

どきどきしりとりなんもじ?

~ソケット通信を利用したしりとりチャットシステム~

プログラミング演習 C グループ : 伊藤 冬子

Fuyuko ITO

1 はじめに

「しりとり」は日本人ならではの言葉遊びである。他の人の言った言葉の語尾から始まる単語を考えるだけのシンプルなゲームであるが、子供から大人までみんなが楽しめる魅力を持っている。また現在では、コンピュータ、ネットワークが共に普及していることから、ネットワークを用いたコミュニケーションが容易となっている。そこで、これらの技術と日本に古くから伝わる言葉遊びであるしりとりを融合させ、新しい形のしりとりを実現するシステムを作成した。

このシステムでは、ソケット通信を利用したチャットシステムを基盤とし、文字数制限や時間制限を組み込むことで、タイトルの「どきどきしりとりなんもじ?」の示すとおりどきどきするような、普通のしりとりでは味わうことのできないゲーム性の高いしりとりを実現した。

2 システム概要

「どきどきしりとりなんもじ?」はチャットを利用した複数人で行うしりとりゲームである。参加者はサーバに接続することでしりとりを開始する。他の参加者もサーバに接続することですでに始まっているしりとりに参加できる。各参加者は発言することでポイントが貯まり、各参加者のレベルは上がっていく。また、参加者はルール違反をするとゲームオーバーとなり強制的に退出させられる。ゲームの目的は、できるだけ長くしりとりを続けレベルアップしていくことである。以下に「どきどきしりとりなんもじ?」のゲーム機能の詳細を示す。

● 参加人数

しりとりチャットには何人でも参加可能である。途中参加も可能で、その場合は参加した時点で最後に入力された単語から始める。なお、しりとりチャットでは発言の順番は決められていないため、誰でもいつでも発言することができる。

● しりとり

一番最初入室した参加者が「しりとり」という単語から始める。なお、語尾に「ん」がついたり、前の単語の語尾と入力した単語の語頭が一致していない、つまりしりとりになっていない場合はルール違反でゲームオーバーとなる。

● 文字数制限

しりとりチャットでは、次に入力する単語の文字数を制限する。文字数は 2~5 文字で、ランダムに提示する。制限文字数と異なる文字数の単語を入力した場合、ルール違反でゲームオーバーとなる。

● 既出単語の使用制限

しりとりチャットでは既出の単語を再度入力することはできない。入力した単語は、既出単語と重複していないか照合するためにデータベースに登録される。データベースと照合し、既出単語を再度入力した参加者はゲームオーバーとなる。

● 入力時間制限

前の単語と文字数を提示してから制限時間内に次の単語を入力しなければならない。制限時間はレベル 1 では 60 秒に設定しており、レベルが上がるごとに短くなる。制限時間内に入力できなかった場合は、ゲームオーバーとなる。

● ポイント制

前の単語と文字数が提示されてから次の単語を入力するまでの時間が規定時間内であればポイントが加算される。加算するポイントは 3pts, 2pts, 1pts のいずれかで、規定時間が短いものほどポイントは大きい。なお、レベルが上がると入力までの規定時間も短くなり、ポイントを獲得することが困難となる。

● レベルアップ

ポイントが 20pts 貯まるとレベルアップする。レベルアップすると単語入力までの制限時間が短くなり、ポイント加算の入力までの規定時間も短くなる。また、後で述べるタンマカードを 1 枚獲得できる。

● タンマカード

タンマカードとは、入力時間制限を無効にするカードで、どうしても制限時間内に単語が思い浮かばない場合に使用できる。タンマカードは累積不可能で、1 枚しか保持できない。

● ゲームオーバー

しりとりにならなかった場合や、文字数制限を違反した場合、既出単語を入力した場合、制限時間内に入力できなかった場合は、その参加者をゲームオーバーとし強制的に退出させる。また、制限時間内に全員が入力できなかった場合には全員を退出させる。なお、次回参加時にはポイント・レベルなどは初期化されるため、ゲームオーバー時からの再開は不可能である。

3 状態遷移

「どきどきしりとりなんもじ?」のベースとなるチャット部分は、一種の状態遷移のモデルとして捉えることができる。チャットは開始状態から参加者の発言によって状態が遷移していき、発言がなくなり参加者全員が退室すると終了状態となる。Fig. 1 にチャットの状態遷移図を示す。



Fig. 1 チャットの状態遷移図

Fig. 1 に示すように、チャット単体では単純な一方通行の状態遷移になる。今回作成した「どきどきしりとりなんもじ?」には、この単純な状態遷移で表されるチャットをベースに、2章に示したしりとり機能やゲーム性を高めるための種々の機能を追加した。「どきどきしりとりなんもじ?」全体の状態遷移を Fig. 2 に示し、Table 1 に Fig. 2 に示す各状態の詳細について示す。各機能については後に詳述する。なお、Fig. 1 では発言するたびに次の状態に遷移しているように表記しているが、Fig. 2 ではコミュニケーションの状態遷移ではなく、システムとしての状態遷移を示している。そのため、発言すると入力待ちである状態 S1 に戻るように表記している。

Fig. 2 の状態遷移について詳述する。基本的なパターンは左側の S1 から S7 までの状態遷移の繰り返しである。しりとりチャットに参加すると S1 の入力待ちの状態になり、前の言葉に対して発言するとそれに対して S2 ~ S5 の判断を行う。すべてクリアすれば S6 において解答時間に応じてポイントを加算し、S7 においてポイントがレベルアップの基準に達しているかどうかの判断を行う。達していなければ同じレベルのまま S1 の入力待ちの状態に遷移する。S7 においてレベルアップをすると、ポイント加算のための解答時間が短くなり、同時にタンマカードを得られる。その後の状態遷移は基本的に前のレベルと同様である。タンマカードを発動した際は、S'6 の制限時間内に解答したかどうかを判断する状態はなくなり、ポイントは解答時間に関係なく一定ポイントを加算し、S'7 のレベルアップの判断の状態に遷移する。

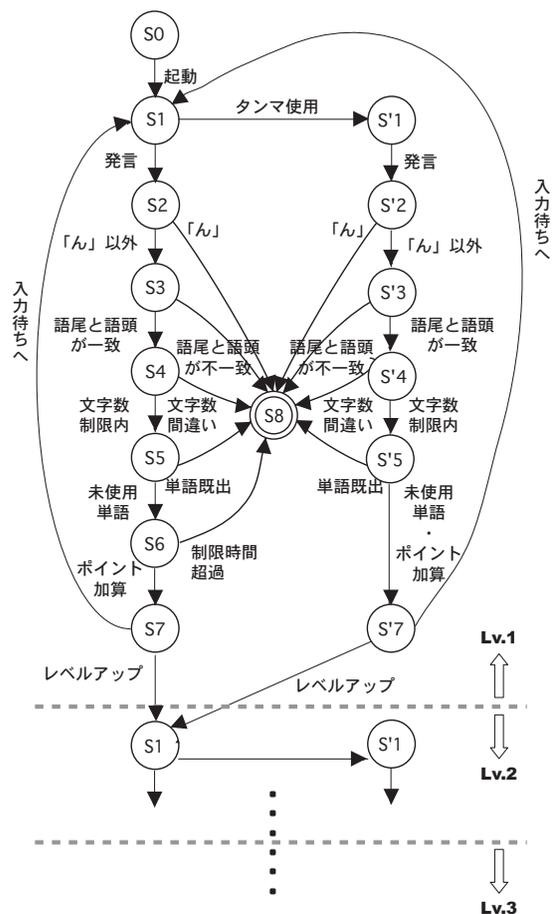


Fig. 2 「どきどきしりとりなんもじ?」の状態遷移図

Table 1 状態遷移図における各状態

状態番号	状態の詳細
S0	開始状態
S1	入力待ち
S2	語尾が「ん」でないか判断
S3	前の言葉の語尾と入力の語頭が一致しているか判断
S4	文字数が制限通りであるか判断
S5	単語が既出かどうか判断
S6	解答時間によるポイントの判断
S7	レベルアップの判断
S8	ゲームオーバー（受理状態）
S'i	タンマカード発動時の Si (i=1~7)

4 システムの構成

4.1 ソケット通信

今回作成したしりとりチャットはソケット通信¹⁾を用いて作成している。ソケットはUNIX (4.2BSD) に用いて作成している。ソケットはUNIX (4.2BSD) に TCP/IP が実装された際に提供された通信の仕組みであり、TCP や UDP で通信を行う際にはソケットを利用してコネクション (通信路) を確立し、通常ファイル入出力と同じ手順 (read/write) によって通信を行う。こ

のソケットを利用して通信を行うには、サーバプログラムとクライアントプログラムの2つが必要となる。Fig. 3にソケット通信のモデル図を示す。

- サーバプログラム
 - ソケットを用意して接続要求を待つ
- クライアントプログラム
 - ソケットを用意してサーバに接続要求を行う

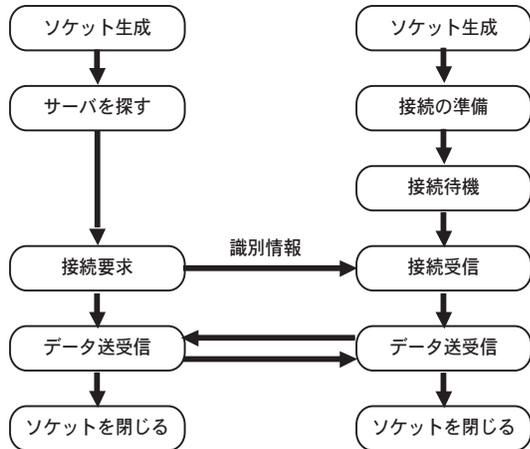


Fig. 3 ソケット通信のモデル図

Fig. 3より、ソケット通信ではまずサーバ・クライアント双方でソケットを生成する。今回作成したしりとりチャットのベースとなるチャットプログラムでは、クライアント側でソケットにサーバのIPアドレス・ポート番号の識別情報を指定し、回線を開くことでサーバとのTCP接続を確立している。サーバはこれを受理し、クライアントに対してこのTCP接続を管理するスレッドを生成する。詳しい構成については4.2節に示す。

4.2 システムモデル

作成したしりとりチャットはチャットプログラムをベースにして、しりとり機能をはじめとする各種機能を組み込んだ。しりとりチャットの具体的なシステムの構成は以下のFig. 4に示す。

本システムではクライアントプログラムとサーバプログラムが存在しており、サーバプログラムを実行すると接続待ち状態となる。クライアントからのTCP接続を受理すればそのクライアントとのTCP接続管理用スレッドを生成する。このスレッド上で、そのクライアントからのメッセージを受け取り、全スレッドにメッセージを送信するメソッドを利用して全クライアントにそのメッセージを送信することができる。同様に、しりとりや各種制限の処理も全クライアントに関係しているため、全クライアントにしりどりの現在の単語や制限文字数などを送信できるようなメソッドをサーバ

プログラムに実装した。これにより、全クライアント間での1つのしりどりの共有を実現した。また、各クライアントとのTCP接続管理用スレッドは、各クライアントによって異なる制限時間やポイント、レベルなどを管理している。つまり基本的には、全クライアントで同期を取る必要のある機能はサーバプログラムのメソッドとして組み込み、各クライアントによって処理の異なる機能は各クライアントごとのスレッドに組み込んでいる。

一方、クライアントプログラムにはしりとり関連の機能の組み込みは行っておらず、一般的なチャットプログラムと同様に入力文字列をサーバに送り、サーバから受信した文字列を表示するのみである。ただし、しりとり本文のみでなくレベルやポイントのデータも扱うため、サーバとクライアントの間で送受信するメッセージの先頭にはデータの種類を示すフラグが埋め込まれている。サーバとクライアントは、フラグを元に、受信したデータの扱いやデータ受信後の処理をどのようにするか決定する。

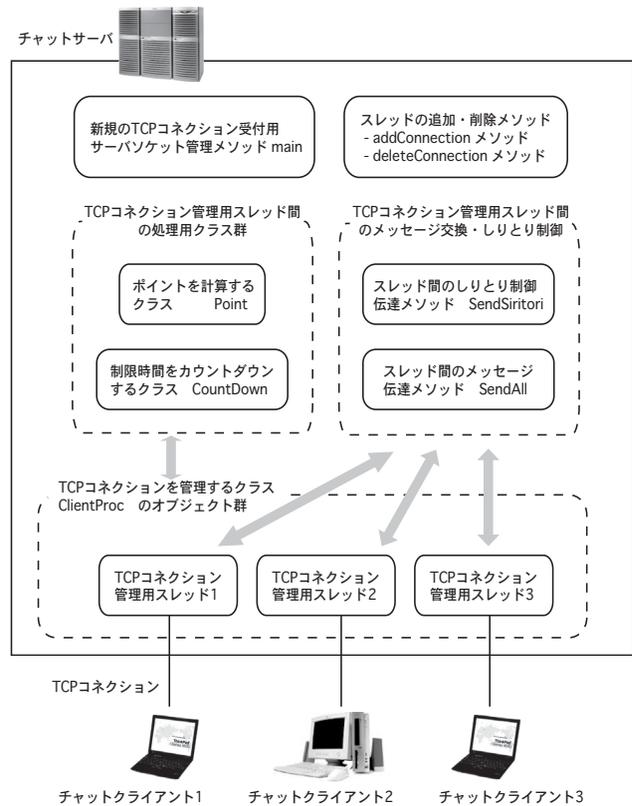


Fig. 4 しりとりチャットのシステムモデル図

5 動作確認

「どきどきしりどりなんもじ?」を実行する際の手順を以下に示す。本システムを使用するには、サーバプログラムとクライアントプログラムを同時に起動する必要がある。まず、サーバプログラムを起動しておき、次に

クライアントプログラムを起動すると、Fig. 5のようなスタート画面が現れる。ここでは、サーバプログラムを起動しているコンピュータのIPアドレス、通信に用いるポート番号、名前、アイコンを入力、選択する。全て入力し終わりスタートボタンを押すと通信が確立され、Fig. 6のプレイ画面に切り替わる。

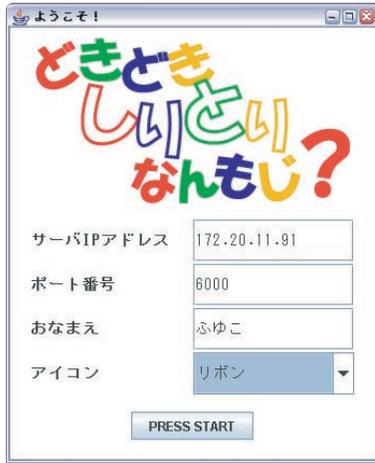


Fig. 5 スタート画面

プレイ画面では、画面上部に入力すべき単語の語頭の1文字とそれ以外の文字がで大きく表示される。例えば、「ま」から始まる5文字の言葉を入力すべき場合はFig. 6のように「ま」と表示される。入力までの制限時間は画面左部分にカウントダウン形式で表示される。この制限時間は、参加者自身が単語を入力するまでクリアされないの、なるべく他人より早く解答をたくさん入力していくことが勝ち残っていく秘訣と言える。参加者の名前、現在のレベルとポイント、次のレベルに上がるために必要なポイント数は画面右上部分に表示されている。獲得できるポイントは画面左側部分に表示されている「ポイント情報」にしたがい、入力時間によって変化する。参加者自身のポイント表示の横に手のひらのマークが表示されているときは、タンマカードが使用可能な状態である。タンマカードは画面下部の「ちょっとタンマ!」ボタンを押すことで使用でき、カウントダウンを止めて入力時間制限を無効にする。プレイ画面上ではカウントダウンが止まり、入力する単語を落ち着いて考えることができる。また、2人のクライアントでしりとりを行っている際のしりとりメッセージの履歴をFig. 7に示す。これよりクライアントが2人以上の場合でも、ネットワークを介して同期を取り、制限文字数などのデータを共有してしりとりチャットを行えることを確認できた。



Fig. 6 プレイ画面

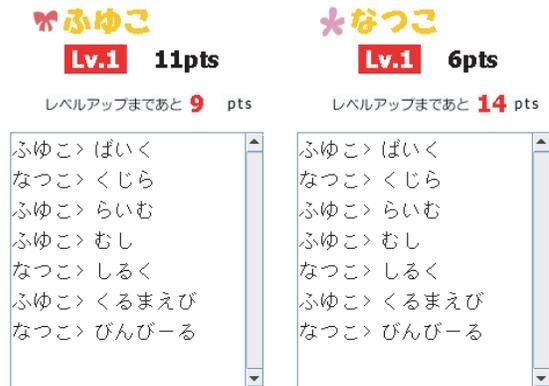


Fig. 7 複数コンピュータでの同時プレイの動作確認

6 おわりに

本報告では、ネットワークを介して全クライアント間で同期をとりながらしりとりを進める「どきどきしりとりなんもじ?」を構築した。基盤となるチャットシステムの実装にはソケット通信を用い、しりとり機能や文字数制限、入力時間制限、レベルアップ機能などの実装も行った。その結果、一般のしりとりと比べ、スリル、スピード感溢れるゲーム性の高いしりとりチャットシステムを実現できた。

参考文献

- 1) 小高 知宏. 基礎からわかる TCP/IP Java ネットワークプログラミング. オーム社. 2002.
- 2) 村上 列. Java スレッド 完全制覇. 技術評論社. 2002.