

WORD

2007.11 From College of Information Science

3

特集

真ゆとり世代。

シユーテイシング

ゲーム講座

桃井はるこライブレポート

ロボコン

「はじめてのおつかい」

雙峰祭

真ゆとり世代。

ロボコン「はじめてのおつかい」... 3

桃井はるこライブレポ 10

できる睡眠 18

シューティングゲーム講座 27

通学路の風景 42

書籍紹介 52

雙峰祭

はじめてのおつかい

文 編集部 ひなたぼこ

つくばロボコン

筑波大学に通っている皆さんは、工学システム学類の筑波ロボットコンテストという授業科目をご存知でしょうか？この授業では、生徒達だけでロボットを作り、毎年雙峰祭で行われるロボットコンテストという大会で、その成果を競う授業です。

ところでこの科目、自由単位扱いなので、卒業までに自由単位を8.5単位とる必要のある情報科学類生にとっては非常に美味しい授業なのですが、**巷ではマイナス6単位と呼ばれる授業科目**だったりします。そう呼ばれる理由を知りたい方は、来年是非とも履修して崖っぷちのスリルを体験してみましょう。

さて今回の記事では、この過酷な科目を見事にやりとげた各チームの素晴らしいロボット達を紹介していきたいと思います。



ルール紹介

早速ロボットの紹介に入りたい所なのですが、ロボコンのルールは毎年異なるので、今年のルールを簡単に説明をしておきましょう。ルールは至ってシンプル…というか、シンプルに説明します。

おつかいを頼まれたマイドロボが、おつかいを済ませて無事にご主人様の元へと戻ってくる！

これが今年の基本的なルールです。実にシンプルですね。あまりのシンプルソリューションに情報科学類生もビックリ！もっと詳細なルールが知りたい方は、下記のつくばロボコンオフィシャルサイトからどうぞ。

つくばロボコンオフィシャルサイト：<http://robocon.esys.tsukuba.ac.jp>

ロボット紹介

それでは、お待ちかねのロボットの紹介に入りたいと思います。なお、各ロボットの紹介は、チーム名の昇順でさせて頂きます。

No.1 AKTTY (TSUGARU)



このチームのロボットは、一切のセンサーを使わずにフィードフォワード（※1）のみでコースの半分を走破するという偉業を成し遂げた、実際に漢なロボットです。調整の賜といったところでしょうか。

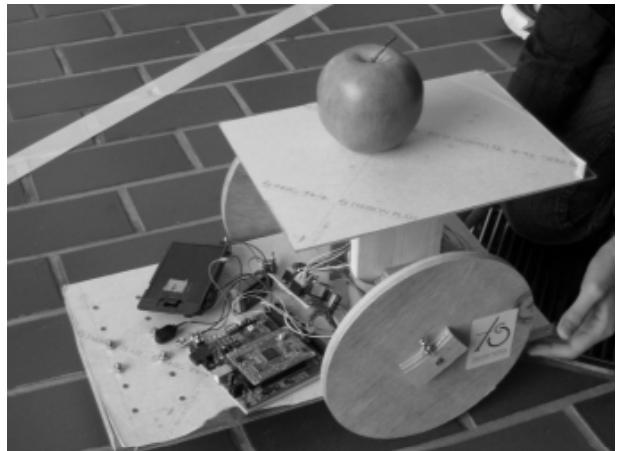
ちなみに、センサーは利用する予定だったらしいのですが、取り付ける時間が無かったとのことです。

No.2 CoD (すたへりんぐら~ど)

木製の大きな車輪が特徴のロボットです。

こちらもセンサーを利用せずにモータの回転数等の情報で位置を判定して走行するという偉業を成し遂げました。

ただ、速度が足りなかつたせいか、コースの半分を走破した時点で、時間切れになったのは非常におしかつたですね。



No.3 SRS (おさむくん)



ロボットの駆動系にキャタピラを利用し、センサーには URG（※2）とライントレース（※3）を使っているロボットです。

ギア比の関係で満足のいく速度が出なかつたのが残念でしたが、非常に安定した走行を見せてくれました。

ちなみにキャタピラは、キャタピラー社の登録商標であつて実際の名称はクローラーなんだとか。

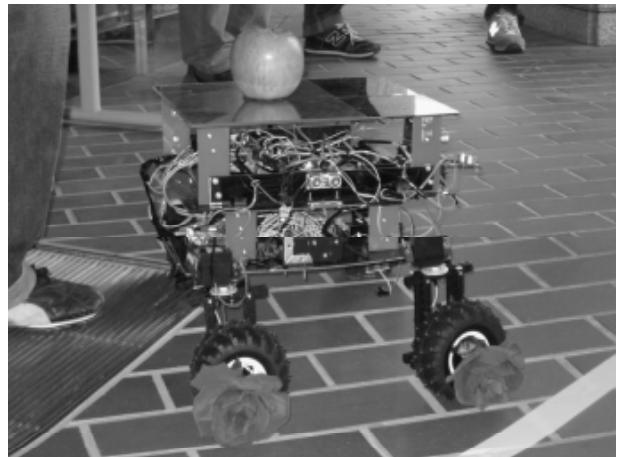
No.4 YAMADA (華子)

車輪のバラが、非常にアートな雰囲気を演出しているロボットです。

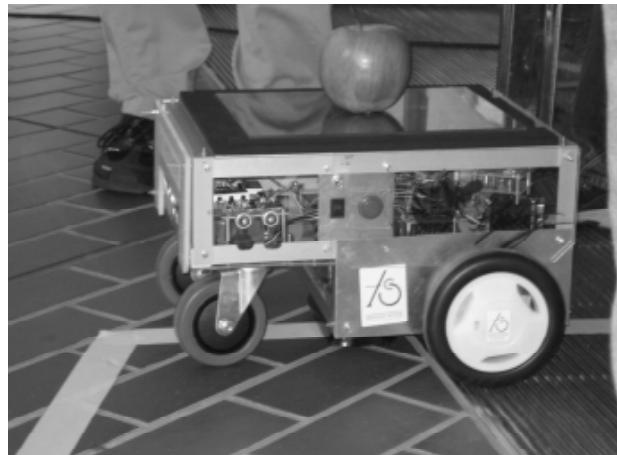
他のチームのようにライントレースは一切使わず、超音波センサー(※4)のみで障害物を検出して走行する作戦だったようです。

しかし、流石に超音波センサーだけでは厳しかったのか、残念ながら完走は出来ませんでした。

本番にてスロープから I can fly し そうな勢いで突き進む姿がとても印象的でしたね。



No.5 オレンヂ (YAMATO)



超音波センサー等の比較的簡単なセンサー類で構成されたロボットです。

シンプルな構成なので、ノートPCが無くともマイコンのみで制御できる所が凄いですね。

ただ、速度が若干遅かったせいか、折り返し地点付近まで走行した時点での時間切れになってしまったのは非常に残念でした。

No.6 かんぴょうキムチゴウヤ入り (かキゴーり)

今年のロボコンで優勝を勝ち取ったロボットです。

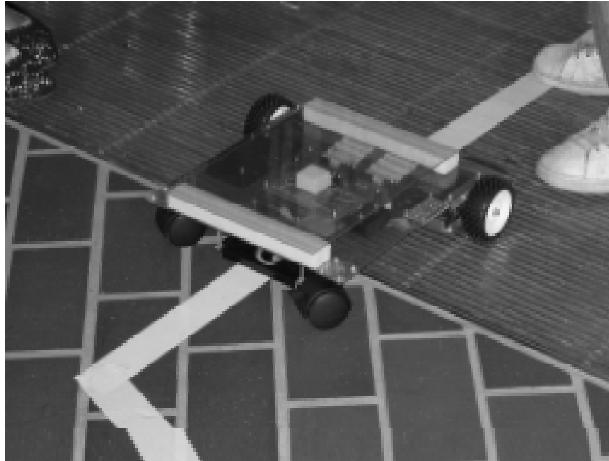
URG やライントレース等を駆使して走行していました。

走行速度は、おそらくロボコン中トップクラスと言えるほどの爆速で、しかもコースを完走したという実に優秀なロボットです。

ただ、速度が速すぎてリンゴが落ちるというアクシデントも…。



No.7 キングポスト (図下んく 2号)



超音波センサーの実装が間に合わず、今回はライントレースのみでの走行とのことでしたが、非常に安定した走行を見せてくれました。

アクリル板の下にはスポンジが入っており、リンゴに優しい仕様になっているようです。

No.8 傀儡廻 (Jameson)

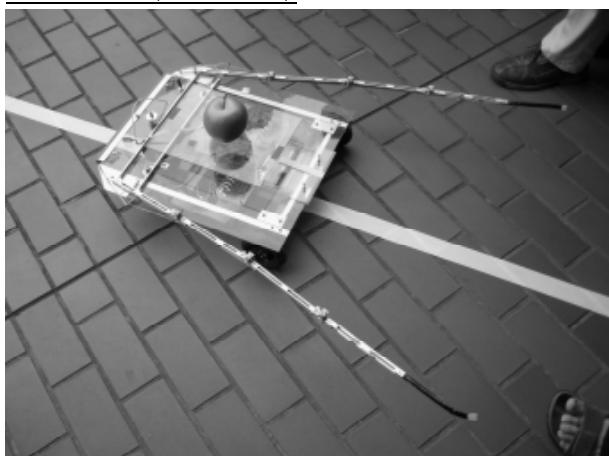
チーム名は「くぐつまわし」と読みます。

大会当日は、まだ製作途中だったため動かすことが出来ず、姿のみのお披露目となりました。

グランドファイナル(※5)では、走行が出来るそうなので、その実力を発揮できるとよいですね。



No.9 なまズ (黒い宝石G)



このロボットは、障害物が腕に当たるとセンサーが反応して、障害物を避けながら走行します。

しかし、センサーの問題が解決出来ず、本番はフィードフォワードのみの走行となってしまいました。

筆者もとても期待していたロボットだけに、非常に残念でした。

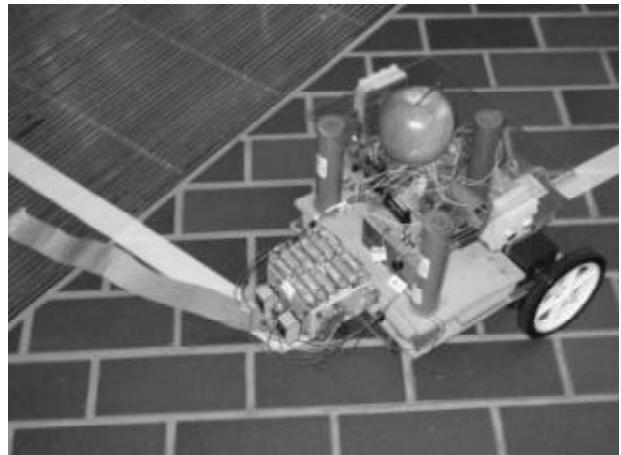
グランドファイナルでの活躍に期待したいところです。

No.10 ねこまつたりら (μ -MAX 1216)

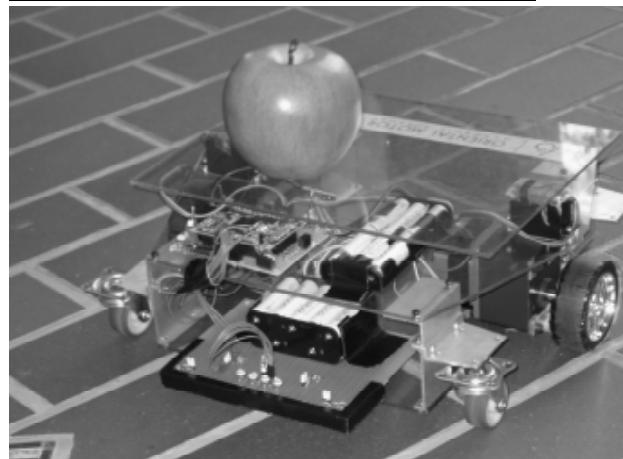
ロボットの走行開始位置は、写真(右写真参照)でいうところの、金網の部分にロボットの一部が含まれていることが必須条件になってます。

このロボットは、そのルールを最大限活用して、最初のコーナーを回避しているのが印象的でしたね。

加速音が常磐線のアレっぽいと思ったのは筆者だけではないはず。



No.11 らっきー☆スターズ (チョココロネ 1号)



チーム名を見た瞬間にこれは！と思った方も多いはず。

それはともかく、ライントレースのみでスロープを走行し、その後の移動はフィードフォワードのみで走行してました。

この作戦は、ロボット製作前から考えていたらしく、非常に挑戦心溢れるチームでしたね。

No.12 校庭ペンギン (Trombe)

「皇帝」ではなく「校庭」ここは非常に重要なポイントです。

URG やラインセンサー等を駆使して走行していました。

URG でコーンを認識しつつ走行するので、走行スピードも速く、大きく期待していたのですが、フィードフォワード走行の調節がいまひとつだったようです。



No.13 夢戦走 (わたるくん)



チーム名は「むせんらん」と読みます。

画像処理、ライントレース、方位センサー（※6）を駆使して走行していました。

処理系のCPUにはCore2Duoを搭載している（※7）らしい…なんという高性能ロボット。

画像処理が日光の影響でうまく働かなかったのが、非常残念でなりませんでした。

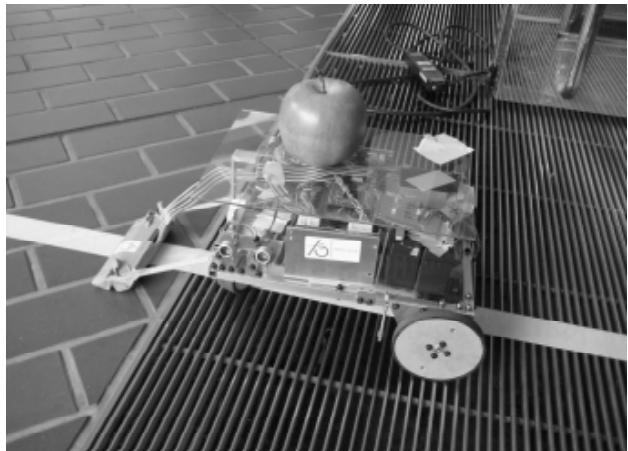
No.14 林檎酢 (オリゴ糖)

チーム名は「りんごす」と読みます。

ラインセンサーと超音波センサーを駆使し走行していました。

車輪にキャスターを使ったためか、L棟の凸凹地面では、カタカタと音をたてながら走行していた姿が印象的でした。

折り返し地点まで無事に走行した数少ないロボットです。



※1 ロボットにあらかじめフィールド情報を与えることで、センサーを利用せずに走行する方法。これを利用するとセンサーを利用せずに走行可能なので、タイムを大幅に縮めることが出来る。しかし、微妙な調節がうまくいかず、非情にもコーンに激突したチームが続出した。

※2 レーザー光により、半円状のフィールドをスキャンし、検出物との距離計測とそのステップ角度により座標を計算し、設定されたエリア内の障害物を検知するセンサーのこと。今年のロボコンでは、北陽電機様にスポンサーになって頂き、センサーを提供して頂きました。

※3 超音波を検出媒体とした非接触検出センサのこと。検出物体で音波が遮断された場合や、検出物体の表面で反射して戻ってくるまでの時間を測定して情報を得ることが出来る。

まとめ

今年のロボコンに出場した各チームの素晴らしいロボット達を簡単に見てきましたが、いかがでしょうか？この紹介記事を読んで、「自分も来年はロボコンを履修するぜ！」と騙されたと思ってくれた方がいたのなら幸いです。

※5 つくばカピオで行われる最終決戦（グランドファイナル）のこと。優勝者は米袋 10kg、お酒、旬の野菜が贈呈される。

※6 地磁気の微弱な方向性を検知して現在の方角を得るセンサーのこと。

※7 ロボット本体に Core2Duo が乗っていたわけではなく、制御していたノート PC に搭載されていたという意味。

ヲタ的雙峰祭

文 編集部 永遠の3年生をまもなく脱出する Niit

桃井はるこが学園祭に来るよ

という情報を入手した瞬間、心臓が止まるかと思った。
マジでやばい。超やばい。
筑波の学園祭では外部から人を呼んでのイベントはなかなか行われない。
声優イベントなんて論外。3年前に田中真弓さんの講演会＆アテレコ体験が行われたということはあったが、このジャンルでの学園祭イベントは東大や電通大、早稲田といった大学からは大きく遅れていたのだ。
しかしここでいきなり桃井はるこか！！！
これは高まらないわけがない。

桃井はるこって誰よ

桃井はるこ。通称モーイ。
一言で言うと声優で歌手でライターでラジオパーソナリティー（全然一言ではない）。
そしてゲームマニア。

それだけでは語れない

彼女はアイドルが大好き。「おニャン子クラブ」などのアイドルファンとして育ち、伝説の地下アイドル^{*1}「水野あおい^{*2}」の熱烈なファンとして地下アイドル界隈に出入りするうちに自分も地下アイドルからスタートし、ライターや声優などとして成功し、ユニット「UNDER17^{*3}」の作詞、作曲、萌え担当として萌えソング界の女王として登り詰めたすごい人なのです。
そんな多彩な桃井はるこさんの半生を描いたドラマ「はるこ UP ☆ DATE」が制作されたりもし

*1 地下アイドル：いわゆるメジャーなアイドル（たとえばモーニング娘。）ではなく、都内の小さなライブハウスや公共施設（会場代が安くすむ。大森スポーツセンターが現在の聖地として有名）などでライブを行ったりするマニア向けのアイドルのこと。ノリとしては80年代の雰囲気をそのまま現代に受け継いでいる。地下アイドルは時にメジャーなアイドルへと成長する場合がある。「場合がある」と言うことはだいたいそうはないということである。後述する水野あおい全盛期は地下アイドルライブの客は都内全域で300人程度居たと言われているが、現在は150人ちょっとである、とはかつての地下アイドル事務所の社長談。なお、AKB48は商業的に計画されたメジャーアイドルであり、決して地下アイドルのジャンルには含まない。いや含んではいけない。

*2 水野あおい：2000年に引退した地下アイドル。アルテミスという地下アイドル事務所に所属し、フリフリのファッションで業界の一時代を築いた。なお、オリジナル曲がカラオケUGAで配信されている。来春武道館ライブが決定している大人気アイドル声優田村ゆかり(31)も水野あおいの大ファンであり、ファッションなど大きな影響を受けているのは知る人ぞ知る話。

*3 UNDER17：桃井はること小池雅也のユニット。「美少女ゲームソングに愛を」を合い言葉に多数のゲーム曲、アニメ曲などを制作し一時代を築いた。2004年、音楽性の違いにより解散。小池氏は現在「MOSAIC.WAV」や織姫よぞらなどA-POP界の音楽に関わるプロデュース業をしている。

ています。ちなみに見てマジ泣きました。

ちなみに過去に行った桃井関係のイベント

- ・2003年夏ナースウィッチ小麦ちゃんまじかるて^{*1} むぎむぎ団決起集会（東京ビッグサイト）
抽選イベントでしたが無事当選。小麦コスのモーイが「愛のメディスン」を歌った。もう死んでも良いと思った。「こむぎ気分でロックンロール」でアニメ中のライブ状況が完璧に再現されてスタッフの人が腹を抱えて笑っていた。素晴らしいライブだった。
- ・2004年春 Broccoli The Live 3（横浜アリーナ）
このときはいろいろ偶然により横浜アリーナの最前列センターというあり得ない席をゲット。
近すぎてむしろ疲れた。
- ・2006年春 DreamParty 春（東京ビッグサイト）
UNDER17解散後初のライブ。色々な想いが心を巡る。

実はあまりライブは行っていないのだ。ライブDVDはたぶん全部持っているが。

歌詞の魅力

モモイの曲の多くは本人が作詞しています。ファンというモノはタレントのいろいろな魅力を見つけて好きになるモノですが、私としては特に歌詞が魅力的。

桃井はるこ、UNDER17の楽曲は特にポップで電波な感じの部分の歌詞が注目されがちなのですが、それよりも同じ過去を抱えた、同じ思いを共有している人たちに強く支持されています。「心からいいと思ってるモノを ^{わら} 曝われた時のショックを覚えてる」（さいごのろっく）でマジ泣きした私が何か。

そんな桃井はるこが筑波大学学園祭に来るというのだからもう大変（俺だけ）・

実は私は秋葉原界隈でヲタク系バンドをやらかしているのですが、畏れ多くも桃井はるこさんの曲を演ったりもしています。また、最近流行の「パンオフ^{*2}」形態の「モモイパンオフ」なるイベントにも顔を出し、燃えたぎるモモイへの気持ちをぶちまけまくっていました。ちなみにこの翌々日にもライブを控え、「うしろ指さされ組^{*3}」を最後の最後の曲としてやる予定でした。

*1 ナースウィッチ小麦ちゃんまじかるて：アニメ「魂狩～soul taker～」からのスピノアウト作品。モモイが中原小麦役を演じ、大ブレイクのきっかけとなった。そう言えばパチンコだかパチスロだかにもなっていた気がする。

*2 パンオフ：メンバーが集まらなくてバンドが組めない、あの曲がやりたいんだけどパートが足りない、などの人たちが楽器を持ち寄って集まり、セッションをしてしまおうというオフ会。なんだか最近流行気味。ステージ上の人や周りと一切関わらずにイベントを楽しめるライブと比べて、他の参加者とのコミュニケーションがあるため引きこもりチックな人にはかなりハードルが高い（と私は勝手に思っている）

*3 うしろ指さされ組：オリジナルはおニャン子クラブから2人組で構成された「うしろ指さされ組」が歌う「うしろ指さされ組」（ごちゃごちゃしてきた）。テレビアニメ「ハイスクール奇面組」のオープニングとして有名。と思ったがもしかしたら今の世代の人には判らないのかもしれない。とにかく、有名な昔のアイドル曲を最近モモイがカバーした曲である。

テンションを上げて行く前準備として最適。というか盛り上がらないわけがない。

チケット予約が始まったのを知ったのは申し込み開始より少し後だったので大慌てでそこら辺の人に任せて最大応募可能枚数5枚×3セット予約。

... とはいえたが、大学会館の収容人数は約 1000 人。冷静に考えたらそんな簡単に満席にはなりません。

見事に私の手には **15枚の入場券が届いてしまいました**w
まあそんなのは私の声優ヲタコネクションにより一瞬で配布完了^{*1}。

そんなわけで当日

池袋のメイド系喫茶^{*2}で知り合ったはずが、実は家が超絶近所だった^{*3}友人T氏とともにとりあえずのメイド喫茶に行ってみる。旧第2学群喫茶を借りて喫茶店営業形態にしているようです。ああそういう手があるのか。しかしながら人が並んで入れない。どうも開店準備に手間取っている模様。まあメイド喫茶には良くあることです^{*4}。とりあえず並んでみたものの、別の友人（同じくメイド喫茶系）との合流時間が近づいてしまい後ろ髪引かれながら泣く泣く撤収。列案内していたメイドさんが割と本気だったので全体的に良い物が出来てたんじゃないでしょうかと推測することしかできないのが非常に悔しい！！！！^{*5}

友人と合流し適当にぶらぶらしながらビール投入。
都内から召還した 10 人ぐらいと待ち合わせたのですが、ものの見事に全員迷子になりました。
まあそりやそうか。

会場入り。当日券も販売され結構人が集まっている模様。会場内で学内の知り合いに色々声を掛けられる。ちなみに席は前から 3 列目左ブロック。まあまあ良い席。
5 席 ×3 ブロック買ったので 3 列目だったのは友人 5 人。残りの 10 枚は 9 列目。
まあ席はこちらの独断と偏見で決めさせていただきましたが（笑）。入場列を事前に形成していた手際もあり、リハ押しで開場が押したにも関わらずオンタイムで開演しました。

*1 転売禁止との事だったので律儀に 1 円ももらっていない。まあ折角なので、筑波大の無駄な広さを味わってみては良いんじゃないかという地元愛の結晶ですよ。あれ俺つくば地元じゃないや。

*2 池袋のメイド喫茶：2004 年 8 月 5 日にオープンした某店。2006 年 11 月に chown するまで推定 50 回以上通った。筑波から池袋に。馬鹿だろ俺。しかしあがえのないモノを得た。人を信じない心とか。

*3 「つくばに住んでるんですよー」『えっ私もつくばなんですよ』「どの辺ですか」『春 4 です』『かすよん！？！？』的な流れ。まさか池袋のそう言う店で同じ丁目に住んでる人に会うとはw

*4 メイド喫茶には良くあることです：開店当日に数時間前から並んでみたものの、手際が悪くてなかなかオープンしないなんてことは本物のメイド喫茶業界でもよくあることです。

*5 前述の通り翌々日は自分のライブ、その前日は練習ということでこの日のこの時間しか私は学園祭を満喫できなかったのです。

今回のライブでは学園祭とはいえ、結構本気の人たちが集まっていたため、特攻服やペンライトの実装度は割と高め。

声優アイドルライブのアイテムとして欠かせないのがいわゆるペンライト。

市場占有度の都合上「サイリューム(CYLUME)」と呼ばれることが多いですが、これは米国オムニグロー社の商品名です。アーティストによってイメージカラーが決まっていることが多く、周りに合わせるのが通という物です。(モモーイの場合特にこれといって決まってはいません)。1本 100 円程度から入手可能ですが、100 円ショップで売っているモノは発光が弱く、持続時間が短い。ドンキホーテで 147 円で売っている「ルミカライト」ブランドのモノがおすすめ。「イベンター」と呼ばれる屈強の戦士達はこれとは違い、電池式のネオン管を使用する事が多いです。彼らはこれをいつも鞄の中に持ち歩いています。しかも数色。一度のライブで使い切るともったいないからですね。電池の値段を考えてもそのうち元が取れます。っていうか元が取れるぐらいライブを行っています。

しかし問題となるのが耐久性。ネオン管を使用しているタイプは衝撃に弱く、暴れている最中にどこかにぶつけてしまったりすることがあります。そのため私は基本的にケミカルライトを使用しています。こちらは多少手荒に扱っても大丈夫なのです。

そんなケミカルライトの中で「ウルトラオレンジ」と呼ばれる秘密兵器があります。これは発光量が半端じゃない。照明レベル。影を作ることが出来ます。元々軍事用に開発されたもので、2km 先からその光を確認できると言うもの。誰が見つけてきたのやら。

しかし、通常のケミカルライトのスペック上の持続時間は 8 時間程度ですが、ウルトラオレンジの場合、2 ~ 3 分で発光力は半減、5 分程度で完全に沈黙します。一曲持たないw。そのためライブ中にキタ-と高まった瞬間に使います。また、曲によっては会場中で一斉に発光させて一気に会場を染め上げるという観客側からの演出の材料としても使われます。UNDER17 のライブで「もっと夢見よう」と言う曲の間奏で一斉に折るという使われ方をしたことでこの商品は一気に知名度を上げました。不用意な発光は周りの迷惑ともなります。節度を守らない空気読めない腐れヲタが使いまくることで揉め事が起こったり、照明効果の妨げになるという理由で持ち込み禁止の処置が取られているライブもありますので注意しましょう。(Lantis/RAMS 系の一部(JAM Project、野川さくら、クローバーなどのライブ))。しかし同じ Lantis 系でも新谷良子ライブでは「ray of sunshine」で合わせて使うという前例があるため OK になってるはず。要するに前例を作った者勝ちなわけですが。

因みにウルトラオレンジは値段もやばい。140 円程度のルミカライトに対し、こちらは 350 円ほどします。まあえらいひとが 1000 本単位でまとめ買いして再配布するのでその方向に知り合いがいれば比較的安く手に入れることもできます。それでも 300 円近いですが、100 本買うとなればだいぶ違ってきます(ツッコミ禁止)。



電池式ネオン。¥900
円程度で入手可能

ケミカルライトを発光させるときは棒の両端を持って軽く曲げます。中身は二重構造になっていて、曲げることで内側のガラスアンプルが折れて 2 液が混合されて発光する仕組みになっています。超余談ですが、私はおもむろにポケットから取り出し、その辺(椅子や壁)に軽く当てて発光させるのが好きです。ほら無駄にジッパーをズボンで擦ってつける人居るじゃないですか。ああいうのw

話がそれました。つまりなんだか無駄にアツい訳です。

ライブレポ

無駄とか言うな。

そんなこんなで開演。

現視研（OB）の2人が登場して注意事項その他前説。

数日前までドイツのアニメ祭りみたいなものに行っていたらしく、帰国後初のイベントということとで「おかえりなさ~い！」で呼び込み。

「声が小さいみたいですねー」。形式美^{*1}。

トークコーナー

数年前、テレビアニメ「げんしけん」放送前にラジオで放送されていた「桃井はるこの現代視覚文化研究会・略してラジオげんしけん」を再現してしまおう、というコンセプト。しかしまんまやると大人の事情に引っかかってしまうのでそれっぽく、というお話。

ラジオ形式でリスナー（設定）からの質問に答える形で進行。司会の人が良い感じにぐだぐだです（ほめ言葉）。「お城みたいな所にファイナルファンタジーのコスプレの人がいてすぐ一格好良かった」とか。総括として、「オタクは世界どこに行っても同じ、ヲタに国境はない！！」との事です。

あとは「モーイが現視研にいたら何をしてますか？」の話で「男装して潜り込む」的な話で。男の友情を味わいたいそうです。

「あと学園祭に呼んでくれて有り難う。来年また呼んで。呼んでくれないと来れないから」との事。

よろしくお願ひします>現視研の偉い人

そんなこんなでトークコーナー終わり。

司会の二人が捌けてモーイのターン！！！！

ライブコーナー

桃井のライブといえば口上。口上といえば桃井。

口上について簡単に説明すると、曲の前奏や間奏中に観客が一斉に台詞を叫ぶことです。たとえば世にも有名な「いちご Go!Go!」という曲の場合

「萌え萌えソングに魅せられて はるこのとりこになりました

一期一会は一度きり みんなで叫ぼう Go!Go!Go!」^{*2}

とか。知らない人には何が起きたのか全く判りませんね。

*1 形式美：「おやくそく」とも言う。テレフォンショッキングのお友達紹介の「えー」だと思えばいい。ちなみにこの手のライブでは「最後の曲です」「えー」だけでは終わらず、誰か1人が「いまきたばっかり！」と叫ぶのがもっと美しいとされます。

その他の形式美：「井上喜久子 17歳です！」「おいおい！」

*2 現在流通版の口上。因みに結成直後には「僕等を縛って離さない 甘いモモイの歌声で 美少女ゲームの 枠こえた 輝く未来へ GoGo!Go!!」と言う版があつたらしい。

これとは別にいわゆるコールという物があります。これは A メロでフレーズが切れるたびに「はるこー」とかいうコールを入れることです。昔のアイドルライブのノリがこの界隈では脈々と受け継がれているのです。

あとは PPPH¹。これは「ぱん・ぱ・ぱん・ひゅー」の略で、いわゆる B メロの時に曲に合わせて 3 回手を叩き、最後のひゅーで右手を突き上げ力の限りジャンプします。この PPPH がヲタクライブ観客の戦闘力を示していると遡っても過言ではありません（過言です）。

因みにこういったモノは「コール表」と呼ばれる紙媒体で周知されていました。最近はネット環境の発達により紙ベースは減ってきているんじゃないかなと思います。

そんなアツくてキモイ感じのライブパートが始まる。

一曲目「くじびきアンバランス」キター。

UNDER17 名義での曲です。まあこの曲をやらないわけはないですよね。

テレビアニメ「げんしけん」の劇中アニメ「くじびきアンバランス」のオープニング曲として作られました。「げんしけん」の 1 話はいきなりこの劇中アニメのオープニングが流れる（オタクの主人公がテレビを見ている）ところから始まるという演出が行われました。私も放送当時は本気でチャンネルを間違えたのかと思いました。この曲のサビにはモーイの曲のほとんどはライブで一緒に楽しめる最優先に作られていてお客さんも一緒に叫ぶことで最高の一体感を得られるようになっています。

余談ですが、「げんしけん」本編のオープニングテーマ「マイペース大王」はあの「日本ブレイク工業社歌」で一世を風靡した萬 Z さんの中の人 manzo さんが歌っています。その繋がりでトーキョーナー中に manzo さんからのメッセージも紹介されました。manzo さんも結構色々なアニメ関係の曲に関わったりしています。

というわけで一曲目から会場の熱気がUP。

二曲目「天罰！エンジェルラビィ」。

ちょっとWまじでwww

UNDER17 の曲の中でもかなり上位に位置する人気曲。同名ゲーム、OVA の主題歌。

定番曲ではあるのだが正直この曲がいきなりここで来るとは割と想定外。会場に来ていた屈強のモモイスト達も驚きを隠せずあわててUOを取り出したり絶叫したりして会場内はカオス。

この曲もとにかくアツい。前奏から「天罰！」で袈裟切りの振り付けや大ジャンプなど一般人どん引きの絵図がいきなり繰り広げられます。そして最強の口上

「エンジェルはるこを見るために徹夜割り込み当たり前
そんなダメダメ僕達に天使の審判下してよ」

あまりに急すぎて微妙に間違えました。モーイの曲は盛り上がるため、アキバ系地下アイドルライブで良く歌われます。このとき歌い手さんや状況に応じて臨機応変に口上を変化させなければなりません。私もこれまでにこの口上を一体何人に向けて放ったか覚えていませんが、あまりにも久しぶりに本人を見たので噛みまくりました。この曲のおかげで会場の空気がさらにヒートアップ。そして運動不足気味の俺の脚大ピンチ。

*1 最近微妙にテレビとかで「ヲタ芸」として取り上げられることがあります、これは我々の定義するところの「ヲタ芸」には含みません。口上も同様。

間髪入れずに三曲目「**God knows...**」。

大ブームになった京都アニメーション制作のテレビアニメ「涼宮ハルヒの憂鬱」の作品中でヒロイン達が学園祭でバンド演奏をした曲。元々はハルヒ役の平野綾が歌っているのですが、先日発売されたカバーアルバムに桃井はるこ版が収録されています。

「学園祭と言えばこの曲でしょう」との事です。

やはり発狂者続出（笑）そして次々に消費されていくウルトラオレンジ。

四曲目「**Let me love you**」。

ここでぐっと雰囲気が変わってしっとり系。

社会問題的アニメ（ということにしたいのですね？）SchoolDaysの曲。

まあ私は見てないのであんまり反応できませんでしたがw

歌い終わったところで

「**Nice boat. ! !**」

ちょ w w w

関係した社会現象的な話を色々。

「仁侠と書いて『にんぎよ』と読むきん！」*1の決めゼリフで

五曲目「**Romantic summer**」。

つい最近まで放送されていたアニメ「瀬戸の花嫁」のオープニングです。

まあこれも来るだろなと思っていました。しかしアニメを全然見ていない罷。

モーイはこのアニメのヒロイン、瀬戸燐を演じています。オリジナルではこの曲はモモーイと戸前留奈役、野川さくらさんがユニット「SUN & LUNAR」として歌っています。今日は勿論一人なのでソロバージョンです。こういうのが聞けるのもライブの醍醐味。

ここで本人曰く「不適切な発言をします」。あーなんか予想ついた。

「任侠と書いてにんぎょう（人形）と読む！」とのことで（当たったw）

六曲目「ルミカ」。

10月から始まった新番組「**Kawaii! JeNny**」～かわいい ジェニー～」のOP曲。サンダーバードチックな（生まれてないのでよく知りませんが）構成で、ジェニ一人形が巨大化して敵と戦うアニメです。ちなみにこの曲はイベント初披露とのこと。これはレア。

ちなみに「ルミカ」とは前述したケミカルライトのうちの一つの商品名「ルミカライト」のことです。モーイは過去に「かがやきサイリューム」という曲も書いています。如何にライブやその関連文化を大事にしているかがよくわかりますね。

此処までの曲のようにコールや振り付けが決まっている曲ではなく完全に初披露の曲です。しかし屈強のモモイスト達は1番のうちに曲の構成を把握し振り付けを覚え、2番の時には既に手拍子もジャンプも手の振りも揃ってしまうというイリュージョン。前の方の席の人たち（含む俺）が率先して色々やるので後ろの方までサイリウムの動きが揃ってきれいに。危うくBメロでPPPHに持って行かれそうになりましたがここは冷静に裏1で。サイリウムできちんと誘導すれば揃う

*1 アニメの中でのヒロインの決めゼリフ

んですよ。会場のお客さんの気持ちが揃っているからこそできる技です。

ライブ終了

ライブ始まる前の予想ではトーク 20 分、ライブ 40 分、セットリストには最新曲、旧定番曲、くじ引きアンバランス、RomanticSummer、あと 1 曲、もしかしたらアンコールとの予想を立てていました。アンコールはイベント運営の都合上完全に時間きっちりで終了のため無し（来年度また無事に呼べるように！！とのことでした）。曲構成はだいたい正解かな（笑）

そんなわけで今年の学園祭は無事終了しました。いや私としてはもはやこれが学園祭の全てでした。偉そうなことを言ってみますがいろんなイベント渡り歩いた私から見て非常に良いイベントだったと思います。やっと筑波の学園祭でもこういうイベントができるようになったんですね。

大学に居るうちに T X は開通するわモーは来るわで長生きはするもんです。

2001 年に入学した時には想像もつきませんでしたね。

…あれ俺今何年生だっけ。

まあそんなことはおいといて。

是非現視研の皆様は東大やら電通大やらに負けない声優イベントをこれからもどっかんどっかんやっちやってください。陰ながらこっそり応援しております。

できる！すいみん

できる！すいみん

文 編集部 いのひろ

こんばんは。いきなりですが、読者の皆さんはちゃんと寝てますか？研究が大切なもわかりますが、ちゃんと寝た方がいろいろ良いことがあるのです。今回は、大学で寝る様々な方法を提案します。



メモリー食い尽くして眠くなってしまった mitty さん（編集部）

寝ないより寝た方がよい

「その言葉、そっくりそのままお前に返すよ」と言われそうですが^{*1}、やっぱり寝ないより寝た方がいろいろ良いのです！米ハーバード大学発行のニュースレター「Harvard Women's Health Watch」は6つの睡眠を取るべき理由を挙げています。

- ・記憶のため：睡眠は学習と記憶を助ける。眠っている間に、脳は「記憶の固定」と呼ばれるプロセスで新しい情報をとどめている。
- ・スリムでいるため：慢性的な睡眠の欠如は、炭水化物を代謝し蓄積する過程に影響を与え、食欲をコントロールするホルモン値を変え、体重増加を招きかねない。

*1 かく言う筆者は、かれこれ1週間以上自分の家で寝てません。。。

- ・安全のため：夜十分に睡眠がとれないと、日中眠くなりやすく、転倒や交通事故、仕事上でのミスのリスクを増加させる。
- ・幸福のため：睡眠の欠如は、かんしゃく、短気、集中力の欠如、不機嫌を招く。また自分が好きな活動をするときに、疲れすぎていることになる。
- ・心臓のため：重度の睡眠障害と高血圧、ストレスホルモン値の増加、不規則な心拍との関係が研究からわかっている。
- ・強くあるため：睡眠の欠乏は免疫系を低下させる。

引用元：よく眠ることが重要な理由（Yahoo! ヘルスケア）

<http://health.yahoo.co.jp/news/detail?idx0=w30060102>

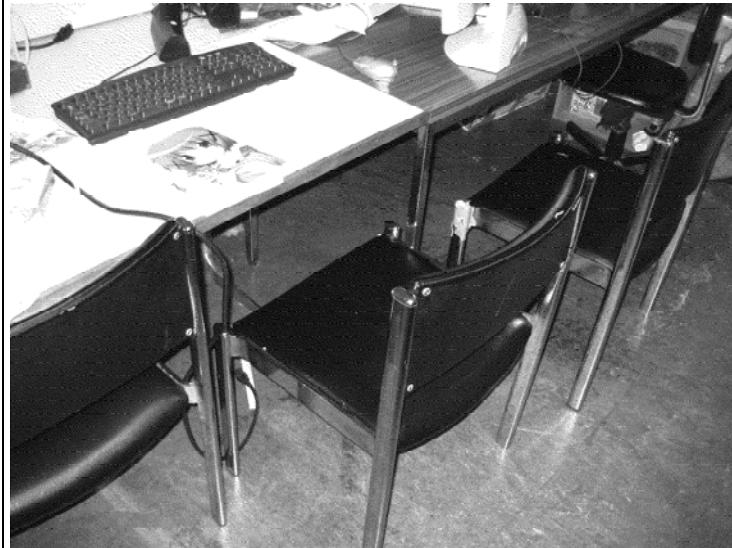
これらの項目を見れば、睡眠の重要性が分かると思います。しかし我々情報学群類・情報科学類生は研究者！開発者！（もしくはその卵）。明日のレポート書いてねー！バイトの納期は明日ー！寝てる暇なんか無いー！単位も来ないー！という方が大半を占めていると考えられます。

そこで今回は「WORD 発、大学（研究室）で寝る方法」を提案します。大学で仮眠を取る一つの方法として、今後参考にしていただければ幸いです。是非！

すいみんアイテム

よく寝るためにそれはそれなりの環境を整えなければなりません。あなたの快適な睡眠をサポートする「すいみんアイテム」を、「お手軽性」「推測される快眠性」などの点から評価し、「顧客満足度」を数値化してみました。なおこの数値化は筆者の独断と偏見によります。

椅子（某編集室）



写真は某編集室の椅子です。足にキャスターはありません。どこにでもある椅子と同じような椅子です。

一般的な椅子だと寝方によってかなり快眠性は変化しそうです。

お手軽性：★★★★★

快眠性：★★★☆☆

リッチな椅子（某編集室）



写真は某編集室にある"リッチ"な椅子です。足にキャスターがついてますが、1つ壊れているのでバランスを崩すと大変危険です。

リクライニング機能があります。

お手軽性：★★☆☆☆

快眠性：★★★★☆

社長性：★★★★★

寝袋



写真は表と裏がファスナーでくっつけられていて、分割できるタイプの寝袋です。寝袋に入らなくても、お腹が冷えないようにすることができます。

これならどこでも寝れそう。

お手軽性：★★★★☆

快眠性：★★★★☆

携帯性：★★★★★

ソファーベッド（某研究室）



ただのベッドぼく見えますが、背もたれを起こすとソファーになります。これは便利！毛布がなかなか良い。

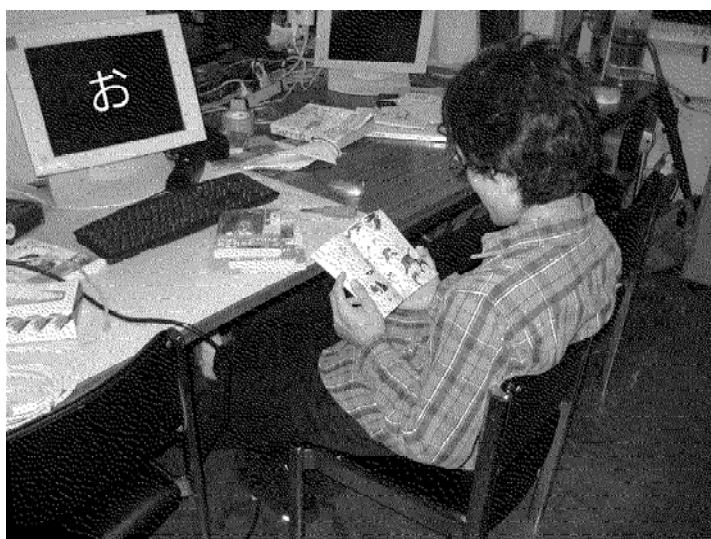
お手軽性：★★☆☆☆

快眠性：★★★★★

実際に寝てみた

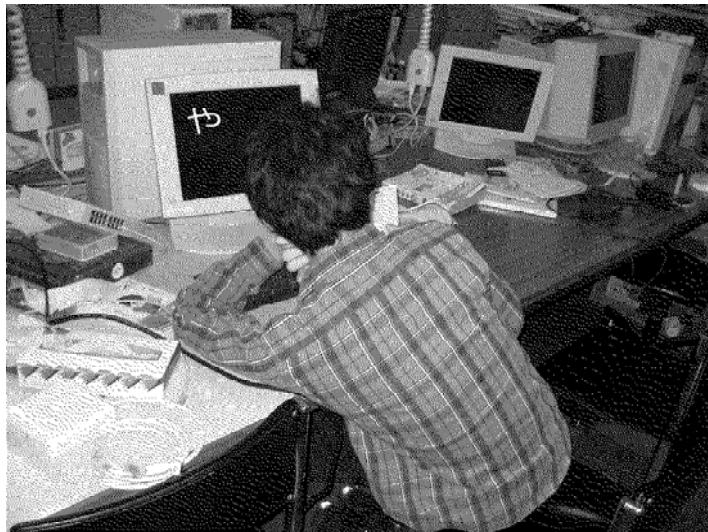
環境を整えたら、後は寝るだけ。実際の快眠度と感想。快眠度は例のごとく筆者の独断と偏見です。

漫画を読んでたら眠くなっちゃった mitty さん

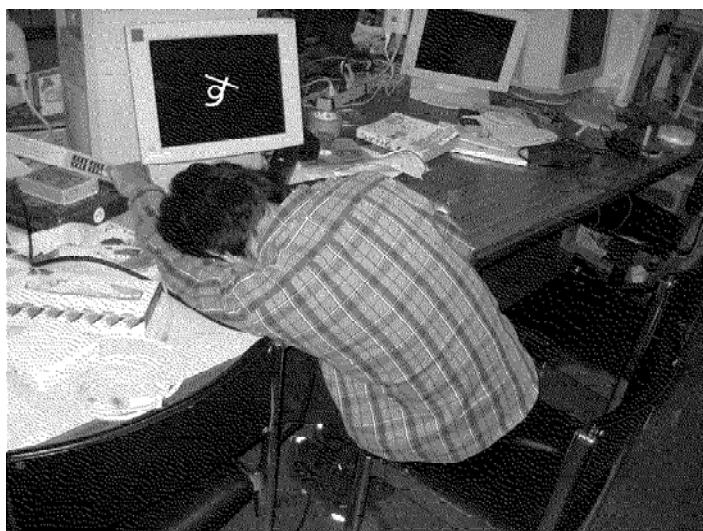


某編集室で漫画を閲覧していた mitty さん（編集部）。

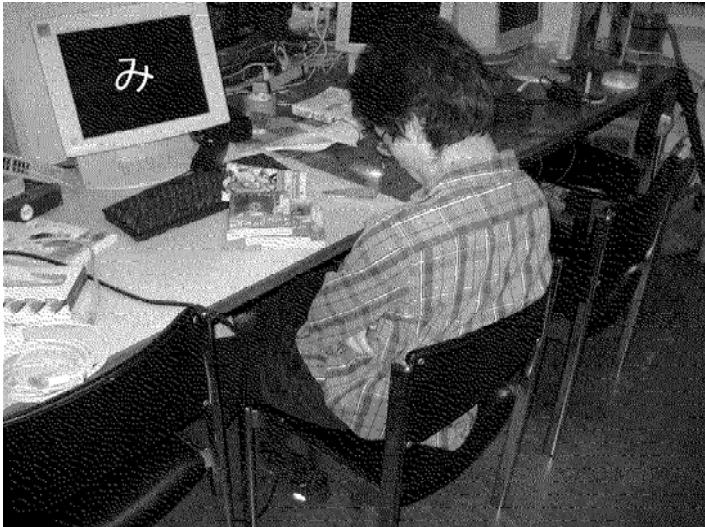
できる！すいみん



何を読んでいるのでしょうか。。。
あれ。。。？眠くなってきたのか？？



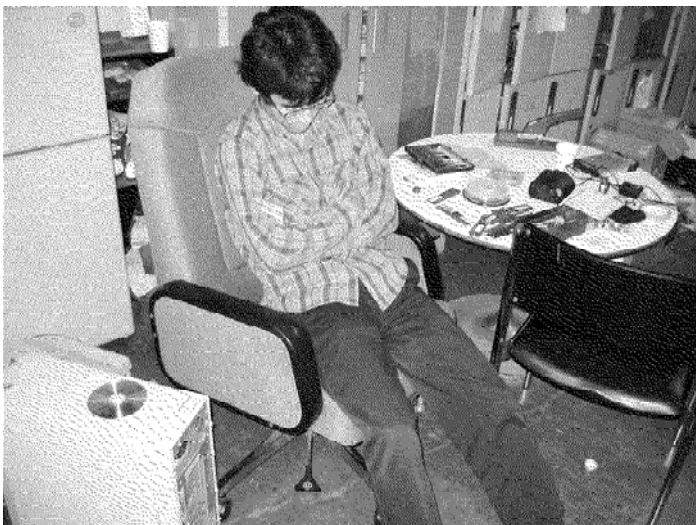
あつ。。。
mitty さーん、起きて一



腕を組んで寝てます。
漫画は完全に諦めたみたいですね。

快眠性：★★☆☆☆

授業に疲れた mitty さん



実はこの"リッチ"な椅子、あまり寝心地はよくありません。
(経験者は語る)

快眠性：★★☆☆☆

研究に疲れたひなたちゃん



注) ひなたちゃん
牛久市、18歳、男性

完全にお休みのようです。寝袋は結構防寒対策がされているので、暖かく寝ることができます。

快眠性：★★★★☆

研究に疲れたひなたちゃん



ひなたちゃん疲れすぎです。そろそろ自宅に帰って寝た方が良いです。

後日、この写真を見て。。。
本人「なんかセクシー」

快眠性：★★★☆☆
セクシー性：★*n
(n → ∞)

研究に疲れたひなたちゃん



今度はソファーベッドで完全にお休みのようです。研究室で寝るんだつたらやっぱりこういう形が一番良いかもしません。

快眠性：★★★★★
予想通りの満点。

トイレに行きたいんだけど寝すぎて寝ちゃった mitty さん



なんというか、まずは寝る前にトイレに行きたいでですよね。行っておかないと、夜、一人でトイレ行くの怖いもんね。

意味が分からない。

快眠性：★★☆☆☆☆☆
トイレ：★★★★★

っと、まーこんな感じで寝てる人を激写して、勝手に快眠性を数値化してみたのですが、すべて屋内ですよね。気分転換したいときは青空の下でお昼寝も良いかもしれません。

できる！すいみん

省スペース、省コストな全く新しいベッド

最後に、この記事を書く2日前に発見した全く新しく、極めて斬新なベッドをお見せいたします（じつは私、ここで初めて写真に出てきます）。



二段ベッド！！！！

椅子を3つ並べただけなのに（省コスト）、2人寝ることができます（省スペース）。いまならこの二段ベッドが1万9800円！お支払い分割手数料はジャ●ネッ●た●たが負担します。

ということで、大学でお休みになるなら、2段ベッドがコスト的にもスペース的にも非常に有効です。是非！

Special Thanks（撮影にご協力いただきました）

mittyさん（WORD編集部）

ひなたぼこさん（WORD編集部）

シューティングゲーム開発講座

文 編集部 yasuharu

目次

1. はじめに <ul style="list-style-type: none"> 1. 開発環境、開発言語 2. 目標 3. 検証環境 	7. プレイヤーの弾の表示 <ul style="list-style-type: none"> 1. 弾の作成 2. 弾の移動
2. 環境の設定	8. 敵とプレイヤーの弾との衝突判定 <ul style="list-style-type: none"> 1. 衝突判定
3. Window の表示	9. 敵の弾の表示 <ul style="list-style-type: none"> 1. 弾の表示 2. 敵の行動 3. 敵の弾とプレイヤーとの衝突判定
4. プレイヤーの機体の表示 <ul style="list-style-type: none"> 1. Player 構造体の定義 2. プレイヤーの機体の表示 	10. 最終章 <ul style="list-style-type: none"> 1. 敵が上から流れてくるようにする 2. プレイヤー、敵、弾を画像を使って表示する 3. 画面のちらつきの制御をする 4. 背景色を変更する 5. プレイヤーの弾に制限をかける
5. プレイヤーの機体の移動 <ul style="list-style-type: none"> 1. タイマーの設定 2. キーの取得 	
6. 敵の機体の表示 <ul style="list-style-type: none"> 1. 敵のチェインを作る 2. 敵の作成 3. 敵の機体の表示 	

(10 章の内容は予定です)

前回までの記事については Web 上に掲載しておりますので、そちらを参照してください。

シューディングゲーム開発講座 第 1 回

<http://yasuharu.net/word/shooting/1/>

シューディングゲーム開発講座 第 2 回

<http://yasuharu.net/word/shooting/2/>

説明のために前回までのソースコードを載せておきます。後からの説明で使用します。42 行目から 108 行目については、今回の説明では必要ないので紙面の都合上掲載していません。申し訳ありませんが、前回までの記事を参照するか Web 上の記事を参照してください。

```

1. #include <windows.h>
2. #define _USE_MATH_DEFINES
3. #include <math.h>
4. #include <time.h>
5.
6. #define CLASS_NAME "Sample"
7. #define TITLE_NAME "Shooting Game"
8. #define TIMER_ID 10000
9. #define MOVE_POINT 5
10. #define MOVE_POINT_NANAME (MOVE_POINT * M_SQRT1_2)
11. #define MAX_ENEMY 100
12. #define ENEMY_SPEED 4

```

```

13.
14. typedef struct WindowInfo
15. {
16.     int Height;
17.     int Width;
18. } WindowInfo;
19.
20. typedef struct Player
21. {
22.     POINT Position;
23.     int Width;
24.     int Height;
25.     int HitPoint;
26.     HDC hImage;
27.     WindowInfo mWindowInfo;
28. } Player;
29.
30. typedef struct Enemy
31. {
32.     POINT Position;
33.     WindowInfo mWindowInfo;
34.     int Width;
35.     int Height;
36.     int Destroy;
37.     HDC hImage;
38.     double Speed;
39.     Player *mPlayer;
40. } Enemy;
41.
42.             /* 紙面の都合上省略 */
43.
44. void SetXToPlayer(Player *mPlayer,int value)
45. {
46.     if(0 <= value && value < mPlayer->mWindowInfo.Width - mPlayer->Width)
47.     {
48.         mPlayer->Position.x = value;
49.     }
50.
51. }
52.
53. void SetYToPlayer(Player *mPlayer,int value)
54. {
55.     if(0 <= value && value < mPlayer->mWindowInfo.Height - mPlayer->Height)
56.     {
57.         mPlayer->Position.y = value;
58.     }
59.
60. }
61.
62. Enemy* CreateEnemy(POINT Position,double Speed,WindowInfo mWindowInfo,Player *mPlayer)
63. {
64.     Enemy *mEnemy;
65.
66.     mEnemy = (Enemy *) (malloc(sizeof(Enemy)));
67.     mEnemy->Destroy = 0;

```

```

132.     mEnemy->Position = Position;
133.     mEnemy->mWindowInfo = mWindowInfo;
134.     mEnemy->Height = 16;
135.     mEnemy->Width = 16;
136.     mEnemy->mPlayer = mPlayer;
137.     mEnemy->Speed = Speed;
138.
139.     return mEnemy;
140. }
141.
142. void DestroyEnemy(Enemy *mEnemy)
143. {
144.     free(mEnemy);
145. }
146.
147. void PlayerMove(Player *mPlayer)
148. {
149.     int Key_Left = 0, Key_Right = 0, Key_Up = 0, Key_Down = 0;
150.
151.     Key_Left = GetKeyState(VK_LEFT);
152.     Key_Right = GetKeyState(VK_RIGHT);
153.     Key_Up = GetKeyState(VK_UP);
154.     Key_Down = GetKeyState(VK_DOWN);
155.
156.     if(Key_Down < 0 && Key_Right < 0)
157.     {
158.         SetXToPlayer(mPlayer, mPlayer->Position.x + MOVE_POINT_NANAME);
159.         SetYToPlayer(mPlayer, mPlayer->Position.y + MOVE_POINT_NANAME);
160.         return;
161.     }
162.     if(Key_Down < 0 && Key_Left < 0)
163.     {
164.         SetXToPlayer(mPlayer, mPlayer->Position.x - MOVE_POINT_NANAME);
165.         SetYToPlayer(mPlayer, mPlayer->Position.y + MOVE_POINT_NANAME);
166.         return;
167.     }
168.     if(Key_Up < 0 && Key_Right < 0)
169.     {
170.         SetXToPlayer(mPlayer, mPlayer->Position.x + MOVE_POINT_NANAME);
171.         SetYToPlayer(mPlayer, mPlayer->Position.y - MOVE_POINT_NANAME);
172.         return;
173.     }
174.     if(Key_Up < 0 && Key_Left < 0)
175.     {
176.         SetXToPlayer(mPlayer, mPlayer->Position.x - MOVE_POINT_NANAME);
177.         SetYToPlayer(mPlayer, mPlayer->Position.y - MOVE_POINT_NANAME);
178.         return;
179.     }
180.
181.     if(Key_Left < 0)
182.     {
183.         SetXToPlayer(mPlayer, mPlayer->Position.x - MOVE_POINT);
184.         return;

```

```

185.      }
186.      if(Key_Right < 0)
187.      {
188.          SetXToPlayer (mPlayer,mPlayer->Position.x + MOVE_POINT);
189.          return;
190.      }
191.      if(Key_Up < 0)
192.      {
193.          SetYToPlayer (mPlayer,mPlayer->Position.y - MOVE_POINT);
194.          return;
195.      }
196.      if(Key_Down < 0)
197.      {
198.          SetYToPlayer (mPlayer,mPlayer->Position.y + MOVE_POINT);
199.          return;
200.      }
201. }
202.
203. LRESULT CALLBACK WindowProc (HWND hWnd,UINT msg,WPARAM wp,LPARAM lp)
204. {
205.     PAINTSTRUCT ps;
206.     HDC hdc;
207.     RECT rect;
208.     static Player mPlayer;
209.     static WindowInfo mWindowInfo;
210.     static Enemy *mEnemy[MAX_ENEMY];
211.     int i;
212.
213.     switch (msg)
214.     {
215.     case WM_CREATE:
216.         GetClientRect (hWnd,&rect);
217.         mWindowInfo.Width = rect.right;
218.         mWindowInfo.Height = rect.bottom;
219.
220.         mPlayer.mWindowInfo = mWindowInfo;
221.         mPlayer.Position.x = 200;
222.         mPlayer.Position.y = 400;
223.         mPlayer.Width = 16;
224.         mPlayer.Height = 16;
225.         mPlayer.HitPoint = 10;
226.
227.         for(i = 0 ; i < MAX_ENEMY ; i++)
228.         {
229.             mEnemy[i] = NULL;
230.         }
231.
232.         srand((unsigned int)time(NULL));
233.
234.         SetTimer (hWnd,TIMER_ID,1000 / 60,NULL);
235.         break;
236.     case WM_PAINT:
237.         hdc = BeginPaint (hWnd,&ps);

```

```

238.     TextOut (hdc,mPlayer.Position.x,mPlayer.Position.y,"■",2);
239.
240.     for(i = 0 ; i < MAX_ENEMY ; i++)
241.     {
242.         if(mEnemy[i] != NULL)
243.         {
244.             TextOut (hdc,mEnemy[i]->Position.x,mEnemy[i]->Position.y,"▼",2);
245.         }
246.     }
247.
248.     EndPaint (hWnd,&ps);
249.
250.     break;
251.
252. case WM_TIMER:
253.     if( (int)wp == TIMER_ID)
254.     {
255.         PlayerMove (&mPlayer);
256.
257.         if((rand() % 60) == 0)
258.         {
259.             for(i = 0 ; i < MAX_ENEMY ; i++)
260.             {
261.                 if(mEnemy[i] == NULL)
262.                 {
263.                     POINT Position;
264.
265.                     Position.y = rand() % mWindowInfo.Height;
266.                     Position.x = rand() % mWindowInfo.Width;
267.
268.                     mEnemy[i] = CreateEnemy (Position,
269.                                         ENEMY_SPEED,mWindowInfo,&mPlayer);
270.                     break;
271.                 }
272.             }
273.
274.             for(i = 0 ; i < MAX_ENEMY ; i++)
275.             {
276.                 if(mEnemy[i] != NULL)
277.                 {
278.                     if(mEnemy[i]->Destroy == 1)
279.                     {
280.                         DestroyEnemy (mEnemy[i]);
281.                         mEnemy[i] = NULL;
282.                     }
283.                 }
284.             }
285.
286.             InvalidateRect (hWnd,NULL,TRUE);
287.         }
288.         break;
289.

```

```

290.     case WM_CLOSE:
291.         for(i = 0 ; i < MAX_ENEMY ; i++)
292.         {
293.             if(mEnemy[i] != NULL)
294.             {
295.                 DestroyEnemy(mEnemy[i]);
296.                 mEnemy[i] = NULL;
297.             }
298.         }
299.
300.         KillTimer(hWnd,TIMER_ID);
301.         DestroyWindow(hWnd);
302.         break;
303.     case WM_DESTROY:
304.         PostQuitMessage(0);
305.         break;
306.     }
307.
308.     return DefWindowProc(hWnd,msg,wp,lp);
309. }
```

7. プレイヤーの弾の表示

1. 弾の作成

この章では、プレイヤーの弾の表示ができるようにしていきます。プレイヤーの弾の表示は、前回行った敵の表示と同じようにして作成します。この章で説明することの多くは、以前に説明したことと同じなので少し省略しながら説明します。

主な流れとしては、構造体を作って、チェインを作って、表示できるようにします。まず、プレイヤーの弾をチェインで管理するために弾の構造体を定義しましょう。名前を Ball 構造体とします。名前が PlayerBall 構造体など、プレイヤーの弾であることを示すような名前となっていないのは、プレイヤーの弾と敵の弾とで同じ構造体を使うためです。両方の弾で異なるのは表示される絵と移動方向だけなので同じ構造体にします。ただ、プログラムの規模によっては、別々の構造体を定義したほうが楽にすむ場合もあります。

弾の構造体のメンバーは、プレイヤーや敵の構造体とほとんど変わりません。ただ、弾はどの方向に飛んでいくのかという情報が必要なので、ベクトルを使って表現します。ベクトルを表現するための構造体を Vector 構造体として定義してあります。コード 1 を見てください。

コード 1 (P.28 40 行目の後に追加)

```

typedef struct Vector
{
    double X;           // ベクトルの X 成分
    double Y;           // ベクトルの Y 成分
} Vector;

typedef struct Ball
{
    POINT Position;      // 座標
    WindowInfo mWindowInfo; // ウィンドウ情報
    int Width;           // 弾の幅
}
```

```

int Height;           // 弾の高さ
int Destroy;         // 破壊フラグ
Vector Direction;   // 移動方向
double Speed;        // 弾の移動スピード

HDC hImage;          // 弾の画像
} Ball;
}

```

Vector 構造体での移動方向の指定は、先に物体ごとに軸を決めておく必要があります。弾の構造体の移動方向は、物体を中心として X 軸正を右方向、Y 軸正を上方向とします。残りの部分は、以前説明したものと同じなので簡単なコメントだけつけておきます。

次に、弾を作成する関数と破壊する関数を作ります。CreateBall 関数でメモリの動的確保を行い、パラメータの設定を行います。DestroyBall 関数では、動的生成したメモリの解放を行います。これらの関数もプレイヤーや敵を作成したときと同じようなコードになります。

コード 2 (P.30 201 行目の後に追加)

```

Ball* CreateBall(POINT Position, Vector Direction,
                  double Speed, HDC hImage, WindowInfo mWindowInfo)
{
    Ball *mBall;

    mBall = (Ball *) (malloc (sizeof(Ball)));
    mBall->Destroy = 0;
    mBall->Position = Position;
    mBall->mWindowInfo = mWindowInfo;
    mBall->Height = 16;
    mBall->Width = 16;
    mBall->Direction = Direction;
    mBall->hImage = hImage;
    mBall->Speed = Speed;

    return mBall;
}

void DestroyBall(Ball *mBall)
{
    free (mBall);
}

```

コード 2 を追加したら、プレイヤーの弾を管理するためのチェインを作成します。同時にチェインを操作する部分を追加していきます。

コード 3 (P.27 12 行目の後に追加)

```

#define MAX_PLAYER_BALL 100
#define PLAYER_BALL_SPEED 4

```

コード 4 (P.30 210 行目の後に追加)

```
static Ball *mPlayerBall[MAX_PLAYER_BALL];
```

コード 5 (P.30 230 行目の後に追加)

```
for (i = 0 ; i < MAX_PLAYER_BALL ; i++)
{
}
```

```

        mPlayerBall[i] = NULL;
    }

コード 6 (P.31 284 行目の後に追加)
if(GetKeyState(VK_SPACE) < 0)
{
    for(i = 0 ; i < MAX_PLAYER_BALL ; i++)
    {
        if(mPlayerBall[i] == NULL)
        {
            Vector v;
            POINT p;

            v.X = 0;
            v.Y = 1;

            p = mPlayer.Position;
            p.y -= mPlayer.Height;

            mPlayerBall[i] = CreateBall(p,v,PLAYER_BALL_SPEED,
                NULL,mWindowInfo);

            break;
        }
    }
}

for(i = 0 ; i < MAX_PLAYER_BALL ; i++)
{
    if(mPlayerBall[i] != NULL)
    {

        if(mPlayerBall[i]->Destroy == 1)
        {
            DestroyBall(mPlayerBall[i]);
            mPlayerBall[i] = NULL;
        }
    }
}

```

```

コード 7 (P.32 298 行目の後に追加)
for(i = 0 ; i < MAX_PLAYER_BALL ; i++)
{
    if(mPlayerBall[i] != NULL)
    {
        DestroyBall(mPlayerBall[i]);
    }
}

```

コード 3～コード 5、コード 7 の部分については敵のチェインを作成するときとほとんど同じです。

コード 6 を見てください。ここでは、スペースキーを押したときにプレイヤーの弾を作成する処理を行っています。

まず、この中で `Vector` 構造体の変数 `v` について説明します。ベクトルの軸を先ほど決めた

ので、それに合わせるようにして移動方向を設定します。プレイヤーの弾は、画面上方向に飛んでいく必要があるので、ベクトルを(0,1)に設定しています。

次に、弾の初期位置についてですが、プレイヤーの正面になるような位置を設定します。今回はプレイヤーの弾とプレイヤーの大きさが同じなので正面になります。しかし、大きさが違う場合にはこのままでは、ずれてしまうため計算を行って正面に調節する必要があります。

最後に CreateBall 関数を使って弾を作成しています。

2. 弾の移動

ここでは、弾の移動と表示を行います。まず、弾の移動を行うために弾を操作するための関数を作成します。コード 8 を見てください。

```
コード 8 (P.33 コード 2 の後に追加)
void SetXToBall (Ball *mBall,int value)
{
    if(0 <= value && value < mBall->mWindowInfo.Width - mBall->Width)
    {
        mBall->Position.x = value;
    } else {
        mBall->Position.x = value;
        mBall->Destroy = 1;
    }
}
void SetYToBall (Ball *mBall,int value)
{
    if(0 <= value && value < mBall->mWindowInfo.Height - mBall->Height)
    {
        mBall->Position.y = value;
    } else {
        mBall->Position.y = value;
        mBall->Destroy = 1;
    }
}
```

大まかには、この関数もプレイヤーの時と同じです。プレイヤーの場合と異なり、弾の場合は画面外に出たら Destroy フラグを 1 (フラグが 0 の時が通常状態、1 の時が破壊された状態) にして破壊します。

これで、弾の作成、削除、移動ができるようになりました。残っているのは、弾の移動をコントロールする部分と描画する部分です。一気に見ていきましょう。

```
コード 9 (P.33 コード 1 の後に追加)
double GetUnitVectorXFromVector (Vector v)
{
    return v.X / sqrt((pow(v.X,2) + pow(v.Y,2)));
}

double GetUnitVectorYFromVector (Vector v)
{
```

```

    return v.Y / sqrt((pow(v.X,2) + pow(v.Y,2)));
}

コード 10 (P.35 コード 8 の後に追加)
Ball* RunBall(Ball* mBall)
{
    SetXToBall(mBall,mBall->Position.x + mBall->Speed *
        GetUnitVectorXFromVector(mBall->Direction));
    SetYToBall(mBall,mBall->Position.y - mBall->Speed *
        GetUnitVectorYFromVector(mBall->Direction));

    return mBall;
}

コード 11 (P.31 246 行目の後に追加)
for(i = 0 ; i < MAX_PLAYER_BALL ; i++)
{
    if(mPlayerBall[i] != NULL)
    {
        TextOut(hdc,mPlayerBall[i]->Position.x,mPlayerBall[i]->Position.y,"●",2);
    }
}

for(i = 0 ; i < MAX_PLAYER_BALL ; i++)
{
    if(mPlayerBall[i] != NULL)
    {

コード 12 (P.34 コード 6 の「if(mPlayerBall[i]->Destroy == 1)」の前に追加)
mPlayerBall[i] = RunBall(mPlayerBall[i]);
if(mPlayerBall[i]->Destroy == 1)
{
    DestroyBall(mPlayerBall[i]);
}

```

上から説明していきます。

まず、コード 9 の部分では、単位ベクトルを計算する関数を定義しています。これは移動方向を計算するのに使います。詳しくは後から説明をします。

次に、コード 10 の部分では、弾の移動のコントロールを行っています。RunBall 内に弾の移動に関する処理を書いておきます。X 座標は、移動方向の単位ベクトルの X 成分と弾の速度の積を加算しています。それに対して、Y 座標は、単位ベクトルの Y 成分と弾の速度の積を減算しています。これは、Windows が使用している描画の軸と Vector の軸が違っているためです。X 軸は同じなのですが、Y 軸の方が軸が逆なので符号を反転させています。この関数は、コード 12 の部分で呼び出されています。ちなみに、この関数の中身を書き換えると弾の軌道が変わるので、例えばワープする弾など突拍子もない弾が作れます。

最後に、コード 11 の部分です。ここが描画を行う部分です。これもプレイヤーの描画の時などと同じです。

以上で、プレイヤーの弾の作成が終わりました。試しに、F5 キーを押して、コンパイルと実行をしてみましょう。成功していたら、スペースキーでプレイヤーから弾が発射するようになっています。

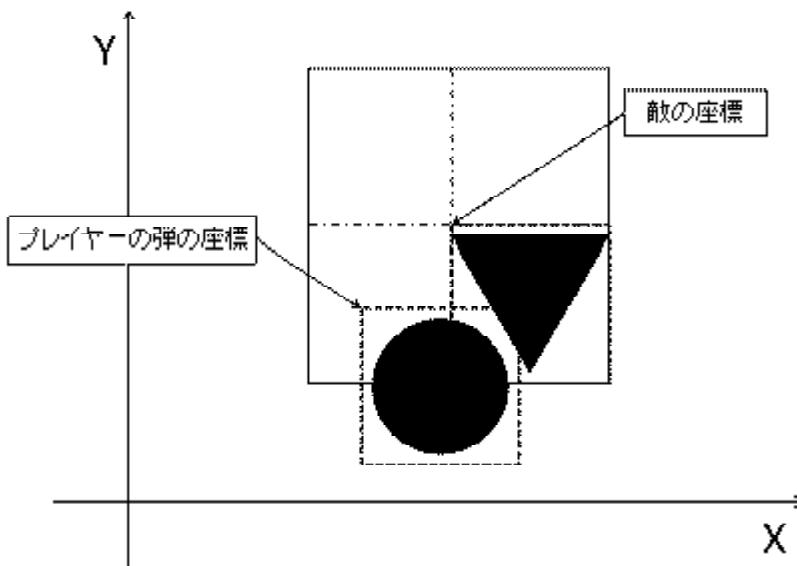
8.敵とプレイヤーの弾との衝突判定

1.衝突判定

敵とプレイヤーの弾が完成し、次にシューティングゲームに必要になってくるのはプレイヤーの弾で敵が破壊される動作です。破壊されるかどうかの判定は、それぞれの物体が衝突しているかどうかで判定します。これを衝突判定と呼んでいます。ここでは、衝突判定について説明していきます。

判定の方法としてはいくつかあり、単純に四角形の範囲を使って判定する方法や円形の範囲を使って判定する方法などがあります。今回は四角形の範囲を使って衝突の判定を行います。

まず、どのようにして判定を行うのか説明します。下の図を見てください。



この図は、座標平面上に敵とプレイヤーの弾を表示している図です。弾や敵の周りについている波線の四角形は各物体の有効範囲です。また、一点鎖線の部分は敵の座標を中心として X 軸と Y 軸を表しています。この図を元に、各物体の X 座標、Y 座標の関係が一定の範囲の中であれば衝突していると判定します。その範囲についてですが、X 軸、Y 軸別々に考えていきます。

まず、X 軸について考えます。図を見ると、次のようにになります。

$$(敵の X 座標) - (敵の幅) \leq (プレイヤーの弾の X 座標) \leq (敵の X 座標) + (敵の幅)$$

また、同じようにして Y 座標について考えると、

$$(敵の Y 座標) - (敵の高さ) \leq (プレイヤーの弾の Y 座標) \leq (敵の Y 座標) + (敵の高さ)$$

となります。

衝突判定を行うタイミングは、WM_TIMER メッセージの部分で移動処理などを行った後です。それでは、実際にコードを見てみましょう。

コード 13 (P.30 211 行目の後に追加)

```

int PlayerBallNumber;
int EnemyNumber;
コード 14 (P.31 286 行目の前に追加)
for (PlayerBallNumber = 0 ; PlayerBallNumber < MAX_PLAYER_BALL ; PlayerBallNumber++)
{
    if (mPlayerBall[PlayerBallNumber] != NULL)
    {
        for (EnemyNumber = 0 ; EnemyNumber < MAX_ENEMY ; EnemyNumber++)
        {
            if (mEnemy[EnemyNumber] != NULL)
            {
                if (
                    mEnemy[EnemyNumber]->Position.x - mEnemy[EnemyNumber]->Width <=
                    mPlayerBall[PlayerBallNumber]->Position.x &&
                    mEnemy[EnemyNumber]->Position.x + mEnemy[EnemyNumber]->Width >=
                    mPlayerBall[PlayerBallNumber]->Position.x &&
                    mEnemy[EnemyNumber]->Position.y - mEnemy[EnemyNumber]->Height <=
                    mPlayerBall[PlayerBallNumber]->Position.y &&
                    mEnemy[EnemyNumber]->Position.y + mEnemy[EnemyNumber]->Height >=
                    mPlayerBall[PlayerBallNumber]->Position.y
                )
                {
                    mPlayerBall[PlayerBallNumber]->Destroy = 1;
                    mEnemy[EnemyNumber]->Destroy = 1;
                }
            }
        }
    }
}
}

```

コード 14 を見てください。mBallPlayer の要素や mEnemy の要素を参照するときには NULL でないかチェックすることを忘れないようにしましょう。真ん中あたりの if 文のところで衝突判定を行っています。先ほど考えた範囲を全て満たすときにプレイヤーの弾と敵を削除します。

これで、衝突判定の処理は終わりました。意外と短くできたと思います。終わったら、F5 キーを押してコンパイルして実行してみましょう。プレイヤーの弾が敵に当たると両方とも破壊されると思います。

9.敵の弾の表示

1.弾の表示

この章では、敵が弾を使ってプレイヤーを攻撃できるようにします。この章が終わった段階でシューディングゲームとして最低限の機能は備わっているはずです。それでは始めていきましょう。

攻撃できるようにするために必要な作業としては、敵の弾を表示できるようにして、敵に行動させ、衝突判定を行うことです。この節では、敵の弾を表示できるようにします。作業の内容などは今までとほとんど変わらないので先にコードを示します。

```

コード 15 (P.30 210 行目の後に追加)
static Ball *mEnemyBall[MAX_PLAYER_BALL];

```

コード 16 (P.30 230 行目の後に追加)

```
for(i = 0 ; i < MAX_ENEMY_BALL ; i++)
{
    mEnemyBall[i] = NULL;
```

コード 17 (P.31 236 行目の後に追加)

```
for(i = 0 ; i < MAX_ENEMY_BALL ; i++)
{
    if(mEnemyBall[i] != NULL)
    {
        TextOut(hdc,mEnemyBall[i]->Position.x,
                mEnemyBall[i]->Position.y,"●",2);
    }
}
```

コード 18 (P.31 284 行目の後に追加)

```
for(i = 0 ; i < MAX_ENEMY_BALL ; i++)
{
    if(mEnemyBall[i] != NULL)
    {

        if(mEnemyBall[i]->Destroy == 1)
        {
            DestroyBall(mEnemyBall[i]);
            mEnemyBall[i] = NULL;
        }
    }
}
```

コード 19 (P.32 298 行目の後に追加)

```
for(i = 0 ; i < MAX_ENEMY_BALL ; i++)
{
    if(mEnemyBall[i] != NULL)
    {
        DestroyBall(mEnemyBall[i]);
    }
}
```

今までと変わったところは特にないと思います。もし、分からぬところなどがあったら前回の記事などを見てみてください。基本的な考え方はそのときと同じです。

2.敵の行動

さて、敵の弾の作成が終わったので、次に敵が行動をするようにします。これには、先ほどの弾の移動の時と同じように RunEnemy 関数というのを作って、その中で行動させます。この関数は、タイマーで一定時間ごとに呼び出されるようにします。さらに、RunEnemy 関数の引数に敵の弾の配列のポインタを渡すようにしておいて、その配列の中に変数を入れることによって敵の行動の中で弾を作成できるようにしておきます。それでは、コードを見ていきましょう。

コード 20 (P.27 12 行目の後に追加)

```
#define ENEMY_BALL_SPEED 3
```

```
if(mEnemy[i] != NULL)
{
```

コード 21 (P.39 コード 18 の「if(mEnemy[i]->Destroy == 1)」の前に追加)

```
mEnemy[i] = RunEnemy(mEnemy[i],mEnemyBall);
-----
if(mEnemy[i]->Destroy == 1)
{
```

コード 22 (P.36 コード 10 の後に追加)

```
Enemy* RunEnemy(Enemy *mEnemy,Ball *mEnemyBall[])
{
    int i;

    if((rand() % 60) == 0)
    {
        for(i = 0 ; i < MAX_ENEMY_BALL ; i++)
        {
            if(mEnemyBall[i] == NULL)
            {
                Vector v;
                v.X = mEnemy->mPlayer->Position.x - mEnemy->Position.x;
                v.Y = mEnemy->Position.y - mEnemy->mPlayer->Position.y;

                mEnemyBall[i] = CreateBall(mEnemy->Position,v,
ENEMY_BALL_SPEED,NULL,mEnemy->mWindowInfo);

                break;
            }
        }
    }

    return mEnemy;
}
```

コード 22 を見てください。RunEnemy 関数では、60 分の 1 の確率で弾の作成を行っています。弾の作成の際に弾の移動方向を指定しています。敵の弾はプレイヤーを狙わないといけないので、プレイヤーの座標から敵の座標を引いて移動方向を決定しています。ただし、Y 座標については軸がずれているので符号を反転させています。移動方向を決定したら CreateBall 関数を使用して弾を作成しています。

これで、敵が弾を撃ってくるようになりました。しっかりとプレイヤーを狙って弾を撃つてくるはずです。F5 キーを押してコンパイルして実行してみましょう。

3. 敵の弾とプレイヤーとの衝突判定

最後に、敵の弾とプレイヤーとの衝突判定を行います。これも、敵とプレイヤーの弾との衝突判定の時とほとんど変わりません。先にコードを見てみましょう。

コード 23 (P.30 211 行目の後に追加)

```
int EnemyBallNumber;
```

コード 24 (P.31 286 行目の前に追加)

```
for(EnemyBallNumber = 0 ; EnemyBallNumber < MAX_ENEMY_BALL ; EnemyBallNumber++)
{
    if(mEnemyBall[EnemyBallNumber] != NULL)
    {
        /* short-circuit evaluation */
        if(
```

```

mEnemyBall[EnemyBallNumber]->Position.y -
    mEnemyBall[EnemyBallNumber]->Height <= mPlayer.Position.y &&
    mEnemyBall[EnemyBallNumber]->Position.y +
        mEnemyBall[EnemyBallNumber]->Height >= mPlayer.Position.y &&
    mEnemyBall[EnemyBallNumber]->Position.x -
        mEnemyBall[EnemyBallNumber]->Width <= mPlayer.Position.x &&
    mEnemyBall[EnemyBallNumber]->Position.x +
        mEnemyBall[EnemyBallNumber]->Width >= mPlayer.Position.x
)
{
    mPlayer.HitPoint--;
    mEnemyBall[EnemyBallNumber]->Destroy = 1;
}
}
}

```

衝突判定の範囲の計算などはやってみてください。今回はプレイヤーに弾がぶつかったら、ヒットポイントを下げるようになります。今のところはヒットポイントは他のところでは使われていませんが、例えばヒットポイントが 0 になったらゲームオーバーにする処理などを入れてみるとゲームらしくなると思います。

さて、いかがでしたでしょうか。今まで 3 回を通しての説明でショーティングゲームの最低限の部分が完成しました。思っていたより簡単でしょうか、あるいは、やはり難しいと思ったでしょうか？どちらにしても、何度も何度も挑戦してみることでだんだんと理解できると思います。次回は、今まで作ってきたショーティングゲームをもっとゲームらしくなるように改造していきます。

○前回の記事の訂正について。

前回の記事で 2 カ所間違이がありました。申し訳ありません。

今回の記事の P.27 10 行目に該当する部分

定数名が間違っていました。次のように変更してください。

(修正前) : #define MOVE_POINT_S (MOVE_POINT * M_SQRT1_2)

(修正後) : #define MOVE_POINT_NANAME (MOVE_POINT * M_SQRT1_2)

今回の記事で P.29 156 行目から 200 行目に該当する部分

押されたキーによってプレイヤーの移動を行う際に、移動を行った後に return していないため、移動量が大きくなってしまいます。各移動を行った後に return してください。

詳しくは、<http://yasuharu.net/word/shooting/2/> にも掲示しております。

今回の記事については、こちらでも公開します。また、記事中に出てきたソースコードについては、章ごとにダウンロードできるようになっています。

Web ページ: <http://yasuharu.net/word/shooting/3/>

間違いの指摘などありましたら、以下の連絡先まで遠慮無く連絡してください。

メールアドレス: word@yasuharu.net

通学路の風景

文 編集部 Goth

どうも(ア)ノ

WORD 編集部・宅通族^{*1} の Goth です。

毎朝〃〃片道 40 分の^{*2} 通学時間、これから季節起きたのがヘヴィです…。

まあその一方で、通学途中、車窓から眺める素晴らしい風景を拝めるわけで、少し得してて気がしないでもない^{*3} 今日この頃。

そこで今回は私、Goth が毎朝眺めている素晴らしい風景をドドンと紹介しちゃいます。

- 8<--- 切 - り - 取 - り - 線 -----

ということで、車に乗り込んで 3 分足らず…。

早速つくば国際大学に到着です!!



失礼。つくば国際大学^{*4} でした。

…土浦市にあるのに「つくば」とはこれ如何に？

ということで、通称「ワンツーファイブ^{*5}」こと国道 125 号を「つくば国際大学」を起点に北上していきます。

*1 宅通族：自宅から大学に通う人々のこと

*2 通学時間：帰りはこの半分で済むのですが…。プチ渋滞が憎い。

*3 気がしないでもない：つまり「得している気はしない」のです。

*4 つくば国際大学：土浦市にある私立大学。偏差値は_____。

*5 ワンツーファイブ：実際こう呼ばれることがある。単に英語にしただけやん…。

- 8<--- 切 - り - 取 - り - 線 -----

北上開始から 5 分たらず、早速右手に元「珍来」、現「麺飯食堂茨珍^{*1}」が見えてきました。



最近改装したばかりなので花輪が飾られています。

しかし、カタカナで「イバチン」って書かれると思わず「イチバン」と読んでしまいそうだ…。多い名前だし。「一番^{*2}」って。

- 8<--- 切 - り - 取 - り - 線 -----

再び北上開始。

次はこちら。



有床診療所開設予定地

18 年度開設予定

*1 茨珍麺飯食堂：元々「珍来」は「総本店(本社:草加市)」「そば坊(本社:土浦市)」「東金店グループ(本社:千葉市)」と三系統あり、それぞれ別会社が同一地方にラーメン店をおくという非常にややこしい状態だったのを、「珍来そば坊」が名称変更を行ったものらしい。

*2 一番：通学路から離れてるのですが筆者の地元には中華はおろか和食、洋食にまで「一番」を謳ってる店があつたりする。さすがにそれは欲張りすぎでは…。

…今年って何年でしたっけ^{*1}？
床^{*2}以前に、床^{*3}がないんですが…。

まあ、予定は未定ということで。

- 8<--- 切 - り - 取 - り - 線 -----

気を取り直してどんどん行きましょう。



地域密着型ショッピングセンター「さん・あぴお^{*4}」でございます!!

駐車場の一部が差し押さえになっているとのもっぱらの噂ですが。

これが廃れ行く土浦^{*5}の姿か…。

- 8<--- 切 - り - 取 - り - 線 -----

延々と 125 号を北上していきます。

右手の方には筑波山。

*1 何年でしたっけ：平成 19 年。

*2 床：とこ【床】1.寝床。2.ゆか。3.畳のしん。4.川の底。川床。5.「床の間」の略。6.床屋。7.苗を育てる所。(旺文社国語辞書より)ここでは入院施設のこと

*3 床：ゆか【床】1.家の中に地面より高く、板などを水平に張った所。2.劇場・寄席などで、太夫が淨瑠璃を語る高座。(旺文社国語辞書より)ここでは 1 の意味。

*4 さん・あぴお：wikipedia にも項目がない地元ショッピングセンター。

*5 廃れ行く土浦：かつては南茨城の覇者であったが、最近では TX によるつくばの台頭で歴史と伝統とレンコンだけの町となりつつある。



なかなか雄大な風景ですなあ。

と、ビニールハウスが見えてきました。

何やら看板が出ています。

見てみましょう。



『今が旬です。いちご狩り』

…今って何月でしたっけ^{*1}？
今が旬なのか…。いちご^{*2}。
人生 20 年目にして驚愕の事実ですよ。

- 8<--- 切 - り - 取 - り - 線 -----

そんな廃れ行く土浦ももう市境。つくば市に突入です。

と、その前になにやら工事中の看板が見えてきました。



写真ではずいぶん離れているように見えますが、一つ目の「300m」の看板と二つ目の「200m」の看板の間に 100m もない、というか高々 20m ちょいしか離れてないのですが…。

どうやらここには空間の歪みが生じているらしい…。

ちなみに、道路拡幅工事だそうです。

まあ、よく混み合ってる所なので是非とも拡幅してほしいところです。
この プチ渋滞 がなければ後 10 分ぐらい家のんびりできそうですよ。

- 8<--- 切 - り - 取 - り - 線 -----

工事現場を抜けつつ、ついに

*1 何月でしたっけ：原稿書いている時点では 10 月。

*2 いちご： いちごの旬は春から初夏、とか思っていたら、実はハウス栽培がメインで、生産量の殆どが 11 ~ 6 月らしい。びっくりだ。



つくば市突入!!

工事看板は土浦市内に立ってましたが、工事自体はつくば市内で行われています。というかほとんど市境でやってるような感じですね。

- 8<--- 切 - り - 取 - り - 線 -----

つくば市入ってすぐに見えてくるのが『和風レストラン キャニオン』。



えっと、メニューは「えびフライ」「かつれつ」「焼肉」「パスタ」「ハンバーグ」。

…「パスタ」？

『和風』 レストランなのに？

店名の「キャニオン」もかなり気になるところですが。

というかこの店、拡幅工事の範囲に入ってるっぽいのですが、潰されないのでしょうか。

まだ営業中の看板は出てましたが。

- 8<--- 切 - り - 取 - り - 線 -----

さて、もうそろそろ 125 号とお別れして国道 200 号^{*1}へと曲がりましょう。

の、前に。

交差点の手前、125 号沿いに筆者的心をグワシッとする交通標語が立っています。

それがこちら!!



『気のゆるみ あつといまにあの世行き』^{*2}

インパクトばっちり。

はい、皆さんもご一緒に。

『気のゆるみ あつといまにあの世行き』
ちゃんと覚えましたか？

*1 国道 200 号：一の矢のすぐ南の道路へとつながります。

*2 気のゆるみ…：よくよく見てみると地元高校の PTA の文字。Google 先生に聞いても検索件数ゼロ、と、どうやらここの PTA の完全オリジナルのようです。すばらしいセンスに感服。

「地獄行き」じゃない所が優しさですね。

- 8<--- 切 - り - 取 - り - 線 -----

つくばりんりんロード^{*1} を越えてずんずんすんでいきまして、



つくば市突入!!

って、あれ？

さっき、つくば市に入らなかつたっけ？

やはりこの界隈には空間の歪みが…。

- 8<--- 切 - り - 取 - り - 線 -----

えと、つくば市入ってから、ややネタ不足^{*2} です…。

まあ、それでもいくつか写真を載せていきましょう。

- 8<--- 切 - り - 取 - り - 線 -----

ということで、非農業県からお越しの都会人の皆様。

これ↓なんだと思いますか？

*1 つくばりんりんロード：筑波鉄道の路線跡に作られた自転車専用道。筆者の中学校にはここを歩いて筑波山まで向かう「筑波登山(行程の大半は平地)」という苦行行事がありました。

*2 ややネタ不足：冒頭では「すばらしい景色を」と書いておきながら結局はネタを探しているあたり、何かを間違っているような気がする。



無人精米所。

いやあ、つくばも田舎だねえ。

ちなみに農業市、土浦でも随所で見かけますが、**不思議**と東京 23 区内で見かけたことないんですよね。ええ、なぜか。

- 8<--- 切 - り - 取 - り - 線 -----

さてさて、写真を撮影している付近は国道 200 号沿いの「栗原」というところで、近くには栗原小学校があります。

三桁国道^{*2}とはいえ車の往来の激しい道路。加えて見通しの悪いカーブにその小学校があります。

と、ということで、こんなものを見つけました。

*1 無人精米所：農家の方々や、農家の方から直接玄米をいただいた人が精米するための施設。よく「米ぬか無料」とか「石抜装置付」などの、付いていると嬉しいのであろう機能の宣伝文句が書かれているが、正直なところ筆者にはよくわからない。最近は玄米のままで出荷できるので、農家の方でも自宅に精米機がないことが多いらしい。

*2 三桁国道：国号のうち番号が三桁になる国道のこと。概して細道が多いが、125 号の様に「激しい」道路も数多く存在する。と、いうかむしろ激しい道路の方が多いやも…。



手を挙げた学童人形。

ただし顔が怖い。

なんともいえない表情であります。この子。

- 8<--- 切 - り - 取 - り - 線 -----

と、いうことで、栗原を超えてやっと筑波大学に到着です。
なんというかあっけないエンディングですが、

撮影のため歩き
全行程約 10km
+) 筆者=運動不足 (式 1.1)
もう体力の限界

式 1.1 より、

もうだめ…。

書籍紹介

文 編集部 suma

第三回

書籍紹介とは

今回で第三回となります書籍紹介のコーナーがやって参りました。今回はこれまでと趣向を変えて洋書を紹介したいと思います。コンピュータ技術書の洋書というと、英語の読解が難しいというイメージを持たれる方がいるかもしれません。しかし、実際のところ高校生でも簡単に読めてしまう洋書も存在します。また、Amazon や紀伊国屋を利用すれば日本語の書籍を買うのと同じくらい手軽に洋書を手に入れることができます。

The Art of Computer Virus Research and Defense

紹介する書籍は「The Art of Computer Virus Research and Defense」です。ウィルスというのはプログラムで、人によって作られるというのは皆さんご存じだと思います。ところで、ウィルス感染の仕組み、アンチウィルスソフトの仕組みについて、どうやって作られているか疑問に思ったり興味を持ったことはありませんか？そんなコアな内容から一般的なウィルスの歴史や種類など、多くのトピックがこの書籍一冊で網羅されています。

具体的には、アンチウィルスソフトがファイルやメモリからのウィルス検出法やアルゴリズム、ウィルスのファイル感染やウィルスがアンチウィルスソフトの検出から逃れるための技術も解説されています。そのため、ウィルスとアンチウィルスソフトの実装技術のようなコアな部分まで知りたい人、アンチウィルスソフトを作ってみたい人にぴったりの書籍ではないでしょうか。

これはと思ったら立ち読みを

少々値が張るため、内容が気になる人はまず次の URL からサンプルを読んでみるとよいでしょう。ぜひ「The Art of Computer Virus Research and Defense」を読んでみてください。美しいウィルス・アンチウィルス技術とバイナリ解析の世界が待っています。

- Strategies of Computer Worms

<http://www.informit.com/articles/article.aspx?p=366891>

- Advanced Code Evolution Techniques and Computer Virus Generator Kits

<http://www.informit.com/articles/article.aspx?p=366890>

書籍名 : The Art of Computer Virus Research and Defense

著者名 : Peter Szor

発 行 : Addison-Wesley Professional

WEB : <http://www.peterszor.com/>

ISBN-13 : 978-0321304544

値 段 : 49.99US / 69.99CAN

頁 数 : 744

発行日 : 2005/02



情報科学類誌

WORD

真・ゆとり世代号

発行者

情報科学類長

編集長

武井 裕也

制作・編集

筑波大学情報学群

情報科学類誌WORD編集部
(第三エリアC棟212室)

印刷

第三学群棟印刷室

2007年11月7日

初版第一刷発行(514部)