Terminology

This specification uses the terms MUST, SHOULD, and MAY as defined in

RFC 2119 [rfc2119]

We also use the following terminology to distinguish between visiting pacman and various game objects:

- LOCAL: the game object is a local game object, and is currently on the local screen.
- AWAY: our pacman is current away on the remote screen.
- REMOTE: a game object on the remote screen that our AWAY pacman might interact with.
- FOREIGN: the other player's pacman, when it is visiting our screen.

The Pacman protocol runs over TCP, using a well known port of 5432.

The player that set the password and run the server becomes the server while the player that join the server with a password becomes the client.

There are 9 types of message: MAZE, GAME STATUS, FOREIGN PACMAN, GO HOME, PACMAN UPDATE, GHOST UPDATE, EAT, FOREIGN PACMAN ATE GHOST.

When the game starts, each player needs to send the other player a copy of its MAZE. During the game, players need to send each other update message about its LOCAL game object to update REMOTE object in the other game player’s model. When one player’s pacman is AWAY, it needs to send message to update the other player about any impact this AWAY pacman has on the REMOTE game object.

Message contents:

The contents of a MAZE message are:

- Type: MAZE
- Value: a list of 31 strings. Each string has a length of 84 chars which are used to represent the maze. Only transmitted once when game starts.

The contents of a GAME_STATUS are:

- Type: GAME_STATUS
- Value: a string that update the other player about the current game status, which MUST be one of the following:
  - WAIT_FOR_NEXT_LEVEL
  - GAME_OVER
  - READY_TO_RESTART

The contents of a FOREIGN_PACMAN:

- Type: FOREIGN_PACMAN
- VALUE: a Boolean PACMAN_ARRIVED used to determine whether the FOREIGN pacman left or arrived. True for arrived and False for left.

The content of a GO_HOME message:

- Type: GO HOME
- Value: a fixed string "GO_HOME" to ask the other player's model to call back its AWAY pacman.

The contents of EAT message:

- Type: EAT
- Value: a tuple of two integers (X,Y positions) to represent the position of the food got eaten in grid square, a Boolean IS_FOREIGN to show whether the food has been eaten in LOCAL or REMOTE game object and another Boolean IS_POWERPILL to show if the food got eaten is a power pill or not.

The contents of FOREIGN_PACMAN_ATE_GHOST message:

- Type: FOREIGN_PACMAN_ATE_GHOST message
- Value: an integer GHOSTNUM of ghost number to update the other player's model that the ghost with this number has been eaten and the other player's LOCAL game object needs to be updated.

The contents of PACMAN_UPDATE:

- Type: PACMAN_UPDATE
- Value: a tuple of two integers (X,Y positions) to represent the position of our AWAY pacman for the other player's model to display. A string DIRECTION to represent the direction of moving for our AWAY pacman and a float SPEED to represent the speed it is moving at.

The contents of GHOST_UPDATE:

- Type: GHOST_UPDATE
- Value: Four lists that consist of number, position, moving direction, moving speed and mode of that particular ghost.

Timing for sending message

- MAZE is only sent once when the connection is established
- Three different types of GAME_STATUS message ares sent under these conditions:
    - WAIT_FORNEXT_LEVEL: when we have finished the current level in LOCAL game object)
    - GAME_OVER: when no life left and game over in LOCAL game object
    - READY_TO_RESTART: when game ends and player has pressed 'r' to show he is ready
- Two different types of FOREIGN_PACMAN messages are sent under these conditions:
    - left: when our AWAY pacman dies in REMOTE game object / when game is reset and our pacman is still AWAY/ when our AWAY pacman step on the tunnel and comes back to LOCAL game object.
    - arrived: when our LOCAL pacman goes through the tunnel and become away

- GO_HOME message is sent when the game goes to next level but there are still FOREIGN pacman in our LOCAL game object.
- Four different types of EAT messages are sent under these conditions:
    - IS_FOREIGN is false, IS_POWERPILL is false: our pacman ate a food in LOCAL game object
    - IS_FOREIGN is true, IS_POWERPILL is false: our pacman ate a food in REMOTE game object
    - IS_FOREIGN is false, IS_POWERPILL is true: our pacman ate a power pill in LOCAL game object
    - IS_FOREIGN is true, IS_POWERPILL is true: our pacman ate a power pill in REMOTE game object
- FOREIGN_PACMAN_ATE_GHOST is sent when our AWAY pacman ate a ghost in REMOTE game object
- PACMAN_UPDATE SHOULD be sent every 20ms when our pacman is AWAY to make the movement of pacman on the remote screen smooth. If a computer cannot maintain 50 frames per second, PACMAN_UPDATE messages MAY be sent once per frame, as it is unreasonable to update the remote computer more often than the local one and it is only for the purpose of display.
- GHOST_UPDATE MUST be sent every 20ms when there is foreign pacman in our LOCAL game object because the other player's model need to use these data to interact with its AWAY pacman.

Message Encoding

Messages are fixed format, ASCII encoded and separated by a newline character. CR characters MUST NOT be sent. More than one message MAY be sent consecutively in a single packet - this may be useful to reduce overhead when sending multiple messages such as both PACMAN_UPDATE and EAT messages.

Message format

- MAZE messages are of variable length, encoded as follows:
    - MAZE: maze=<string1> <string2> …… <string 31><newline>
    - "MAZE" MUST be capitalized, and MUST be at the start of the connection or immediately follow a newline character. The fields MUST be separated by a single space character. The fields MUST be in this order and there MUST be 31 fields exactly. The protocol MUST ignore any unknown fields.
    - <string1><string2>…<string31> are decimal ASCII encoding of the string used to represent that row of the maze, and each string MUST be of the length 84
- GAME_STATUS messages are of variable length, encoded as follows:
    - STATUS: status=<game_status><newline>
    - "STATUS" MUST be capitalized, and MUST immediately follow a newline character. The protocol MUST ignore any unknown fields.
    - <game_status> is a decimal ASCII encoding of string used to represent the game status. It MUST be one of these three: "WAIT_FOR_NEXT_LEVEL", "GAME_OVER" or "READY_TO_RESTART".
- FOREIGN_PACMAN messages are of variable length, encoded as follows:

  - FOREIGN: arrived=<foreign_pacman_arrived><newline>
  - "FOREIGN" MUST be capitalized, and MUST immediately follow a newline character. The protocol MUST ignore any unknown fields.
  - <foreign_pacman_arrived> is a decimal encoding of Boolean used to represent whether our pacman enter or exit the REMOTE game object. It MUST be true when our pacman enters the REMOTE game object and false when our pacman exits the REMOTE game object and this value MUST be different from the Boolean which has the value of true if our pacman is AWAY and false for it is local
- GO_HOME messages are of fixed length, encoded as follows:
  - GOHOME: home=<go_home><newline>
  - "GOHOME" MUST be capitalized, and MUST immediately follow a newline character. The protocol MUST ignore any unknown fields.
  - <go_home> is a decimal ASCII encoding of a string. This string MUST be "GO_HOME"
- EAT messages are of variable length, encoded as follows:
  - EAT: xposition=<x_pos> yposition=<y_pos> isforeign=<is_foreign> ispill=<is_powerpill><newline>
  - "EAT" MUST be capitalized, and MUST immediately follow a newline character. The fields MUST be separated by a single space character. The fields can be in any order. The protocol MUST ignore any unknown fields.
  - <x_pos> is a decimal encoding of an integer used to represent the row in grid squares of the food. It MUST be in between 0 and 30
  - <y_pos> is a decimal encoding of an integer used to represent the column in grid squares of the food. It MUST be in between 0 and 83
  - <is_foreign> is a decimal encoding of a Boolean used to represent if our pacman is AWAY. It has the value of true if our pacman is AWAY and false if our pacman is LOCAL
  - <is_powerpill> is a decimal encoding of a Boolean used to represent if our pacman ate a power pill. It has the value of true if our pacman ate a power pill and false if our pacman ate a normal food
- FOREIGN_PACMAN_ATE_GHOST messages are of variable length, encoded as follows:
  - ATEGHOST: ghostnumber=<ghostnum><newline>
  - "ATEGHOST" MUST be capitalized, and MUST immediately follow a newline character. The protocol MUST ignore any unknown fields.
  - <ghostnum> is a decimal encoding of an integer used to represent the number of the ghost our pacman just art. It MUST be in the range of 0 to 3 as there are only 4 ghosts and the ghost with this number MUST not be frightened.
- PACMAN_UPDATE messages are of variable length, encoded as follows:
  - PACMAN: xposition=<x_pos> yposition=<y_pos> direction=<direction> speed=<speed><newline>
  - "PACMAN" MUST be capitalized, and MUST immediately follow a newline character. The fields MUST be separated by a single space character. The fields can be in any order. The protocol MUST ignore any unknown fields.

  - <x_pos> is a decimal encoding of an integer used to represent the row in grid squares of our AWAY pacman. It MUST be in between 0 and 30
  - <y_pos> is a decimal encoding of an integer used to represent the column in grid squares of our AWAY pacman. It MUST be in between 0 and 83
  - <direction> is a decimal ASCII encoding of a string used to represent the direction our pacman is moving in, it can only be one of the following four: "LEFT", "RIGHT", "UP", "DOWN"
  - <speed> is a decimal encoding of a float used to represent the speed our pacman is moving at.
- GHOST_UPDATE messages are of variable length, encoded as follows:
  - GHOST: ghostone=[ghostnum=<ghostnum1> x_position=<x_pos1> y_position=<y_pos1> direction=<direction1> speed=<speed1> mode=<mode1>] ghosttwo=[ghostnum=<ghostnum2> x_position=<x_pos2> y_position=<y_pos2> direction=<direction2> speed=<speed2> mode=<mode2>]… ghostfour=[ghostnum=<ghostnum4> x_position=<x_pos4> y_position=<y_pos4> direction=<direction4> speed=<speed4> mode=<mode4>]<newline>
  - "GHOST" MUST be capitalized, and MUST immediately follow a newline character. The fields MUST be separated by a single space character. The fields can be in any order. The protocol MUST ignore any unknown fields.
  - <ghostnum> is a decimal encoding of an integer used to represent the number of the ghost our pacman just art. It MUST be in the range of 0 to 3 as there are only 4 ghosts and the ghost with this number MUST not be frightened.
  - <x_pos> is a decimal encoding of an integer used to represent the row in grid squares of this ghost. It MUST be in between 0 and 30
  - <y_pos> is a decimal encoding of an integer used to represent the column in grid squares of this ghost. It MUST be in between 0 and 83
  - <direction> is a decimal ASCII encoding of a string used to represent the direction this ghost is moving in, it can only be one of the following four: "LEFT", "RIGHT", "UP", "DOWN"
  - <speed> is a decimal encoding of a float used to represent the speed this ghost is moving at.
  - <mode> is a decimal ASCII encoding of a string used to represent the mode of this ghost, it MUST be "FRIGHTEN" or "CHASE"