

## Pacman Protocol Specification

Student Number: 20001402

### 1. Terminology

The Pacman protocol runs over TCP, using a port of 9872. There are 12 message types:

- 1.1. RECEIVED\_MAZE
- 1.2. FOREIGN\_PACMAN\_ARRIVED
- 1.3. FOREIGN\_PACMAN\_LEFT
- 1.4. FOREIGN\_PACMAN\_DIED
- 1.5. PACMAN\_GO\_HOME
- 1.6. PACMAN\_UPDATE
- 1.7. GHOST\_UPDATE
- 1.8. FOREIGN\_GHOST\_ATE\_GHOST
- 1.9. EAT
- 1.10. LIVES\_UPDATE
- 1.11. LIVES\_UPDATE
- 1.12. STATUS\_UPDATE

Apart from that, there are four capitalized terms with specific meanings.

- 1.1. LOCAL: a game object which is currently on the local screen
- 1.2. REMOTE: a game object which is currently on the remote screen
- 1.3. AWAY: our pacman is on the remote screen now
- 1.4. FOREIGN: the other pacman is on the local screen now

There are four situations about the two pacmans in the view of our pacman and local screen:

When the game starts or restarts, the local and remote computer sends each other a MAZE\_UPDATE message. On each screen, the left side displays LOCAL situation and right side displays REMOTE situation.

When the score or lives of our pacman changes, if it is LOCAL, the local model sends a SCORE\_UPDATE to the remote model. If it is AWAY, the remote model sends a SCORE\_UPDATE to the local model.

When the score or lives of the other pacman changes, if it is REMOTE, the remote model sends a SCORE\_UPDATE to the local model. If it is FOREIGN, the local model sends a SCORE\_UPDATE to the remote model.

For the anytime of the four situations, each computer continuously sends GHOST\_UPDATE about LOCAL to the other computer.

#### 1.1. our pacman is LOCAL and the other is REMOTE

- 1.1.1. When our pacman died, local model sends FOREIGN\_PACMAN\_DIED to the remote model.

- 1.1.2. When our pacman moves, local model sends PACMAN\_UPDATE to the remote model.
- 1.1.3. When our pacman eats ghosts, local model sends FOREIGN\_PACMAN\_ATE\_GHOST to the remote model.
- 1.1.4. When our pacman eats food or powerpills, local model sends EAT to the remote model.

Hint that when our pacman is LOCAL, there are three different possibilities.

The first one is that our pacman is initially LOCAL.

The second one is that the our pacman becomes AWAY that local model sends a FOREIGN\_PACMAN\_ARRIVED and STATUS\_UPDATE to the remote model, and it comes back to become LOCAL. AS a result local model sends FOREIGN\_PACMAN\_LEFT and STATUS\_UPDATE to the remote model.

The third one is that our pacman becomes AWAY that local model sends a FOREIGN\_PACMAN\_ARRIVED and STATUS\_UPDATE to the remote model. However the remote model sends PACMAN\_GO\_HOME, so our pacman comes back and becomes LOCAL. The local then sends FOREIGN\_PACMAN\_LEFT and STATUS\_UPDATE to the remote model.

Hint that when the other pacman experiences the same thing as our pacman, the remote model would send the corresponding message to the local model.

## 1.2. our pacman is AWAY and the other is REMOTE

- 1.2.1. When either pacman died, remote model sends FOREIGN\_PACMAN\_DIED to the local model.
- 1.2.2. When either pacman moves, remote model sends PACMAN\_UPDATE to the local model.
- 1.2.3. When either pacman eats ghosts, remote model sends FOREIGN\_PACMAN\_ATE\_GHOST to the local model.
- 1.2.4. When either pacman eats food or powerpills, remote model sends EAT to the local model.

Hint that when our pacman becomes AWAY after it traverses the tunnel. Local model sends FOREIGN-PACMAN\_ARRIVED to the remote model as it goes to the remote screen.

## 1.3. our pacman is AWAY and the other is FOREIGN

- 1.3.1. When our pacman died, remote model sends FOREIGN\_PACMAN\_DIED to the local model.
- 1.3.2. When our pacman moves, remote model sends PACMAN\_UPDATE to the local model.
- 1.3.3. When our pacman eats ghosts, remote model sends FOREIGN\_PACMAN\_ATE\_GHOST to the local model.
- 1.3.4. When our pacman eats food or powerpills, remote model sends EAT to the local model.

Hint that when our pacman becomes AWAY after it traverses the tunnel. Local model sends FOREIGN-PACMAN\_ARRIVED to the remote model as it goes to the remote screen. The other pacman is FOREIGN so as it goes to the local screen, the remote model sends FOREIGN-PACMAN\_ARRIVED to the local model.

Hint that when the other pacman experiences the same thing as our pacman, the local model would send the corresponding message to the remote model.

#### 1.4. our pacman is LOCAL and the other is FOREIGN

1.4.1. When either pacman died, local model sends FOREIGN\_PACMAN\_DIED to the remote model.

1.4.2. When either pacman moves, local model sends PACMAN\_UPDATE to the remote model.

1.4.3. When either pacman eats ghosts, local model sends FOREIGN\_PACMAN\_ATE\_GHOST to the remote model.

1.4.4. When either pacman eats food or powerpills, local model sends EAT to the remote model.

Hint that when the other pacman becomes FOREIGN after it traverses the tunnel. Remote model sends FOREIGN-PACMAN\_ARRIVED to the local model as it goes to the local screen.

#### 1.5. There are six different statuses of the game mode. As the status changes the STATUS\_UPDATE message is used

1.1. STARTUP

1.2. CHASE

1.3. FRIGHTEEN

1.4. GAME\_OVER

1.5. NEXT\_LEVEL\_WAIT

1.6. READY\_TO\_RESTART

Either model is in CHASE or FRIGHTEEN states it becomes SRARTUP state and sends STATUS\_UPDATE to the other model, as the other model becomes STARTUP the game starts.

As either player loses all the lives, the model which the player's pacman visits becomes GAME\_OVER and sends STATUS\_UPDATE to the other. The other model then becomes GAME\_OVER state.

As both models are GAME\_OVER state, either player presses 'r' to restart, the model which the player controlled becomes READY\_TO\_RESTART state and sends STATUS\_UPDATE to the other model. As the other player presses 'r' the other model becomes READY\_TO\_RESTART.

When the level of either model is finished, the model becomes NEXT\_LEVEL\_WAIT and sends STATUS\_UPDATE to the other model. It does not affect the level of the other model.

## 2. Message Contents

### 2.1 RECEIVED\_MAZE

2.1.1. maze: the shape of each maze.

### 2.2. PACMAN\_UPDATE

The contents of a PACMAN\_UPDATE are:

2.2.1. pos: The X and Y position of the pacman. X is distance along the screen from 0 to 1023, 0 is the left end of either player's screen and 1023 is the right. Y is distance down the screen from 0 to 1023, 0 is the top end of either player's screen and 1023 is the bottom.

2.2.2. dir: five different directions of the pacman which are UP, LEFT, RIGHT, DOWN, NONE

2.2.3. speed: the speed of the pacman which is either 0 or 1 units per frame.

### 2.3. GHOST\_UPDATE

2.3.1. ghostnum: how many ghosts are on the screen now. It would be a number in the range of 0 to 4.

2.3.2. pos: The X and Y position of the ghost. X is distance along the screen from 0 to 1023, 0 is the left end of either player's screen and 1023 is the right. Y is distance down the screen from 0 to 1023, 0 is the top end of either player's screen and 1023 is the bottom.

2.3.3. dirn: five different directions of the ghost which are UP, LEFT, RIGHT, DOWN, NONE

2.3.4. speed: the speed of the ghost which is either 0 or 1 units per frame.

2.3.5. mode: three different modes for each ghost that each mode represents a different picture of ghost. Apart from that there are three modes representing ghost die, ghost scared and ghost scared ending.

### 2.4. FOREIGN\_PACMAN\_ATE\_GHOST

2.4.1 ghostnum: if the number of ghosts on the screen decreases 1, a ghost is killed by the pacman which means the ghost in FRIGHTEN state touches the pacman.

### 2.5. EAT

2.5.1. pos: The X and Y position of the food. X is distance along the screen from 0 to 1023, 0 is the left end of either player's screen and 1023 is the right. Y is distance down the screen from 0 to 1023, 0 is the top end of either player's screen and 1023 is the bottom

2.5.2. is\_foreign: the position of the food is on the local screen or remote screen. If it is on the local screen after it is eaten the number of food of local screen minus 1. If it is on the remote screen after it is eaten the number of food of remote screen minus 1.

2.5.3 is\_powerpill: whether the food is powerpill. If the food is not powerpill the number of food minus 1. If the food is powerpill the number of food minus 1 and ghost changes to ghost scared state.

### 2.6. SCORE\_UPDATE

2.6.1. score: when the pacman eats food, powerpill or ghost, the score of the pacman shown on the screen increases.

## 2.7. LIVES\_UPDATE

2.7.1. lives: when the pacman died the lives minus 1.

## 2.8. STATUS\_UPDATE

2.8.1 status: the status of local model it presents now.

## 2.9. FOREIGN\_PACMAN\_DIED

2.9.1 self: when the pacman touches the ghost which is in CHASE state, the status of the pacman becomes died.

## 2.10. FOREIGN\_PACMAN\_ARRIVED

2.10.1. self: the status of the pacman becomes foreign.

## 2.11. FOREIGN\_PACMAN\_LEFT

2.11.1. self: the status of the pacman becomes remote.

## 2.12. PACMAN\_GO\_HOME

2.12.1. self: the status of the pacman becomes local.

## 3. Message Encode

Messages are fixed format, binary encoded, with all integer fields send in network byte order (i.e, big endian order).

### 3.1. MAZE\_UPDATE

```
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 |                unused                |
+++++
|u2|                maze0                |
+++++
|u3|                maze1                |
+++++
|u4|                maze2                |
+++++
```

3.1.1. T: 4 bits type field Type = MAZE\_UPDATE has decimal value 1

3.1.2. unused: 28 bits, not used, but needed to maintain byte alignment. Must be set to zero

in this version of the protocol

3.1.3. u2 = unused2: 2 bits, not used, but needed to maintain byte alignment.

3.1.4. maze0: 30 bits, giving an unsigned integer in big-endian byte order

3.1.5. u3 = unused3: 2 bits, not used, but needed to maintain byte alignment.

3.1.6. maze1: 30 bits, giving an unsigned integer in big-endian byte order

3.1.7. u4 = unused4: 2 bits, not used, but needed to maintain byte order.

3.1.8. maze2: 30 bits, giving an unsigned integer in big-endian byte order

### 3.2. PACMAN\_UPDATE

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 | unused | x pos | y pos | dir | s |

+++++

3.2.1. T: 4 bits type field Type = PACMAN\_UPDATE has decimal value 2

3.2.2. unused: 4 bits, not used, but needed to maintain byte alignment. Must be set to zero in this version of the protocol

3.2.3. x pos: 10 bits, giving an unsigned integer in big-endian byte order

3.2.4. y pos: 10 bits, giving an unsigned integer in big-endian byte order

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

3.2.6. s = speed: 1bit, giving an unsigned integer in big-endian byte order

### 3.3. GHOST\_UPDATE

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 | u | g | x pos | y pos | dir | s |

+++++

| mode | unused2 |

+++++

3.3.1. T: 4 bits type field Type = GHOST\_UPDATE has decimal value 3

3.3.2. u = unused: 2 bits, not used, but needed to maintain byte alignment. Must be set to zero in this version of the protocol

3.3.3. g = ghostnum: 2 bits, giving an unsigned integer in big-endian byte order

3.3.4. x pos: 10 bits, giving an unsigned integer in big-endian byte order

3.3.5. y pos: 10 bits, giving an unsigned integer in big-endian byte order

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

3.3.7. s = speed: 1 bit, giving an unsigned integer in big-endian byte order

3.3.8. mode: 4 bits, giving an unsigned integer in big-endian byte order

3.3.9. unused2: 28 bits, not used, but needed to maintain byte alignment.

### 3.4. FOREIGN\_PACMAN\_ATE\_GHOST

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 | unused | g |

+++++

3.4.1. T: 4 bits type field Type = FOREIGN\_PACMAN\_ATE\_GHOST has decimal value 4

3.4.2. unused: 26 bits, not used, but needed to maintain byte alignment. Must be set to zero in this version of the protocol

3.4.3. g = ghostnum: 2 bits, giving an unsigned integer in big-endian byte order

### 3.5. EAT

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 | unused | x pos | y pos | f | p |

+++++

3.5.1. T: 4 bits type field Type = EAT has decimal value 5

3.5.2. unused: 6 bits, not used, but needed to maintain byte alignment. Must be set to zero in this version of the protocol

3.5.3. x pos: 10 bits, giving an unsigned integer in big-endian byte order

3.5.4. y pos: 10 bits, giving an unsigned integer in big-endian byte order

3.5.5. is\_foreign: 1 bit, giving an unsigned integer in big-endian byte order

3.5.6. is\_powerpill: 1 bit, giving an unsigned integer in big-endian byte order

### 3.6. SCORE\_UPDATE

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 | unused | score |

+++++

3.6.1. T: 4 bits type field Type = SCORE\_UPDATE has decimal value 6

3.6.2. unused: 14 bits, not used, but needed to maintain byte alignment. Must be set to zero in this version of the protocol

3.6.3. score: 14 bits, giving an unsigned integer in big-endian byte order

### 3.7. LIVES\_UPDATE

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 | unused | lives |

+++++

3.6.1. T: 4 bits type field Type = LIVES\_UPDATE has decimal value 7

3.6.2. unused: 25 bits, not used, but needed to maintain byte alignment. Must be set to zero in this version of the protocol

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

### 3.8. STATUS\_UPDATE

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 | unused | s |

+++++

- 3.8.1. T: 4 bits type field Type = STATUS\_UPDATE has decimal value 8
- 3.8.2. unused: 25 bits, not used, but needed to maintain byte alignment. Must be set to zero in this version of the protocol
- 3.8.3. s = status: 3 bits, giving an unsigned integer in big-endian byte order

## 2.9. FOREIGN\_PACMAN\_DIED

```

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 |               unused               | s |
+++++

```

- 3.8.1. T: 4 bits type field Type = FOREIGN\_PACMAN\_DIED has decimal value 9
- 3.8.2. unused: 25 bits, not used, but needed to maintain byte alignment. Must be set to zero in this version of the protocol
- 3.8.3. s = self: 3 bits, giving an unsigned integer in big-endian byte order

## 2.10. FOREIGN\_PACMAN\_ARRIVED

```

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 |               unused               | s |
+++++

```

- 3.8.1. T: 4 bits type field Type = FOREIGN\_PACMAN\_ARRIVED has decimal value 10
- 3.8.2. unused: 25 bits, not used, but needed to maintain byte alignment. Must be set to zero in this version of the protocol
- 3.8.3. s = self: 3 bits, giving an unsigned integer in big-endian byte order

## 2.11. FOREIGN\_PACMAN\_LEFT

```

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 |               unused               | s |
+++++

```

- 3.8.1. T: 4 bits type field Type = FOREIGN\_PACMAN\_LEFT has decimal value 11
- 3.8.2. unused: 25 bits, not used, but needed to maintain byte alignment. Must be set to zero in this version of the protocol
- 3.8.3. s = self: 3 bits, giving an unsigned integer in big-endian byte order

## 2.12. PACMAN\_GO\_HOME

```

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 |               unused               | s |
+++++

```

- 3.8.1. T: 4 bits type field Type = PACMAN\_GO\_HOME has decimal value 12
- 3.8.2. unused: 25 bits, not used, but needed to maintain byte alignment. Must be set to zero in this version of the protocol
- 3.8.3. s = self: 3 bits, giving an unsigned integer in big-endian byte order



