキルについて毎度こういうのだ。

「とりあえずぐぐればいいと思う。」

「プログラミングスキルは６割経験で残り４割は未知の問題に対し新しく解法を０から生み出せるセンスだと思う。」

アンケートと基礎学力テストから見る近年の大学生

せれん

はじめに

近年、日本の大学は大学全入時代と呼ばれる大学教育の質の低下、定員割れなどの問題や高校のゆとり教育、入試形態の多様化による大学生のレベルの低さが新聞・インターネットなど各メディアで報道されています。そこで、東京六大学の一つでもある明治大学の学力レベルを明治大学エレクトロニクス研究部 (以下エレ研) の部会中に抜き打ちで行った基礎学力テストを用いて検証しました。また、同時に集計したアンケートから近年の大学生の傾向をまとめてみました。

~~要はエレ研馬鹿すぎﾜﾗﾀｗｗｗｗがやりたかっただけです。~~

部員の分布と傾向

部会中に行ったアンケートとテストを回答した学年とその学科を図 1 にまとめました。 (有効回答 48 人) 2010 年度エレ研では一年生 (B1) が多く入部したためテストを受けた部員も一年生がだいぶ多くなっています。今回、修士二年生 (M2) からの回答は得られませんでした。

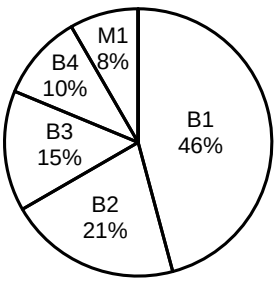


図1 部員の学年分布

図 2 は学生の学科分布になります。エレ研では例年電生 (電気電子生命学科) と情報 (情報科学科) が多く、アンケートもこの二つの学科だけで半分近く占めています。

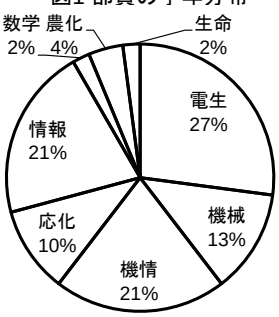


図2 部員の学科分布

図 3 では学生の所属班を集計したものです。所属班とはこの機関誌の前項にまとめられたハード班、ソフト班のことを示しています。筆者が B1 の頃は両方の班に所属する人はあまりいませんでしたが、最近では多くの子が両方の班に所属していることをここで知り、最近のエレ研民の真面目さに驚かされました。

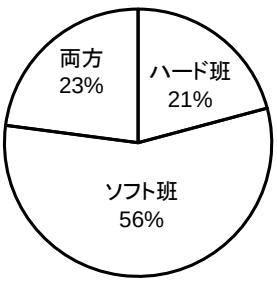


図3 部員の所属班

大学入試について

部員が入学した大学入試形態を図 4 にまとめました。筆者も AO 推薦によって明治大学に入学し、周囲の友人も推薦入試が多いことに気づいていましたが、この推薦入試の合計が一般受験を上回る結果には非常に驚きました。図 4 中の内部推薦で一番多かったのは明大中野で、次に明大明治で、明大八王子はいませんでした。

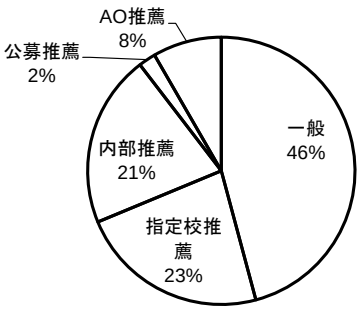


図4 入試形態

また、同時に行った『明治大学が第一志望か？』という質問に NO で答えたのは一般入試の部員が多く、所謂滑り止めと言う形で明治大学入学した人が多いと感じました。

大学生活について

ここからはアンケート中の大学生活に関する設問のまとめになります。

Q.1 アルバイトをしているか？

YES:19 人　　NO:29 人

意外にバイトをしている人が少ない・・・理系は文系に比べて忙しいことは有名で筆者も文系との格差を感じていますが、バイトをしていない人は趣味に投じるお金はどうしているんだろう・・・

Q.2 実家から通学しているか？

YES:39 人　　NO:9 人

明治大学というと地方出身者が多いというのが昔の校風でもありましたが、最近では違うようです。

Q.3 大学の講義内容が想像していたものと違っていたか？

YES:16 人　　NO:32 人

具体例として

B4:陽さん:『すげぇフリー』

B1:先輩面さん:『もっと大人数で授業を受けると思ってた』

B1:Miske さん:『語学が多すぎる』

B1:参謀さん:『高校と同じような授業形式だった』 etc….

明治大学では語学が英語+第二外国語（ドイツ語,フランス語,ロシア語,中国語）の二つが必修単位で、それに伴ってクラス分けが 40 人程度になっているため、大講堂の授業が少なく驚いた人がいたようです。

Q.4 大学生活は想像していたものと違っていたか？

YES:21 人　　NO:28 人

具体例として

B1:K さん:『忙しい』

B3:とまと 100% さん:『遊べない』

B3:T さん:『エレ研に入ると思わなかった。』

B4:ポンチョさん:『女性分が少ない』 etc…

レポートや授業で忙しく、想像していた大学生活より遊べないと感じた人が多い他、学科によって女性が数えるほどしかいないため女性が少ないと感じた人がいたようです。

Q.5 サークルに入ってよかったと感じている。

YES:48 人　　NO:0 人

この質問で NO が一人でもいたらどうしようかと。(´・ω・`)

筆者が学科や混合クラス、研究室で様々な人と話すと、サークルに所属していた人は自己主張を明確にできたり、周りと調和をとることができる人が多いと常々感じていました。せっかく大学に入ったわけですからサークルに入り友人を増やしたほうがいいのではないかとこの質問の結果を通して感じました。

基礎学力テスト

ここからは部会中に行った基礎学力テストの結果になります。

試験科目は国語・数学・社会・理科・英語の五科目、テストの内容は中学～高校レベルに設定し、25 問 25 分間の試験にて行いました。最高点は B1:あしゅらさんの 82 点、最低点は B3:tack さん、M1:Y さんの 20 点で、平均点は 45.38 点でした。図 5 は学年別の平均点をまとめたものです。ここでは思った以上に差は出ずに、平均点近い得点になりました。B3 は少し低めですが・・・。

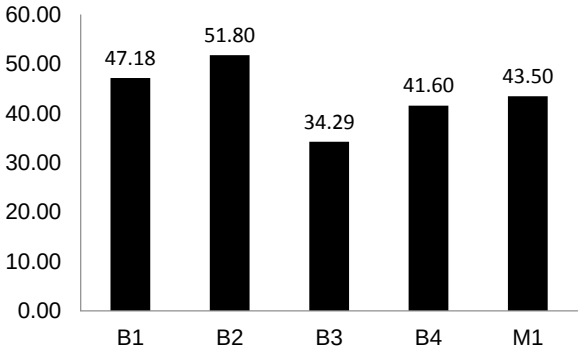


図5 平均点の学年別分布

次の図 6 は入学試験形態別の平均点の分布になります。図 5 と違いこちらでは結果がよくわかります。AHO 入試 (アホ入試) とも呼ばれる AO 入試の平均点が極端に低く、一般受験と比べ 18 点も差が開いています。

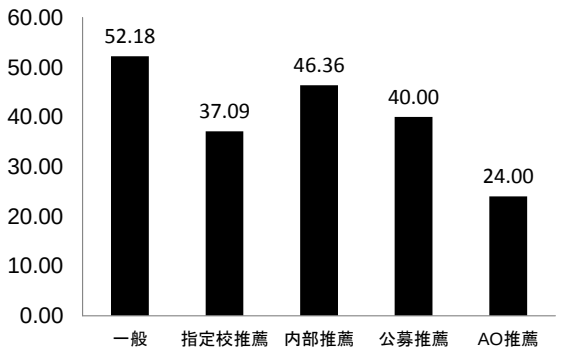


図6 入試形態別平均点分布

最後に、図 7 は科目別の平均点分布になります。どの問題も中学レベルから高校レベルですが、理系学生ということで数学と理科は少し難しめに設定しました。それでも、理科の平均点が一番高い結果となり一理系学生として安心しています。また、明治大学の学生は英語ができないとよく言われますが、この結果もそのことを示しており、点取り問題を多く設定したにも関わらず低い平均点になりました。

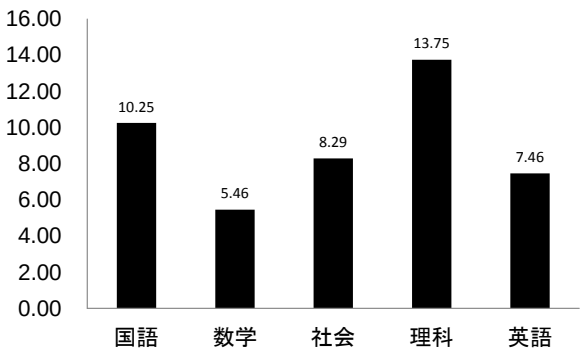


図7 科目別平均点

国語 [平均点:10.25 点,最高点:B1:I さん他 20 点(満点)]

国語は漢字の読み書き問題を 5 問出題しました。

問.1 次の下線部の漢字をひらがなに直しなさい。 正答率 95.8%

エレ研は彼を<u>侮</u>っていた。 解答：(あなどって)

問.3 次の下線部のカタカナを漢字に直しなさい。 正答率 43.8%

エレ研は<u>ユイショ</u>あるサークルだ。 解答：(由緒)

問.5 次の下線部のカタカナを漢字と送り仮名で直しなさい。 正答率 12.5%

部室の平和が<u>オビヤカサ</u>れかねない。 解答：(脅かさ)

問.1 の読み問題の正答率は 95%を超えるものでしたが、問.2 以降の書き問題に正答率が激減、誤答例としては・・・

問.3 B2:こたにさん:結緒

B2:yoshioka さん:唯緒

問.5 B2:yoshioka さん:驚やかさ

たしかにいろいろ惜しい・・・

高度情報化社会に乗ったエレ研民はパソコンや携帯電話によって漢字を書く機会が減って漢字の書きとり問題ができなくなっているように感じました。試験中にクレームがでましたが、どの問題も高校初級レベルです^o^

数学 [平均点:5.46 点,最高点:B1:kudi さん,あしゅらさん 16 点]

数学は小学生レベルの計算から、高校の三角関数などから 5 問出題。理系学生ということで少し難しめの問題にしました。

問.2 次の x を求めなさい。 正答率 31.3%

$$2.4\times\left(x-\frac{1}{4}\right)\div\frac{1}{3}=6$$

x=($\frac{13}{12}$)

問.5 次の( )内を埋めなさい。 正答率 43.8%

$$(\cos^2\alpha)=\frac{1+\cos 2\alpha}{2}$$

問.2 の分数の問題は小学校レベルェ・・・点取り問題として設定したのに、ひどい正答率・・・。問.5 の加法定理の問題は B1 の正答率が高かったのですが学年が上がるごとに正答率が激減、B4,M1 で正解者はいませんでした。加法定理は大学に入って以降ほとんど使わないため、忘れてしまった人が多いようです。誤答例としては・・・

問.2　　B1:H さん:7.5

B2:O さん:7.25

問.5　　B4:ナベさん:tan2α

B4:陽さん:cosα

問.2 の分数の問題があまりに誤答が多すぎて正解が間違っているのか何回も確認したレベルｗｗｗ。大学に入ってから関数電卓や Excel を使った計算に慣れて、分数の計算ができなくなってしまったのだろうか・・・うーん^^;;

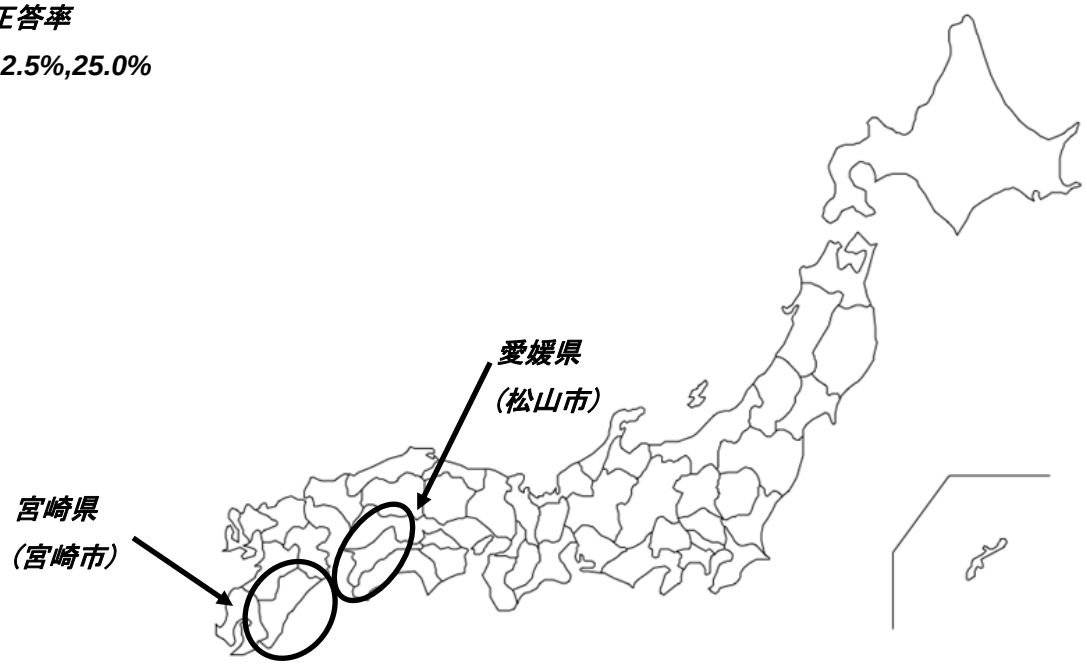
社会 [平均点:8.29 点,最高点:B2:yoshioka さん:20 点(満点)]

社会は日本の都道府県と県庁所在地、ヨーロッパの国と首都を出題しました。

問.2,3 右下の日本地図に愛媛県、宮崎県に印を付け、県庁所在地を答えなさい。

正答率

12.5%,25.0%



問.4　下の地図にイタリアに印を付け、首都を答えなさい。　正答率 41.7%

当初社会の問題は南アメリカと中東あたりから出題しようと考えていました。しかし、日頃のエレ研民の会話から日本の都道府県すら怪しいのでは・・・と思い出してみるとひどい正答率・・・。ヨーロッパのイタリアと言えば長靴型と小学生でも知っていると思っていました・。・；。誤答例としては・・・

問.2　　B2:Agate さん他:愛媛県を愛知県に○、県庁所在地を名古屋市
B2:T さん:愛媛県を奈良県に○、県庁所在地を名古屋市
問.4　　B3:とまと 100% さん他:イタリアの首都をパリ

ひどい・・・エレ研民の 12.5%が愛媛県を愛知県と勘違いしていました。愛媛県の県庁所在地を名古屋市と書く人が多くて、愛媛県に申し訳ない気持ちになりました。問.4 ではイタリアの首都をパリと答えた人が 3 人も・・・イタリアと一緒にフランスを出題していたらどうなってたんだろう・・・。
また、問.5 として出題したスペインの場所と首都を答えられた人は 2 人だけでエレ研民の教養のなさを大きく感じました。

理科 [平均点:13.75 点,最高点:B4:ナベさん他:20 点(満点)]

理科は高校レベルの物理、化学、生物から幅広く 5 題出題しましたが、だいぶ高い平均点^^。特に物理の問題の正答率は高く大学受験を物理選択した人間が多いことが影響したものと感じました。

問.1 次の問に単位を省略せずに答えなさい。 正答率 89.6%
20.0Ωの抵抗に 3.0V の電圧を加えたとき、流れる電流は？ 解答 : (0.15A)
問.3 次の( )の言葉を埋めなさい。 正答率 77.1%
元素番号 7 番は(N)で、単体分子は空気中に約(70~80)%含まれている。
問.5 次の臓器で人間が複数持っているものを全て選べ。 正答率 66.7%
(1) 胃 (2) 肺 (3) 心臓 (4) 肝臓 (5) 腎臓 (6) 脾臓 解答 : (2 と 5)

正答率としては物理＞化学＞生物の順番。もう少し難しい問題を出してもいいかなと感じたほどです。採点をしていて社会のひどさに真面目に受験していないのでは・・・と感じたのですが、どうやら真面目に受けてくれていたようです。誤答例としては・・・

問.3 B3:とまと 100%さん:元素番号 7 番= (C) 、空気中に約 (5%)
問.5 B1:Miske さん: (肺,腎臓,脾臓)
B1:NAOKI さん: (肺,肝臓,腎臓)

結構惜しい解答がちらほら。炭素の単体分子は空気中には存在しないけど・・・二酸化炭素と勘違いしたと思われる回答でした。問.5 で脾臓を選んだ人は脾臓そのものを知らない様子でした。
お酒大好きなエレ研民でも人間に肝臓は 1 個しかありません^^;;飲みすぎには注意。

英語 [平均点:7.46 点,最高点:B1:I さん:20 点(満点)]

英語は中学初級レベルの単語から高校レベルの英作問題を 5 題出題しましたが、低い平均点になりました。特に英作、並び替え問題の正答率が低かったです。

問.1 次の単語を英単語に直しなさい。 正答率 64.6%

水曜日 (Wednesday) 土曜日 (Saturday)

問.3 次の英文を和訳しなさい。 正答率 8.3%

He was not a political figure but a religious one.

解答 : (彼は政治家ではなくて宗教家だった。)

問.5 次の日本語文を英訳しなさい。 正答率 18.8%

彼は自分の生き方をサークルに合わせることができなかった。

解答 : (He could not adapt his way of life to the circle.)他

問.1 に点取り問題として置いた英単語問題の正答率が非常に悪く残念なものでした。明治の学生は英語ができないと言われていますが、水曜日を英語で書けないようじゃ言われても仕方ないかな・・・誤答例としては・・・

問.1 B1:Miske さん:水曜日 (Wensday) 土曜日 (Surturdey)

B3:もにふらさん:水曜日 (Wendday)

M1:Y さん:土曜日 (Satuday)

問.3 B1:Misake さん:(彼はまともな性格でなく異常者だ。)

B1:N さん:(彼のは贋作のフィギュアだが、それは唯一のものだ。)

問.5 B1:ゴロウさん:(He is freedom.)

問.1 のスペルミスがひどすぎるｗｗあまりにスペルミスが多くてゲシュタルト崩壊しそうになった^^;;。問.3 は N さんの回答が秀逸すぎて採点中にコーヒー噴きそうに・・figure=フィギュア・・・・うーん・・・。問.5 ゴロウさんの『He is freedom.』たしかにその通り・・大喜利なら正解です。 筆者も英語は苦手ですがもうちょっとがんばって欲しいと感じました。

おわりに

今回アンケートと基礎学力テストを通じて感じたのは大学に入学し、大学教育を学ぶうちに各学科の専攻科目の知識に勉学が集中し、一般常識である社会や漢字の知識が薄れること、受験勉強から離れる事で基礎英語力が低下しているように感じました。この原因として、ほとんどの学生が授業・レポートによって時間が奪われ、忙しいと感じていることやカリキュラム上 B3 以降に英語の授業が無いことなどが考えられます。筆者も他人の事を蔑むほど一般常識があるわけではありませんが、東京六大学の明治大学の学生として最低限の知識・教養を身につけておくべきではないかと感じました。

エレ研民の特徴として一つのことに打ち込み、集中することができる人が多いことがこのサークルの良い点と感じています。このサークルを通し、同じ志を持った友人・先輩・後輩を作り協調性を伸ばすことで、卒業後も立派な人になることを信じています。

謝辞

この自己満足とくだらない興味のために貴重な部会の時間、機関誌のスペースを提供していただいた後輩、同期、先輩方に深く感謝いたします。最後に、明治大学エレクトロニクス研究部ならびに皆様方の今後益々のご健闘、ご発展を心よりお祈り申し上げます。

2010 年 12 月 9 日　　せれん

2010 年 12 月 19 日 発行
明治大学 理科部連合会所属 エレクトロニクス研究部

幹事長 塚本 耕司
副幹事長 齊木 雅博、田辺 優
会計 岩田 純一

編集責任者 塚本 耕司

エレクトロニクス研究部 ウェブサイト
http://www.isc.meiji.ac.jp/~eleken/