

■班企画書

■年■月■日(金) 12:00 提出

1 ゲーム内容

1.1 概略

本ゲームは、潜水艦同士が互いに攻撃し合い、自分以外を撃沈する内容になる。

プレイヤーはゲーム内に用意されたメニューなどを使い、自機である潜水艦を操作する。潜水艦に命令できる行動は、下記に従う。

命令	内容
移動	潜水艦を前後に動作させる
回頭	潜水艦を左右に回転させる
潜航	海中に潜る
浮上	海上に出る
探索	ソナーを利用し、探索を開始する
使用	選択されたアイテムを使用する
攻撃	魚雷を発射する

ゲームフィールドは、海上と海中の2面に分ける。各フィールドにさまざまな属性を付加し、多様性を持たせるようにする。例えば、海上では空気の入れ替えができる反面、敵に見つかりやすいなどが該当する。

また、画面には軽いふかんを利用した内容(潜水艦など)を描画する。

表1にゲームの仕様とルールを示しておく。

表 1: ゲームのおおまかな仕様とルール

ゲームジャンル: 海中アクションゲーム プレイヤー人数: 2 ~ 4 人	ゲームに対するルール	1. 自分以外の全潜水艦が撃沈した時点で終了する
	自機に対するルール	1. 武器は、2種類まで 2. 潜航中は一定時間経過後、ペナルティを受ける

1.2 コンセプト

携帯ゲーム用に開発をするわけであるから、簡単な操作方法でゲームを楽しめるように工夫する。特に注意することは、どんな人でも直感的にわかり操作できることである。さらにいえば、精神的に疲れにくいようなゲームを目標とする。

1.3 ゲーム画像



図 1: ゲーム画面 (全体)

ゲーム画面は主に次の 6 つのウィンドウにより構成される。

- ① メインウィンドウ (Main Window) ... 自艦の周辺を表示し、操作を行なう
- ② マップウィンドウ (Map Window)... 自艦の位置を表示する
- ③ ソナーウィンドウ (Sonar Window)... 敵の位置と深度を表示する
- ④ ユニットウィンドウ (Unit Window)... 自艦の状態、アイテムを表示する
- ⑤ アクティブウィンドウ (Active Window)... 行動可能量を表示する
- ⑥ ガイドウィンドウ (Guide Window)... ユーザの補助を行なう

これらウィンドウの詳細について述べる。

① メインウィンドウ



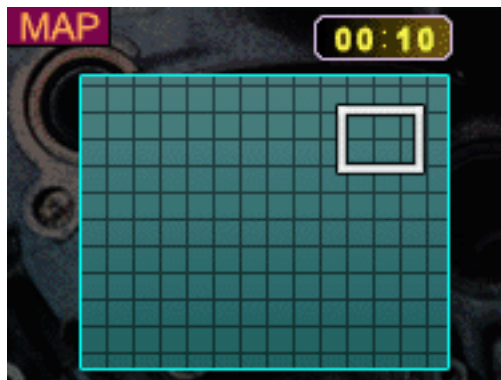
このウィンドウには、主に、自艦、敵艦、アイテムの3種類が表示される。このほかに、攻撃時などには魚雷が表示されるが、操作には直接影響がないため割愛する。

まず、自艦の基本操作は、このウィンドウ内で行う。海の領域(艦を除く)をタップすることにより、艦を移動させることができる。また、艦をタップすると、メニューパネルが表示される。閉じるには、メニューパネル以外の領域をタップすれば良い。これらの詳しい操作方法は「操作方法」で述べる。

次に、敵艦はその深度に応じて表示される。通常ソナーでは、敵艦を個別に判別することは、不可能である。しかし、この画面では、目視により判別することができる。

最後に、漂流中のアイテムをタップすることにより、アイテムの詳細情報が表示される。これは、ユーザのアイテム取得判断の目安となる。取得は、艦を移動させ、重ね合わせることでできる。

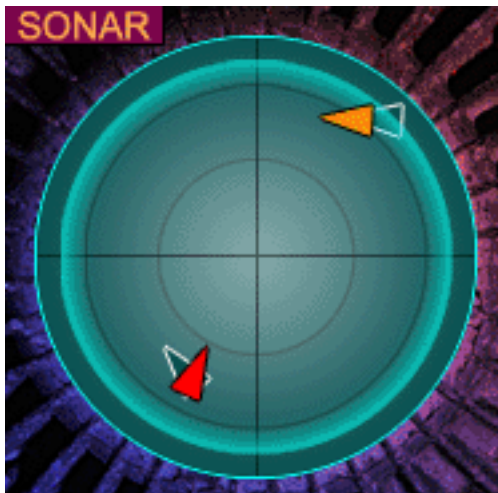
② マップウィンドウ



グリッド平面上が、艦の活動可能範囲である。ゲーム中は、この平面外に移動することはできない。また、白い枠内が、メインウィンドウに表示されるエリアである。艦の移動と同時に、この枠も移動することになる。グリッド上には、アイテム位置が点滅表示されるが、こういったアイテムが詳しい情報を得ることはできない。そのため、取得するには、実際に近くまで移動する必要がある。

なお、右上部に表示されている数字は、ゲームの経過時間を示す。

③ ソナーウィンドウ



ソナーは、敵艦の位置、動きを捕捉し、レーダのような役目を果たす。方位は、マップウィンドウと同一であり、艦を回頭しても、北を指す方向は変化しない。円の中心は自艦の位置であり、敵艦は、その相対距離に応じて、中心から遠ざかる。また、自艦との相対深度に応じて色分けされ、敵艦の方が深い場合は青、浅い場合は赤、同じ場合は黄で表される。つまり、自艦が海上の場合、敵艦は黄もしくは青(深)であり、海中の場合には、黄もしくは赤(浅)となる。

ただし、これら敵情報は、必ずしも表示されるわけではない。敵艦が、海中で待機状態である場合、その機影を捕らえることは出来ない。そこで、ピンガー(探信音)を放つことによって、正確に情報を得ることが可能である。

捕捉された艦は、待機状態であっても一定時間は、その姿をさらけ出すことになる。注意しなければならないことは、ピンガーを放った艦も同様に捕捉される点である。つまり、事前に相手より有利な位置に移動しておくことが重要である。

④ ユニットウィンドウ



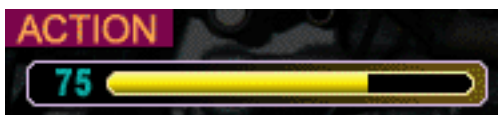
LIFE ゲージは、艦の耐久力を示し、敵の攻撃を受ける度に減少する。

AIR ゲージは、潜航時間に応じて、その状態(色)が変化し、赤色に変化すると左端から棒が伸び始め、右端に達すると、LIFE ゲージが1ポイント減少する。そして、再び棒が伸び始める。これを回避するには、艦を浮上させればよい。ただし、浮上中は、待機状態であっても敵艦のソナーに捕捉されることになる。

アイテムパネルには、所持するアイテムが表示される。アイテムは、その効果により大きく3つの属性に分類され、左パネルから順に、攻撃系(例：ホーミング)、防御系(例：デコイ)、能力向上系(例：ステルス)が、配置される。各属性は1種類しか保持できないため、同じ属性をもつものを取得した場合は、上書きされることになる。また、各アイテムには、使用回数が設けられており、同じアイテムを取得した場合には、この数値が回復することになる。

取得したアイテムを使用するには、後述するメニューパネルの操作を行えばよい、ただし、使用できるアイテムは、1回の操作で1種類であり、同時に複数使用することはできない。そのため、あらかじめ選択しておく必要がある。(アイテム選択は、常時切替えが可能である。)

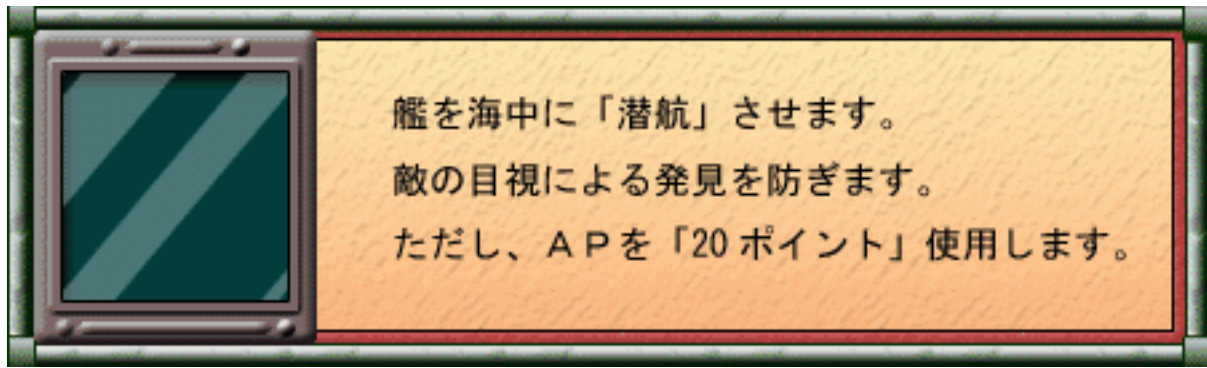
⑤ アクティブウィンドウ



AP(Active Point) ゲージは、艦の行動可能ポイントを示している。攻撃、浮上(潜航)、探索、アイテムを使用するとポイントが一定値減少する。また、高速移動の場合も同様である。このポイントが不足すると、これら行動を取ることが

できなくなる。低速移動中でも、徐々に回復するが、早期に回復するためには、その場に待機する必要がある。

⑥ ガイドウィンドウ



このウィンドウは、ゲームの進行状況の解説、及び各ボタンの補助説明を行なう。これにより、ユーザは、PDAの小さな画面でも、ゲームに馴染むことができる。

右側のメッセージパネルでは、ボタン操作、各パネルの補助説明を行う。ユーザがマニュアルを未読の状態でも、ゲーム中に操作方法を学習することができる。また、イベント発生時には、ユーザにその情報を知らせる。例えば、敵から魚雷が発射された場合には、その接近を知らせ、対処方法を解説する。

左側のパネルには、オペレータが表示される。イベントが発生すると、キャラクターの表情が変化し、キャラクターの背景は、艦の深度に応じて背景が変化する。

1.4 操作方法

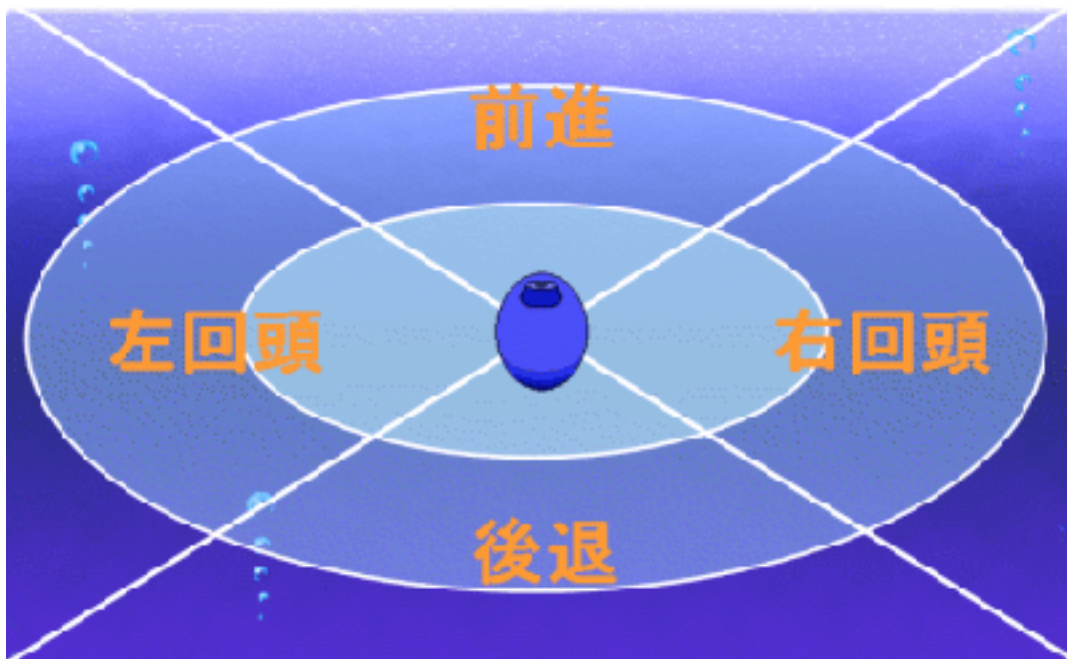


図 2: 操作方法 (移動)

図 2 及び図 3 はゲームのメイン画面の概略図である。端末が PDA であるため、基本的に全ての操作をペン型の入力デバイスを用いて行う。そのため、当ゲームの基本動作である移動はこの画面上をタップすることによって行う。

図 2 が移動時のタップ領域の区分である。水平方向の動作はタップする領域によって決定され、艦の方向に対して相対的な領域に分割される。当ゲームでは、移動によって艦が回頭すると、背景は変化せず艦自体の表示方向が変わる。そのため、このタップ領域は艦の方向に対するものであり、例えば艦が左方向を向いている場合、艦の前方にあたる画面左側の領域が前進となる。

又、タップする位置の艦からの距離によって移動速度が段階的に変化し、艦からの距離が近ければ遅く、遠ざかる程速く移動する。移動速度は現段階で 3 段階程度の変化を予定している。

画面をタップするとペン先が離されるまで移動を続け、後退は前進に対して若干低速になる。又、左右回頭の場合には艦からのタップ位置は回転速度に影響する。艦が回頭すると、前述の様に艦の方向に合わせてタップ領域も回転するが、回頭行動中にはこれは適応されない。つまり、回頭を開始するとそれによってペン先が艦の前方の領域に入っても、ペン先が離さなければそのまま回転を続けることになる。

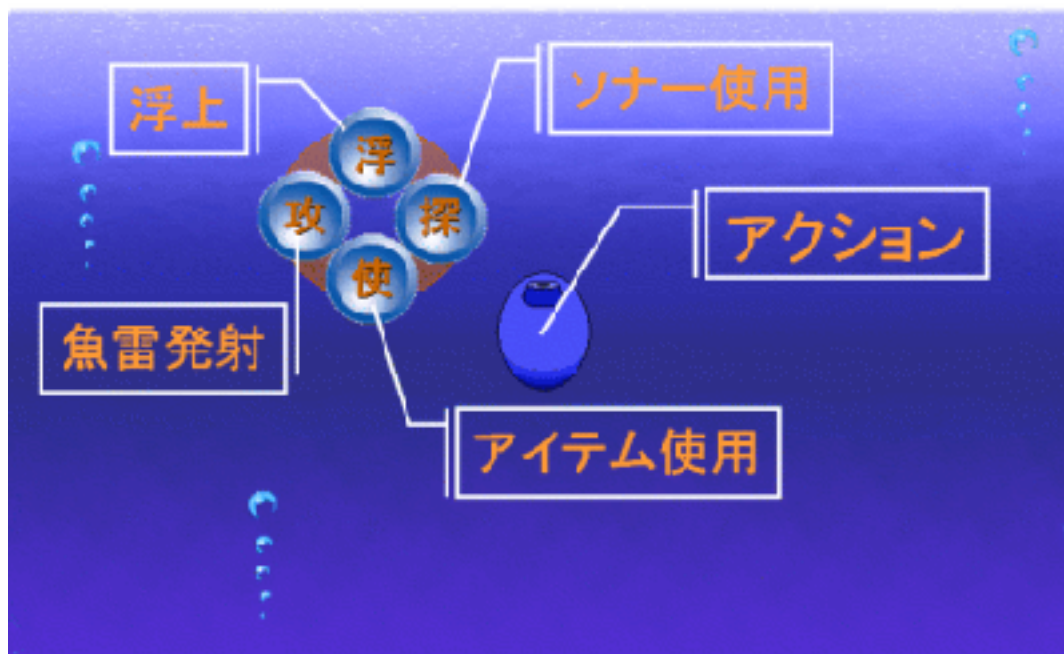


図 3: 操作方法 (アクション)

移動以外のアクションは、艦をタップすることで現れるメニューボタンによって操作する。図3がメニュー表示状態の略図である。

メニューが表示されると移動が出来なくなるが、ボタンをタップすることで各アクションを実行できる。現段階では4つのメニューボタンが存在し、「攻」が攻撃で魚雷の発射などを行い、「使」がアイテム使用、「探」がアクティブソナーによる索敵、「浮」及び「潜」が浮上と潜航で状態によって表示が変化する。

又、当ゲームの端末であるPDAは画面が非常に小さく、必然的に非常に細かい操作が要求される。そのため、各アクションはメニューボタンのタップのみで行えるようにするなど、各操作を極力シンプルに行えるように考慮している。

1.5 世界観の設定

ゲームに対してより親しみやすくなるように、簡単な世界観を設定している。

西暦 30xx 年 世界は水没していた。
人々は居住空間を海底に移し、人々の移動手段は潜水艦になった。
限られた居住空間を争って魚雷を発射することもあった。
そんな現状を見兼ねたお偉いさん達は、オリンピックの水中版
「マリニピック」の開催を提案する。
勝者には合法的に領地が与えられる。
マリニピックの開催を受け、世界中から強者達が揃った。

2 データ構造とモジュールの説明

① ゲームデータの構造

共通の構造		
データ型	変数名	内容
int	type	オブジェクトの型 (キャラ, アイテム, 兵器, エフェクト, etc...)
unsigned short	id	オブジェクトの番号
Point	pos	オブジェクトの座標
int	depth	深さ (0 or 1)

キャラ固有の構造		
データ型	変数名	内容
unsigned short	hp	キャラの耐久度
int	air	酸素濃度
char*	name	プレイヤー名

② 各モジュールの説明

以下に作成するモジュールを説明する。サーバおよびクライアントは、このモジュールを組合せ作成する。

1. ネットワークモジュール
サーバ・クライアント間の通信に対する処理を行うモジュール。
2. コマンド処理モジュール
ネットワーク上でのデータ送受信に対するプロトコルの解析及び生成を行うモジュール。
3. GUI モジュール
ゲーム進行時のパラメータの視覚化及び種々のオブジェクトを表示するモジュール。
4. イベント処理モジュール
X-Window で発生するイベントを処理するモジュール。
5. ゲームモジュール
ゲームを構成するための具体的な計算を行うモジュール。

3 各モジュールの外部関数

3.1 サーバ

以下に、サーバの各モジュールの外部関数を説明する。

1. ネットワークモジュール [server_net.c]

int SetUpServer(int)	
関数名	SetUpServer()
機能	通信のための初期設定をする関数
引数	cnum — クライアント数
返り値	成否 (正值 : 正常終了 / 負値 : エラー)

Bool SendRecvManager(void)	
関数名	SendRecvManager()
機能	通信処理のメインモジュール関数
引数	なし
返り値	終了フラグ (真値: 処理継続 / 偽値: 終了)

int RecvData(int, void*, int)	
関数名	RecvData()
機能	クライアントからのデータを受信する関数
引数	pos — 受信クライアントの ID data — 受信データ格納用バッファ dataSize — 最大受信データサイズ
返り値	受信データサイズ (負値: エラー)

void SendData(int, void*, int)	
関数名	SendData()
機能	クライアントへデータを送信する関数
引数	pos — 送信クライアントの ID data — 送信データ dataSize — 送信データサイズ
返り値	なし

voie Ending(int)	
関数名	Ending()
機能	クライアントとの接続を切断し, プログラムを終了する関数
引数	signum — シグナルナンバー
返り値	なし

2. コマンド処理モジュール [server_command.c]

Bool ExecuteCommand(int*)	
関数名	ExecuteCommand()
機能	受信したデータを解析して, 処理する関数
引数	client — クライアント ID
返り値	終了フラグ (真値: 処理継続 / 偽値: 終了)

void SetSendCommand(char, int, int)	
関数名	SetSendCommand()
機能	コマンドをセットして, 対象のクライアントへ送信する関数
引数	command — コマンド clientID — クライアント ID transID — 送信対象クライアント
返り値	なし

void SetDataBlock(char*, const char*, ...)	
関数名	SetDataBlock()
機能	送信用のデータブロックにデータをセットする関数
引数	dataBlock — データブロック format — データフォーマット ... — セットデータ
返り値	なし

void GetDataBlock(char*, const char*, ...)	
関数名	GetDataBlock()
機能	受信したデータブロックからデータを取得する関数
引数	dataBlock — データブロック format — データフォーマット ... — 取得データ用バッファ
返り値	なし

3.2 クライアント

以下に、クライアントの各モジュールの外部関数を説明する。

1. ネットワークモジュール [client_net.c]

int SetUpClient(char*, char*)	
関数名	SetUpClient()
機能	通信のための初期設定をする関数
引数	hostName — サーバのホスト名 clientName — ログイン名
返り値	成否 (真値: 正常終了 / 偽値: エラー)

Bool SendRecvManager(void)	
関数名	SendRecvManager()
機能	通信処理のメインモジュール関数
引数	なし
返り値	終了フラグ (真値: 処理継続 / 偽値: 終了)

void RecvAllName(void)	
関数名	RecvAllName()
機能	サーバから全クライアントのユーザ名を受信する関数
引数	なし
返り値	なし

int RecvData(int, void*, int)	
関数名	RecvData()
機能	サーバからのデータを受信する関数
引数	data — 受信データ格納用バッファ dataSize — 最大受信データサイズ
返り値	受信データサイズ (負値: エラー)

void SendData(int, void*, int)	
関数名	SendData()
機能	サーバへデータを送信する関数
引数	data — 送信データ dataSize — 送信データサイズ
返り値	なし

void Ending(int)	
関数名	Ending()
機能	ソケットを閉じて、プログラムを正常終了する関数
引数	signum — シグナルナンバー
返り値	なし

2. コマンド処理モジュール [client_command.c]

Bool ExecuteCommand(void)	
関数名	ExecuteCommand()
機能	サーバから受信したデータのコマンドを解析し、処理する関数
引数	なし
返り値	終了フラグ (真値: 処理継続 / 偽値: 終了)

void SetSendCommand(int, int)	
関数名	SetSendCommand()
機能	コマンドをセットして、サーバへ送信する関数
引数	command — コマンド value — コマンドの引数値
返り値	なし

void SetDataBlock(char*, const char*, ...)	
関数名	SetDataBlock()
機能	送信用のデータブロックにデータをセットする関数
引数	dataBlock — データブロック format — データフォーマット ... — セットデータ
返り値	なし

void GetDataBlock(char*, const char*, ...)	
関数名	GetDataBlock()
機能	受信したデータブロックからデータを取得する関数
引数	dataBlock — データブロック format — データフォーマット ... — 取得データ用バッファ
返り値	なし

3. ユーザインタフェースモジュール [client_win.c]

void InitWindows(void)	
関数名	InitWindows()
機能	ウィンドウの初期化処理を行なう関数
引数	なし
返り値	なし

Bool WindowEvent(void)	
関数名	WindowEvent()
機能	イベントを取得し，処理する関数
引数	なし
返り値	終了フラグ (真値: 処理継続 / 偽値: 終了)

void DrawMessage(char*, int, Bool)	
関数名	DrawMessage()
機能	ウィンドウの固定領域にメッセージを表示する関数
引数	message — メッセージ area_id — 領域の ID expose — 再描画フラグ
返り値	なし

void UpdateDisplay(void)	
関数名	UpdateDisplay()
機能	ディスプレイの描画状態を更新する関数
引数	なし
返り値	なし

4 コマンドプロトコルの説明

4.1 サーバ送信 クライアント受信のプロトコル

サーバからクライアントへのデータ伝送には特にコマンドを指定せず，第2節①の「ゲームデータの構造」に示すデータ構造に基づいたプロトコルにより構成される．

4.2 クライアント送信 サーバ受信のプロトコル

コマンド	A (int)
機能	攻撃コマンド (魚雷発射)
引数	武器の ID
対象	サーバ
タイミング	アクティブゲージが貯った状態で，ユーザによって攻撃コマンドが選択された時

コマンド	M (int)
機能	移動コマンド (フィールド移動)
引数	タップしたパネルの ID
対象	サーバ
タイミング	ユーザがメイン画面上の移動コマンド認識領域をタップした時

コマンド	S
機能	索敵コマンド (アクティブソナー使用)
引数	なし
対象	サーバ
タイミング	アクティブゲージが貯った状態で、ユーザによって索敵コマンドが選択された時

コマンド	I (int)
機能	アイテムコマンド (アイテム使用)
引数	アイテムの ID
対象	サーバ
タイミング	アクティブゲージが貯った状態で、ユーザによってアイテムコマンドが選択された時

コマンド	F (int)
機能	浮上潜航コマンド (レイヤー切替え)
引数	レイヤーの ID (0: 水中 / 1: 水上)
対象	サーバ
タイミング	アクティブゲージが貯った状態で、ユーザによって浮上潜航コマンドが選択された時

5 各人の担当

ネットワーク担当: 山内
GUI 担当: 後藤田, 山形
ゲーム担当: 中川

6 ガントチャート



1ヶ月	2ヶ月	3ヶ月	6ヶ月	2005年																														
2005/01/01～2005/01/31				1月																														
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
渡嶋田 中	ゲーム内容の把握及び理解																	パワーポイント		最終調整														
	GPSの実装とゲームのGPS対応																																	
中川 健二	ゲーム内容の把握及び理解																	パワーポイント		最終調整														
	GPSの実装とゲームのGPS対応																																	
山形 達平	ゲーム内容の把握及び理解																	パワーポイント		最終調整														
	GPSの実装とゲームのGPS対応																																	
山内 学	ゲーム内容の把握及び理解																	パワーポイント		最終調整														
	GPSの実装とゲームのGPS対応																																	

参考文献

[1] 光原弘幸, ソフトウェア設計及び実験 2(ユーザインターフェイス 1) ユーザインターフェイス概論と X-Window プログラミング基礎, 徳島大学工学部知能情報工学科, 2004/10/1

[2] 得重仁 毛利公美, ソフトウェア設計および実験 2 -ネットワークプログラミング応用-, 徳島大学工学部知能情報工学科, 2004/10/8

[3] 光原弘幸, ソフトウェア設計及び実験 2(ユーザインターフェイス 2) ゲームとユーザインターフェイス-ゲームと X-Window プログラミング, 徳島大学工学部知能情報工学科, 2004/10/15

[4] 泓田正雄, ソフトウェア設計及び実験 2 モジュール化, 徳島大学工学部知能情報工学科, 2004/10/22