2011年度 卒業論文

ゲームによるプログラミング学習補助の提案とその効果の検証

大阪産業大学 工学部 情報システム工学科
情報教育システム研究室

08H056 坂田龍之介

目次

1　はじめに

現在、大学などの教育機関において学生にいかにプログラミングを学習させるかが注目されている [1–3]。しかし、プログラミングはおろか情報の基礎ですら十分に学習できていないプログラミングの初学生にとってプログラミングの講座をすんなりと頭に受け入れ、自分の知識として取り込むことは困難である。何故ならプログラミングというものは初学者にとって全く新たな概念であり、プログラムを作成するにあたって必要な考え方、いわゆるアルゴリズムを構築する力が絶対的に不足していると推測される。

その傾向は大阪産業大学工学部情報システム工学科でのプログラミング演習にも当てはまる。基本的に演習で行うプログラミングの講座は淡々とテキストの解説を行うもので、実習時間においてもテキストに記されたサンプルを打ち込んだり、テストを行ったりとその行程は楽しくないと感じる人のほうが多いと推測される。さらに多人数の学生を少人数の先生、ティーチングアシスタントが教える形にならざるをえないために個々の細かな疑問や躓きに対応しきれないといった問題も生じる。

プログラミングが何たるかを解説することは非常に有意義であり、習得するには反復練習は不可避なものである。だが、まず学習者自身がプログラミングは楽しいものだと認識し、すすんで学習する土台を作ることが必要になると推測される。

2　本研究の目的

本研究の目的は大きく分けると 2 つある。1 つ目はプログラミングに対する苦手意識や難しいものという先入観をなくして楽しいものだと思わせることで学生のモチベーションを向上させることである。2 つ目はゲームをプレイし、攻略するにあたって必要になる論理的思考力を養うことである。

どんなに優れた学生でも、物事に対して全くの意欲がない状態では学習効果は芳しくないことは自明の理である。逆に意欲的に取り組めば学習効果は増加する。プログラミングを楽しいものだと認識し、自らが意欲的に取り組ませることができれば授業や講義における学習効果も大きくなるのではないかと仮定した。

また、論理的思考力を養い、プログラミングを行うにあたっての類似的な思考を前もって体験しておくことでスムーズに授業や講義でのプログラミングへ入ることが出来ると仮定している。ここで言う論理的思考力とはプログラミングを行う上で、どうすれば目的を達成することが出来るかという道筋を考え、その考えた道筋を実現するにはどういった動作が必要で、その動作を実現するにはどうすればよいか。という一連の流れを考える力のことである。

本研究では入門から中級者程度までの学生を対象に、プログラミングに対するモチベーションの向上、そして基本的な論理的思考力の向上の二点を主目的とし、この目的を達成するべくゲームプログラムを作成し、学習補助効果の有意性を検証した。

3 先行研究

プログラミングの補助学習については先行研究がすでに存在している。ここではその中でもゲームによってプログラミングの学習補助を行う以下 2 つの研究について考察した。

3.1 プログラミング入門支援ゲームの試作と評価

この研究 [1] では平面上でキャラクターを動かし、論理的思考力を養えるような魔法の本やジャンプシューズなどを駆使してなるべく最短の手順でゴールすることを目指すゲームである。

入門者に対してプログラミングの知識を必要としないゲームをプレイしてもらい、事前にプログラミング的思考の体験をすることで授業や演習による理解が促進され、プログラミングへの敷居を下げる効果が期待でき、更にゲームとして取り組ませることでモチベーションの向上を図るとしている。ここでは入門者用として GUI[*1] を用いた純然たるゲームが用いられている。入門者にとっては適しているが、入門からステップアップする学習者にとっては更に効果的な方法があると考えた。GUI で視覚的にわかりやすく取っ付き易いため、学習者の学習意欲を高めやすい。しかしながら同時にハードルが低いと越えた時の成果もまた低いものになってしまう可能性もはらんでおり、また実際にプログラミングを入門として行う際は CUI[*2] プログラミングから入るのがほとんどであるため、GUI を用いたゲームだと CUI プログラミングとゲームとの乖離が発生し、演習を行う際の学習者の意欲低下につながるおそれがあると考えた。

3.2 ゲームを題材としたプログラミング教育支援システムの開発

この研究 [2] ではアメリカからの輸入ゲームである Battleship のルールを簡略したものを用い、より少ない攻撃回数で軍艦を殲滅できる攻撃アルゴリズムを考えるものである。

さらに、ゲームプログラムを別々の役割をもたせた Practice、Verification、Score の 3 つを開発し、これらをガイドラインとすることでプログラミングの各プロセスを繰り返し行い易くしている。Practice でまずはルールの確認をして目標とするべきアルゴリズムを明確にした後、Verification でその目標と現状の差異を確認できる。そしてアルゴリズムの成績を Score によって視覚化しているため、目安がわかりやすく目標値を設定しやすいなどの効果が期待できる。だが、CUI 上でマス目を出現させて行う方法ではゲーム自体が単調で味気のないものと見られがちで、ゲーム自体の難易度も高めであるために初学者にとっては難しいものになるおそれがある。

3.3 ボードゲームの戦略プログラミングを題材とした Java 演習の支援システムの開発

この研究 [3] は五五ゲームと呼ばれる五目並べに石取りの要素を加えたボードゲームを用いて、その戦略アルゴリズムをプログラミングして対戦させるものである。

ボードゲームという身近な題材を使い取っ付き易いものにしておき、さらにゲームルールをあまり知られていないものにすることで予備知識の有無による差を小さくしている。ボードゲームという性質上勝敗が明確で勝利条件を満たす回答が一つではないため、より強い戦略アルゴリズムを試行錯誤して組み上げていくことが可能になっている。さらにトーナメント形式で対戦を行うことにより多くの対戦過程を学生に提示し、対戦結果から戦略を修正する機会が多くあり、また競争による意欲の向上も図ることが出来る。しかしボードゲームという一見レトロで単調なもののため、学習用の支援システムと捉えられがちで純粋なゲームとして遊ぶことが難しいおそれがある。

[*1] グラフィカルユーザインタフェースのこと。画像等のグラフィカルな表示で情報が出力される。

[*2] キャラクタユーザーインタフェースのこと。文字のみで情報が出力される。

3.4 プログラミング学習補助ゲームにとって必要な要素

以上の先行研究から、プログラミング学習補助ゲームは初学者にとってとっつきやすいグラフィカルでゲーム性の高いものを用いてモチベーションの向上をはかり、さらに論理的思考力の向上を狙うアルゴリズムをプレイヤーに組ませる必要がある。それはさらにひと目で目標を達成できたかどうかを確認できる形が望ましい。

したがって、プログラミング学習補助ゲームにとって必要な要素は本研究では以下のようになると考えた。

3.4.1 論理的思考力の育成

プログラミングを行う際に、言語ごとの仕様や文法といった言語依存の問題を除くと、重要なのは論理的思考力いわゆるアルゴリズムを考える力が重要になってくる。これは要求されたプログラムの仕様を把握し、その仕様を実現するためにはどのような記述が必要か、また効率的に組み上げるにはどうするべきかを考える力のことで、たとえ文法や言語仕様がわかっていたとしてもこの能力が皆無であれば 1 からプログラムを組み上げることはできないからである。

さらに、この力は言語仕様などと違い暗記による学習があまり意味を成さない。プログラムを書く、もしくはそれに類似する体験を積ませることで育成されていく力のために、こういったプログラミング学習補助によって育成するのが望ましいとされている。

3.4.2 モチベーションの向上

これはプログラミングに限ったことではないが、学習を行う上でやる気、つまりモチベーションは相当に重要である。モチベーションが低ければ学習をする気も沸かず、強制的に学習させられても学習効果は当然低い。学校での授業内となると時間も多くは割けないために更にシビアになる。

逆にプログラミングは楽しいものだと認識させることが出来れば上記のシチュエーションのみならず、授業時間外のあらゆる時間でも自己学習に励む可能性が上昇する。あらゆる面でメリットが大きいためにモチベーションの維持、向上は極めて重要な要素だといえる。

3.4.3 総合的なステップアップへの足がかり

前述した 2 つの先行研究 [1, 2] では、例えばごく入門者から中級者へのステップアップ時の考慮はあまりされているとはいえない。入門から中級へステップアップするには少なからず障害となる壁を乗り越えなければならない。その時に躓いたり、わからないまま先にすすんでしまうと結果的にそれが足かせとなってしまう。上の段階へとスムーズにステップアップし、且つ広い範囲でサポートすることが出来ればプログラミング学習は現状よりもさらに手近で興味深い分野になるのではないだろうか。

4 ゲーム内容

前述したプログラミング学習補助ゲームにとって必要な要素の 3 つを踏まえ、この 3 つの要素をなるべく高レベルで満たしうるゲームプログラムを開発することにした。

ゲームタイトルは DT 防衛戦である。

DT は DefenceTank の略称であり、二足歩行型戦闘装甲車両の通称である。舞台となる国の自衛が主目的となる軍隊において、無限軌道型の戦車に変わり、更に汎用性の高い戦車を求める声が大きくなったため試作されたものである。高い機動力、俊敏性を備え、多種多様な装備を組み合わせることが可能で、市街地での戦闘行為や人員救助をはじめ様々な任務に対応できるようになっている。

なお、ゲーム内では DefendTank と DefenceTank の二通りの呼称で呼ばれることがあるが、これはあくまで非公開情報に基づいた通称であるため、正確な名称がないからである。

本ゲームはストーリー進行とチュートリアルをおりまぜながら、ユーザーに DT の AI プログラムを記述してもらいそれを実際のロボットに読み込ませ、装備されている武器や移動を組み合わせて敵機を撃破しながら進んでいくストーリー進行形のゲームとなる。

ここで言う AI は自律的に敵に対して攻撃や回避を行うプログラムのことで、DT にこれを読み込ませて戦わせることで無人の DT でも戦闘行動を取ることが可能になっている。元々は有人のリーダー機体を補佐したり、有人機では危険すぎて遂行できない任務を完遂するために搭載されたシステムだが、本ゲーム内では DT を操縦する知識も訓練も受けていない人間が AI をプログラミングすることによって DT を操縦する目的で使用される。

開発言語のコンパイラを用いてユーザーに記述してもらう AI のエラーなどを検出するため、開発言語はユーザー[*3]が触れる機会の多い C 言語で開発することとした [4]。

またデータの保持には小規模のアプリケーションに適している SQLite3 を用いた [5, 6]。

本プログラム自体は 3 つのパートに分かれており、それはそれぞれストーリー進行パート、チュートリアルパート、バトルパートとなる。さらにチュートリアルパートとバトルパートの間に、プログラム外でユーザーに実際にエディタと仕様ドキュメントを照らしあわせながらコードを記述してもらうコーディングパートを加え、全部で 4 つのパートによりゲームは進行していく。この 4 つのパートはそれぞれ以下の役割を持ち、この 4 パートを一巡することで 1 ステージとなる。ステージごとに目標となるテーマを決定しておき、それを実現しうる AI を記述できていれば戦闘に勝利でき、そのステージをクリアし次のステージへと進むことが出来る。プレイヤーは各ステージを AI を記述してステージをクリアしていき、最終的にエンディングを見るのが目標である。

前述した 4 つのパートはストーリー進行パート、チュートリアルパート、コーディングパート、バトルパートの順で進行していく。

ゲーム中の画面を図 1 に示す。

[*3] ここで言うユーザーは特に大阪産業大学工学部情報システム工学科 1,2 回生、またはそれと同程度の学生を指す。

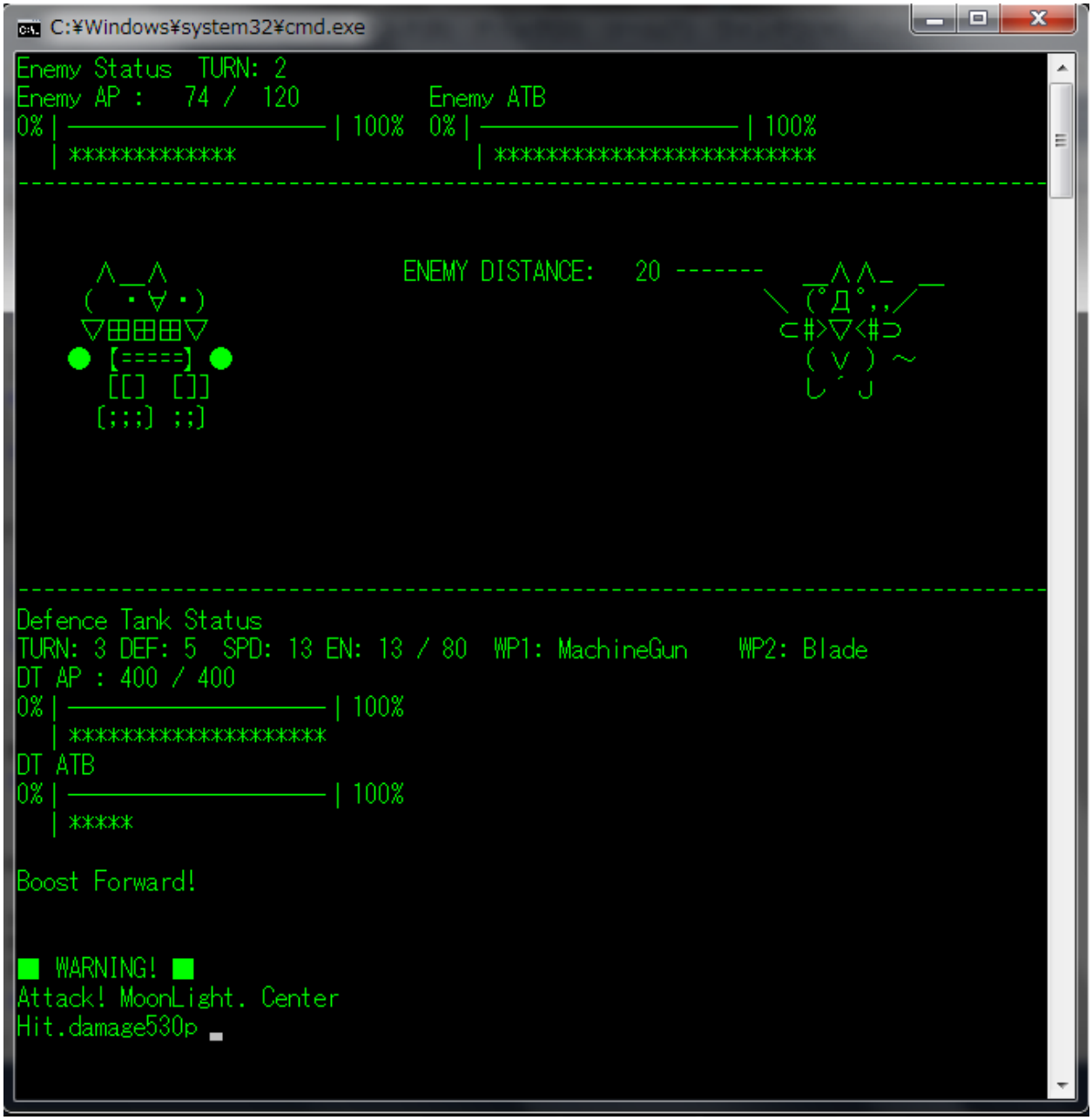

図 1　ゲームの中心部分であるバトルパートの画面。目的に沿った強い AI をチュートリアルにそって組み上げ、自機と敵機を戦わせ勝利していくことが目的となる。

4.1　ストーリー進行パート

このパートではストーリーの進行を行う。ストーリーをゲーム内に組み込むことで、没入感や使命感が加味され、プレイヤーを途中で飽きさせないようにする効果が期待できる。さらにストーリー進行時には図のようにアスキーアートを用いてキャラクターを表示させている。これにより文字だけという CUI 環境でもある程度グラフィカルな表示が可能になり、視覚的にとっつきやすくしている。ストーリーを追うごとに自然にゲーム部分の難易度やチュートリアル部分で設定されているプログラミングの進度をすすめることが可能である。ストーリー進行時の画面を図 2 に示す。

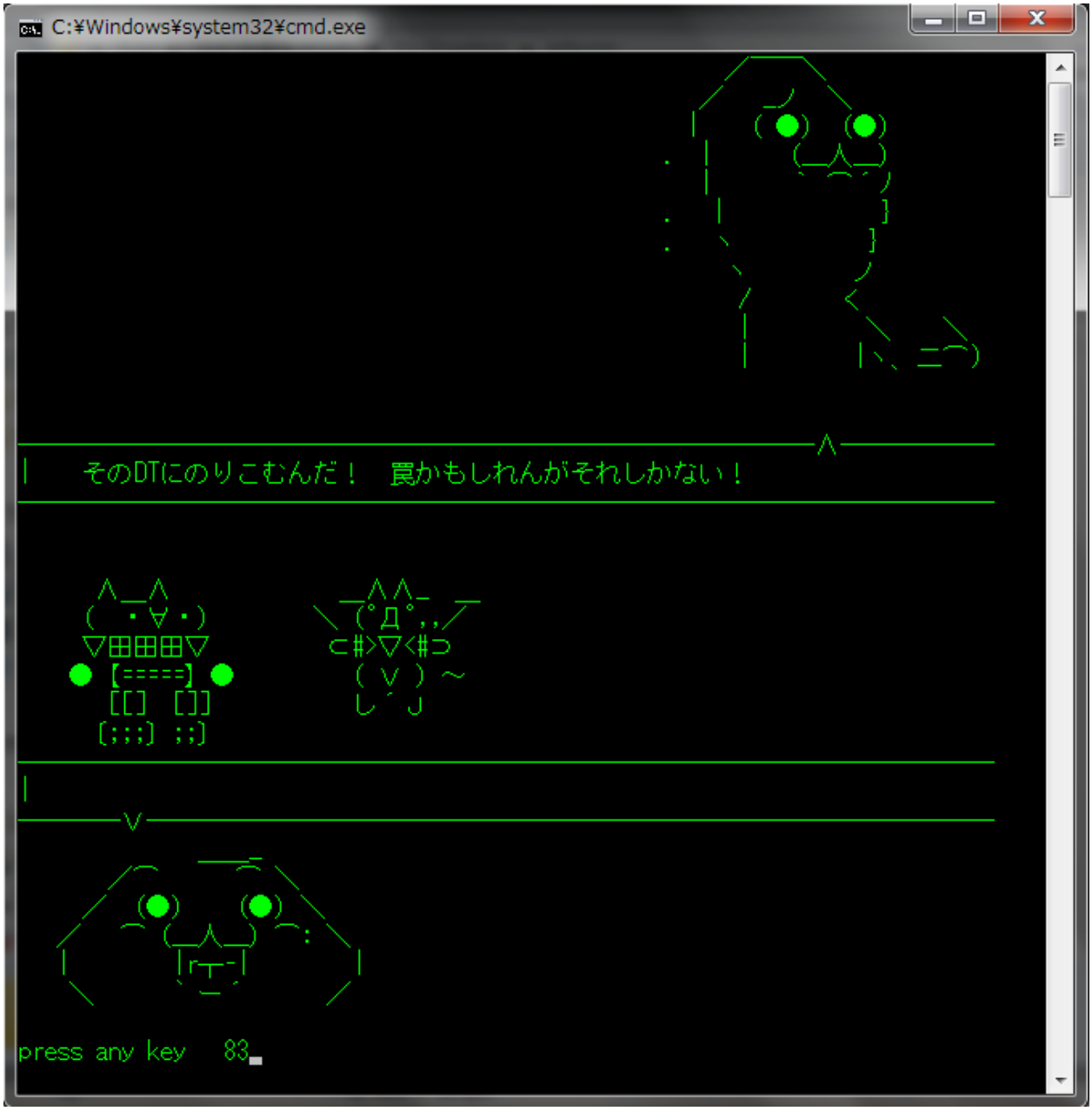

図 2　ストーリー進行時の画面。キャラクターが二人一度に登場し、その会話のやり取りでストーリーが進んでいく方式。

4.2 チュートリアルパート

このパートでは前述したストーリーパートを引き継いで、実際にコーディングする上でのヒントや方法を示す。ストーリー上で登場したキャラクターがそのまま説明を行う。そのためストーリーに没入したままチュートリアルに入ることができ、更に自分のタイミングで行送りすることが可能になるために自分のペースでチュートリアルを咀嚼していくことが可能である。上部にはもう一人のキャラクターの代わりにサンプル AI のコードや前回に使用された AI のコードが表示される。

これによりチュートリアルを読みながら実際のコードと照らしあわせて考えることが可能になる。チュートリアル進行時の画面の例を図 3 に示す。

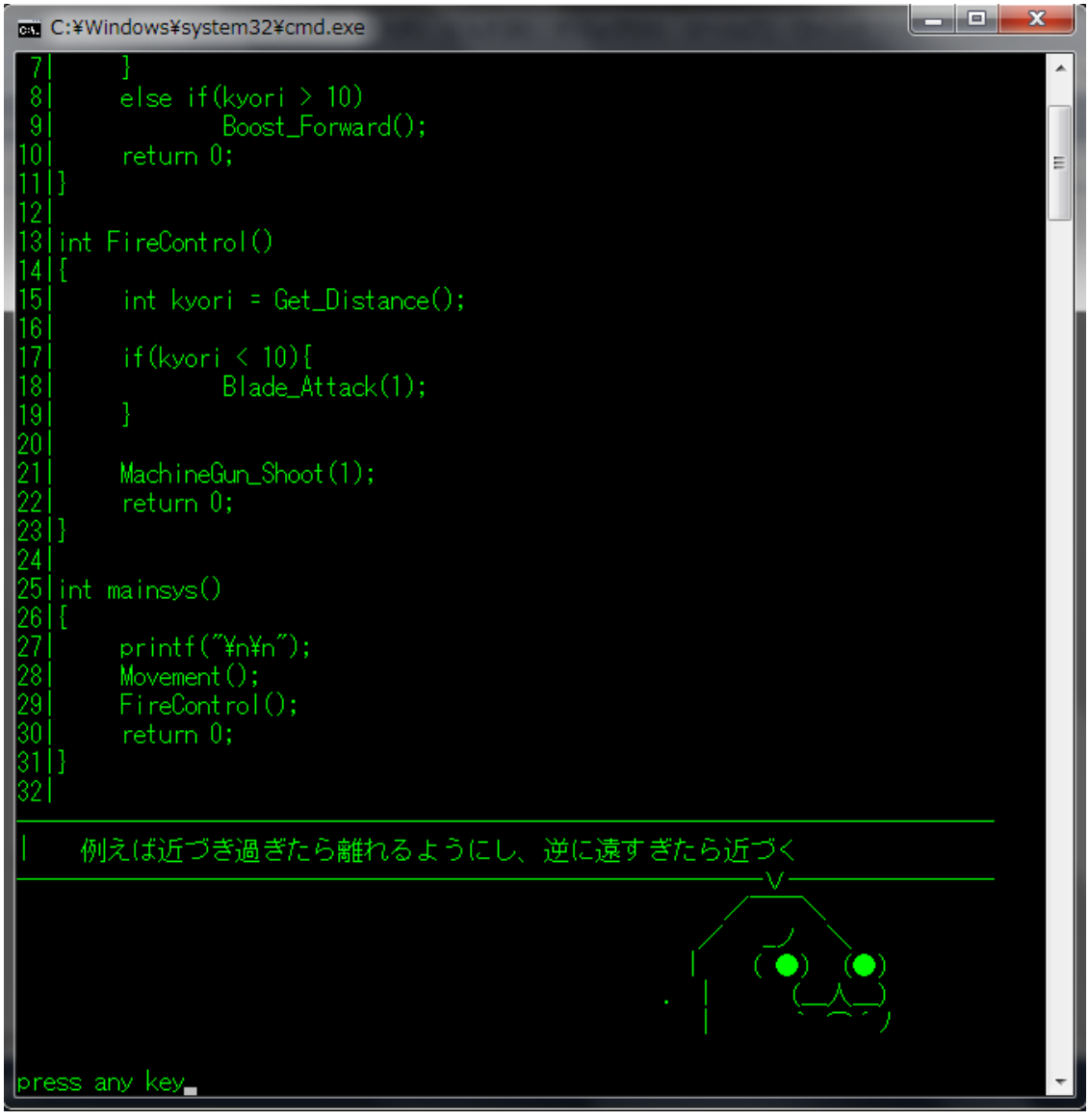

図 3　チュートリアル進行時の画面。基本的にはストーリー進行部分と連続した形で表示されるため、インタフェースもストーリーパートのものを引き継いでいる。チュートリアルを行うために上部には代わりにソースコードが表示される。ここに表示されるコードはサンプルや目標とすべきコードなど、チュートリアル内容によって変化する。

4.3 コーディングパート

このパートではチュートリアルパートを受け、プレイヤーがチュートリアルにあった範囲を実際にコーディングしていくパートである。コーディングには実際に使われている各種エディタを使用する。コーディング時には、チュートリアルパートが終了した際に表示されるヒントと、付随しているドキュメントをプレイヤーが読みながら仕様を確認し、AI を記述していく形を採用した。

学習という面において、入門ないしそれ以上のプログラミングとなるとドキュメントやテキストを読みながら記述していくケースは増えていくため、その訓練も兼ねた方法である。

この AI は本プログラム内部に組み込まれてソースコードとして取り扱われるので、AI を反映させるにはコンパイルし直す必要がある。プレイヤーに AI を記入させてコンパイルさせる一連の流れも、実際のコーディング手順になぞらえさせている。

コーディング時の画面の例を図 4 に示す。

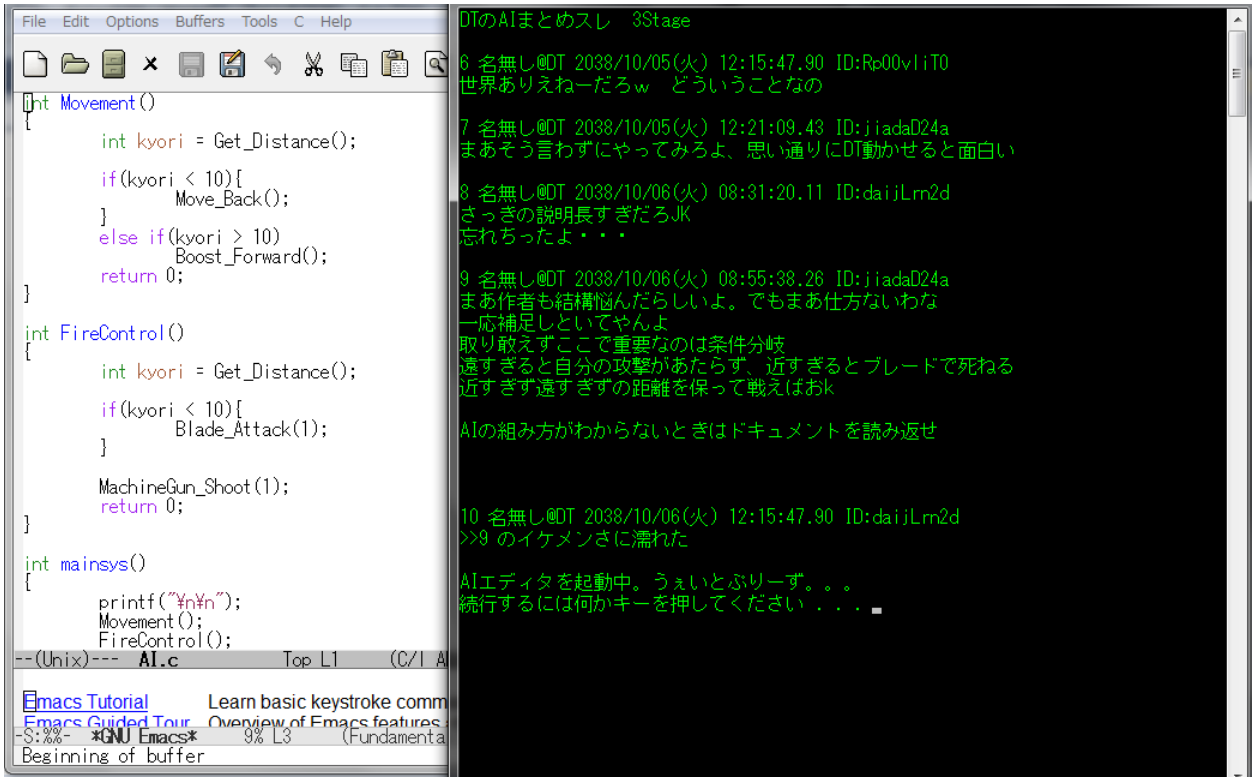

図 4　コーディング時の画面。このようにチュートリアルパートで最後に表示されるまとめを読みながらエディタを開いてコーディングすることが可能である。実際にはまとめと同梱されているドキュメントも開いてコーディングを行うことが望ましい。

4.4 バトルパート

このパートではコーディングパートにおいて記述した AI を用いて実際に敵機 AI と戦わせるパートである。後述するルールに従い、先に AP[*4]を 0 にしたほうが勝利となる。

予めストーリー進行パートやチュートリアルパートで示されている敵機 AI の弱点をつくように AI を記述できていればほぼ勝利できるように調整してあるため、コードの優劣がはっきりとわかりやすく、さらに回答は一つではないことが理解できる。またバトルはターン制で行われ、自機 AI の行動と敵機 AI の行動やステータスは一つ一つが表示されるので自分が組んだ AI が実際にどういった動きをしているのかが理解しやすい。バトル時の画面の例を図 5 に示す。

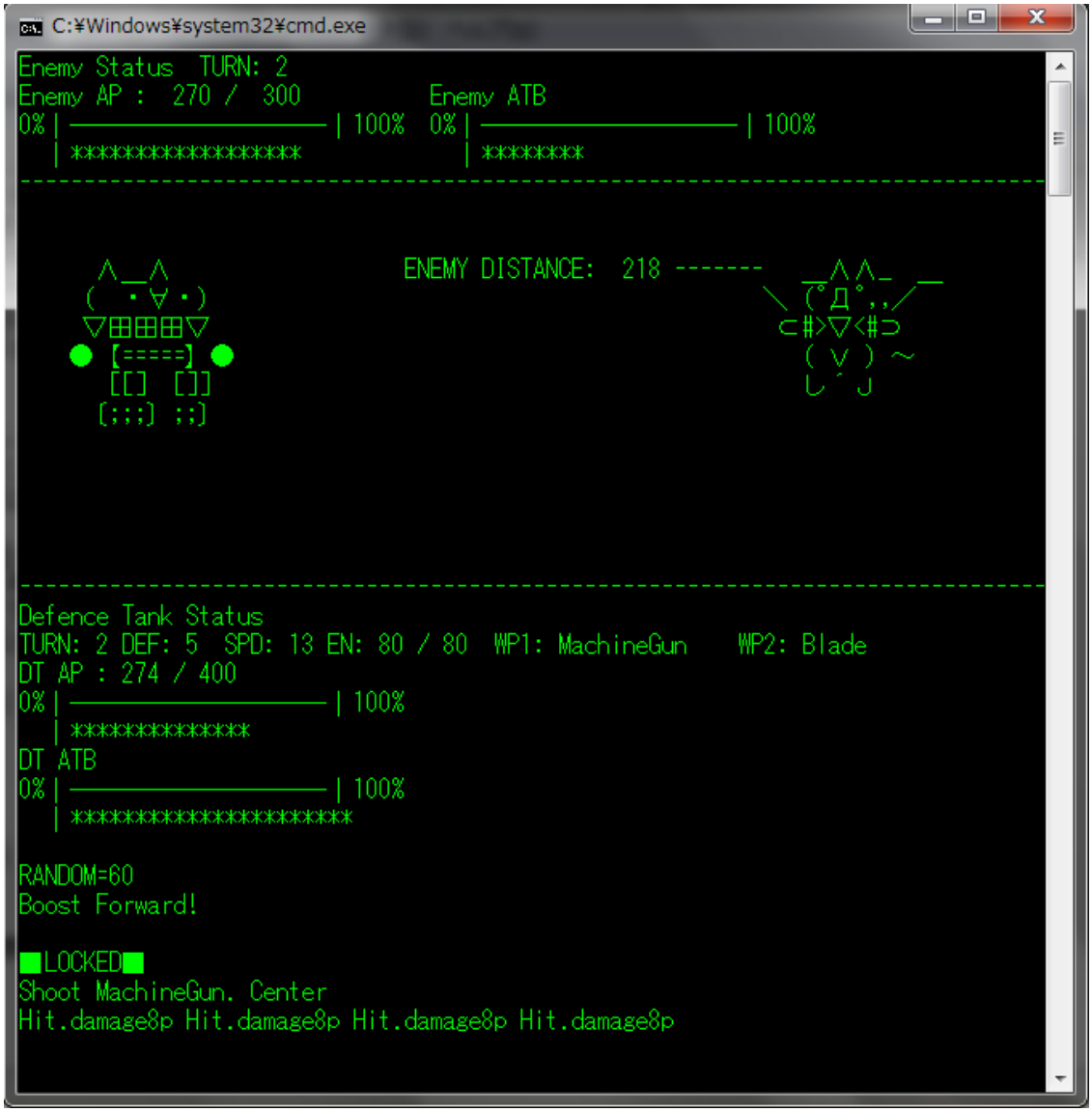

図 5　バトル時の画面。上部が敵のステータス、下部分が自機ステータスとなる。自機ステータスの下はそのターンに実行された行動が表示されるため自分が書いた AI がどのような行動を実際にしているか確かめやすい。

[*4] アーマーポイントの略。装甲値のこと。

4.5 ゲームルール

バトルパートで行われる戦闘は DT 同士によるロボットの戦闘となる。ロボットを構成する部品は以下の 3 種類であり、これらを組み合わせたものを戦わせる。各部品には防御力などのパラメータが存在しており、それらを総合的に判断してロボットをどう戦わせるのかを考える。

Engine　ロボットの動力源であり、移動速度や攻撃出力などに総合的に関与している。
Armor　ロボットの装甲であり、防御力や耐久力であるアーマーポイントを決定している。
Weapon　ロボットが装備している武器で二種類装備でき、攻撃力や攻撃回数、射程距離などを決定する。

以上の 3 種類を組み合わせて DT は構成されている。これに頭脳部分の AI を組み合わせて DT の完成となる。

4.6 AI 構成

プレイヤーに記述してもらい、DT に搭載する AI は移動系の関数を記述する Movement、武器の使用など火器管制を行う FireControl、この 2 つをまとめ、実際に実行する mainsys の 3 つに分けられており、プレイヤーは主に Movement と FireControl の中に予め用意された関数を記入していくことで作成される。

しかしながら行動にはすべてエネルギーの消費が不可避なため、現在エネルギー量を超えての行動はできない。そのためいかに効率よく最小限の関数を用いて AI を構築していくかが重要である。

4.7 ターン制戦闘

バトルを行う上での勝利条件は各 DT に装備されている兵装を用いて敵のアーマーポイントを 0 にすることである。ただし、いつでもリアルタイムに行動できるわけではなく、時間ごとに一定量溜まっていく行動ゲージが先に 100 になった機体から行動する。行動する内容は AI に記述された mainsys 関数を実行し、同時に行動した方の行動ゲージを 0 に戻す。以上を 1 ターンとし、以後これを繰り返してバトルを行う。

4.8 パラメータ

戦闘の際に関係する環境パラメータは以下のとおりである。

距離　自機と敵機との相対距離である。距離が離れるほど威力や命中率に影響を及ぼす。
位置　ロボットの装甲であり、防御力や耐久力であるアーマーポイントを決定している。

戦闘の際に関係する機体パラメータは以下のとおりである。

EnergyPower　エネルギー容量のことで、DT の行動には全てエネルギーを消費しこの値を超えて行動はできない。
OutPutPower　エネルギー出力のことで、移動速度やスピードをはじめ一部の武器の攻撃力などに関与する。
AP　ArmorPoint、つまり装甲耐久値を表し、この数値が 0 になると戦闘不能となる。
防御力　装甲防御力を表し、攻撃はすべてこの値をダメージ値から減算したものが最終的なダメージになる。
スピード　行動可能速度のことで、行動ゲージのたまる速さに関係している。

また、戦闘の際に使用する武器パラメータは以下のとおりである。

攻撃力　武器攻撃一発あたりのダメージ量を表す。
攻撃回数　武器攻撃が一回の攻撃関数呼び出しで何回攻撃できるかを表す。
射程距離　武器の射程距離であり、これが長いほど距離の威力減衰現象や命中率減衰現象の影響を受けにくい。
命中率　武器の命中率であり、この確率で相手に命中するかどうかが決定される。
回避時固有命中率低下値　照準した位置と相手の実位置が違っていた場合にこの値の分だけ命中率が減少する。

4.9 各パラメータに起因する現象

以下のゲーム的要素を加味することで、AI プログラムを構築する際に考えるべき部分が増える。これにより複雑な AI を考える力を自然に身につけることが期待できる。

4.9.1 威力減衰現象

最終的に相手に与えるダメージは与えるダメージを X とした場合の計算式を以下に示す。

$$X = \frac{\text{武器の攻撃力} - \text{相手の防御力} + (Engine\ \text{の}\ OutputPower)}{50} * (\text{武器係数} - \frac{\text{相手との相対距離}}{\text{武器の射程距離}})$$

これにより、武器射程の短い武器は射程が離れれば離れるほど武器の攻撃力は落ちていく。そのため近づく必要が生じるので、プレイヤーは AI 構築時に効率よく接近するためにはどうすればよいかを考える必要が出てくる。

4.9.2 命中率減衰現象

最終的に命中判定のしきい値はしきい値を Y とした場合の計算式を以下に示す。

$$Y = \text{武器命中率} + 10 - \frac{\text{相手との相対距離} + 1}{\text{武器の射程距離} * 5}$$

この Y をしきい値とし、ランダム値と比較することにより命中判定が行われる。この計算式により距離が離れれば離れるほど命中率が低下することが分かる。

更にこれは攻撃する位置と相手の位置が一致していた場合であり、攻撃位置と相手の位置が違っていた場合はここからさらに武器それぞれの回避時固有命中率低下値のぶん低下する。

これにより、きちんと正しい攻撃位置を指定することで勝率が大幅に上昇するためプレイヤーは AI に攻撃位置を考慮する必要が生じる。

5　評価実験

このゲームがプログラミング教育補助に有効かどうかを評価するために本学の一回生を対象に試験的に DT 防衛戦をプレイしてもらい、アンケート調査を行った。

5.1　アンケート調査

アンケートは本学の一回生に対し電子メールで配布した。アンケートを取る狙いはプログラミング学習に対する苦手意識の解消やモチベーションの向上を計れたかどうかに主軸を置き集計を行い、分析する。以下にアンケート内容を示す。

1. プログラミングは好きですか？
    - 1-1 三度の飯よりも大好き。
    - 1-2 結構好き。
    - 1-3 別に普通。
    - 1-4 課題とかで言われたらやるけど嫌い。
    - 1-5 大嫌い。見たくもねえ。
2. あなたのプログラミング能力はどれくらいですか？
    - 2-1 めちゃくちゃ出来る。実際コーディングとかのバイトも経験ある。
    - 2-2 割とできる。趣味でアプリ作ったりしてるよ。学校の授業程度は余裕。
    - 2-3 普通。学校の授業くらいならなんとか。
    - 2-4 正直プログラミングわからない。めちゃむずい。
    - 2-5 プログラミングって何？
3. このゲームをプレイして、ストーリーは面白かったですか？
    - 3-1 めちゃ面白かった。先が気になった。
    - 3-2 わりと面白かった。
    - 3-3 可もなく不可もなく。
    - 3-4 あんまり面白くなかった。あんた文才無いんじゃない？
    - 3-5 本気でつまらなかった。不快感が湧いた。

4. このゲームをプレイして、ゲーム部分 AI 記述部分は面白かったですか？
    - 4-1 とても面白かった。
    - 4-2 わりと面白かった。
    - 4-3 普通。どっちでもない。
    - 4-4 あんまりおもしろくない。
    - 4-5 おもしろくない。何これやる気あんの？
5. このゲームの難易度はどうでしたか。
    - 5-1 簡単だった。
    - 5-2 普通にクリアはできた。
    - 5-3 敵が強すぎてクリアできない。
    - 5-4 AI 書いてもエラーが出て実行できない。
    - 5-5 よくわかんない。どうすればいいのかわかんない。
6. このゲームをプレイしてからプログラミングに対する気持ちの変化はありましたか？
    - 6-1 楽しいものだと思えるようになった
    - 6-2 面白くなりそうかもしれない。
    - 6-3 以前と変わらず。
    - 6-4 逆に面白くないものだと思うようになった。
    - 6-5 え、これってプログラミング関係あったの？

他に何か意見や感想があれば以下に書いてください。

以下の図はアンケート結果を集計したものである。質問 5 までは右に行くほど肯定的な結果、左に行くほど否定的な結果となる。

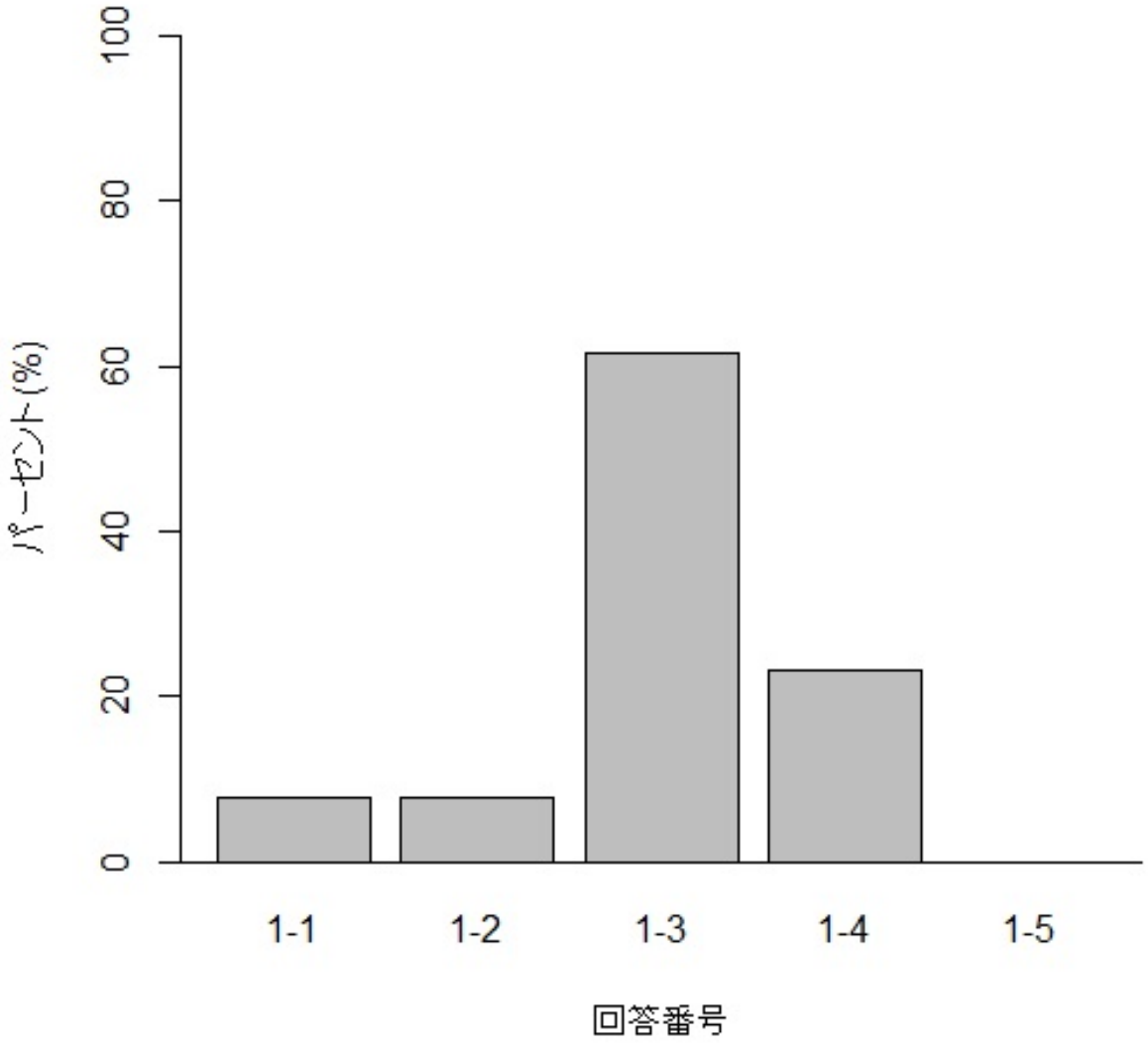

図 6　アンケート 1 プログラミングに対する印象に関する質問への解答

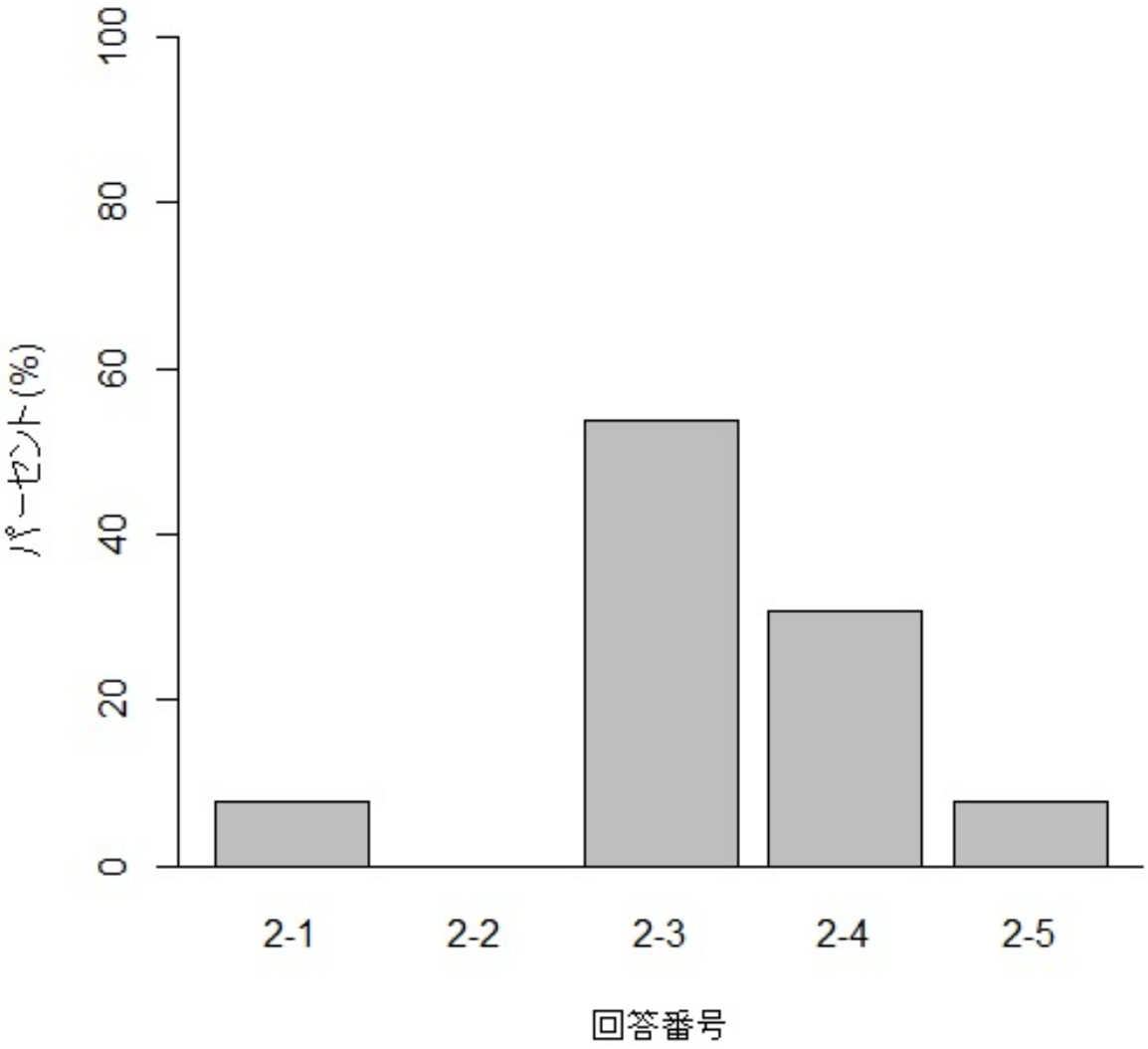

図 7　アンケート 2 プログラミングに対する能力に関する質問への解答

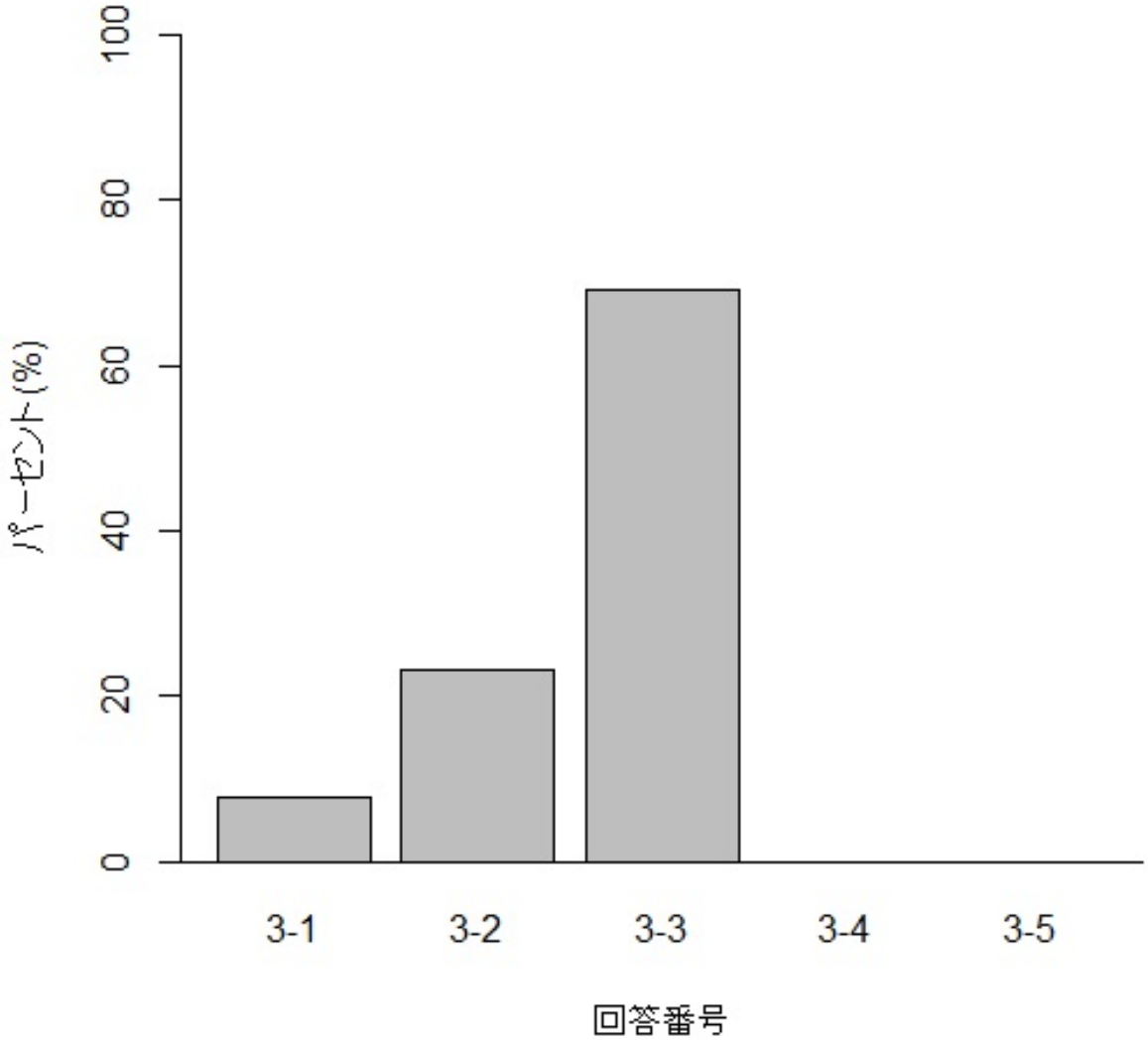

図 8　アンケート 3 ストーリーパートの没入感や依存性に関する質問への解答

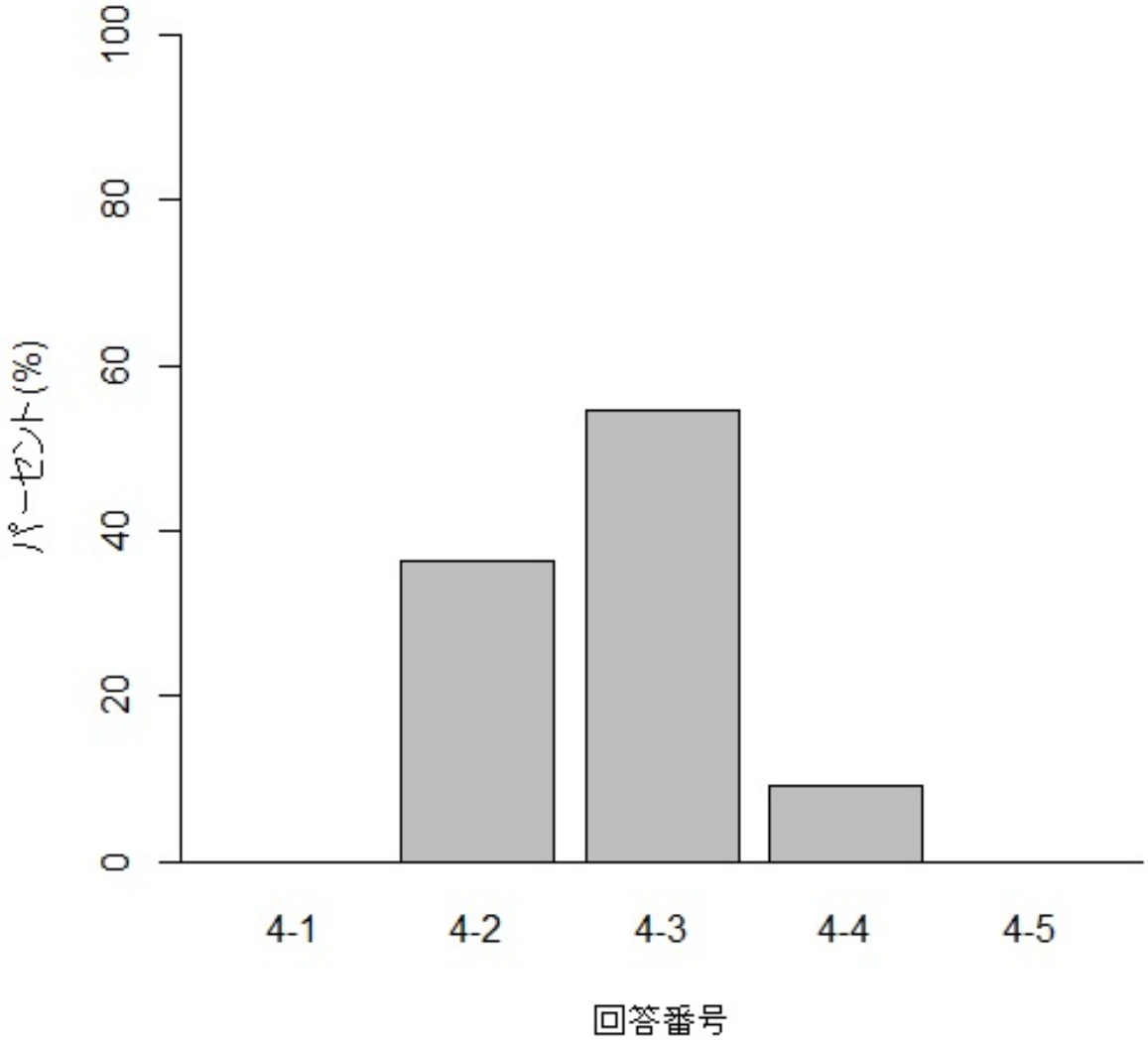

図 9　アンケート 4 コーディングパート、ゲームパートの面白さに関する質問への解答

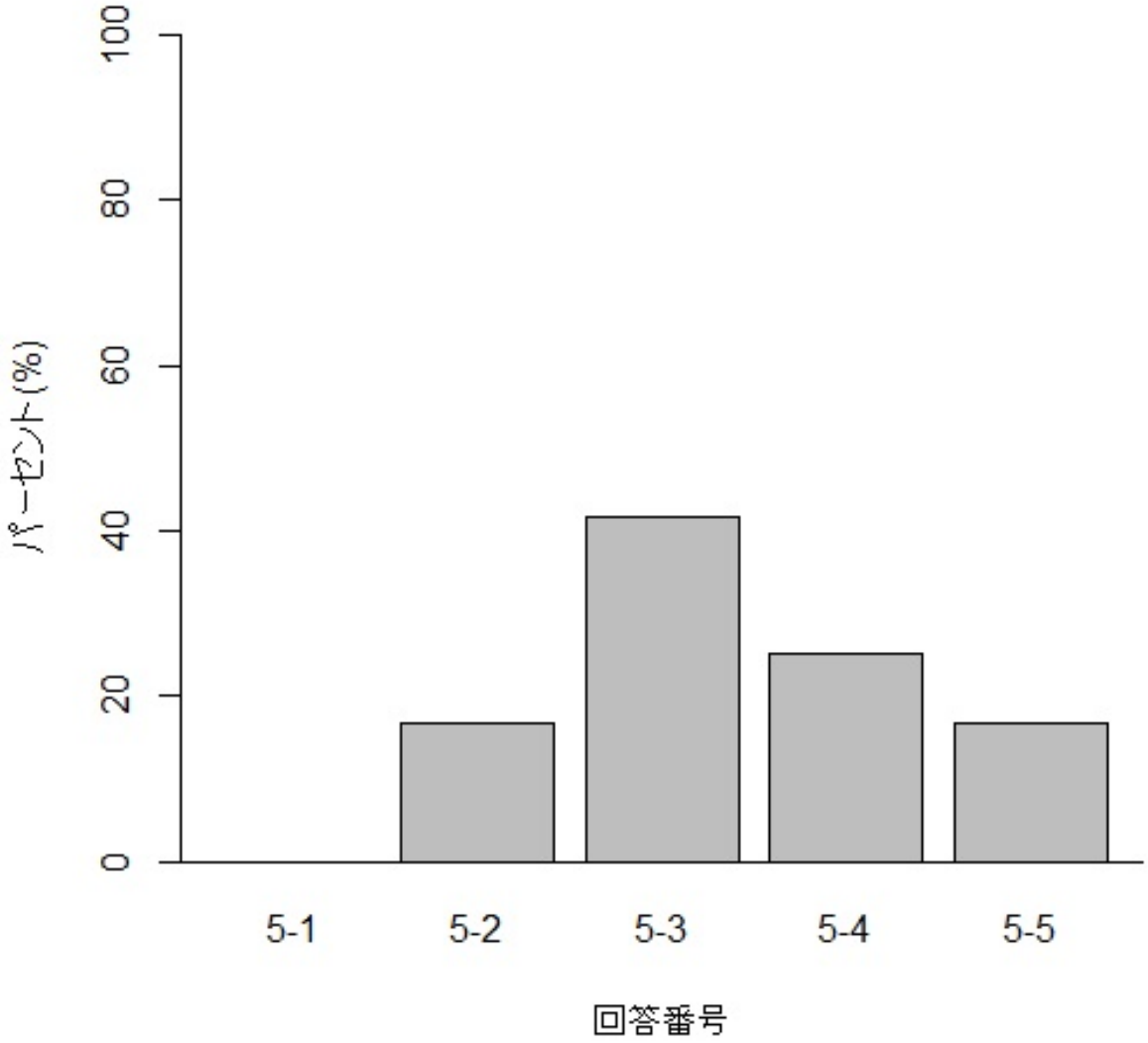

図 10　アンケート 5 ゲームの難易度に関する質問への解答

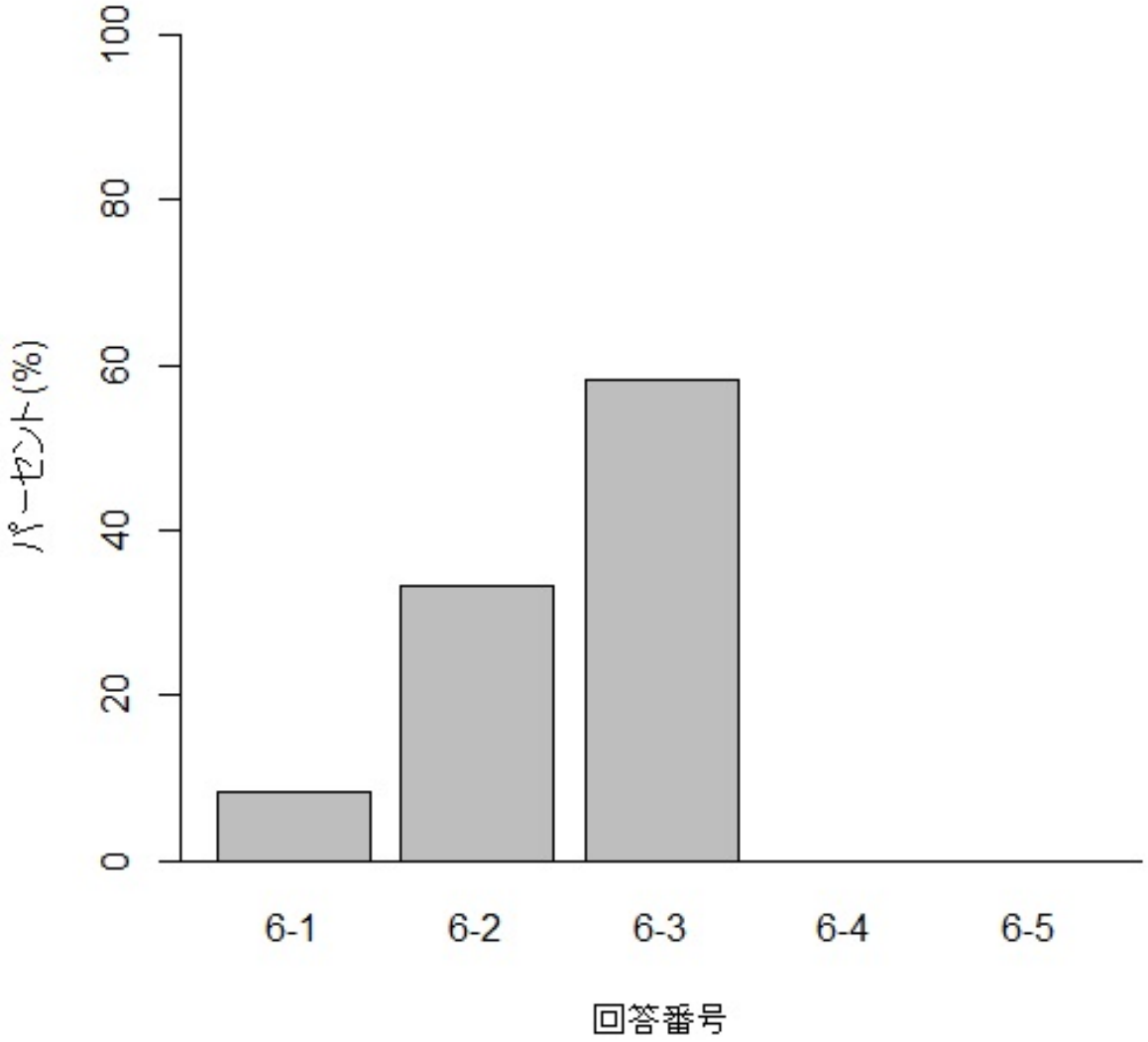

図 11　アンケート 6 ゲームプレイの前後でプログラミングに対する印象の変化に関する質問への解答

また、アンケート 7 での感想や意見は次のとおりであった。

- Excepreesもそうだが Microsoft Visual C++ は起動が遅いと膨大なセキュリティアップデートするイライラ欠陥ソフトです。Microsoft Visual C++ はダブルクリックでコンパイルしてアプリケーション起動できる機能があったのにそれで起動できなかったから面倒でした。この点を改善してほしい。
- いつもと違うレポートだからおもしろかった。ときどきこういうのをしたいと思いました。
- いつもと違って、ゲームの簡単なプログラミングなどは面白かった。ゲーム自体はクリアできませんでしたが・・・
- AI の組み方がよくわかりませんでした。
- TBS のゲームを作ってくれたるぅたそ司令官には∠ (`・ω・´) ケイレイ

ここでは、以上のアンケート結果より、その結果に対する考察を述べていく。

5.1.1 論理的思考力の向上に対する効果

まず図 7 を見ると、殆どの回答がプログラミングが苦手、もしくは普通のどちらかであった。そのため論理的な思考力の向上は必要である。図 11 では解答 4,5 が全くなく、おおよそプログラミングが楽しくなりそうだという回答を得ることができた。この事から、論理的思考力の向上はあったものと思われる。しかしながら、今回の研究ではこのゲームのプレイの前後で定量的に論理的思考力を測っていなかったため論理的思考力が定量的に向上したとは断言することはできない。

5.1.2 モチベーションの向上に対する効果

図 6,7 を見ると、プログラミングが好きだという回答はあまりなく、どちらでもないか嫌いだという回答が多い傾向にあることが分かる。それに関連してプログラミングが得意だという回答も殆ど無く、全体的にプログラミングに対してあまり良い印象を持っていないことが分かる。まずはこの苦手意識を払拭する必要がある。

図 8 のストーリーパートの没入感に対する質問への解答ではどちらでもないという解答が多かったものの面白く無いという意見はなく、逆に面白かったという意見は少数であるものの得ることができた。この事からゲームストーリーを純粋に楽しむことで自然とプログラミング学習を行い、なおかつモチベーションの向上を図ることが出来ているといえる。図 9 では、概ねポジティブな結果を得ることができたため、プログラミングに対して苦手意識を持っていても単純にプログラミングをこなしていくよりも、ゲーム内で自然にプログラミングを行うことでモチベーションを維持したまま取り組めている。さらにストーリーパート、チュートリアルパートを引き継いでのコーディング、ゲームパートになっているため目的が明確であることもモチベーションを維持してコーディングに取り組めた理由の一つになっていると考えた。また、記述式で回答してもらった質問 7 の回答ではいつもと違う課題形式で面白かったという意見が得られたことからもモチベーションを向上させることができたと推測できる。以上から、プログラミングをうまくゲーム内に取り込むことでプログラミングに対する苦手意識を抑え、モチベーションの向上を図ることができたといえる。

5.1.3 難易度の設定に関する問題

図 10 の質問 5 では難しかった、最後まで進めなかった、という回答が多くあった。この原因としてはゲーム自体の難易度設定が難しすぎた事が考えられる。よって敵が強すぎて、適切な AI プログラムを記述することが出来なかったという部分ではアルゴリズムを考える能力の要求が高すぎたのが一点、AI のコーディングに対するヒントの量が少なかったことが一点、この二点が難易度設定の問題として考えられる。そのため最後まで進めずに途中で挫折してしまった人が多く、その人がどちらでもないという意見を多く回答していたと推測できる。

5.1.4 ゲームフローチャートの明確さに関する問題

図 10 の回答で、やり方が分からなかったという回答の原因として、ゲーム進行のフローチャートが明確でないことも考えられる。現在のゲーム仕様ではストーリーパート、チュートリアルパートまで進めた後一回ゲームプログラムが終了し、そこからエディタに操作が移って、AI プログラムの編集が終了したらコンパイルして、ゲームパートに移らなければならないため、パート間のつながりがわかりにくくなってしまっている。そのためエディタで編集した後どうすればいいのかわからなかったり、コンパイルを通す方法がわからなかったりしたためにやり方が分からなかったという回答があったのではないかと推測できる。

6　まとめ

以上の結果と考察により、モチベーション向上に対する効果はあり、純粋に楽しみながらゲームをプレイし、AIプログラミングもそのモチベーションを維持したまま取り組むことができたという結果を得ることができた。しかし、論理的思考力の向上に対する効果は定量的に測定することが出来ず効果があったかどうかを定量的に示すことは出来なかった。またゲーム自体の難易度が難しい、ゲーム進行のフローチャートが不明瞭である点から、プレイした学生がゲームを最後まで進めることが出来ずに行き詰ってしまうことが確認できた。

今後の課題としては、以下の項目が考えられる。

- ゲーム難易度の調整
- ゲーム進行フローチャートの明確化

6.1　ゲーム難易度の調整

結果と考察よりゲーム難易度が難しすぎ、AI プログラムを構築できないという結果に至ったため、よりわかりやすくプログラムを構築するためのヒントを増やし、更に細かい段階に分けて要求する難易度を上げていくことでゲーム難易度の調整をはかる。そうすることにより途中で行き詰ってしまう可能性が低くなり、モチベーションの低下を防ぐことが出来ると推測できる。

6.2　ゲーム進行フローチャートの明確化

現在の仕様で分離しているコーディングパートとバトルパートをつなげ、プレイする学生ができるだけ少ない動作で連続してゲームをプレイ出来るようにゲーム進行フローチャートを明確にする。これによりゲームプレイする上で煩わしい操作をせずにゲームに没頭できるため、さらなるモチベーションの向上をはかることが出来ると考えられる。

謝辞

本研究を進めていく上で、大垣斉講師に御指導及び御協力頂きました。また、研究室の OB として指導して下さった水野貴弘先輩や、東川諒央先輩には御助言を賜りました。ここに深く感謝の意を表します。

参考文献

[1] 辻大地. プログラミング入門支援ゲームの試作と評価. Master's thesis, 大阪産業大学, 2010.

[2] 東川涼央. ゲームを題材としたプログラミング教育支援システムの開発. 大阪産業大学, 2009.

[3] 尾崎敏範. ボードゲームの戦略プログラミングを題材とした java 演習の支援システムの開発. 情報処理学会 研究報告, 2006.

[4] 柴田望洋. 新板 明快 C 言語 入門編. ソフトバンククリエイティブ株式会社, 2004.

[5] Sqlite に c 言語でアクセス 基礎編. http://idocsq.net/page/505.

[6] Sqlite home page. http://www.sqlite.org/index.html.

付録 A　付録

A.1　ソースコード

A.1.1　本プログラム

リスト 1　sotuken.c

```
1
2
3  #define STORY story.txt
4  #define ONELINE 500
5  #define BUFFSIZE 1000
6  #define WEAPON_SIZE 1000
7  #define TAB \t\t\t\t
8  #define DISTANCE 300
9
10 #include <stdio.h>
11 #include <stdlib.h>
12 #include <conio.h>
13 #include <string.h>
14 #include <Windows.h>
15 #include <time.h>
16 #include "sqlite3.h"
17 #include <mbstring.h>
18
19 #include "AI.c"
20 #include "AIStage1.c"
21 #include "AIenemy.c"
22
23 enum ERRORTAG{
24   STORY_FILE_OPEN_SUCCESS,
25   STORY_FILE_OPEN_FAILED,
26   DT_FILE_OPEN_SUCCESS,
27   DT_FILE_OPEN_FAILED
28 };enum ERRORTAG etag;
29
30 enum DT_NUMBER{
31   DT_NUMBER1,
32   DT_NUMBER2
33 };enum DT_NUMBER dtnum;
34
35 enum HITFLAG{
36   HIT,
37   MISS
38 };enum HITFLAG hitflag;
39
40 enum DOA{
41   CONTINUE,
42   GAMEOVER
43 };enum DOA doa;
44
45 typedef struct{
46   int no;
47    unsigned char name[WEAPON_SIZE];
48   int power;
49   int time;
50   int hit;
51   int range;
52   int slot;
53 } weapon_t;
54   weapon_t weapon[WEAPON_SIZE] = {0,NULL,0,0,0,0,0};
55   weapon_t DT1weapon = {0,NULL,0,0,0,0,0};
56   weapon_t DT1weapon2 = {0,NULL,0,0,0,0,0};
57   weapon_t DT2weapon = {0,NULL,0,0,0,0,0};
58   weapon_t DT2weapon2 = {0,NULL,0,0,0,0,0};
59
60 typedef struct{
61   int no;
62   unsigned char name[WEAPON_SIZE];
63   int defencepoint;
64   int armorpoint;
```

```
65 |   int slot;
66 | } armor_t;
67 |   armor_t armor[WEAPON_SIZE] = {0,NULL,0,0,0};
68 |   armor_t DT1armor = {0,NULL,0,0,0};
69 |   armor_t DT2armor = {0,NULL,0,0,0};
70 |
71 | typedef struct{
72 |   int no;
73 |   unsigned char name[WEAPON_SIZE];
74 |   int outputpower;
75 |   int energypower;
76 | } engine_t;
77 |   engine_t engine[WEAPON_SIZE] = {0,NULL,0,0,0};
78 |   engine_t DT1engine = {0,NULL,0,0};
79 |   engine_t DT2engine = {0,NULL,0,0};
80 |
81 | typedef struct{
82 |   unsigned char name[WEAPON_SIZE];
83 |   int stage;
84 |   int weapon1;
85 |   int weapon2;
86 |   int weapon3;
87 |   int armor;
88 |   int engine;
89 | } dtdata_t;
90 |   dtdata_t dtdata = {NULL,0,0,0,0,0,0};
91 |   dtdata_t dtdata2 = {NULL,0,0,0,0,0,0};
92 |
93 |
94 | char story[5000];
95 | char story2[5000];
96 | char matome[5000];
97 | char AAbuff[5000];
98 | char AAbuff2[5000];
99 |
100 | char *talkbuff[500];
101 | char *talkbuff2[500];
102 | int DTAP1,DTAP2 = 0;
103 | int atb1,atb2 = 0;
104 | int winner = 0;
105 | int turn,turn_e = 0;
106 | char storynumber,AAnumber = NULL;
107 |
108 | int DT1place[2] = {0,1};
109 | int DT2place[2] = {0,1};
110 | int Distance = 300;
111 | int DT1Energy = 0;
112 | int DT2Energy = 0;
113 |
114 |
115 | //NNNO//////////////////StoryTeller///////////////////////
116 | int StoryReader(char *story ,const char *storynum)
117 | {
118 |   FILE *sfp;
119 |   char buff[BUFFSIZE];
120 |   char *bp = buff;
121 |
122 |   sfp = fopen(storynum,"r");
123 |   if(sfp == NULL)
124 |   {
125 |     return STORY_FILE_OPEN_FAILED;
126 |   }
127 |   else
128 |   {
129 |     while(fgets(bp,ONELINE,sfp) != NULL)
130 |       strcat(story,bp);
131 |
132 |     fclose(sfp);
133 |     return STORY_FILE_OPEN_SUCCESS;
134 |   }
135 | }
136 |
137 | int AAReader(char *AAbuff ,const char *AAnum)
138 | {
139 |   FILE *sfp;
```

```
140 |   char buff[BUFFSIZE] = {NULL};
141 |   char *bp = buff;
142 |
143 |   sfp = fopen(AAnum,"r");
144 |   if(sfp == NULL)
145 |   {
146 |     return STORY_FILE_OPEN_FAILED;
147 |   }
148 |   else
149 |   {
150 |     AAbuff == NULL;
151 |     while(fgets(bp,ONELINE,sfp) != NULL)
152 |     {
153 |       strcat(AAbuff,bp);
154 |     }
155 |
156 |     fclose(sfp);
157 |     return STORY_FILE_OPEN_SUCCESS;
158 |   }
159 | }
160 |
161 |
162 |
163 | int StoryWriter(char *story)
164 | {
165 |   char key[BUFFSIZE] = {NULL};
166 |   char *tp;
167 |   int i = 0;
168 |
169 |   tp = strtok(story, "\n");
170 |   puts(tp);
171 |   puts("▽");
172 |   //Beep(3000,100);
173 |   Sleep(1800);
174 |   while(tp != NULL){
175 |     tp = strtok( NULL,"\n" );
176 |         if ( tp != NULL ) puts( tp );
177 |     puts("▽");
178 |       //Beep(3000,100);
179 |     Sleep(1800);
180 |   }
181 | }
182 |
183 | int StoryTalkSet(char *story ,char **talkbuff)
184 | {
185 |   char key[BUFFSIZE] = {NULL};
186 |   char *tp;
187 |   int i = 0;
188 |   tp = strtok(story, "\n");
189 |   talkbuff[0] = tp;
190 |   i++;
191 |   while(tp != NULL){
192 |     tp = strtok(NULL,"\n");
193 |     talkbuff[i] = tp;
194 |     i++;
195 |   }
196 | }
197 | int AAWriter(char *AAbuff)
198 | {
199 |   printf("%s",AAbuff);
200 |   return 0;
201 | }
202 |
203 | void StoryTeller(const char *storynumber,const char *AAnumber )
204 | {
205 |   if(StoryReader(story,storynumber) == STORY_FILE_OPEN_SUCCESS)
206 |   StoryWriter(story);
207 |   else if(StoryReader(story,storynumber) == STORY_FILE_OPEN_FAILED)
208 |   {
209 |     printf("Story file open failed.\n"); exit;
210 |   }
211 |   if(AAnumber != NULL)
212 |   {
213 |     AAReader(AAbuff,AAnumber);
214 |     AAWriter(AAbuff);
```

```
  }
  else;
}
void init()
{
  *story = NULL;
  *story2 = NULL;
  *AAbuff = NULL;
  *AAbuff2 = NULL;
  *talkbuff = NULL;
  *talkbuff2 = NULL;
  DTAP1,DTAP2 = 0;
  atb1,atb2 = 0;
  winner = 0;
  turn = 0;
  storynumber,AAnumber = NULL;
}

void Flash(int time)
{
  int p,q=0;
  for(q=0;q<time;q++)
  {
    Sleep(300);
    system("cls");
    for(p=0;p<30;p++)
      printf("■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■\n
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■\n");
    Sleep(300);
    system("cls");
  }

}

void Kisei()
{
  int i=0;
  printf("21 名無し@DT 2038/10/06(火) 88:88:88 ID:daijLrn2d \n
ふははははは、こんなものは規制、規制だ！ \n何が匿名の力だ、こんなものねじ伏せてやるううううう");
  Sleep(2000);
  //Beep(400,20000);
  for(i=0;i<3;i++)
  {
    system("cls");
    Matome("matome6.txt");
    Sleep(50);
  }
}


int AIview(const char *AI_file )
{
  FILE *fp;
  char buff[500] = {NULL};
  char string[1000] = {NULL};
  int i =0;
  fp = fopen(AI_file,"r");
  printf("---------------------------------------------\n");
  while(fgets(buff,500,fp) != NULL)
  {
    i++;
    printf("%2d|%s",i,buff);
  }
  return 0;
}
int AIshow(const char *AI_file )
{
  FILE *fp;
  char buff[500] = {NULL};
  char string[1000] = {NULL};
  int i =0;
  fp = fopen(AI_file,"r");
  system("cls");
  Sleep(100);
  printf("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t        AI DATA LOADING...");
```

```
  Sleep(1500);
  system("cls");
  printf("――――――――――――――――――――――――――――――――――――――――\n");
  while(fgets(buff,500,fp) != NULL)
  {
    i++;
    printf("%2d|%s",i,buff);
    Sleep(50);
  }
  Sleep(100);
  printf("\n\n\n\t\t\t        AI DATA LOAD COMPLETE!\n");
  Sleep(1500);
  return 0;
}

void AIedit()
{
    printf("AIエディタを起動中。うぇいとぷりーず。。。\n");
    system("start TeraPad.exe AI.c");
}

void StoryWindow1()
{
  int i,j = 0;
  int p,q = 0;

  init();
  StoryReader(story2,"story1yaruo.txt");
  StoryTalkSet(story2,talkbuff2);
  StoryReader(story,"story1yaranai.txt");
  StoryTalkSet(story,talkbuff);
  AAReader(AAbuff2,"br.txt");

  for(i=0;i<90;i++)
  {
    if(i == 70)
    {
      Flash(2);
    }
    else
    {
      system("cls");
      if(i==8)
      {
        *AAbuff2 = NULL;
        AAReader(AAbuff2,"AA4.txt");
      }
      else if(i==38)
      {
        *AAbuff2 = NULL;
        AAReader(AAbuff2,"br.txt");
      }
      else if(i==41)
      {
        *AAbuff2 = NULL;
        AAReader(AAbuff2,"AA7.txt");
      }
      else if(i==47)
      {
        *AAbuff2 = NULL;
        AAReader(AAbuff2,"AA4.txt");
      }
        AAWriter(AAbuff2);
      printf("\n――――――――――――――――――――――――――――――∧―――――――\n");
      printf("|    %s\n",talkbuff[i]);
      printf("――――――――――――――――――――――――――――――――――――――――\n");
      if(i >= 60)
      {
        printf("\n\n");
        printf("    ∧_∧             _∧∧_  _ \n");
        printf("   ( ・∀・)        ＼(  澁    ,,／\n");
        printf("    ▽田田田▽          ⊂#>▽<#⊃ \n");
        printf("   ●【=====】●          (  V  ) ～\n");
        printf("     [[]  []]            し ´ J \n");
        printf("    〔;;;〕;;〕\n");
```

```
365 |       }
366 |       else
367 |         printf("\n\n\n\n\n\n\n\n\n");
368 |       printf("―――――――――――――――――――――――――――――――――――――――\n");
369 |       printf("|    %s\n",talkbuff2[i]);
370 |       printf("―――――V―――――――――――――――――――――――――――――――――\n");
371 |       *AAbuff = NULL;
372 |       //if(i>=11 && i<=12 ||i>=14 && i<=18)AAReader(AAbuff,"AA5.txt");
373 |       //else AAReader(AAbuff,"AA3.txt");
374 |       AAReader(AAbuff,"AA5.txt");
375 |
376 |       AAWriter(AAbuff);
377 |       printf("\npress any key   %d",i+1);getch();
378 |     }
379 |   }
380 |   system("cls");
381 | }
382 |
383 |
384 | void StoryWindow2()
385 | {
386 |   int i,j = 0;
387 |
388 |   init();
389 |   StoryReader(story2,"story2yaruo.txt");
390 |   StoryTalkSet(story2,talkbuff2);
391 |   StoryReader(story,"story2yaranai.txt");
392 |   StoryTalkSet(story,talkbuff);
393 |   AAReader(AAbuff2,"AA4.txt");
394 |
395 |
396 |   for(i=0;i<21;i++)
397 |   {
398 |     system("cls");
399 |     AAWriter(AAbuff2);
400 |     printf("―――――――――――――――――――――――――――――――――∧―――――\n");
401 |     printf("|    %s\n",talkbuff[i]);
402 |     printf("―――――――――――――――――――――――――――――――――――――――\n");
403 |     printf("\n\n\n\n\n\n\n\n\n");
404 |     printf("―――――――――――――――――――――――――――――――――――――――\n");
405 |     printf("|    %s\n",talkbuff2[i]);
406 |     printf("―――――V―――――――――――――――――――――――――――――――――\n");
407 |     *AAbuff = NULL;
408 |     if(i>=14 && i<=15 ||i>=17 && i<=21)AAReader(AAbuff,"AA5.txt");
409 |     else AAReader(AAbuff,"AA3.txt");
410 |
411 |     AAWriter(AAbuff);
412 |     if(i==20)printf("\nWait please...");
413 |     else printf("\npress any key");getch();
414 |   }
415 | }
416 | void AIWindow2()
417 | {
418 |   int i=0;
419 |   init();
420 |   StoryReader(story2,"story2setumei.txt");
421 |   StoryTalkSet(story2,talkbuff2);
422 |   AAReader(AAbuff2,"AA6.txt");
423 |   for(i=0;talkbuff2[i]!=NULL;i++)
424 |   {
425 |     system("cls");
426 |     AIview("AIsamp.c");
427 |     printf("―――――――――――――――――――――――――――――――――――――――\n");
428 |     printf("|    %s\n",talkbuff2[i]);
429 |     printf("―――――――――――――――――――――――――――――――V―――――――\n");
430 |     AAWriter(AAbuff2);
431 |     printf("press any key");getch();
432 |   }
433 | }
434 |
435 | void AIWindowB()
436 | {
437 |   int i=0;
438 |   init();
439 |   StoryReader(story2,"battletext.txt");
```

```
440 |   StoryTalkSet(story2,talkbuff2);
441 |   AAReader(AAbuff2,"AA6.txt");
442 |   for(i=0;talkbuff2[i]!=NULL;i++)
443 |   {
444 |     system("cls");
445 |     AIview("battle.txt");
446 |     printf("――――――――――――――――――――――――――――――――――――――――\n");
447 |     printf("|    %s\n",talkbuff2[i]);
448 |     printf("――――――――――――――――――――――――――――――――V―――――――\n");
449 |     AAWriter(AAbuff2);
450 |     printf("press any key");getch();
451 |   }
452 | }
453 |
454 | void StoryWindow3()
455 | {
456 |   int i,j = 0;
457 |
458 |   init();
459 |   StoryReader(story2,"story3yaruo.txt");
460 |   StoryTalkSet(story2,talkbuff2);
461 |   StoryReader(story,"story3yaranai.txt");
462 |   StoryTalkSet(story,talkbuff);
463 |   AAReader(AAbuff2,"AA4.txt");
464 |
465 |
466 |   for(i=0;i<30;i++)
467 |   {
468 |     system("cls");
469 |     AAWriter(AAbuff2);
470 |     printf("――――――――――――――――――――――――――――――――――^―――――\n");
471 |     printf("|    %s\n",talkbuff[i]);
472 |     printf("――――――――――――――――――――――――――――――――――――――――\n");
473 |     printf("\n\n\n\n\n\n\n\n\n\n");
474 |     printf("――――――――――――――――――――――――――――――――――――――――\n");
475 |     printf("|    %s\n",talkbuff2[i]);
476 |     printf("―――――V――――――――――――――――――――――――――――――――\n");
477 |     *AAbuff = NULL;
478 |     if(i>=11)AAReader(AAbuff,"AA5.txt");
479 |     else AAReader(AAbuff,"AA3.txt");
480 |
481 |     AAWriter(AAbuff);
482 |     if(i==20)printf("\nWait please...");
483 |     else printf("\npress any key");getch();
484 |   }
485 | }
486 |
487 | void AIWindow3()
488 | {
489 |   int i=0;
490 |   init();
491 |   StoryReader(story2,"story3setumei.txt");
492 |   StoryTalkSet(story2,talkbuff2);
493 |   AAReader(AAbuff2,"AA6.txt");
494 |   for(i=0;talkbuff2[i]!=NULL;i++)
495 |   {
496 |     system("cls");
497 |     if(i>=57) AIview("AIStage3.c");
498 |     else      AIview("AI.c");
499 |
500 |     printf("――――――――――――――――――――――――――――――――――――――――\n");
501 |     printf("|    %s\n",talkbuff2[i]);
502 |     printf("――――――――――――――――――――――――――――――――V―――――――\n");
503 |     AAWriter(AAbuff2);
504 |     printf("press any key");getch();
505 |   }
506 | }
507 |
508 |
509 | void StoryWindow4()
510 | {
511 |   int i,j = 0;
512 |
513 |   init();
514 |   StoryReader(story2,"story4yaruo.txt");
```

```
515 |   StoryTalkSet(story2,talkbuff2);
516 |   StoryReader(story,"story4yaranai.txt");
517 |   StoryTalkSet(story,talkbuff);
518 |   AAReader(AAbuff2,"AA4.txt");
519 |
520 |
521 |   for(i=0;i<56;i++)
522 |   {
523 |     system("cls");
524 |     AAWriter(AAbuff2);
525 |     printf("―――――――――――――――――――――――――――――――∧―――――\n");
526 |     printf("|    %s\n",talkbuff[i]);
527 |     printf("―――――――――――――――――――――――――――――――――――――――\n");
528 |     printf("\n\n\n\n\n\n\n\n\n");
529 |     printf("―――――――――――――――――――――――――――――――――――――――\n");
530 |     printf("|    %s\n",talkbuff2[i]);
531 |     printf("―――――V―――――――――――――――――――――――――――――――\n");
532 |     *AAbuff = NULL;
533 |     if(i>=6&&i<=15)AAReader(AAbuff,"AA8.txt");
534 |     else AAReader(AAbuff,"AA5.txt");
535 |
536 |     AAWriter(AAbuff);
537 |     if(i==57)printf("\nWait please...");
538 |     else printf("\npress any key");getch();
539 |   }
540 | }
541 |
542 | void AIWindow4()
543 | {
544 |   int i=0;
545 |   init();
546 |   StoryReader(story2,"story4setumei.txt");
547 |   StoryTalkSet(story2,talkbuff2);
548 |   AAReader(AAbuff2,"AA6.txt");
549 |   for(i=0;talkbuff2[i]!=NULL;i++)
550 |   {
551 |     system("cls");
552 |     if(i>=57) AIview("AIStage4.c");
553 |     else      AIview("AI.c");
554 |
555 |     printf("―――――――――――――――――――――――――――――――――――――――\n");
556 |     printf("|    %s\n",talkbuff2[i]);
557 |     printf("―――――――――――――――――――――――――――――――V―――――――\n");
558 |     AAWriter(AAbuff2);
559 |     printf("press any key");getch();
560 |   }
561 | }
562 | void StoryWindow5()
563 | {
564 |   int i,j = 0;
565 |
566 |   init();
567 |   StoryReader(story2,"story5yaruo.txt");
568 |   StoryTalkSet(story2,talkbuff2);
569 |   StoryReader(story,"story5yaranai.txt");
570 |   StoryTalkSet(story,talkbuff);
571 |   for(i=0;i<61;i++)
572 |   {
573 |     system("cls");
574 |     if(i>=35 && i<=46 ||  i>=57)
575 |     {
576 |       *AAbuff2 = NULL;
577 |       AAReader(AAbuff2,"AA10.txt");
578 |     }
579 |     else if(i>=46 && i<=57)
580 |     {
581 |       *AAbuff2 = NULL;
582 |       AAReader(AAbuff2,"AA9.txt");
583 |     }
584 |     else if(i<=28)
585 |     {
586 |       *AAbuff2 = NULL;
587 |       AAReader(AAbuff2,"AA4.txt");
588 |     }
589 |     else if(i==29)
```

```
590 |     {
591 |       *AAbuff2 = NULL;
592 |       AAReader(AAbuff2,"br.txt");
593 |     }
594 |     AAWriter(AAbuff2);
595 |     printf("―――――――――――――――――――――――――――――――――∧―――――\n");
596 |     printf("|    %s\n",talkbuff[i]);
597 |     printf("―――――――――――――――――――――――――――――――――――――――\n");
598 |     printf("\n\n\n\n\n\n\n\n\n");
599 |     printf("―――――――――――――――――――――――――――――――――――――――\n");
600 |     printf("|    %s\n",talkbuff2[i]);
601 |     printf("―――――V―――――――――――――――――――――――――――――――\n");
602 |     *AAbuff = NULL;
603 |     if(i<=15)AAReader(AAbuff,"AA5.txt");
604 |     else if(i>=16 && i<=32)AAReader(AAbuff,"AA3.txt");
605 |     else AAReader(AAbuff,"AA8.txt");
606 |
607 |     AAWriter(AAbuff);
608 |     if(i==60)printf("\nWait please...");
609 |     else printf("\npress any key");getch();
610 |   }
611 | }
612 |
613 |
614 | void StoryWindow6()
615 | {
616 |   int i,j = 0;
617 |
618 |   init();
619 |   StoryReader(story2,"story6yaruo.txt");
620 |   StoryTalkSet(story2,talkbuff2);
621 |   StoryReader(story,"story6yaranai.txt");
622 |   StoryTalkSet(story,talkbuff);
623 |   for(i=0;i<52;i++)
624 |   {
625 |     if(i==40)
626 |     {
627 |       system("cls");
628 |       Sleep(3000);
629 |     }
630 |     system("cls");
631 |     if(i>=0 && i<=15)
632 |     {
633 |       *AAbuff2 = NULL;
634 |       AAReader(AAbuff2,"AA10.txt");
635 |     }
636 |     else if(i>=17 && i<=35)
637 |     {
638 |       *AAbuff2 = NULL;
639 |       AAReader(AAbuff2,"AA9.txt");
640 |     }
641 |     else if(i==16 || i>=31)
642 |     {
643 |       *AAbuff2 = NULL;
644 |       AAReader(AAbuff2,"br.txt");
645 |     }
646 |     AAWriter(AAbuff2);
647 |     printf("―――――――――――――――――――――――――――――――――∧―――――\n");
648 |     printf("|    %s\n",talkbuff[i]);
649 |     printf("―――――――――――――――――――――――――――――――――――――――\n");
650 |     printf("\n\n\n\n\n\n\n\n\n");
651 |     printf("―――――――――――――――――――――――――――――――――――――――\n");
652 |     printf("|    %s\n",talkbuff2[i]);
653 |     printf("―――――V―――――――――――――――――――――――――――――――\n");
654 |     *AAbuff = NULL;
655 |     if(i>=20||i<=26)AAReader(AAbuff,"AA5.txt");
656 |     else AAReader(AAbuff,"AA8.txt");
657 |
658 |     AAWriter(AAbuff);
659 |     if(i==60)printf("\nWait please...");
660 |     else printf("\npress any key");getch();
661 |   }
662 | }
663 |
664 | void Ending()
```

```
{
  int i=0;
  init();
  StoryReader(story2,"endtext.txt");
  StoryTalkSet(story2,talkbuff2);
  AAReader(AAbuff2,"AA5.txt");
  for(i=0;talkbuff2[i]!=NULL;i++)
  {
    system("cls");
    AIview("end.txt");
    printf("――――――――――――――――――――――――――――――――――――――\n");
    printf("|    %s\n",talkbuff2[i]);
    printf("――――――――――――――――――――――――――――――V――――――――\n");
    AAWriter(AAbuff2);
    printf("press any key");getch();
  }
}

int Matome(const char *storynum)
{
  FILE *sfp;
  char buff[BUFFSIZE];
  char *bp = buff;

  sfp = fopen(storynum,"r");
  if(sfp == NULL)
  {
    return STORY_FILE_OPEN_FAILED;
  }
  else
  {
    system("cls");
    Sleep(2000);
    while(fgets(bp,ONELINE,sfp) != NULL)
      printf("%s",bp);

    fclose(sfp);
    return STORY_FILE_OPEN_SUCCESS;
  }
}

int YoN()
{
  int ans = 0;
  printf("YES = 1   NO = 0");
  while(TRUE)
  {
    scanf("%d",&ans);
    if(ans==1)
      return 1;
    else if(ans==0)
      return 0;
    else;
  }
}

int Story_Skipper()
{
  printf("直前のやる夫達の会話を聞きますか？聞かない場合はそのまま戦闘を始めます");
  if(YoN()==0)
    ;
  else if(YoN()==1)
    ;
}


//////////////////////Data_Base//////////////////////
void copydtname(const unsigned char *src)
{
  int p = 0;
  int count = 0;

  for(p=0;p<8;p++)
  {
     (dtdata.name[p]) = (src[p]);
```

```
740 |     count++;
741 |   }
742 |   *(dtdata.name+count) = (unsigned char)'\0';
743 | }
744 |
745 |
746 | int DTLoad()
747 | {
748 |   int i=0;
749 |
750 |   sqlite3 *db;
751 |   sqlite3_stmt *st_handle=NULL;
752 |
753 |   int error_code = sqlite3_open("sotuken.db",&db);
754 |   sqlite3_reset(st_handle);
755 |   sqlite3_prepare(db,"SELECT * FROM dtdata;",-1,&st_handle,NULL);
756 |   printf("DataBase error code is %d\n",error_code);
757 |   Sleep(400);
758 |   printf("Save Data Reading.\n");
759 |   Sleep(400);
760 |     sqlite3_step(st_handle);
761 |     copydtname(sqlite3_column_text(st_handle,0));
762 |     dtdata.stage    = sqlite3_column_int(st_handle,1);
763 |     dtdata.weapon1  = sqlite3_column_int(st_handle,2);
764 |     dtdata.weapon2  = sqlite3_column_int(st_handle,3);
765 |     dtdata.weapon3  = sqlite3_column_int(st_handle,4);
766 |     dtdata.armor    = sqlite3_column_int(st_handle,5);
767 |     dtdata.engine   = sqlite3_column_int(st_handle,6);
768 |
769 |   sqlite3_finalize(st_handle);
770 |   sqlite3_close(db);
771 |   printf("Save Data Read Complete.\n");
772 |   printf("DT Name:%-10s  Stage:%-1d  weapon1:%-1d  weapon2:%-1d
773 | weapon3:%-1d  armor:%-1d  engine:%-1d\n\n",dtdata.name,dtdata.stage,dtdata.weapon1,
774 | dtdata.weapon2,dtdata.weapon3,dtdata.armor,dtdata.engine);
775 |   Sleep(2000);
776 |   return error_code;
777 |
778 | }
779 | int DTSave()
780 | {
781 |   int i=0;
782 |
783 |   sqlite3 *db;
784 |   sqlite3_stmt *st_handle=NULL;
785 |
786 |   int error_code = sqlite3_open("sotuken.db",&db);
787 |   sqlite3_reset(st_handle);
788 |   sqlite3_prepare(db,"UPDATE dtdata set stage=?,weapon1=?,weapon2=?,
789 | weapon3=?,armor=?,engine=?;",-1,&st_handle,NULL);
790 |   sqlite3_bind_int(st_handle,1,dtdata.stage);
791 |   sqlite3_bind_int(st_handle,2,dtdata.weapon1);
792 |   sqlite3_bind_int(st_handle,3,dtdata.weapon2);
793 |   sqlite3_bind_int(st_handle,4,dtdata.weapon3);
794 |   sqlite3_bind_int(st_handle,5,dtdata.armor);
795 |   sqlite3_bind_int(st_handle,6,dtdata.engine);
796 |   sqlite3_step(st_handle);
797 |   sqlite3_finalize(st_handle);
798 |   sqlite3_close(db);
799 |
800 | }
801 |
802 | void copyweaponname(int i,const unsigned char *src)
803 | {
804 |   int p = 0;
805 |   int count = 0;
806 |
807 |   for(p=0;p<_mbslen(src);p++)
808 |   {
809 |      (weapon[i].name[p]) = (src[p]);
810 |     count++;
811 |   }
812 |   *(weapon[i].name+count) = (unsigned char)'\0';
813 | }
814 | int ReadWeapon()
```

```
815 | {
816 |   int i=0;
817 |
818 |   sqlite3 *db;
819 |   sqlite3_stmt *st_handle=NULL;
820 |
821 |   int error_code = sqlite3_open("sotuken.db",&db);
822 |   sqlite3_reset(st_handle);
823 |   sqlite3_prepare(db,"SELECT * FROM weapon;",-1,&st_handle,NULL);
824 |   printf("error code is %d\n",error_code);
825 |   printf("Weapon Data Reading.\n");
826 |
827 |   for(i=0;i<6;i++)
828 |   {
829 |     sqlite3_step(st_handle);
830 |     weapon[i].no    = sqlite3_column_int(st_handle,0);
831 |     copyweaponname(i,sqlite3_column_text(st_handle,1));
832 |     weapon[i].power = sqlite3_column_int(st_handle,2);
833 |     weapon[i].time  = sqlite3_column_int(st_handle,3);
834 |     weapon[i].hit   = sqlite3_column_int(st_handle,4);
835 |     weapon[i].range = sqlite3_column_int(st_handle,5);
836 |     weapon[i].slot  = sqlite3_column_int(st_handle,6);
837 |   }
838 |   sqlite3_finalize(st_handle);
839 |   sqlite3_close(db);
840 |   printf("Weapon Data Read Complete.\n");
841 |   return error_code;
842 | }
843 | void printweapon()
844 | {
845 |   int i = 0;
846 |   for(i=0;i<6;i++)
847 |   {
848 |     printf("No:%-2d  Name:%-10s  Power:%-3d  Time:%-3d  Hit:%-3d  Range:%-4d\n"
849 | ,weapon[i].no,weapon[i].name,weapon[i].power,weapon[i].time,weapon[i].hit,weapon[i].range);
850 |     Sleep(50);
851 |   }
852 | }
853 |
854 | void copyenginename(int i,const unsigned char *src)
855 | {
856 |   int p = 0;
857 |   int count = 0;
858 |
859 |   for(p=0;p<_mbslen(src);p++)
860 |   {
861 |      (engine[i].name[p]) = (src[p]);
862 |     count++;
863 |   }
864 |   *(engine[i].name+count) = (unsigned char)'\0';
865 | }
866 | int ReadEngine(void)
867 | {
868 |   int i=0;
869 |   sqlite3 *db;
870 |   sqlite3_stmt *st_handle=NULL;
871 |
872 |   sqlite3_open( "sotuken.db" ,&db );
873 |   sqlite3_reset(st_handle);
874 |   sqlite3_prepare(db,"SELECT * FROM engine;",-1,&st_handle, NULL);
875 |
876 |   for(i=0;i<7;i++)
877 |   {
878 |     sqlite3_step(st_handle);
879 |     engine[i].no      = sqlite3_column_int(st_handle,0);
880 |     copyenginename(i,sqlite3_column_text(st_handle,1));
881 |     engine[i].outputpower = sqlite3_column_int(st_handle,2);
882 |     engine[i].energypower = sqlite3_column_int(st_handle,3);
883 |   }
884 |   sqlite3_finalize(st_handle);
885 |   sqlite3_close(db);
886 |   printf("Engine Data Read Complete.\n");
887 |   return 0;
888 | }
889 | void printengine()
```

```
{
  int i = 0;
  for(i=0;i<7;i++)
  {
    printf("No:%-2d  Name:%-10s  OutputPower:%-3d  EnergyPower:%-3d\n",
engine[i].no,engine[i].name,engine[i].outputpower,engine[i].energypower);
    Sleep(50);
  }
}

void copyarmorname(int i,const unsigned char *src)
{
  int p = 0;
  int count = 0;

  for(p=0;p<_mbslen(src);p++)
  {
     (armor[i].name[p]) = (src[p]);
    count++;
  }
  *(armor[i].name+count) = (unsigned char)'\0';
}
int ReadArmor(void)
{
  int i=0;
  sqlite3 *db;
  sqlite3_stmt *st_handle=NULL;

  sqlite3_open( "sotuken.db" ,&db );
  sqlite3_reset(st_handle);
  sqlite3_prepare(db,"SELECT * FROM armor;",-1,&st_handle, NULL);
  for(i=0;i<6;i++)
  {
    sqlite3_step(st_handle);
    armor[i].no     = sqlite3_column_int(st_handle,0);
    copyarmorname(i,sqlite3_column_text(st_handle,1));
    armor[i].defencepoint = sqlite3_column_int(st_handle,2);
    armor[i].armorpoint = sqlite3_column_int(st_handle,3);
    armor[i].slot     = sqlite3_column_int(st_handle,4);
  }
  sqlite3_finalize(st_handle);
  sqlite3_close(db);
  printf("Armor Data Read Complete.\n");

  return 0;
}
void printarmor()
{
  int i = 0;
  for(i=0;i<6;i++)
  {
    printf("No:%-2d  Name:%-10s  ArmorPoint:%-3d  DefencePoint:%-3d  Slot:%-2d\n",
armor[i].no,armor[i].name,armor[i].armorpoint,armor[i].defencepoint,armor[i].slot);
    Sleep(50);
  }
}


void equipweapon(int DT2w1, int DT2w2)
{
  DT1weapon = weapon[dtdata.weapon1];
  DT1weapon2= weapon[dtdata.weapon2];
  DT2weapon = weapon[DT2w1];
  DT2weapon2 = weapon[DT2w2];
}
void equipengine(int DT2e)
{
  DT1engine = engine[dtdata.engine];
  DT2engine = engine[DT2e];
}
void equiparmor(int DT2a)
{
  DT1armor = armor[dtdata.armor];
  DT2armor = armor[DT2a];
}
```

```
965 |
966 | void equip(int DT2w1,int DT2w2,int DT2e, int DT2a)
967 | {
968 |   equipweapon(DT2w1,DT2w2);
969 |   equipengine(DT2e);
970 |   equiparmor(DT2a);
971 | }
972 |
973 | void ItemLoader()
974 | {
975 |   ReadWeapon();
976 |   printweapon();
977 |   Sleep(500);
978 |
979 |   ReadEngine();
980 |   printengine();
981 |   Sleep(500);
982 |
983 |   ReadArmor();
984 |   printarmor();
985 |   Sleep(500);
986 | }
987 |
988 |
989 | ////////////////////Game////////////////////////////
990 |
991 | void GamePlaceF()
992 | {
993 |   printf("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n");
994 | }
995 | void GamePlaceB()
996 | {
997 |   printf("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n");
998 | }
999 |
1000 | void GameWaitWindow()
1001 | {
1002 |   int i=0;
1003 |   for(i=0;i<3;i++)
1004 |   {
1005 |
1006 |   system("cls");
1007 |   Sleep(1000);
1008 |   printf("\n\n\n\n\n\n\n\n\n\n\n\t\t  Defend Tank Operation System Starting.\n");
1009 |   Sleep(1000);
1010 |   }
1011 |   system("cls");
1012 | }
1013 |
1014 | void CreateMap()
1015 | {
1016 |   int i = 0;
1017 |   Distance = abs(Distance);
1018 |   if(DT1place[1] == 1 || DT2place[1] == 1)
1019 |   {
1020 |     printf("\n\n");
1021 |     printf("    ∧__∧                ENEMY DISTANCE: %4d -------    _ ∧ ∧ _  _ \n",Distance);
1022 |     printf("   ( ・∀・)                                          ＼ ( 滋    ,,／\n");
1023 |     printf("    ▽田田田▽                                          ⊂#>▽<#⊃ \n");
1024 |     printf("   ●【=====】●                                         ( V ) 〜\n");
1025 |     printf("     [[]  []]                                          し ´ J \n");
1026 |     printf("    {;;;} ;;} \n\n\n");
1027 |   }
1028 |   else if(DT1place[1] == 0)
1029 |   {
1030 |     printf("\n\n");
1031 |     printf("      ∧＿∧  \n");
1032 |     printf("    (  ・∀・)  \n");
1033 |     printf("    ▽田田田▽              ENEMY DISTANCE: %4d -------    _ ∧ ∧ _  _ \n",Distance);
1034 |     printf("   ●【=====】●                                         ＼ ( 滋    ,,／\n");
1035 |     printf("      [[]  []]                                          ⊂#>▽<#⊃ \n");
1036 |     printf("    {;;;} ;;}                                          ( V ) 〜\n");
1037 |     printf("                                                       し ´ J \n");
1038 |     printf("    \n");
1039 |   }
```

```
1040 |
1041 |   else if(DT1place[1] == 2)
1042 |   {
1043 |     printf("\n\n");
1044 |     printf("                              ENEMY DISTANCE: %4d -------    ＿ Λ Λ _  ＿ \n",Distance);
1045 |     printf("                                                          ＼ ( 澁    ,,／\n");
1046 |     printf("      Λ ＿ Λ                                               ⊂#>▽<#⊃ \n");
1047 |     printf("    (  ・∀・)                                                ( V ) 〜\n");
1048 |     printf("    ▽田田田▽                                                し ´ J \n");
1049 |     printf("    ●【=====】● \n");
1050 |     printf("     [[]  []]  \n");
1051 |     printf("    〔;;;〕;;〕\n");
1052 |   }
1053 |
1054 |   /*if(DT1place[1] == 0)
1055 |   {
1056 |     printf("□□■");
1057 |     for(i=0;i<28;i++)printf("□");
1058 |     printf("\n");
1059 |     for(i=0;i<distance;i++)printf("□");
1060 |     printf("■");
1061 |     for(i=0;i<distance;i++)printf("□");
1062 |     printf("\n");
1063 |     for(i=0;i<distance;i++)printf("□");
1064 |     for(i=0;i<=distance;i++)printf("□");
1065 |   }
1066 |   else if(DT1place[1] == 0)
1067 |   {
1068 |
1069 |   }*/
1070 |
1071 | }
1072 | void ShowStatusDT1()
1073 | {
1074 |   printf("Defence Tank Status\n");
1075 |   printf("TURN: %d\tDEF: %d\tSPD: %d\tEN: %d / %d  WP1:
1076 | %s\tWP2: %s\t\n",turn,DT1armor.defencepoint,DT1engine.outputpower/10,
1077 | DT1Energy,DT1engine.energypower,DT1weapon.name,DT1weapon2.name);
1078 | }
1079 | void ShowStatusDT2()
1080 | {
1081 |   printf("Enemy Status  TURN: %d\n",turn_e);
1082 | }
1083 |
1084 | int DTStatus(void)
1085 | {
1086 |   int i,j = 0;
1087 |   ShowStatusDT2();
1088 |   printf("Enemy AP : %4d / %4d\t\tEnemy ATB\n",DTAP2,DT2armor.armorpoint);
1089 |   printf("0%% | ― ― ― ― ― ― ― ― ― ― | 100%%\t0%% | ― ― ― ― ― ― ― ― ― ― | 100%%\n   | ");
1090 |   for(i=0;i<(((double)((double)DTAP2/(double)DT2armor.armorpoint)*100)/5);i++)
1091 |   printf("*");
1092 |   for(j=0;j<30-(((double)((double)DTAP2/(double)DT2armor.armorpoint)*100)/5);j++)
1093 | printf(" ");
1094 |   printf("|");
1095 |   for(i=0;i<(int)(atb2/5);i++)    printf("*");
1096 |   printf("\n-------------------------------------
1097 | ----------------------------------------");
1098 |   CreateMap();
1099 |   printf("\n\n\n---------------------------------
1100 | ----------------------------------------");
1101 |
1102 |
1103 |   ShowStatusDT1();
1104 |   printf("DT AP : %d / %d\n",DTAP1,DT1armor.armorpoint);
1105 |   printf("0%% | ― ― ― ― ― ― ― ― ― ― | 100%%\n   | ");
1106 |   for(i=0;i<(((double)((double)DTAP1/(double)DT1armor.armorpoint)*100)/5);i++)
1107 |     printf("*");
1108 |   printf("\nDT ATB\n");
1109 |   printf("0%% | ― ― ― ― ― ― ― ― ― ― | 100%%\n   | ");
1110 |   for(i=0;i<(int)(atb1/5);i++)    printf("*");
1111 |
1112 |   Sleep(100);
1113 |   return 0;
1114 | }
```

```
1115 |
1116 |
1117 | enum DT_NUMBER TurnSystem()
1118 | {
1119 |   int i,j = 0;
1120 |   int spd1 = DT1engine.outputpower / 10;
1121 |   int spd2 = DT2engine.outputpower / 10;
1122 |
1123 |   while(TRUE)
1124 |   {
1125 |      atb1 += spd1;
1126 |      atb2 += spd2;
1127 |      if(DT1Energy < (DT1engine.energypower-(DT1engine.outputpower / 10)))
1128 |        DT1Energy += (DT1engine.outputpower / 10);
1129 |      else if(DT1Energy >= (DT1engine.energypower-(DT1engine.outputpower / 10))
1130 | && DT1Energy <= DT1engine.energypower)
1131 |        DT1Energy += DT1engine.energypower-DT1Energy;
1132 |
1133 |      if(DT2Energy < (DT2engine.energypower-(DT2engine.outputpower / 10)))
1134 |        DT2Energy += (DT2engine.outputpower / 10);
1135 |      else if(DT2Energy >= (DT2engine.energypower-(DT2engine.outputpower / 10))
1136 |  && DT2Energy <= DT2engine.energypower)
1137 |        DT2Energy += DT2engine.energypower-DT2Energy;
1138 |      DTStatus();
1139 |
1140 |      if(atb1 >= 100 && atb2 < 100)
1141 |      {
1142 |       atb1-=100;
1143 |       return DT_NUMBER1;
1144 |      }
1145 |      else if(atb2 >= 100 && atb1 < 100)
1146 |      {
1147 |       atb2-=100;
1148 |       return DT_NUMBER2;
1149 |      }
1150 |      else if(atb1 >= 100 && atb2 >= 100)
1151 |      {
1152 |       if(atb1 < atb2)
1153 |       {
1154 |         atb2-=100;
1155 |         return DT_NUMBER2;
1156 |       }
1157 |       else
1158 |       {
1159 |         atb1-=100;
1160 |         return DT_NUMBER1;
1161 |       }
1162 |      }
1163 |      else system("cls");
1164 |   }
1165 | }
1166 |
1167 | enum HITFLAG Hitter(weapon_t DTweapon,int enemy_place, int attack_place)
1168 | {
1169 |   int rnum = 0+(int)(rand()*(100 - 0 + 1.0)/(1.0 + RAND_MAX));
1170 |   int dishit = (Distance+1) / DT1weapon.range * 5;
1171 |   if(enemy_place == attack_place)
1172 |   {
1173 |     if(rnum > DTweapon.hit+10-dishit)
1174 |       return MISS;
1175 |     else
1176 |       return HIT;
1177 |   }
1178 |   else
1179 |   {
1180 |     if(rnum > DTweapon.hit-30-dishit)
1181 |       return MISS;
1182 |     else
1183 |       return HIT;
1184 |   }
1185 | }
1186 |
1187 | int Turn_Player(weapon_t DT1weapon,weapon_t DT2weapon,int stage)
1188 | {
1189 |   enum DT_NUMBER num = TurnSystem();
```

```
1190 |   int i = 0;
1191 |
1192 |   if(num == DT_NUMBER1)
1193 |   {
1194 |     if(dtdata.stage == 0)
1195 |     {
1196 |       mainsys1();
1197 |       turn++;
1198 |     }
1199 |     else
1200 |     {
1201 |       mainsys();
1202 |       turn++;
1203 |     }
1204 |   }
1205 |   else if(num == DT_NUMBER2)
1206 |   {
1207 |     mainsys_e(stage);
1208 |     turn_e++;
1209 |   }
1210 |   Sleep(1200);
1211 |   system("cls");
1212 |   return 0;
1213 | }
1214 |
1215 | char* placename(int DTplace)
1216 | {
1217 |   if(DTplace == 0)
1218 |     return "Left";
1219 |   else if(DTplace == 1)
1220 |     return "Center";
1221 |   else
1222 |     return "Right";
1223 | }
1224 |
1225 | int SetAP()
1226 | {
1227 |   DTAP1 = DT1armor.armorpoint;
1228 |   DTAP2 = DT2armor.armorpoint;
1229 |
1230 |   DT1Energy = DT1engine.energypower;
1231 |   DT2Energy = DT2engine.energypower;
1232 | }
1233 |
1234 | void destroy(char *losename)
1235 | {
1236 |   printf("%s is destroy! \n",losename);
1237 | }
1238 |
1239 | void GameOver()
1240 | {
1241 |   printf("その後、やる夫達の行方を知るものは誰もいなかった…\n");
1242 |   printf("そしてこの国から全ての萌が消え失せた・・・\n\n");
1243 |   printf("やる夫達のやり取りにヒントが隠されている。よく読みなおしてみよう！\n");
1244 | }
1245 |
1246 | ////////////////////AI_DTSystem//////////////////////////////
1247 | int Move_Forward(void)
1248 | {
1249 |   if(DT1Energy >= 30)
1250 |   {
1251 |     Distance -= 20+(DT1engine.outputpower/10);
1252 |     printf("Move Forward\n");
1253 |     Sleep(50);
1254 |     DT1Energy -= 30;
1255 |     return 0;
1256 |   }
1257 |   else
1258 |   {
1259 |     printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
1260 |     Sleep(100);
1261 |     return 0;
1262 |   }
1263 | }
1264 | int Boost_Forward(void)
```

```
{
  if(DT1Energy >= 50)
  {
    Distance -= 30+(DT1engine.outputpower/5);
    printf("Boost Forward!\n");
    Sleep(100);
    DT1Energy -= 50;
    return 0;
  }
  else
  {
    printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
    Sleep(100);
    return 0;
  }
}
int Move_Back(void)
{
  if(DT1Energy >= 30)
  {
    Distance += 20+(DT1engine.outputpower/10);
    printf("Move Back\n");
    Sleep(50);
    DT1Energy -= 30;
    return 0;
  }
  else
  {
    printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
    Sleep(100);
    return 0;
  }
}
int Boost_Back(void)
{
  if(DT1Energy >= 50)
  {
    Distance += 30+(DT1engine.outputpower/5);
    printf("Boost Back!\n");
    Sleep(100);
    DT1Energy -= 50;
    return 0;
  }
  else
  {
    printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
    Sleep(100);
    return 0;
  }
}

int Step_Left(void)
{
  if(DT1Energy >= 20)
  {
    DT1place[1] = 0;
    printf("Step Left\n");
    Sleep(50);
    DT1Energy -= 20;
    return 0;
  }
  else
  {
    printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
    Sleep(100);
    return 0;
  }
}

int Step_Right(void)
{
  if(DT1Energy >= 20)
  {
    DT1place[1] = 2;
    printf("Step Right\n");
```

```
1340 |     Sleep(50);
1341 |     DT1Energy -= 20;
1342 |     return 0;
1343 |   }
1344 |   else
1345 |   {
1346 |     printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
1347 |     Sleep(100);
1348 |     return 0;
1349 |   }
1350 | }
1351 |
1352 | int Step_Center(void)
1353 | {
1354 |   if(DT1Energy >= 20)
1355 |   {
1356 |     DT1place[1] = 1;
1357 |     printf("Step Center\n");
1358 |     Sleep(50);
1359 |     DT1Energy -= 20;
1360 |     return 0;
1361 |   }
1362 |   else
1363 |   {
1364 |     printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
1365 |     Sleep(100);
1366 |     return 0;
1367 |   }
1368 | }
1369 |
1370 |
1371 |
1372 |
1373 | //////////////////////AI_FireControl//////////////////////////
1374 | int Rifle_Shoot(int place)
1375 | {
1376 |   int i = 0;
1377 |   int damage = 0;
1378 |   int Hit = 0;
1379 |   if(DT1Energy >= 30)
1380 |   {
1381 |     printf("\n■LOCKED■\n");
1382 |     //Beep(800,500);//Beep(800,500);
1383 |     Sleep(300);
1384 |     printf("Shoot %s. %s\n",DT1weapon.name,placename(place));
1385 |     DT1Energy -= 30;
1386 |     for(i=0;i<DT1weapon.time;i++)
1387 |     {
1388 |       if(Hit = Hitter(DT1weapon,DT2place[1],place) == HIT)
1389 |       {
1390 |         damage = ((DT1weapon.power - DT2armor.defencepoint +
1391 | (DT1engine.outputpower/20))*(1.5-((double)Distance/(double)DT1weapon.range)));
1392 |         DTAP2 -= damage;
1393 |         printf("Hit.damage%dp ",damage);
1394 |         Sleep(100);
1395 |         //Beep(200,500);
1396 |       }
1397 |       else if(Hitter(DT1weapon,DT2place[1],place) == MISS)
1398 |       {
1399 |         printf("Miss. ");
1400 |         Sleep(100);
1401 |       }
1402 |     }
1403 |   }
1404 |   else
1405 |   {
1406 |     printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
1407 |     return 0;
1408 |   }
1409 |   return 0;
1410 | }
1411 |
1412 | int MachineGun_Shoot(int place)
1413 | {
1414 |   int i = 0;
```

```
1415 |   int damage = 0;
1416 |   int Hit = 0;
1417 |   if(DT1Energy >= 30)
1418 |   {
1419 |     printf("\n■LOCKED■\n");
1420 |     //Beep(800,500);//Beep(800,500);
1421 |     Sleep(300);
1422 |     printf("Shoot %s. %s\n",DT1weapon.name,placename(place));
1423 |     DT1Energy -= 30;
1424 |     for(i=0;i<DT1weapon.time;i++)
1425 |     {
1426 |       if(Hit = Hitter(DT1weapon,DT2place[1],place) == HIT)
1427 |       {
1428 |         damage = ((DT1weapon.power - DT2armor.defencepoint +
1429 | (DT1engine.outputpower/50))*(1.05-((double)Distance/(double)DT1weapon.range)));
1430 |         DTAP2 -= damage;
1431 |         printf("Hit.damage%dp ",damage);
1432 |         Sleep(100);
1433 |         //Beep(200,200);
1434 |       }
1435 |       else if(Hitter(DT1weapon,DT2place[1],place) == MISS)
1436 |       {
1437 |         printf("Miss. ");
1438 |         Sleep(100);
1439 |       }
1440 |     }
1441 |   }
1442 |   else
1443 |   {
1444 |     printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
1445 |     return 0;
1446 |   }
1447 |   return 0;
1448 |
1449 | }
1450 |
1451 | int Blade_Attack(int place)
1452 | {
1453 |   int i = 0;
1454 |   int damage = 0;
1455 |   int Hit = 0;
1456 |   if(DT1Energy >= 20)
1457 |   {
1458 |     printf("\n■LOCKED■\n");
1459 |     //Beep(800,500);//Beep(800,500);
1460 |     Sleep(300);
1461 |     printf("Attack! %s. %s\n",DT1weapon2.name,placename(place));
1462 |     DT1Energy -= 20;
1463 |     for(i=0;i<DT1weapon2.time;i++)
1464 |     {
1465 |       if(Distance <= DT1weapon2.range)
1466 |       {
1467 |         if(Hit = Hitter(DT1weapon2,DT2place[1],place) == HIT)
1468 |         {
1469 |           damage = ((DT1weapon2.power)*(1.45-((double)
1470 | Distance/(double)DT1weapon2.range)));
1471 |           DTAP2 -= damage;
1472 |           printf("Hit.damage%dp ",damage);
1473 |           Sleep(100);
1474 |           //Beep(200,500);
1475 |         }
1476 |         else if(Hitter(DT1weapon2,DT2place[1],place) == MISS)
1477 |         {
1478 |           printf("Miss. ");
1479 |           Sleep(100);
1480 |         }
1481 |       }
1482 |       else
1483 |       {
1484 |         printf("Miss. ");
1485 |         Sleep(100);
1486 |       }
1487 |     }
1488 |   }
1489 |   else
```

```
1490 |   {
1491 |     printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
1492 |     return 0;
1493 |   }
1494 |   return 0;
1495 | }
1496 |
1497 | int Missile_Shoot(int place)
1498 | {
1499 |   int i = 0;
1500 |   int damage = 0;
1501 |   int Hit = 0;
1502 |   if(DT1Energy >= 50)
1503 |   {
1504 |     printf("\n■LOCKED■\n");
1505 |     //Beep(800,500);//Beep(800,500);
1506 |     Sleep(300);
1507 |     printf("Shoot %s. %s\n",DT1weapon.name,placename(place));
1508 |     DT1Energy -= 50;
1509 |     for(i=0;i<DT1weapon.time;i++)
1510 |     {
1511 |       if(Hit = Hitter(DT1weapon,DT2place[1],place) == HIT)
1512 |       {
1513 |         damage = ((DT1weapon.power - DT2armor.defencepoint)
1514 | *(1.2-((double)Distance/(double)DT1weapon.range)));
1515 |         DTAP2 -= damage;
1516 |         printf("Hit.damage%dp ",damage);
1517 |         Sleep(100);
1518 |         //Beep(200,500);
1519 |       }
1520 |       else if(Hitter(DT1weapon,DT2place[1],place) == MISS)
1521 |       {
1522 |         printf("Miss. ");
1523 |         Sleep(100);
1524 |       }
1525 |     }
1526 |   }
1527 |   else
1528 |   {
1529 |     printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
1530 |     return 0;
1531 |   }
1532 |   return 0;
1533 |
1534 | }
1535 |
1536 | void HandGan_Shoot()
1537 | {
1538 |
1539 | }
1540 |
1541 |
1542 | ////////////////////other//////////////////////////
1543 | int RANDOM()
1544 | {
1545 |   int rnum;
1546 |   srand((unsigned)time(NULL));
1547 |   rnum = 0+(int)rand()%100;
1548 |   printf("RANDOM=%d\n",rnum);
1549 |   return rnum;
1550 | }
1551 |
1552 | int Get_Distance()
1553 | {
1554 |   int distance = Distance;
1555 |   return distance;
1556 | }
1557 |
1558 | int Get_Enemyplace()
1559 | {
1560 |   int enemyplace = DT2place[1];
1561 |   return enemyplace;
1562 | }
1563 |
1564 | int Get_Energy()
```

```
{
  int energy = DT1Energy;
  return energy;
}
int Get_Turn()
{
  int Turn = turn;
  return Turn;
}

int Get_Enemy_Turn()
{
  int Enemy_Turn = turn_e;
  return Enemy_Turn;
}


//////////////////////AI_DTSystem_e//////////////////////////////
int Move_Forward_e(void)
{
  if(DT2Energy >= 30)
  {
    Distance -= 20+(DT2engine.outputpower/10);
    printf("Move Forward\n");
    Sleep(50);
    DT2Energy -= 30;
    return 0;
  }
  else
  {
    printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
    Sleep(100);
    return 0;
  }
}
int Boost_Forward_e(void)
{
  if(DT2Energy >= 50)
  {
    Distance -= 30+(DT2engine.outputpower/5);
    printf("Boost Forward!\n");
    Sleep(100);
    DT2Energy -= 50;
    return 0;
  }
  else
  {
    printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
    Sleep(100);
    return 0;
  }
}
int Move_Back_e(void)
{
  if(DT2Energy >= 30)
  {
    Distance += 20+(DT2engine.outputpower/10);
    printf("Move Back\n");
    Sleep(50);
    DT2Energy -= 30;
    return 0;
  }
  else
  {
    printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
    Sleep(100);
    return 0;
  }
}
int Boost_Back_e(void)
{
  if(DT2Energy >= 50)
  {
    Distance += 30+(DT2engine.outputpower/5);
```

```
1640 |     printf("Boost Back!\n");
1641 |     Sleep(100);
1642 |     DT2Energy -= 50;
1643 |     return 0;
1644 |   }
1645 |   else
1646 |   {
1647 |     printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
1648 |     Sleep(100);
1649 |     return 0;
1650 |   }
1651 | }
1652 |
1653 | int Step_Left_e(void)
1654 | {
1655 |   if(DT2Energy >= 20)
1656 |   {
1657 |     DT2place[1] = 0;
1658 |     printf("Step Left\n");
1659 |     Sleep(50);
1660 |     DT2Energy -= 20;
1661 |     return 0;
1662 |   }
1663 |   else
1664 |   {
1665 |     printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
1666 |     Sleep(100);
1667 |     return 0;
1668 |   }
1669 | }
1670 |
1671 | int Step_Right_e(void)
1672 | {
1673 |   if(DT2Energy >= 20)
1674 |   {
1675 |     DT2place[1] = 2;
1676 |     printf("Step Right\n");
1677 |     Sleep(50);
1678 |     DT2Energy -= 20;
1679 |     return 0;
1680 |   }
1681 |   else
1682 |   {
1683 |     printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
1684 |     Sleep(100);
1685 |     return 0;
1686 |   }
1687 | }
1688 |
1689 | int Step_Center_e(void)
1690 | {
1691 |   if(DT2Energy >= 20)
1692 |   {
1693 |     DT2place[1] = 1;
1694 |     printf("Step Center\n");
1695 |     Sleep(50);
1696 |     DT2Energy -= 20;
1697 |     return 0;
1698 |   }
1699 |   else
1700 |   {
1701 |     printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
1702 |     Sleep(100);
1703 |     return 0;
1704 |   }
1705 | }
1706 |
1707 |
1708 |
1709 |
1710 | //////////////////////AI_FireControl_e//////////////////////////
1711 | int Rifle_Shoot_e(int place)
1712 | {
1713 |   int i = 0;
1714 |   int damage = 0;
```

```
1715 |   int Hit = 0;
1716 |   if(DT2Energy >= 30)
1717 |   {
1718 |     printf("\n\n■ WARNING! ■\n");
1719 |     //Beep(1000,300);//Beep(1000,300);//Beep(1000,300);
1720 |     Sleep(300);
1721 |     printf("DT2Attacked %s!\n",DT2weapon.name);
1722 |     DT2Energy -= 30;
1723 |     for(i=0;i<DT2weapon.time;i++)
1724 |     {
1725 |       if(Hit = Hitter(DT2weapon,DT1place[1],place) == HIT)
1726 |       {
1727 |         damage = ((DT2weapon.power - DT1armor.defencepoint
1728 |  + (DT2engine.outputpower/16))*(1.3-((double)Distance/(double)DT2weapon.range)));
1729 |         DTAP1 -= damage;
1730 |         printf("Hit.damage%dp ",damage);
1731 |         Sleep(100);
1732 |         //Beep(200,500);
1733 |       }
1734 |       else
1735 |       {
1736 |         printf("Miss. ");
1737 |         Sleep(100);
1738 |       }
1739 |     }
1740 |   }
1741 |   else
1742 |   {
1743 |     printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
1744 |     return 0;
1745 |   }
1746 |   return 0;
1747 | }
1748 |
1749 | int MachineGun_Shoot_e(int place)
1750 | {
1751 |   int i = 0;
1752 |   int damage = 0;
1753 |   int Hit = 0;
1754 |   if(DT2Energy >= 30)
1755 |   {
1756 |     printf("\n\n■ WARNING! ■\n");
1757 |     //Beep(1000,300);//Beep(1000,300);//Beep(1000,300);
1758 |     Sleep(300);
1759 |     printf("DT2Attacked %s!\n",DT2weapon.name);
1760 |     DT2Energy -= 30;
1761 |     for(i=0;i<DT2weapon.time;i++)
1762 |     {
1763 |       if(Hit = Hitter(DT2weapon,DT1place[1],place) == HIT)
1764 |       {
1765 |         damage = ((DT2weapon.power - DT1armor.defencepoint
1766 |  + (DT2engine.outputpower/25))*(1.1-((double)Distance/(double)DT2weapon.range)));
1767 |         DTAP1 -= damage;
1768 |         printf("Hit.damage%dp ",damage);
1769 |         Sleep(100);
1770 |         //Beep(200,200);
1771 |       }
1772 |       else if(Hitter(DT2weapon,DT1place[1],place) == MISS)
1773 |       {
1774 |         printf("Miss. ");
1775 |         Sleep(100);
1776 |       }
1777 |     }
1778 |   }
1779 |   else
1780 |   {
1781 |     printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
1782 |     return 0;
1783 |   }
1784 |   return 0;
1785 |
1786 | }
1787 |
1788 | int Blade_Attack_e(int place)
1789 | {
```

```
1790 |   int i = 0;
1791 |   int damage = 0;
1792 |   int Hit = 0;
1793 |   if(DT2Energy >= 20)
1794 |   {
1795 |     printf("\n\n■ WARNING! ■\n");
1796 |     //Beep(1000,300);//Beep(1000,300);//Beep(1000,300);
1797 |     Sleep(300);
1798 |     printf("Attack! %s. %s\n",DT2weapon2.name,placename(place));
1799 |     DT2Energy -= 20;
1800 |     for(i=0;i<DT2weapon2.time;i++)
1801 |     {
1802 |       if(Distance <= DT2weapon2.range)
1803 |       {
1804 |         if(Hit = Hitter(DT2weapon2,DT1place[1],place) == HIT)
1805 |         {
1806 |           damage = ((DT2weapon2.power)*
1807 | (1.45-((double)Distance/(double)DT2weapon2.range)));
1808 |           DTAP1 -= damage;
1809 |           printf("Hit.damage%dp ",damage);
1810 |           Sleep(100);
1811 |           //Beep(200,500);
1812 |         }
1813 |         else if(Hitter(DT2weapon2,DT1place[1],place) == MISS)
1814 |         {
1815 |           printf("Miss. ");
1816 |           Sleep(100);
1817 |         }
1818 |       }
1819 |       else
1820 |       {
1821 |         printf("Miss. ");
1822 |         Sleep(100);
1823 |       }
1824 |     }
1825 |   }
1826 |   else
1827 |   {
1828 |     printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
1829 |     return 0;
1830 |   }
1831 |   return 0;
1832 | }
1833 |
1834 | int Moon_Attack_e(int place)
1835 | {
1836 |   int i = 0;
1837 |   int damage = 0;
1838 |   int Hit = 0;
1839 |   if(DT2Energy >= 30)
1840 |   {
1841 |     printf("\n\n■ WARNING! ■\n");
1842 |     //Beep(1000,300);//Beep(1000,300);//Beep(1000,300);
1843 |     Sleep(300);
1844 |     printf("Attack! %s. %s\n",DT2weapon.name,placename(place));
1845 |     DT2Energy -= 20;
1846 |     for(i=0;i<DT2weapon.time;i++)
1847 |     {
1848 |       if(Distance <= DT2weapon.range)
1849 |       {
1850 |         if(Hit = Hitter(DT2weapon,DT1place[1],place) == HIT)
1851 |         {
1852 |           damage = ((DT2weapon.power - DT1armor.
1853 | defencepoint)*(1.4-((double)Distance/(double)DT2weapon2.range)));
1854 |           DTAP1 -= damage;
1855 |           printf("Hit.damage%dp ",damage);
1856 |           Sleep(100);
1857 |           //Beep(200,500);
1858 |         }
1859 |         else if(Hitter(DT2weapon,DT1place[1],place) == MISS)
1860 |         {
1861 |           printf("Miss. ");
1862 |           Sleep(100);
1863 |         }
1864 |       }
```

```
1865        else
1866        {
1867          printf("Miss. ");
1868          Sleep(100);
1869        }
1870      }
1871    }
1872    else
1873    {
1874      printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
1875      return 0;
1876    }
1877    return 0;
1878  }
1879
1880  int Missile_Shoot_e(int place)
1881  {
1882    int i = 0;
1883    int damage = 0;
1884    int Hit = 0;
1885    if(DT2Energy >= 50)
1886    {
1887      printf("\n\n■ WARNING! ■\n");
1888      //Beep(1000,300);//Beep(1000,300);//Beep(1000,300);
1889      Sleep(300);
1890      printf("Shoot %s. %s\n",DT2weapon.name,placename(place));
1891      DT2Energy -= 50;
1892      for(i=0;i<DT2weapon.time;i++)
1893      {
1894        if(Hit = Hitter(DT2weapon,DT1place[1],place) == HIT)
1895        {
1896          damage = ((DT2weapon.power - DT1armor.defencepoint)*(0.8));
1897          DTAP1 -= damage;
1898          printf("Hit.damage%dp ",damage);
1899          Sleep(100);
1900          //Beep(200,500);
1901        }
1902        else if(Hitter(DT2weapon,DT1place[1],place) == MISS)
1903        {
1904          printf("Miss. ");
1905          Sleep(100);
1906        }
1907      }
1908    }
1909    else
1910    {
1911      printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
1912      return 0;
1913    }
1914    return 0;
1915
1916  }
1917
1918  int HandGun_Shoot_e(int place)
1919  {
1920    int i = 0;
1921    int damage = 0;
1922    int Hit = 0;
1923    if(DT2Energy >= 30)
1924    {
1925      printf("\n\n■ WARNING! ■\n");
1926      //Beep(1000,300);//Beep(1000,300);//Beep(1000,300);
1927      Sleep(300);
1928      printf("DT2Attacked %s!\n",DT2weapon2.name);
1929      DT2Energy -= 30;
1930      for(i=0;i<DT2weapon2.time;i++)
1931      {
1932        if(Hit = Hitter(DT2weapon2,DT1place[1],place) == HIT)
1933        {
1934          damage = ((DT2weapon2.power - DT1armor.defencepoint
1935   + (DT2engine.outputpower/48))*(1.0-((double)Distance/(double)DT2weapon.range)));
1936          DTAP1 -= damage;
1937          printf("Hit.damage%dp ",damage);
1938          Sleep(100);
1939          //Beep(200,500);
```

```
1940        }
1941        else
1942        {
1943          printf("Miss. ");
1944          Sleep(100);
1945        }
1946      }
1947    }
1948    else
1949    {
1950      printf("■ WARNING! ■ Energy Too Low! Can Not Action!\n");
1951      return 0;
1952    }
1953    return 0;
1954  }
1955
1956
1957  ////////////////////other_e//////////////////////////
1958  int RANDOM_e()
1959  {
1960    int rnum;
1961    srand((unsigned)time(NULL));
1962    rnum = 0+(int)rand()%100;
1963    printf("RANDOM=%d\n",rnum);
1964    return rnum;
1965  }
1966
1967  int Get_Distance_e()
1968  {
1969    int distance = Distance;
1970    return distance;
1971  }
1972
1973  int Get_Enemyplace_e()
1974  {
1975    int enemyplace = DT1place[1];
1976    return enemyplace;
1977  }
1978
1979  int Get_Energy_e()
1980  {
1981    int energy = DT2Energy;
1982    return energy;
1983  }
1984
1985  int Get_Turn_e()
1986  {
1987    int Turn_e = turn_e;
1988    return Turn_e;
1989  }
1990
1991
1992  ////////////////////Debug_Tool///////////////////////
1993  void StageSkipper()
1994  {
1995    int ans = 0;
1996    printf("【Debug Command】select stage.(normal:-1)");
1997    scanf("%d",&ans);
1998    if(ans != -1)
1999    {
2000      DTSave();
2001      dtdata.stage = ans;
2002    }
2003    else;
2004  }
2005
2006  void Equip_Selector()
2007  {
2008    int ans = 0;
2009    printf("【Debug Command】Change Equip?(YES = 1  NO = 0)");
2010    scanf("%d",&ans);
2011
2012    if(ans == 1)
2013    {
2014      ItemLoader();
```

```
2015 |     printf("\narmor?  ");
2016 |     scanf("%d",&dtdata.armor);
2017 |
2018 |     printf("\nengine?  ");
2019 |     scanf("%d",&dtdata.engine);
2020 |
2021 |     printf("\nweapon1?  ");
2022 |     scanf("%d",&dtdata.weapon1);
2023 |
2024 |     printf("\nweapon2?  ");
2025 |     scanf("%d",&dtdata.weapon2);
2026 |     DTSave();
2027 |   }
2028 |   else;
2029 | }
2030 |
2031 | void StageSkip()
2032 | {
2033 |   int ans = 0;
2034 |   printf("始めからやり直したい場合は0を、\n続きからしたい場合は0以外
2035 | を入力してください(0以外は何を入れても前回の続きから始まります)。\n");
2036 |   scanf("%d",&ans);
2037 |   if(ans == 0)
2038 |   {
2039 |     dtdata.stage = 0;
2040 |     DTSave();
2041 |   }
2042 |   else;
2043 |   return;
2044 | }
2045 |
2046 |
2047 |
2048 |
2049 | ////////////////////GameMain//////////////////////////////
2050 |
2051 | enum DOA GameMain(int stage)
2052 | {
2053 |   int i = 0;
2054 |
2055 |   printf("Game Start!");
2056 |   Sleep(500);
2057 |   GameWaitWindow();
2058 |   SetAP();
2059 |   while(DTAP1 > 0 && DTAP2 > 0)
2060 |     Turn_Player(DT1weapon,DT2weapon,stage);
2061 |   if(DTAP1 > DTAP2)
2062 |   {
2063 |     winner = 1;
2064 |     destroy("Enemy");
2065 |     return CONTINUE;
2066 |   }
2067 |   else
2068 |   {
2069 |     destroy("DT1");
2070 |     GameOver();
2071 |     return GAMEOVER;
2072 |   }
2073 |
2074 | }
2075 |
2076 | ////////////////////Stage//////////////////////
2077 | void Stage0()
2078 | {
2079 |     init();
2080 |     DTLoad();
2081 |     AAReader(AAbuff,"AA2.txt");
2082 |     AAWriter(AAbuff);
2083 |     Sleep(2000);
2084 |     system("cls");
2085 |
2086 | }
2087 | void Stage1()
2088 | {
2089 |     init();
```

```
2090 |     printf("STAGE  %d\n",dtdata.stage);
2091 |     StoryWindow1();
2092 |     ItemLoader();
2093 |     equip(0,0,0,0);
2094 |     AIshow("AIStage1.c");
2095 |     if(GameMain(1)==CONTINUE)
2096 |     {
2097 |       Sleep(2000);
2098 |       dtdata.stage = 1;
2099 |       DTSave();
2100 |     }
2101 |     else
2102 |       exit(1);
2103 |
2104 | }
2105 | void Stage2()
2106 | {
2107 |     init();
2108 |     StoryWindow2();
2109 |     Sleep(2000);
2110 |     system("cls");
2111 |     AIWindowB();
2112 |     AIWindow2();
2113 |     Matome("matome2.txt");
2114 |     AIedit();
2115 |     dtdata.stage=2;
2116 |     DTSave();
2117 | }
2118 | void Stage3()
2119 | {
2120 |     init();
2121 |     ItemLoader();
2122 |     equip(0,0,0,0);
2123 |     AIshow("AI.c");
2124 |     if(GameMain(3)==CONTINUE)
2125 |     {
2126 |       dtdata.stage=3;
2127 |       DTSave();
2128 |       printf("STAGE  %d\n",dtdata.stage);
2129 |       Sleep(2000);
2130 |     }
2131 |     else
2132 |       exit(1);
2133 | }
2134 | void Stage4()
2135 | {
2136 |     init();
2137 |     StoryWindow3();
2138 |     AIWindow3();
2139 |     Matome("matome3.txt");
2140 |     AIedit();
2141 |     dtdata.stage=4;
2142 |     DTSave();
2143 |     Sleep(2000);
2144 | }
2145 |
2146 | void Stage5()
2147 | {
2148 |     init();
2149 |     ItemLoader();
2150 |     equip(5,1,2,1);
2151 |     AIshow("AI.c");
2152 |     if(GameMain(5)==CONTINUE)
2153 |     {
2154 |       dtdata.stage=5;
2155 |       DTSave();
2156 |       Sleep(2000);
2157 |     }
2158 |     else
2159 |       exit(1);
2160 | }
2161 |
2162 | void Stage6()
2163 | {
2164 |     init();
```

```
2165 |     StoryWindow4();
2166 |     AIWindow4();
2167 |     Matome("matome4.txt");
2168 |     AIedit();
2169 |     dtdata.stage=6;
2170 |     DTSave();
2171 |     Sleep(2000);
2172 | }
2173 |
2174 | void Stage7()
2175 | {
2176 |     init();
2177 |     ItemLoader();
2178 |     equip(0,1,0,0);
2179 |     AIshow("AI.c");
2180 |     if(GameMain(6)==CONTINUE)
2181 |     {
2182 |       dtdata.stage=7;
2183 |       DTSave();
2184 |       Sleep(2000);
2185 |     }
2186 |     else
2187 |       exit(1);
2188 | }
2189 |
2190 | void Stage8()
2191 | {
2192 |     init();
2193 |     StoryWindow5();
2194 |     Matome("matome5.txt");
2195 |     Sleep(5000);
2196 |     Kisei();
2197 |     AIedit();
2198 |     dtdata.stage=8;
2199 |     DTSave();
2200 |     Sleep(2000);
2201 | }
2202 |
2203 | void Stage9()
2204 | {
2205 |     init();
2206 |     ItemLoader();
2207 |     equip(2,4,6,5);
2208 |     AIshow("AI.c");
2209 |     if(GameMain(8)==CONTINUE)
2210 |     {
2211 |       dtdata.stage=9;
2212 |       DTSave();
2213 |       Sleep(2000);
2214 |     }
2215 |     else
2216 |       exit(1);
2217 | }
2218 | void Stage10()
2219 | {
2220 |     init();
2221 |     StoryWindow6();
2222 |     dtdata.stage=0;
2223 |     DTSave();
2224 |     Sleep(2000);
2225 | }
2226 |
2227 | ////////////////////Main//////////////////////////
2228 |
2229 | int main(void)
2230 | {
2231 |     Stage0();
2232 |     StageSkip();
2233 |     //Equip_Selector();
2234 |     switch(dtdata.stage)
2235 |     {
2236 |       case 0:
2237 |         Stage1();
2238 |         Stage2();
2239 |         break;
```

```
2240 |       case 1:
2241 |         Stage2();
2242 |         break;
2243 |       case 2:
2244 |         Stage3();
2245 |         Stage4();
2246 |         break;
2247 |       case 3:
2248 |         Stage4();
2249 |         break;
2250 |       case 4:
2251 |         Stage5();
2252 |         Stage6();
2253 |         break;
2254 |       case 5:
2255 |         Stage6();
2256 |         break;
2257 |       case 6:
2258 |         Stage7();
2259 |         Stage8();
2260 |         break;
2261 |       case 7:
2262 |         Stage8();
2263 |         break;
2264 |       case 8:
2265 |         Stage9();
2266 |         Stage10();
2267 |         Ending();
2268 |         break;
2269 |       case 9:
2270 |         Stage10();
2271 |         Ending();
2272 |         break;
2273 |     }
2274 |   return 0;
2275 | }
```

A.1.2 AI サンプルコード

リスト 2　AIsample

```
int Movement1()
{
  int i = 0;
  int num = RANDOM();
  int distance = Get_Distance();

  if(distance >= 171)
  {
    Boost_Forward();
  }
  else if(distance <= 170)
  {
    Move_Forward();
  }
  else if(distance <= 30);


  if(num == 0 || num == 2)
    Step_Left();
  else if(num == 1 || num == 3)
    Step_Right();
  else if(num == 4 || num == 5)
    Step_Center();
  else;
  return 0;
}

int FireControl1()
{
  int distance = Get_Distance();
  if(distance >= 50)
    MachineGun_Shoot(1);
  else if(distance <= 49)
    Blade_Attack(1);

  return 0;
}

int mainsys1()
{
  printf("\n\n");
  Movement1();
  FireControl1();
  return 0;
}
```

A.2 チュートリアル説明文

リスト 3　setumei2

```
1
2  やる夫、取り敢えずこれが現在お前の乗っているDTに読み込まれているAIだ。
3  プログラミング、なんて聞くと難しく聞こえるかもしれないが
4  要はお前がDTをどういったふうに動かしたいのかを書き込むためのものだ。
5
6  じゃあ、取り敢えずどこにどういったことが表示されているのか説明するぞ。
7  次にint ~~~~~()といった部分が1,6,12行目にあるだろう。
8  これは関数、ファンクションといって各機関の機能をまとめてあるものだ。
9  例えば移動関係の関数ならint Movement()の{}内に
10 銃器を撃ったりする関数ならint FireControl()の{}内に
11 書きこむんだ。
12 間違ってもFireControlの部分に移動の関数を
13 書いてはいけないぞ。正しい場所に正しい関数を書いていくんだ。
14 そして最後のMain()の中に書かれた関数を上から順番に実行していく。
15 いくらMovement()の中に書こうが、mainsys()でMovement()を
16 呼び出さなければ実行されないからな。気を付けろよ。
17 最初は動いて銃を撃つだけでいい。
18 相手はライフル、こっちはマシンガンだ。距離を開ければこちらが不利になる。
19 前方に移動してマシンガンを撃ちこんでいくんだ。
20 具体的には
21 int Movement()の中にMove_Forward();、
22 int FireControl()の中にMachineGun_Shoot();を
23 取り敢えず書いてみろ。
24 はじめから難しい物を組む必要はない。ゆっくり慣れていくんだ。
25 あとはint mainsys()でMovement(),FireControl()を15行目から
26 書いていくんだ。intは要らないが、各関数の最後に;を忘れないようにな。
27 編集はこれを使ってみろ。今からAIをこれで編集して、保存して、
28 もう一度DT防衛線！でDTStartを実行すれば戦闘が始まるぞ。
```

リスト 4　setumei3

```
1
2  よし、いいかやる夫。
3  これが現在のAIだ。さっき戦ったままのAIだな。
4  次に戦う相手はどうやら近距離戦に特化しているタイプのようだ。
5  近づいたら最後、レーザーブレードで容赦無く切り刻まれるぞ。
6  逆に言えば相手を近づかせなければいい。
7  開始からずっと後ろに後退して、後退しながら撃ちこめばいい。
8  かと言って後退しすぎてもこちらの攻撃が無意味になってしまうからな。
9  Move_BackとBoost_Backをうまく使い分けるんだ。
10 使い分けるには「条件分岐」だ。
11 例えば近づき過ぎたら離れるようにし、逆に遠すぎたら近づく
12 ようにするんだ。要するに
13 「もし相手との距離が～以上ならMove_Forward」
14 「もし相手との距離が～以下ならMove_Back」
15 「それ以外なら何もしない」
16 といった風に考えるといい。
17 これをAIに記述する形に直すと
18 if(distance > 300){Move_Forward()}
19 else if(distance < 150){Move_Back()}
20 else;
21 こんな風になる。条件分岐の始めをifで書き、条件を()の中に書く。
22 その条件が満たされるときに{}内の行動を実行する、というわけだ。
23 条件の書き方は数学でもお馴染みの等号、不等号みたいなものを使う。
24 正確には条件演算子と言う。これで数値なんかを比較する。
25 例えばdistance < 150という文はdistance（現在の敵までの距離）
26 が150よりも小さい時、という意味になる。
27 ちなみにdistanceが150の時に実行させたいときは
28 distance == 150となる。「=」ではないことに気をつけるんだ。
29
30 後は距離のとり方だが、DTには各種センサー類が搭載されている。
31 現在の位置はもちろん、相手までの距離、残りエネルギーなどを
32 すべてモニタリングしている。
33 その値は当然AIにおいて利用することが出来るぞ。
34 センサーから値を取得するには、例えば相手との距離なら
35 Get_Distance()で取得できる。
36 ただ、直接これを記述するだけでは値を使用することはできない。
37 AI側でセンサーの値を利用できるように変数と呼ばれるものが必要だ。
38 変数はセンサーから読み取った情報を格納しておく役割を持つ。
39 つまり、センサーから取り出した距離の情報をAIの変数に格納するわけだ。
40 具体的に書くと
41 int distance = Get_Distance()
42 となる。intとは整数のことだ。このDTはすべて整数で処理を行なっているので
43 気にする必要はない。
44 整数を格納する変数distanceを作成し、
45 Get_Distance()を実行して返された値をdistanceに格納する。
46 こうすることで取得した情報が使えるようになるぞ。
47 さっきの条件分岐で等号が「==」だったのも、「=」は代入、格納の意味を
48 持っているからだ。等しい時は==、値を格納するときは=を使うんだぞ。
49 実はdistanceという変数名は必ずしもこれじゃなくていい。
50 変数名は自由に決められるから、例えば
51 int weight = Get_Distance()
52 でも問題なく距離をセンサーから取得することが出来る。
53 しかし、距離を「重量」なんて変数に格納したらわかりにくいだろう。
54 やる夫や俺が読んでもわかりやすいように、変数名には一目見て
55 どんな変数か分かるような名前を用いるんだぞ。
56
57 長くなってしまったがコレを頭に叩きこめば条件分岐を用いた
58 AIが書けるはずだ。
59
60 ・・・念のためにサンプルを表示させておくぞ。
61 いいかやる夫、ここはもう敵地だ。お前の家を襲ったタイプとは
62 比較にならないくらいに強化されている。
63 「HelloWorld」は終了、ということだ。がんばれよ。
64 ここで言ったことは要約されているものがドキュメントにも書いてある。
65 途中でわからなくなったらドキュメントを見るんだ。
66 前と同じようにAIを編集して保存した後、
67 DT防衛線！を実行すれば戦闘が始まるからな。
```

リスト 5　setumei4

```
1
2  やる夫、こいつは最初に戦った奴とは操縦者の熟練度が
3  断然違う。
4  やる夫的な立場で言えば相手のAIがとても高度に組まれている。
5  最初に戦った時のAIでは確実にこちらが負けてしまうぞ。
6  だがチャンスがないわけでもない。
7  相手の動きをよく見ていると、一定の癖があるようだ。
8  大人になってからのクセはなかなか抜けないからな・・・
9  奴はどうやら
10 「右・右・左・左・右・右・左・左」
11 を繰り返しているようだ。
12 右に二回、左に二回移動するのを繰り返しているわけだな。
13 標準でもDTはロックした敵を自動的に追尾して照準補正をかける。
14 しかし、あんなふうに急激に動かれると自動補正だけでは追いきれずに
15 かわされてしまうんだ。
16 そういう時にはShoot_MachineGun()に追加命令を与える。
17 今までは()に1が入っていただろう。あれは中央を狙えという
18 追加命令をShoot_MachineGunに出していたんだ。
19 同じようにやる夫から見て左を狙うには0、右を狙うには2を入れればいい。
20 この数字のことをShoot_MachineGunの引数、という。
21 Shoot_MachineGunが実際に発砲するときに使う追加の値のようなものだ。
22 これで左を狙ったり右を狙ったりすることが可能になるぞ。
23 他にも奴はブレードも装備している。近づきすぎないように戦うんだ。
24 そうだやる夫、相手が横に回避するように、こちらもサイドステップで
25 相手の攻撃をかわすことができるぞ。
26 Step_Center()、Step_Right(),Step_Left()で
27 それぞれ中央、右、左に移動できる。
28 相手が攻撃したところと今やる夫がいる位置が違っていれば
29 相手の命中率を大きく下げることが出来る。
30 有効に活用してくれ。
31 ただ、1ターンに使用出来るEnergyは決まっているからな。
32 Energy切れを起こすとそのターンはそこから先の行動が一切できない。
33 よく考えて、計算しながら組んでいくんだぞ。
34 前回でセンサーの値を使用した時のように、Energy残数も
35 モニタリングされているから使用することが出来る。
36 Get_Energy()を使うんだ。変数に格納するのを忘れないようにな。
37 各関数の使用エネルギー量や変数の記述の仕方はドキュメントに
38 書かれている。わからなければそれを見ながら組み上げていくんだ。
39 頑張れよ、やる夫。
```

A.3 会話内容文

A.3.1 Stage1

リスト 6 Stage1yaruo

```
1
2  あー、今日も暇だお・・・
3  何かすることないのかお？
4  とりあえずブラウザ立ち上げるお
5  カタカタカタ・・・
6  ・・・
7  なん・・・だと・・・
8  そんな・・・ありえないお・・・！
9  とととととりあえずやらない夫に言ってみよう
10
11 やらない夫、これを見るんだお！
12
13
14 日本＼(^o^)／
15
16 しかもこれ、もう全国で施行されてるみたいなんだお！
17
18
19
20 ばっちりだお！　HDDがVHSみたいにケースに収まってるお！
21
22 い、いやだお！　リナたんや唯たんやエリオたんにもう会えなくなるなんて
23 死んだほうがましだお！
24
25
26 そんな・・・理不尽すぎるお・・・
27 やる夫のシアたんが・・・たまひよが・・・桜子さんが・・・orz
28 ニッポンのアニメとかは世界的にクールで通ってるんじゃなかったのかお！
29
30
31
32 くそ・・・あの知事め・・・自分はエロ小説書いてるくせにずるいお
33
34 おや・・・誰かきたようだ。ちょっと見てくるお。すかぽは一応スマホで
35 繋ぎ直しておくお。
36
37
38 きっと素敵な女の子が家を訪ねてやってきたんだお！
39
40
41 （どたどた・・・）
42 はいはい、こんな時間に誰だお？
43
44 くぁwせｄｒｆｔｇｙふじこｌｐ
45
46 い、いやあのそのそれはちょっとあのなにももってないお
47
48 ・・・お、お・・・
49
50 おまいらにやる夫の嫁たちは渡さないお！俺の嫁はやる夫が守るお！
51
52 やらない夫、何とかしてくれお！
53
54 わかってるお！　庭から逃げれば余裕だお・・・！
55
56
57 警察って言ったけど、軍服着て小銃さげてたお！　どういうことなの・・・
58
59 ここまで来れば・・・
60 ひぃ、ひぃ・・・運動不足のやる夫にはつらいお・・・
61
62
63 いいお。これで映っ・・・！
64
65 ガンダ●・・・！？
66
67 そういえばガンオタの蒼豆＠種厨ｗ氏から聞いたことがあるお・・・
```

68
69 自衛隊が本格的に二足歩行型戦車を作ってるっていう、とんでもない噂だお
70 まさか本当・・・なのかお・・・
71
72 ま、まじかお・・・
73 え、ちょ、まっ
74 うわあああああ
75
76 な、何が起こったんだお・・・
77 ちょ、おま
78 家が・・・
79 跡形もなく吹き飛んでるお・・・
80
81
82
83 そ、それが・・・
84 なんか動いてない方のＤＴが目の前でかばってくれたんだお・・・
85 コックピット部分が空いてるお
86
87
88 ｏｋ把握だお！
89
90 （ぷしゅー、がちゃん）
91 よ、よし、乗り込んだお
92
93 乗り込んだけどどうするんだお！
94
95 ん、何だお、勝手に動き出して・・？！
96
97
98
99
100

リスト 7　Stage1yaranai

```
1
2
3
4
5
6
7
8  ひょこっ
9  どうしたやる夫？
10
11 これは・・・
12 アニメ単純所持禁止法・・・だと・・・
13
14 これはひどい
15
16 本当だな、今ざっとウェブ上を見て回ったが
17 日本のオタクどもが次々に逮捕されている。
18 やる夫、たしかお前物心ついた頃から全部のアニメ録画してなかったか？
19
20 ・・・それ、結構まずくないか？　差し押さえられるぞ
21
22
23 気持ちはわからなくもないが・・・この条文、アニメだけじゃなく
24 当然ながらエロゲや漫画なんかもアウトみたいだな。
25
26
27
28 オレンジに偏りすぎだろ。
29 首都で施行されていた条例が全国に強化されて飛び火したようだな。
30 あくまで目立たないように水面下で準備をしていたらしい。
31
32 消されるぞやる夫。まあ当然のごとく小説は除外されているみたいだな
33
34
35 フラグびんびん立ってるんだが
36
37 ねーよｗｗｗｗ
38
39
40
41 あの、すいません。警察ですが
42
43 ちょっとお話お伺いしてもよろしいですか？
44
45 ・・・？
46
47 ！　こいつ違反者か！　ひっとらえろ！
48
49 まずは家の外に逃げろ！　スマホとヘッドセットはわすれるなよ！
50
51 逃げ出したらなるべく家から遠ざかるんだ！　嫌な予感がする
52 手口が強引すぎるんだ！
53
54 わからん・・・何が起こっているんだ
55
56
57 がんばれ。ん、なんか変な音がするぞ・・・
58 やる夫、ビデオチャットで周りを映してくれないか
59 なんだ・・・これは・・・？
60
61 お台場じゃないんだからありえないだろう。
62 二機あるな。一つは動いてないのか・・・？
63
64 ・・・？
65
66
67 ググッてきた。これはどうやら自衛隊の新秘密兵器
68 DefenceTank・・・通称DTだそうだ。
69
70 ！　おい、やる夫、家から今すぐ離れるんだ！！
71
72 ―――か、やる夫
73 大丈夫か、やる夫！
74
```

75 |
76 | ありえんだろ、常識的に考えて・・・
77 | 一般人相手の、ただのアニオタになんで・・・
78 | それよりもやる夫、なんで無事なんだ？
79 |
80 |
81 | なん・・・だと・・・
82 | やる夫、考えている時間はない。もう一機は次の砲撃準備をしている。
83 | そのDTにのりこむんだ！　罠かもしれんがそれしかない！
84 |
85 |
86 |
87 | よし、取り敢えずそれで即死はなくなったはずだ。
88 |
89 |
90 |

A.3.2 Stage2

リスト 8　Stage2yaruo

```
1
2  た、倒せたのかお・・・
3  もうやだお、死にたくないお・・・
4
5
6  そ、そんなこと言ったって無理に決まってるお・・・
7  奴らは本職だお、自室警備員のやる夫にはかないっこないお・・・
8  まず操縦できるわけがないお！
9  まず操縦できるわけがないお！
10
11
12 そんなの余計に無理だお！AIってどうやって組むんだお・・・
13
14
15
16 誰かにやってもらえばいいお！　やる夫は家に帰って2ch張り付いてるお！
17 ・・・・・・
18
19 AIってどうすれば組めるんだお、早く教えろやらない夫！
20 やってやるお！　ニートの自由を取り戻すのはやる夫だお！
21 やってやるお！　ニートの自由を取り戻すのはやる夫だお！
22
```

リスト 9　Stage2yaranai

```
1 |
2 |
3 | やる夫、考えている暇はない。死にたくなかったら戦うんだ！
4 | 相手の増援が来ている、このままじゃおしまいだぞ！
5 |
6 |
7 |
8 | イキナリDTを動かすなんて無理だろ・・・常識的に考えて
9 | 大丈夫だ、今回のDTにはAIが搭載されている。AIを即席で組みさえ
10 | すれば十分に戦えるはずだ！
11 |
12 | AIの組み方は教えてやる、きちんと聞いて勉強するんだ。
13 | 幸い奴らが攻めてくるまでに多少の時間がある。取り敢えず動くように
14 | するだけでいい。
15 |
16 | お前の家はさっき吹き飛ばされただろ。
17 | お前が勝てばネット界の英雄になれるぞ。どうだ。
18 |
19 | （わかりやすいやつ・・・）
20 | その意気だやる夫。
```

A.3.3 Stage3

リスト 10　Stage3yaruo

```
1
2  や、やったお・・・
3  もうやだお・・・やっぱり無理だお・・・
4
5
6  こんな事ならまじめに勉強して政治家にでもなればよかったお
7
8
9
10
11 了解だお。もうここまで来たら腹をくくるお・・・
12
13 （ずーん　ずーん）
14
15 お、見えてきたお
16
17 でもDTが攻めてくるとは思ってないだろうお。
18
19 やらない夫イケメン！
20 マニュアルがあればもうちょっとましに動けるお・・・！
21
22
23 了解だお。
24
25 この丸い円が書いてある機械かお？　なんか接近してきてるお！
26
27
28 大丈夫だお。どうすればいいお？
29
30
31
32
```

リスト 11　Stage3yaranai

```
1
2  やったか！
3
4  もう後戻りはできないぞやる夫。
5  戦わないと俺達の世界に明日は来ないんだ。
6
7  今更悔やんでも仕方ないだろう。
8  さて、取り敢えずここまでの強行手段を取るわけが知りたいな。
9  やる夫、このまま首都真ん中まで攻めこむぞ。
10 どのみちここでしのいでいてもジリ貧になるだけだ。
11
12 だな。それが懸命だ。
13
14
15
16 やはりそれなりの警備体制がしかれているな。
17
18 うむ、そうだな。
19 そうだやる夫、移動時間の間にDTのAIを書くマニュアルを手に入れておいた。
20
21
22 かなりぎくしゃくしていたからな。此処から先は敵の攻撃も激しくなるだろう。
23 取り敢えずURLをスマホに転送しておくぞ。気になったら目を通しておいてくれ。
24
25 ん、やる夫、レーダーを見るんだ。
26
27 敵さんのお出ましか。やる夫、さっきのドキュメントは目を通して開いておけ。
28 ドキュメントがある前提で話をすすめるぞ。
29
30 どうやら奴は近接戦闘型のDTのようだ。装備は・・・
31 ブレードのみのようだな。
32 よしやる夫、いい考えがある。
```

A.3.4 Stage4

リスト 12 Stage4yaruo

```
1
2  よし、やったお・・・！
3  なんとかなるもんだお！
4
5
6
7  そ、そんな褒めても何もでてこないお！
8  やる夫はただの・・・ヒキヲタニートだお・・・
9
10 そんなの知らないお。働きたくないでござる
11
12 ・・・・・・
13 ・・・
14 ・・・帰ったら考えるお。生きて帰ることが先決だお。
15
16
17 オッケーだお。
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35 組みたいものをどうやって組み上げるかを考える力をつければ
36 もしも言語や環境が変わってしまってもその言語とかの仕様を
37 覚えなおすだけでいい、ってことだおね。
38
39
40
41
42 敵みたいだお。普通に最初戦った機体とほぼ同じみたいだお・・・？
43
44
45 射程に入ったお。標的ロック・・・
46 ファイア！
47
48 な・・・左にかわされたお！
49
50
51
52 りょ、了解だお。
53
54
55 ふう、ここなら大丈夫そうだお。あいつも追って来てないお。
56
57
58
59
```

リスト 13　Stage4yaranai

```
1  |
2  | よし、やったか！
3  |
4  | やる夫、すごいじゃないか。
5  | この短時間で複雑なAIを組めるようになってきている。
6  | 結構素質あるぞ。
7  |
8  |
9  | なあやる夫、なんで仕事やめたんだ・・・？
10 |
11 | 勝手な意見かもしれないが、もう一度頑張ってみろよ・・・
12 |
13 |
14 |
15 | ・・・だな、悪かった。
16 | よしやる夫、先へ進むぞ。
17 |
18 | やる夫、歩きながら聞いていてくれ。
19 | やる夫がさっき書いたプログラムは基本的に一回移動して
20 | 一回攻撃するようなものだったはずだ。
21 | 確認はしていないので独自に組んだのなら更にハイレベルな組み方も
22 | 出来ているだろうが、それは無視して話をすすめるぞ。
23 | さっき使った条件分岐は、やり方によってはもっと複雑な行動を
24 | 取ることだって出来る。
25 | 例えば一回移動して一回攻撃していたところを
26 | 有効射程に近づくまでエネルギー量のぶん限界まで移動して
27 | 有効射程入ったらその距離をキープしながら複数回攻撃する
28 | なんてことも可能だ。
29 | 要するにプログラムを組むに当たって重要なのは
30 | ロジックやアルゴリズム、ごく簡単に言うと自分が望む通りのプログラム
31 | を組むにはどうすればいいのかという道筋を考えることだ。
32 | たとえAIを組み上げるときの関数群を覚えたって、DTの種類が違って
33 | くるとどのみち覚えなおさなければならないからな。
34 | もちろん仕様を暗記することが重要じゃないとは言っていないぞ。
35 |
36 |
37 |
38 | そういうことだ。
39 | まあはじめはなんだかんだ言って難しいものなんだがな。
40 |
41 | おっと、やる夫、レーダーが反応しているぞ。
42 |
43 | やる夫、よく見るんだ。どうやら機体は同じでも中に乗っている
44 | 兵士は相当な手練らしい。
45 |
46 |
47 |
48 | やはり一筋縄ではいかないようだな。
49 | やる夫、相手からの反撃を食らう前に一回距離をとれ。
50 | 後方300の場所に廃ビルがある。その影に隠れてAIを書き込むんだ。
51 | 幸い慎重派なようだ。あちらからはそうそう追ってこないだろう。
52 |
53 |
54 |
55 |
56 | うむ。じゃあ説明に入るぞ。
57 |
58 |
59 |
60 |
```

A.3.5 Stage5

リスト 14 Stage5yaruo

```
1
2  つ、強い・・・でもやったお！
3  でも、乗ってる人が違うだけであんなにも変わるものなのかお・・・
4
5
6
7
8
9
10
11
12 なんかやらない夫が上司みたいだお・・・
13
14
15
16
17 やらない夫・・・！
18 やる夫もやらない夫が友達でよかったお！
19 むしろ心の友だお！
20
21
22 ・・・やらない夫・・・？
23
24 やらない夫、なにかあったのかお！？
25
26
27 やらない夫、やらない夫ー！
28
29 やらない夫おおおおおおおおおおおおおおーーーーーー！
30
31 ぐすっ、やらない夫・・・
32 やらない夫の仇は必ず取るお・・・！
33
34 ここが国会議事堂・・・
35 ん、誰か居るお・・・！
36
37
38 よくもこんなキチガイ政策を・・・！
39
40
41
42
43 こんなコトして誰得なんだお！こんなことのためにみんな犠牲に・・・！
44
45
46
47 あ、あなたは・・・！
48 ろうぜん閣下！
49
50
51
52
53 閣下は見えない所でちゃんと努力していたんだ、グググッてからモノを言えお！
54 実績も実は報道されていないだけで多数あるんだお！
55
56
57
58
59
60
61
62 ・・・！
```

リスト 15　Stage5yaranai

```
1
2  ktkr!
3
4  だな。それはAIも同じだ。機体が同じでもAIが違えば全く違った
5  戦い方にすることが出来る。
6  前にも言ったが、大切なのは自分がどういうふうにDTを動かしたいかを
7  まず頭の中で組み上げる事だ。
8  次にどうしたらそのとおりにAIを組み上げることが出来るのかを模索する。
9  書き方や関数の使い方なんてものはだいたい調べればわかるが、
10 考え方は自分で訓練していくしか無いからな。
11 いずれはお前がこの国を引っ張っていくんだ。責任重大だぞ。
12
13 それだけお前のことを評価しているんだ。正直見なおしたぞ。
14 ふだん引きこもってくすぶってばかりいるからだめだこいつはなく何とかしないと
15 と思ってたが・・・
16 やる夫、お前が友達でほんとうに良かったと思ってるよ。
17 この事件が終わったら又一緒に遊びに行こうぜ・・・！
18
19
20
21 よせよ、ケツの穴が痒くなっちまうぜ・・・ゴホッ
22
23 だから・・・ゴホゴホッ、早く世界に萌を取り戻してくれ・・・
24
25 やられた、狙撃だ。ちょっと通信を逆探知されていたらしい・・・
26 さすが自衛隊だな・・・ぴくせるまり・・・とは違う
27
28 やる夫・・・裸エプロンは・・・最高だな・・・
29
30 ブツッ　ザザザー
31 ザー
32
33
34
35
36 ふはははは、とうとうやってきたな小僧！
37 どうだいまの気分は？ 悔しいか？ ええ？
38
39 キチガイとは心外だな。見ろ、国中から萌えがなくなった！
40 素晴らしいとは思わないかね！
41 この国には演歌と小説さえあればいい！
42 私が、私がこの国を変えてやるのだー！　ふふふ、ふはははは！
43
44 そんなものは些細な犠牲にしかならん。それにだ。
45 元はといえばこいつが悪いんだろう。
46 こいつがトップになって国は大きく傾いた！
47
48
49 く、すまんな若者よ、こんな辛い役回りをさせて・・・
50 いくら気に入らないことがあっても、やっていいことと悪いことがある。
51 知事よ、お前はそんなこともわからないのか・・・！
52 はっ、何を言い出すかと思えば・・・！ お前こそ何もしていないだろう！
53
54
55 若者よ、それは自慢するべきことでも誇るべきことでもない・・・
56 ただ、自分がやらねばならぬと思ったからやったまでのことだ・・・
57 この国を思い行動することに、どうしてこの国を傷つけなければ
58 ならないのだ。目をさますのは知事よ、お前のほうだ・・・！
59 うるさいうるさいうるさい！　私はこの国の王になる！
60 お前らは日の目をみることなく抹消されるのだ！
61 さあ、現世に別れは済ませたか？
62 行くぞ、小僧！
```

A.3.6 Stage6

リスト 16 Stage6yaruo

```
1
2  や、やったお・・・！
3  勝ったんだお、やる夫の勝ちだお！
4
5
6  やらない夫・・・嫁たち・・・ゲーム・・・
7  やる夫の大切なものを傷つけようとしたからだお！
8  ヒキヲタニートも怒ると怖いんだお！
9
10 ・・・趣味は人それぞれ。誰かが干渉したり、優劣を付けられるものじゃないお。
11 むしろそれぞれを引き立てあって、活性化させることこそ、メジャーな趣味も
12 マイナーな趣味も共存できる道だと思うお。
13 おまいの演歌や小説だって嫌う人はいるお。でもそれを規制したり頭ごなしに
14 迫害するのはお門違いなんだお！
15
16
17
18 ふぅ・・・
19 閣下、大丈夫ですかお！？
20
21
22 いや、大したことはしてないお・・・
23 やる夫は大切な物を守りたかっただけだお・・・
24
25
26
27
28
29
30
31
32 ・・・・・・
33
34
35
36
37
38 やる夫は・・・！
39 ・・・・・・・・・・・・
40 ・・・・・・・
41 ・・・
42
43
44
45
46 はっ
47 ・・・・・・
48 何だ夢か
49 ・・・とりあえずブラウザ立ち上げるお
50 カタカタカタ・・・
51
52 ・・・ハロワ行くか
53
```

リスト 17　Stage6yaranai

```
1
2  ぐう・・・っ
3
4  やりおるな・・・
5  小僧のくせに・・・
6
7
8
9  ふん、そんなものにすがって生きている貴様に道を潰されるとはな
10
11
12
13
14
15 ・・・・・・
16 どうやらわしは・・・大きな間違いを犯していたようだ
17 そうか、そういうことか・・・
18 （ばたりこ）
19
20 あ、ああ、大丈夫だ・・・よくこの漢の暴走を止めてくれた。
21 この国を代表して例を言うぞ。
22
23
24 ・・・そうか・・・
25
26 やる夫くんよ、この世の中にはいろいろな人がいる。
27 趣味に時間を費やすもの、狂ったように仕事に打ち込むもの、
28 ただ時の流れ行くままその日暮らしを楽しむもの。
29 時には志を同じくしたものが協力しあい、また時には裏切り陥れ利益を奪いあう。
30 君は否応なくこの人生の流れに放り出されなければならない。
31 遅かれ、早かれだ。
32
33 今この困難を乗り越えた君なら、この流れを自分のものに出来るだろう。
34 こんな所でくすぶっている場合じゃない。下流の世界はもっと広いぞ。
35 見てみたいと思わないか。自分で泳いでみたいと思わないか。
36 決めるのは君自身だ。さあ、もういけ。
37 こんな所で立ち止まっているわけではないだろう？
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
```