Pacman Protocol Specification

Student number: 22009088

Terminology

This specification uses the terms MUST, SHOULD and MAY as defined in RFC 2119 [rfc2119].

The protocol uses TCP and UDP for different messages with the same port of 5432.

There are 6 messages types:

1. Maze update.
2. Pacman update.
3. Ghost update.
4. Pacman died.
5. Eat.
6. Status update.

Following terminology are used to distinguish between pacmans and game objects.
LOCAL: The game object is local.
AWAY: Our pacman is away on the other screen.
REMOTE: The game object on the other screen which can interact with the AWAY pacman.
FOREIGN: The other pacman on our screen.

Player 1 vs Player 2

Each computer recognizes themselves as Player 1, and the other one as Player 2. The scores on the top are shown accordingly.

Our pacman is always yellow, while the other pacman is always pink. (This information does not need to be transmitted through the protocol).

Coordinate system

The canvas width is 650 pixels. 0 represents the left edge of the screen, and 649 represents the right edge of the screen.

The canvas height is 800 pixels. 0 represents the top edge of the screen, and 799 represents the bottom edge of the screen.

The x, y coordinates for all the game objects are based on this coordinate system.

Messages

Type 1: Maze update

Protocol used: TCP

Content:

The LOCAL maze is divided into 28 columns and 31 rows. Each square can be represented by x and y positions in the maze, from (0, 0), (0, 1)... (1, 0), (1, 1)... to (27, 30). Each square has a side length of 20 pixels, and the square on the top left corner is 50 pixels away from both the top edge and the left edge of LOCAL screen.

The value, which is an unsigned integer in the range 0-5 are sent for each square:
0: Empty.
1: Wall.
2: Food.
3: Powerpill.
4: Left tunnel.
5: Right tunnel.

Timing:
Maze update messages SHOULD be sent once at the start of the game, after restarting the game, or when LOCAL game moves to a new level.

Package:
Each message of this type SHOULD be sent in one package.

Maze update messages consist of 330 bytes, encode as follows:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| T |                       sequence number                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|0,0 |0,1 |0,2 | 0,3 | 0, 4 | 0,5 | 0, 6 | 0,7 | 0,8 | 0, 9 | 0,10     (1 more bit is needed)
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-

...

0                               1
0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| 27,27    | 27, 28    |    27, 29    | 27, 30    |        U         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

-T: 3 bit type field. Type Maze update has value 0.

-sequence number: 29 bit unsigned integer with big endian byte order. Its initial value is 0. For each new message sent, the value is increased by 1. If the value reaches (2^29) - 1, it wraps back to 0.

-0,0: 3 bit unsigned integer representing the value of the square at position (0, 0).

-0,1: 3 bit unsigned integer representing the value of the square at position (0, 1).

...

-27,30: 3 bit unsigned integer representing the value of the square at position (27, 30).

-U: 4 bit unused to maintain byte alignment. MUST be set to 0. The receiver MUST ignore these bits.

Type 2: Pacman update

Protocol used: UDP

Content:

The x coordinate of our pacman, an unsigned integer in the range 0-649. The unit is pixel.

The y coordinate of our pacman, an unsigned integer in the range 0-799. The unit is pixel.

The direction of our pacman, an unsigned integer in the range 0-3. 0 means facing up, 1 means facing right, 2 means facing down and 3 means facing left.

The speed of our pacman, a signed floating point number. The unit for the speed is pixel per frame, which is pixel per 1/60 second. The speed SHOULD be non-negative, as 0 means stop and positive means moving in its direction.

The screen which our pacman is on, an unsigned integer in the range 0-1. 0 means that our pacman is LOCAL, and 1 means that our pacman is AWAY on the REMOTE screen.

Our current score, an unsigned integer in the range 0 - (2^32)-1.

Timing:
Pacman update messages SHOULD be sent once for every frame of the game, which is 60 times per second.

Package:
Each message of this type SHOULD share one package with one message of Type 3: Ghost update (below) at that frame, since the timing for both messages is once per frame.

Pacman update messages consist of 16 bytes, encode as follows:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| T |                   sequence number                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| x coordinate   |   y coordinate   | D |S|        U          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                           speed                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                           score                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

-T: 3 bit type field. Type Pacman update has value 1.

-sequence number: 29 bit unsigned integer with big endian byte order. Its initial value is 0. For each new message sent, the value is increased by 1. If the value reaches (2^29) - 1, it wraps back to 0.

-x coordinate: a 10 bit unsigned integer with big endian byte order.

-y coordinate: a 10 bit unsigned integer with big endian byte order.

-D: direction of our pacman, a 2 bit unsigned integer.

-S: the screen that our pacman is on, a 1 bit unsigned integer.

-U: 9 bit unused to maintain byte alignment. MUST be set to 0. The receiver MUST ignore these bits.

-speed: a 32 bit signed floating point number using IEEE-754 format with big endian byte order.

-score: a 32 bit unsigned integer with big endian byte order.

Type 3: Ghost update

Protocol used: UDP

Content:

The x coordinates of four LOCAL ghosts, unsigned integers in the range 0-649. The unit is pixel.

The y coordinates of four LOCAL ghosts, unsigned integers in the range 0-799. The unit is pixel.

The directions of four LOCAL ghosts, unsigned integers in the range 0-3. 0 means facing up, 1 means facing right, 2 means facing down and 3 means facing left.

The speeds of four LOCAL ghosts, signed floating point numbers. The unit for the speed is pixel per frame, which is pixel per 1/60 second. The speed SHOULD be non-negative, as 0 means stop and positive means moving in its direction.

The modes of four LOCAL ghosts, unsigned integers in the range 0-4:
0: Scatter.
1: Chase.
2: Frighten.
3: Frighten and trapped.
4: Eyes.

Timing:
Ghost update messages SHOULD be sent once for every frame of the game, which is 60 times per second.

Package:
Each message of this type SHOULD share one package with one message of Type 2: Pacman update at that frame, since the timing for both messages is once per frame.

Ghost update messages consist of 36 bytes, encode as follows:

```
0                   1                   2                   3
```

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| T |                    sequence number                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| x coordinate 1 |    y coordinate 1 | D1| M1 |      U           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                         speed 1                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-

...

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| x coordinate 4 |    y coordinate 4 | D4| M4 |      U           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                         speed 4                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

-T: 3 bit type field. Type Ghost update has value 2.

-sequence number: 29 bit unsigned integer with big endian byte order. Its initial value is 0. For each new message sent, the value is increased by 1. If the value reaches (2^29) - 1, it wraps back to 0.

-x coordinate 1: a 10 bit unsigned integer with big endian byte order for LOCAL ghost 1.

-y coordinate 1: a 10 bit unsigned integer with big endian byte order for LOCAL ghost 1.

-D1: direction of LOCAL ghost 1, a 2 bit unsigned integer.

-M1: the mode of LOCAL ghost 1, a 3 bit unsigned integer.

-U: 7 bit unused to maintain byte alignment. MUST be set to 0. The receiver MUST ignore these bits.

-speed 1: a 32 bit signed floating point number using IEEE-754 format with big endian byte order for LOCAL ghost 1.

Type 4: Pacman died

Protocol used: TCP

Content:

The x coordinate of our pacman, an unsigned integer in the range 0-649. The unit is pixel.

The y coordinate of our pacman, an unsigned integer in the range 0-799. The unit is pixel.

The screen which our pacman is on, an unsigned integer in the range 0-1. 0 means that our pacman is LOCAL, and 1 means that our pacman is AWAY on the REMOTE screen. When our pacman is LOCAL, it can only be killed by LOCAL ghosts. While when it is away, it can only be killed by REMOTE ghosts.

The number of the ghost which kills our pacman, an unsigned integer in the range 0-3.

The remaining lives of our pacman, an unsigned integer in the range 0-5.

Timing:
Pacman died messages SHOULD be sent once when our pacman is killed.

Package:
Each message of this type SHOULD be sent in one package.

Pacman died messages consist of 8 bytes, encode as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| T |                    sequence number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| x coordinate    |   y coordinate   |S| N | L   |     U      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

-T: 3 bit type field. Type Pacman died has value 3.

-sequence number: 29 bit unsigned integer with big endian byte order. Its initial value is 0. For each new message sent, the value is increased by 1. If the value reaches (2^29) - 1, it wraps back to 0.

-x coordinate: a 10 bit unsigned integer with big endian byte order.

-y coordinate: a 10 bit unsigned integer with big endian byte order.

-S: the screen that our pacman is on, a 1 bit unsigned integer.

-N: the number of ghost which kills our pacman, a 2 bit unsigned integer.

-L: the remaining lives of our pacman, a 3 bit unsigned integer.

-U: 6 bit unused to maintain byte alignment. MUST be set to 0. The receiver MUST ignore these bits.

Type 5: Eat

Protocol used: TCP

Content:

The x coordinate of our pacman, an unsigned integer in the range 0-649. The unit is pixel.

The y coordinate of our pacman, an unsigned integer in the range 0-799. The unit is pixel.

The screen which our pacman is on, an unsigned integer in the range 0-1. 0 means that our pacman is LOCAL, and 1 means that our pacman is AWAY on the REMOTE screen. When our pacman is LOCAL, it can only eat LOCAL foods, powerpills and ghosts. While when it is away, it can only eat REMOTE foods, powerpills and ghosts.

The type of things our pacman eats, an unsigned integer in the range 0-5:
0: Food.
1: Powerpill.
2: Ghost 1.
3: Ghost 2.
4: Ghost 3.
5: Ghost 4.

Timing:
Eat messages SHOULD be sent once when our pacman eats a food, a powerpill or a ghost.

Package:
Each message of this type SHOULD be sent in one package.

Eat messages consist of 7 bytes, encode as follows:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| T |                    sequence number                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| x coordinate     |     y coordinate     |S| E     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

-T: 3 bit type field. Type Eat has value 4.

-sequence number: 29 bit unsigned integer with big endian byte order. Its initial value is 0. For each new message sent, the value is increased by 1. If the value reaches (2^29) - 1, it wraps back to 0.

-x coordinate: a 10 bit unsigned integer with big endian byte order.

-y coordinate: a 10 bit unsigned integer with big endian byte order.

-S: the screen that our pacman is on, a 1 bit unsigned integer.

-E: the type of things our pacman eats, a 3 bit unsigned integer.

Type 6: Status update

Protocol used: TCP

Content:

The mode of LOCAL game, an unsigned integer in the range 0-5:

0: STARTUP. The two computers are first connected, or LOCAL game is in READY_TO_RESTART mode (below) and has received a message of this type, showing that the other game is also in READY_TO_RESTART mode. When the game is in this mode, a message of type 1: Maze update SHOULD be sent, and the jingle is played. At the beginning, each pacman has 5 lives. After playing the jingle, receiving a type 1 message and the Acknowledge Character of our type 1 message from the other computer, move to CHASE mode.

1: CHASE. The game is in process, and the ghosts are in chase mode.

2: FRIGHTEN. The game is in process. A powerpill is eaten, and the ghosts are in frighten mode.

3: GAME_OVER. Our pacman has 0 lives. The receiver MUST also move to GAME_OVER mode.

4: NEXT_LEVEL_WAIT. The level on the LOCAL screen is cleaned. A jingle is played, and any FOREIGN pacman is forcibly sent back, so its position is reset. When the game is in this mode, a message of type 1: Maze update SHOULD be sent to inform the other computer the maze for new level.

5: READY_TO_RESTART. LOCAL GAME_OVER mode game gets an “r” press from the player.

Timing:
Status update messages SHOULD be sent once when the mode of LOCAL game changes.

Package:
Each message of this type SHOULD be sent in one package.

Status update messages consist of 5 bytes, encode as follows:

```
0                   1                   2                 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| T |                       sequence number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| M |    U      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

-T: 3 bit type field. Type Status update has value 5.

-sequence number: 29 bit unsigned integer with big endian byte order. Its initial value is 0. For each new message sent, the value is increased by 1. If the value reaches (2^29) - 1, it wraps back to 0.

-M: the mode of LOCAL game, a 3 bit unsigned integer.

-U: 5 bit unused to maintain byte alignment. MUST be set to 0. The receiver MUST ignore these bits.

Validity

If any fields in any messages received have a value that is not in the range clarified in the content part of that type of message, that package MUST be discarded to avoid potential malicious attack.

Sequence Number

UDP is used for two types of messages: Pacman update and Ghost update. These data may be lost, or they may arrive in the wrong order. To avoid updating using the outdated messages, the receiver SHOULD keep track of the greatest sequence number received.

If a UDP message with lower sequence number than the record is received, it MUST be discarded. The data loss is acceptable since the UDP package SHOULD be sent once per frame. Losing one intermediate position and speed is not serious.

After the whole TCP package is received, the sequence number in that message is compared with the record. If the sequence number is lower, the message MUST be handled normally. Each TCP package contains important information which cannot be lost. TCP messages may be late due to latency, but little lag in actions like eating or being killed is not very serious. They do not destroy the playability of the game.

When comparing the sequence numbers, pay attention to the wrap. Sequence numbers like 0 - 100 SHOULD be considered greater than (2^29) - 1.