

**Student Number: 23163657**

=====  
**Pacman Protocol Specification**  
=====

**Terminology**  
=====

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

This specification also uses the terms LOCAL, AWAY, REMOTE and FOREIGN as defined in assignments/assignment5/assignment.pdf on GitHub.

This specification also defines additional terms to state the computer the software instance is running on:

- LOCAL\_GAME : The LOCAL game instance that is running on the local host.
- REMOTE\_GAME : The REMOTE game instance that is connected to the LOCAL game instance.

This specification defines the term CONNECTION as the connection between either:

- The server and client if the Client-Server Mode (defined in assignments/assignment5/assignment.pdf on GitHub) is being used.
- The two peers connected via the relay server if the Two clients, one relay server (defined in assignments/assignment5/assignment.pdf on GitHub) is being used.

All ranges specified in this specification are inclusive of the boundaries and are the only valid integers. (Example: range from 0 to 5 = 0, 1, 2, 3, 4, 5)

In this specification, all numbers that are used for field values are represented as decimal value integer equivalents of the binary values to make it simpler for the implementer to understand.

This specification assumes that for computer one, computer one is the LOCAL\_GAME and computer two is the REMOTE\_GAME and for computer two, computer two is the LOCAL\_GAME and computer one is the REMOTE\_GAME. All messages in this specification are sent from a LOCAL\_GAME to the REMOTE\_GAME or vice versa so each message is sent by each computer to each other except the SYNC message type, that is only sent by one of the two computers.

This specification assumes the 650 pixels x 800 pixels canvas to be divided into 28 columns and 31 rows. Thus, all PosX or PosY fields are x-coordinates and y-coordinates respectively, of a 28 x 31 grid. The x-coordinates range from 0 to 27 and the y-coordinates range from 0 to 30.

The Pacman Protocol runs over UDP and TCP, using a well known port of 5432.

### **Message Encoding**

=====

This specification uses binary encoding to transmit data over the network. Binary encoding uses a compact representation of data which results in smaller sized data being transmitted, this makes the transmission faster than using textual encoding. Using binary encoding is more efficient than using textual encoding as binary data is machine-readable whereas textual data is human-readable so it will have to be converted to binary data before the computer reads it.

This specification uses the big-endian byte order for all the messages as most modern networks specify big-endian as the standard byte order. An additional length field is not required as the message type is fixed to the first byte being sent over the network first, then the second byte, then the third byte... .

### **ACKNOWLEDGEMENT**

=====

An ACKNOWLEDGEMENT message needs to be sent to ensure that a message of specific message types have been received by the REMOTE\_GAME, as this MUST happen for the game to function smoothly. If an ACKNOWLEDGEMENT message is not received by the LOCAL\_GAME within the span of 100ms, it assumes that the packet sent to the REMOTE\_GAME, that contains the message, has been lost and will resend the packet. If the LOCAL\_GAME receives an ACKNOWLEDGEMENT message for its message then it will take no actions and continue with what it was doing.

### **Contents**

-----

Type : ACKNOWLEDGEMENT

Stype : This field specifies the type of the message for which the ACKNOWLEDGEMENT message is being sent. It contains an integer



Obj : This field specifies the type of the object at the location. It contains an integer value in the range 0 to 9, each corresponding to a specific type of object:

- ```
- 0 : Horizontal Wall ( ' - ' )
- 1 : Vertical Wall ( ' | ' )
- 2 : Left to Right Wall Corner ( ' / ' )
- 3 : Right to Left Wall Corner ( ' \ ' )
- 4 : Empty space
- 5 : Food ( ' . ' )
- 6 : Power-pill ( ' * ' )
- 7 : Tunnel A
- 8 : Tunnel B
- 9 : Inaccessible Area ( ' # ' )
```

Row : This field specifies the row where the object is positioned. It contains an integer value in the range 0 to 30, where 0 indicates the first row at the top of the LOCAL\_GAME and 30 indicates the last row at the bottom of the LOCAL\_GAME.

Column : This field specifies the column where the object is positioned. It contains an integer value in the range 0 to 27, where 0 indicates the leftmost column of the LOCAL\_GAME and 27 indicates the rightmost column of the LOCAL\_GAME.

## Timing

-----

These messages MUST be first sent at the beginning of the CONNECTION. Any other messages MUST wait until all these messages have been received by the REMOTE GAME.

These messages **MUST** be sent again if the maze needs to be updated  
(Example: before a new level begins)

## Format

-----

MAZE SETUP messages consist of three bytes, encoded as follows:

[illegible]



-----

[illegible]

Sequence number : a 12 bit unsigned integer, incremented by one for every new message sent. If it reaches  $(2^{12})-1$ , it wraps back round to 0.

TA : 3 bit unsigned integer with decimal value 5, in big-endian byte order.

## PACMAN UPDATE

This message is sent across the network using UDP and there is no need for an ACKNOWLEDGEMENT message as the message is sent very frequently so in case a packet is lost, the next packet will reach the REMOTE\_GAME soon until which it can use the previous message to animate the Pacman. This can be done using the PosX, PosY and Dir fields of the previous message.

-----

```
Type : PACMAN_UPDATE
```

PosX : This field specifies the x-coordinate of the position of the LOCAL\_GAME's Pacman on the LOCAL\_GAME's maze or REMOTE\_GAME's maze. It contains an integer value in the range 0 to 27, where 0 indicates the leftmost column of the LOCAL\_GAME or REMOTE\_GAME and 27 indicates the rightmost column of the LOCAL\_GAME or REMOTE\_GAME.

PosY : This field specifies the y-coordinate of the position of the LOCAL\_GAME's Pacman on the LOCAL\_GAME's maze or REMOTE\_GAME's maze. It contains an integer value in the range 0 to 30, where 0 indicates the first row at the top of the LOCAL\_GAME or REMOTE\_GAME and 30 indicates the last row at the bottom of the LOCAL\_GAME or REMOTE\_GAME.

Dir : This field specifies the direction of the LOCAL\_GAME's Pacman towards which it is facing. It contains an integer value in the range 0 to 3, each corresponding to a direction:

- 0 : North
- 1 : East
- 2 : South
- 3 : West

L : This field specifies if the Pacman is either LOCAL or AWAY. It helps the LOCAL\_GAME and REMOTE\_GAME to interpret where to display the Pacman. It contains an integer value in the range 0 to 1, where 0 indicates the Pacman is LOCAL and 1 indicates the Pacman is AWAY for the LOCAL GAME (i.e. FOREIGN for the REMOTE GAME).

## Timing

\_\_\_\_\_

While a LOCAL Pacman is moving, the PACMAN\_UPDATE message SHOULD be sent every 20ms (Average response time for human sight). If the LOCAL\_GAME's computer is not capable of maintaining 50 frames per second, PACMAN\_UPDATE message MAY be sent once per frame, as it is unreasonable to update the REMOTE\_GAME's computer more often than the LOCAL\_GAME's computer.

### Format

\_\_\_\_\_

PACMAN UPDATE consists of 6 bytes, encoded 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 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| +    | - | + | - | + | - | + | - | + | - | +               | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - |
| Type |   |   |   |   |   |   |   |   |   | Sequence number |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| +    | - | + | - | + | - | + | - | + | - | +               | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - |

```

+--+
|  PosX  |  PosY  | Dir |L| U |
+--+

```

Type: 4 bit type field. Type=PACMAN\_UPDATE has decimal value 3.

Sequence number : a 28 bit unsigned integer, incremented by one for every new message sent. If it reaches  $(2^{28})-1$ , it wraps back round to 0.

PosX : 5 bits, giving an unsigned integer in big-endian byte order.

PosY : 5 bits, giving an unsigned integer in big-endian byte order.

Dir : 2 bits, giving an unsigned integer in big-endian byte order.

L : 1 bit, giving an unsigned integer in big-endian byte order.

U (Unused) : 3 bits, not used, but needed to maintain byte alignment. MUST be set to zero.

## **PACMAN\_ARRIVED**

=====

This message notifies the REMOTE\_GAME about the LOCAL\_GAME's Pacman becoming FOREIGN to the REMOTE\_GAME so that the REMOTE\_GAME knows that the FOREIGN Pacman is now going to interact with the REMOTE\_GAME's objects (i.e. Ghosts, Food and Power-pills) and not with the LOCAL\_GAME's objects.

This message MUST reach the REMOTE\_GAME or else it will not know that the LOCAL\_GAME's Pacman has arrived and will not let the LOCAL\_GAME's Pacman interact with its objects. This will result in the LOCAL\_GAME's Pacman passing through Ghosts, Food and Power-pills without affecting the game. So, this message MUST be sent across the network using UDP and an ACKNOWLEDGEMENT message should be sent by the REMOTE\_GAME back to the LOCAL\_GAME.

## **Contents**

-----

Type : PACMAN\_ARRIVED

## **Timing**

-----



This message should be sent when a Pacman becomes AWAY to the LOCAL\_GAME (i.e. FOREIGN to the REMOTE\_GAME). This message MUST be sent during these two cases:

- When the Pacman exits LOCAL\_GAME via Tunnel A, hence it enters the REMOTE\_GAME through Tunnel B. This is when the PACMAN\_UPDATE message fields has these values:
  - PosX : In the range 25 to 30
  - PosY : 14
  - Dir : 3
  - L : 1
- When the Pacman exits LOCAL\_GAME via Tunnel B, hence it enters the REMOTE\_GAME through Tunnel A. This is when the PACMAN\_UPDATE message fields has these values:
  - PosX : In the range 0 to 5
  - PosY : 14
  - Dir : 1
  - L : 1

When a Pacman enters the REMOTE\_GAME's maze, the message is being sent 5 times to ensure the REMOTE\_GAME receives this message as soon as possible to be to take action upon it, as if it was just sent once and the packet is lost, this will cause a delay and this can not be risked for this message.

## Format

\_\_\_\_\_

PACMAN ARRIVED consists of 2 bytes, encoded as follows:

[illegible]

Type: 4 bit type field. Type=PACMAN ARRIVED has decimal value 4.

Sequence Number : a 12 bit unsigned integer, incremented by one for every new message sent. If it reaches  $(2^{12})-1$ , it wraps back round to 0.

PACMAN LEFT

=====

This message notifies the REMOTE\_GAME about the LOCAL\_GAME's Pacman becoming LOCAL, so that the REMOTE\_GAME knows that the Pacman will interact with the LOCAL\_GAME's objects and not with the REMOTE\_GAME's objects.

This message MUST reach the REMOTE\_GAME or else it will not know that the LOCAL\_GAME's Pacman has left and will result in the Pacman interacting with REMOTE\_GAME's objects that are on the corresponding positions of the Pacman on the LOCAL\_GAME. So, this message MUST be sent across the network using UDP and an ACKNOWLEDGEMENT message should be sent by the REMOTE\_GAME back to the LOCAL\_GAME.

## Contents

-----

Type : PACMAN LEFT

## Timing

\_\_\_\_\_

This message MUST be sent when a Pacman becomes LOCAL to the LOCAL GAME. This message MUST be sent during these two cases:

- When the Pacman exits REMOTE\_GAME via Tunnel A, hence it enters the LOCAL\_GAME through Tunnel B. This is when the PACMAN\_UPDATE message fields has these values:
  - PosX : In the range 25 to 30
  - PosY : 14
  - Dir : 3
  - L : 0
- When the Pacman exits REMOTE\_GAME via Tunnel B, hence it enters the LOCAL\_GAME through Tunnel A. This is when the PACMAN\_UPDATE message fields has these values:
  - PosX : In the range 0 to 5
  - PosY : 14
  - Dir : 1
  - L : 0

When a Pacman exits the REMOTE\_GAME's maze, the message is being sent 5 times to ensure the REMOTE\_GAME receives this message as soon as possible to be to take action upon it, as if it was just sent once and the packet is lost, this will cause a delay and this can not be risked for this message.

### Format

\_\_\_\_\_

PACMAN LEFT consists of 2 bytes, encoded as follows:

[illegible]

Type: 4 bit type field. Type=PACMAN LEFT has decimal value 5.

Sequence Number : a 12 bit unsigned integer, incremented by one for every new message sent. If it reaches  $(2^{12})-1$ , it wraps back round to 0.

## **PACMAN\_DIED**

=====

This message MUST be sent to the REMOTE\_GAME so that it knows that the LOCAL\_GAME's Pacman has died and updates the REMOTE\_GAME's screen for either the LOCAL\_GAME's maze clone or the REMOTE\_GAME's maze.

This message MUST reach the REMOTE\_GAME or else it will not know that the LOCAL\_GAME's Pacman has died and will not update its screen. So, this message MUST be sent across the network using UDP and an ACKNOWLEDGEMENT message should be sent by the REMOTE\_GAME back to the LOCAL\_GAME.

## **Contents**

-----

Type : PACMAN\_DIED

## **Timing**

-----

This message MUST be sent every time the LOCAL\_GAME's Pacman dies. For the Pacman to die, the current STATUS\_UPDATE status MUST be CHASE and the Pacman MUST interact with a LOCAL\_GAME's Ghost on its maze or a REMOTE\_GAME's Ghost on its maze.

## **Format**

-----

PACMAN\_DIED consists of 1 byte, encoded 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 |  |  |  |  |  |
| + | +    | + | + | + | + | +  | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + |  |  |  |  |  |
|   | Type |   |   |   |   | SN |   |   |   |   | U |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |
| + | +    | + | + | + | + | +  | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + | + |  |  |  |  |  |

Type: 4 bit type field. Type=PACMAN\_DIED has decimal value 6.

SN (Sequence Number) : a 3 bit unsigned integer, incremented by one for every new message sent.

U (Unused) : 1 bits, not used, but needed to maintain byte alignment. MUST be set to zero.

## **PACMAN\_GO\_HOME**



frequently so in case a packet is lost, the next packet will reach the REMOTE\_GAME soon until which it can use the previous message to animate the Ghosts. This can be done using the PosX, PosY and Dir fields of the previous message.

## **Contents**

-----

Type : GHOST\_UPDATE

N : This field specifies which ghost the message is allocated for. It contains an integer value in the range 0 to 3, where each integer is allocated to the 4 different Ghosts.

PosX : This field specifies the x-coordinate of the position of the LOCAL\_GAME's Ghosts on the LOCAL\_GAME's maze. It contains an integer value in the range 0 to 27, where 0 indicates the leftmost column of the LOCAL\_GAME and 27 indicates the rightmost column of the LOCAL\_GAME.

PosY : This field specifies the y-coordinate of the position of the LOCAL\_GAME's Ghosts on the LOCAL\_GAME's maze. It contains an integer value in the range 0 to 30, where 0 indicates the first row at the top of the LOCAL\_GAME and 30 indicates the last row at the bottom of the LOCAL\_GAME.

Dir : This field specifies the direction of the LOCAL\_GAME's Pacman towards which it is facing. It contains an integer value in the range 0 to 3, each corresponding to a direction:

- 0 : North
- 1 : East
- 2 : South
- 3 : West

## **Timing**

-----

While a LOCAL Ghosts is moving, the GHOST\_UPDATE message SHOULD be sent every 20ms (Average response time for human sight). Four of GHOST\_UPDATE messages MUST be sent every time, one message for each Ghost. If the LOCAL\_GAME's computer is not capable of maintaining 50 frames per second, GHOST\_UPDATE message MAY be sent once per frame, as it is unreasonable to update the REMOTE\_GAME's computer more often than the LOCAL\_GAME's computer.

## **Format**

-----

GHOST\_UPDATE consists of 6 bytes, encoded 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
+--+
| Type |                               Sequence number |
+--+
+--+
| N |   PosX   |   PosY   | Dir |U|
+--+

```

Type: 4 bit type field. Type=GHOST\_UPDATE has decimal value 8.

Sequence number : a 28 bit unsigned integer, incremented by one for every new message sent. If it reaches  $(2^{28})-1$ , it wraps back round to 0.

N : 2 bits, giving an unsigned integer in big-endian byte order.

PosX : 5 bits, giving an unsigned integer in big-endian byte order.

PosY : 5 bits, giving an unsigned integer in big-endian byte order.

Dir : 2 bits, giving an unsigned integer in big-endian byte order.

U (Unused) : 2 bits, not used, but needed to maintain byte alignment. MUST be set to zero.

## EAT

=====

This message needs to be sent to the REMOTE\_GAME so that it can update its screen for either the LOCAL\_GAME's maze clone when the LOCAL\_GAME's Pacman eats a LOCAL Food, Power-pill or Ghost, or the REMOTE\_GAME's maze when the AWAY Pacman eats a REMOTE Food or Power-pill.

This message MUST reach the REMOTE\_GAME or else it will not know that the AWAY Pacman is eating objects on the REMOTE\_GAME's maze or it will not be able to update the LOCAL\_GAME's maze clone. So, this message MUST be sent across the network using UDP and an ACKNOWLEDGEMENT message should be sent by the REMOTE\_GAME back to the LOCAL\_GAME.

## Contents

-----

Type : EAT

ETy : This field specifies what the LOCAL\_GAME's Pacman is eating and whether it is a LOCAL or AWAY object. It contains an integer

value in the range 0 to 4, each corresponding to a specific type of object and location:

- 0 : LOCAL Food
- 1 : LOCAL Power-pill
- 2 : LOCAL Ghosts
- 3 : AWAY Food
- 4 : AWAY Power-pill

PosX : This field specifies the x-coordinate of the position of the Food, Power-pill or Ghost that has been eaten by the LOCAL\_GAME's Pacman. It contains an integer value in the range 0 to 27, where 0 indicates the leftmost column of the LOCAL\_GAME and 27 indicates the rightmost column of the LOCAL\_GAME.

PosY : This field specifies the y-coordinate of the position of the Food, Power-pill or Ghost that has been eaten by the LOCAL\_GAME's Pacman. It contains an integer value in the range 0 to 30, where 0 indicates the first row at the top of the LOCAL\_GAME and 30 indicates the last row at the bottom of the LOCAL\_GAME.

### Timing

-----

These messages MUST be sent every time the LOCAL\_GAME's Pacman interacts with a non-empty cell on either the LOCAL\_GAME's maze or the AWAY\_GAME's maze. For a message to be sent where the decimal integer value of the E\_Type field is 2, the current STATUS\_UPDATE status MUST be FRIGHTEN.

### Format

-----

EAT consists of 6 bytes, encoded 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 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Type |   |   |   |   |   |   |   |   |   | Sequence Number |   |   |   |   |      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| ETy  |   |   |   |   |   |   |   |   |   | PosX            |   |   |   |   | PosY |   |   |   |   | U |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

Type : 4 bit type field. Type=EAT has decimal value 9.

Sequence Number : a 28 bit unsigned integer, incremented by one for every new message sent. If it reaches  $(2^{28})-1$ , it wraps back round to 0.

ETy : 3 bits, giving an unsigned integer in big-endian byte order.

PosY : 5 bits, giving an unsigned integer in big-endian byte order.

## FOREIGN PACMAN ATE GHOST

This message MUST be sent to the REMOTE\_GAME so that it knows that its Pacman ate a Ghost on the LOCAL\_GAME's maze so that it can update its score,

## Contents

```
Type : FOREIGN_PACMAN_ATE_GHOST
```

## Timing

This message MUST be sent every time when a FOREIGN Pacman interacts with a Ghost on the LOCAL GAME's maze.

FOREIGN PACMAN ATE GHOST consists of 2 bytes, encoded as follows:

Type: 4 bit type field. Type=PACMAN GO HOME has decimal value 10.



Sequence Number : a 10 bit unsigned integer, incremented by one for every new message sent.

=====

This message is sent across the network using UDP and there is no need of an ACKNOWLEDGEMENT message as the message is sent very frequently so in case a packet is lost, the next packet will reach the REMOTE GAME soon and the score will eventually be updated.

-----

Score : This field is the LOCAL\_GAME's score. It contains an integer value in the range 0 to 16,777,215.

\_\_\_\_\_

\_\_\_\_\_

[illegible]

Type : 4 bit type field. Type=SCORE UPDATE has decimal value 11.

Sequence Number : a 28 bit unsigned integer, incremented by one for every new message sent. If it reaches  $(2^{28})-1$ , it wraps back round to 0.

Score : 24 bits, giving an unsigned integer in big-endian byte order.

## **LIVES\_UPDATES**

=====

This messages MUST be sent to the REMOTE\_GAME so that it can keep a count of the LOCAL\_GAME's lives so that it know when to end the game if the LOCAL\_GAME runs out of lives and to display the LOCAL\_GAME's lives on its screen.

This message MUST reach the REMOTE\_GAME or else it will not know that the game is over if the LOCAL\_GAME runs out of lives. So, this message MUST be sent across the network using UDP and an ACKNOWLEDGEMENT message should be sent by the REMOTE\_GAME back to the LOCAL\_GAME.

## **Contents**

-----

Type : LIVES\_UPDATES

## **Timing**

-----

This message MUST be sent to the REMOTE\_GAME every time the LOCAL\_GAME's Pacman dies.

## **Format**

-----

LIVES\_UPDATES consists of 1 byte, encoded 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 |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| + | -    | + | - | + | - | +  | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|   | Type |   |   |   |   | SN |   |   |   |   | U |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| + | -    | + | - | + | - | +  | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - | + | - |  |   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Type: 4 bit type field. Type=LIVES\_UPDATE has decimal value 12.

SN (Sequence Number) : a 3 bit unsigned integer, incremented by one for every new message sent.

U (Unused) : 1 bit, not used, but needed to maintain byte alignment. MUST be set to zero.

## **STATUS\_UPDATE**



SN (Sequence Number) : a 9 bit unsigned integer, incremented by one for every new message sent.

### **Sequence Numbers**

=====

Due to the use of UDP, messages may be lost or arrive out of order so, the

REMOTE\_GAME keeps track of the sequence number of the last message received of each type. If it receives a message of a specific

type with a lower sequence number than the last one received, the message MUST be discarded. When performing this comparison, care must be taken to account for the potential for sequence numbers to Wrap. If it receives a message with a sequence number more than once, then it should discard the message but also send an ACKNOWLEDGEMENT message again as it is possible the ACKNOWLEDGEMENT message it sent before did not reach the LOCAL\_GAME due to packet loss.

### **Message Range Check**

=====

For all message types with field values that have a range of integers, any value outside the range should be discarded as it is probably due malicious activity.