



Garry's mod Death run

1112926 資訊工程學系 大二 陳家銓



大綱



作品架構、符合條件說明

架構圖、運用技術介紹

DEMO與遊戲介紹

程式執行、玩法介紹

遭遇的困難、挑戰

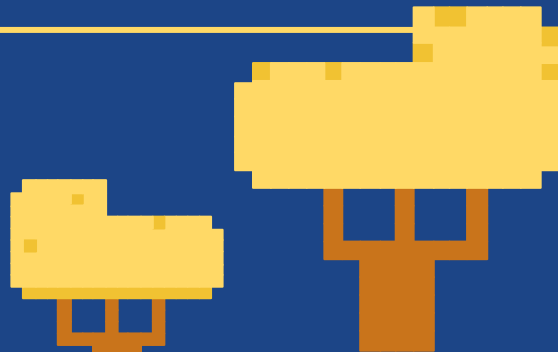
卡爛、耍笨

重點程式碼說明

網路實作與遊戲設計

心得

一點點自己的想法





01

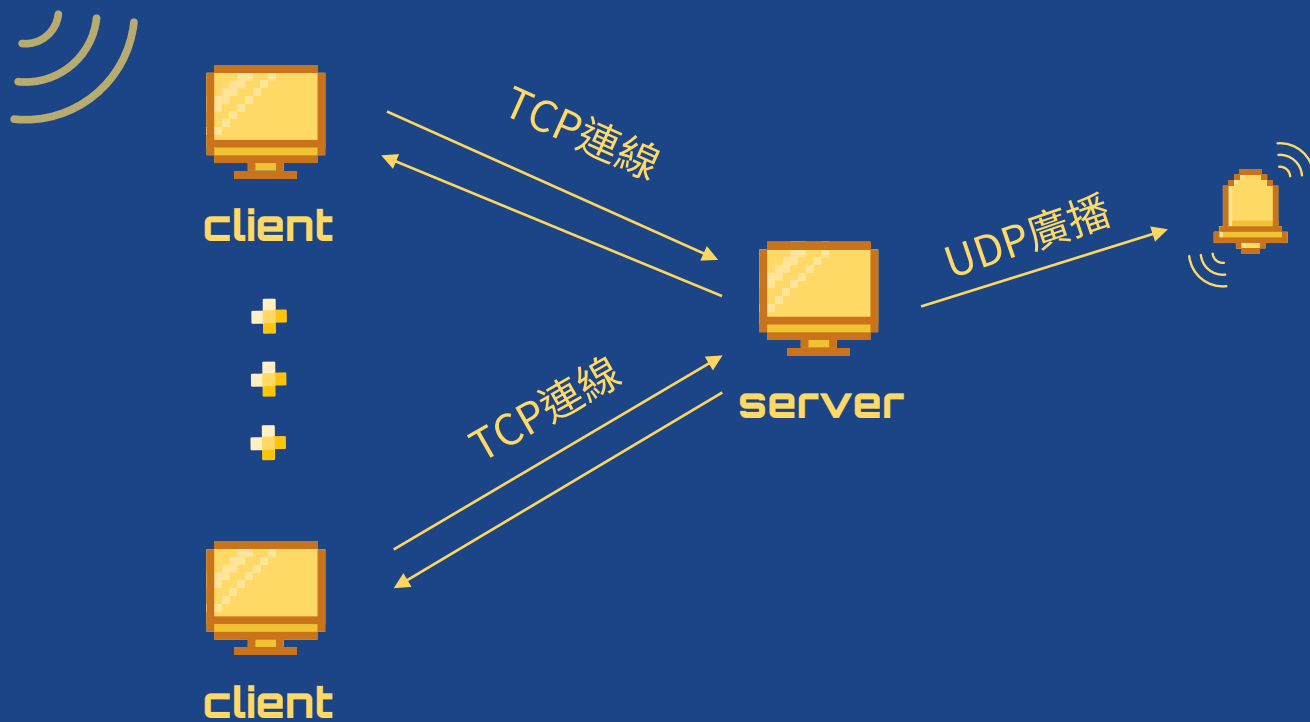


作品架構、符合條件說明

架構圖、運用技術介紹



基本架構圖





封包規格與分類



一般封包規格

長度為13個字
第一個字代表封包類型
後續每四個為一個參數
ex: 0111122223333



TCP連線

server

特殊封包 傳送玩家ID 長度為1

client

更新玩家位置: 0 玩家ID X座標 Y座標
殺手陷阱觸發: 1 陷阱ID 選擇123



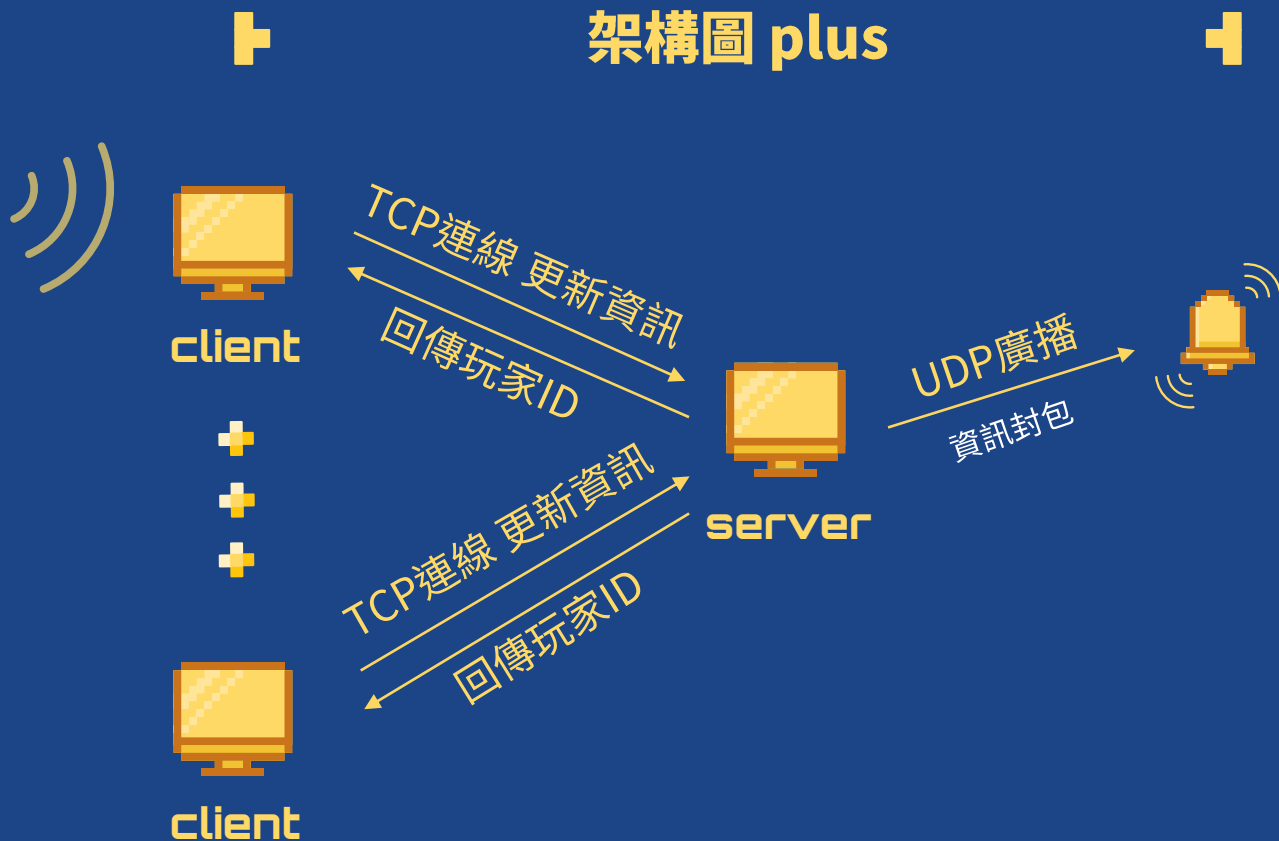
UDP廣播

server

玩家位置: 0 玩家ID X座標 Y座標
玩家退出: 1 玩家ID
切換觀戰: 2 玩家ID
陷阱警告: 3 陷阱ID 選擇123
陷阱觸發: 4 陷阱ID 選擇123
陷阱恢復: 5 陷阱ID 選擇123
遊戲結束: 6 勝利方

玩家位置
玩家退出
切換觀戰
陷阱警告
陷阱觸發
陷阱恢復
遊戲結束

架構圖 plus



Server架構圖



client

TCP連線 更新資訊



回傳玩家ID



server

UDP廣播



資訊封包



執行項目：

- 計算玩家狀態
- 檢查玩家連線狀況(thread)
- 計算並更新陷阱狀態(thread)
- 提供client TCP連線
- 傳送玩家ID
- 接收client的TCP封包
- 更新遊戲狀況
- 廣播遊戲狀況

使用技術：

- TCP、UDP
- Boardcast廣播
- Multi-threads多緒
- Non-blocking非攔阻模式

使用語言：C

`check_socket_alive()`

每5秒檢查一次
socket連線狀況
如果檢測到斷線就
回報狀況

`trap_action()`

發出陷阱警告廣播
等待一秒
發出陷阱啟動廣播
等待兩秒
發出陷阱恢復廣播

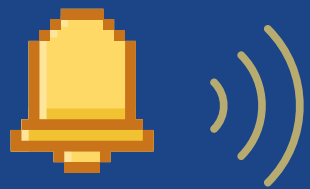
`main()`

初始化
提供TCP連線

`check_socket_alive()`

如果有client進來就回傳玩家ID
查看client端是否有傳送資料
如果是位置封包
就更新資料跟廣播資訊
順便判斷是否遊戲結束
如果是陷阱封包
就啟動 `trap_action()`
順便判斷是否遊戲結束

Client架構圖



玩家位置
玩家退出
切換觀戰
陷阱警告
陷阱觸發
陷阱恢復
遊戲結束



client

TCP連線 更新資訊

回傳玩家ID



server

執行項目：

接收廣播UDP(thread)
運行陷阱動畫(thread)
用TCP與server連線
解讀封包更新遊戲狀態
計算鏡頭位置
畫面渲染

使用技術：

TCP、UDP
Multi-threads多緒
pygame(GUI)

使用語言：Python

main()

初始化、TCP連線

recvData()

如果按下對應按鍵
就用TCP傳送資料

recvData()

解讀封包

0 1 2更新玩家資訊

3 4 5更新陷阱 BOOM()

6 更新game_over狀態

渲染畫面 update_screen()

update_screen()

計算鏡頭位置
繪出圖像和圖形

BOOM()

陷阱啟動後
播放動畫



使用技術與條件說明



Server端

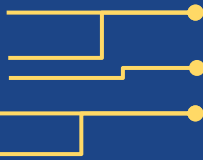


達成條件

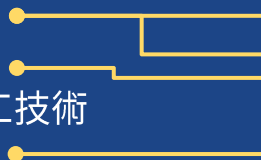


Client端

用TCP與client端連線
用廣播傳送資訊(UDP)
non-blocking
多線程



用到TCP與UDP
用到broadcast
server須用到多工技術
client須用GUI



用TCP與server端連線
UDP接收廣播資訊
多線程
用pygame寫GUI



02

DEMO與遊戲介紹

程式執行、玩法介紹



陷阱格

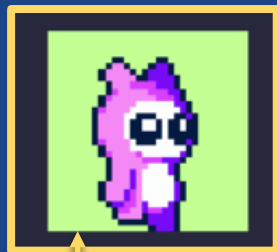


安全

警告

危險

玩家

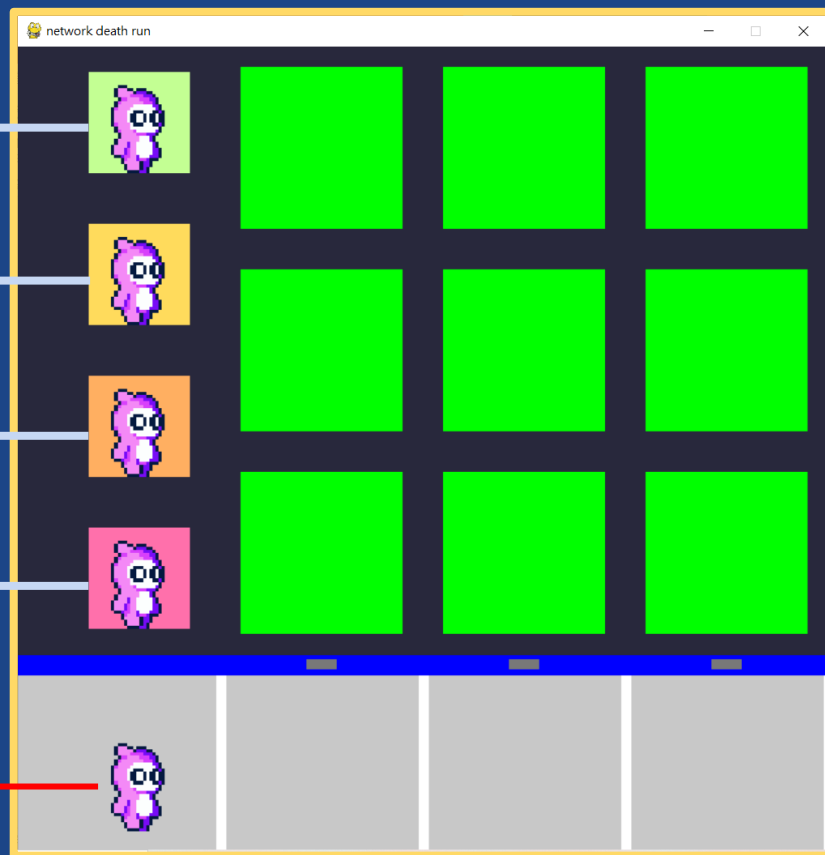


碰撞箱

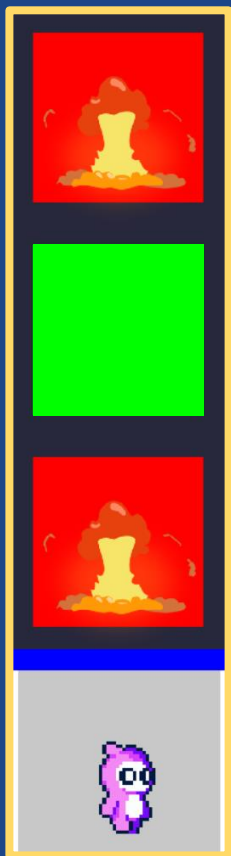
跑者

殺手

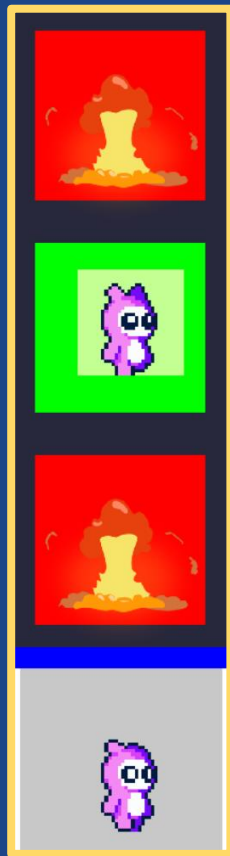
遊戲畫面(可同時2~5人遊玩)



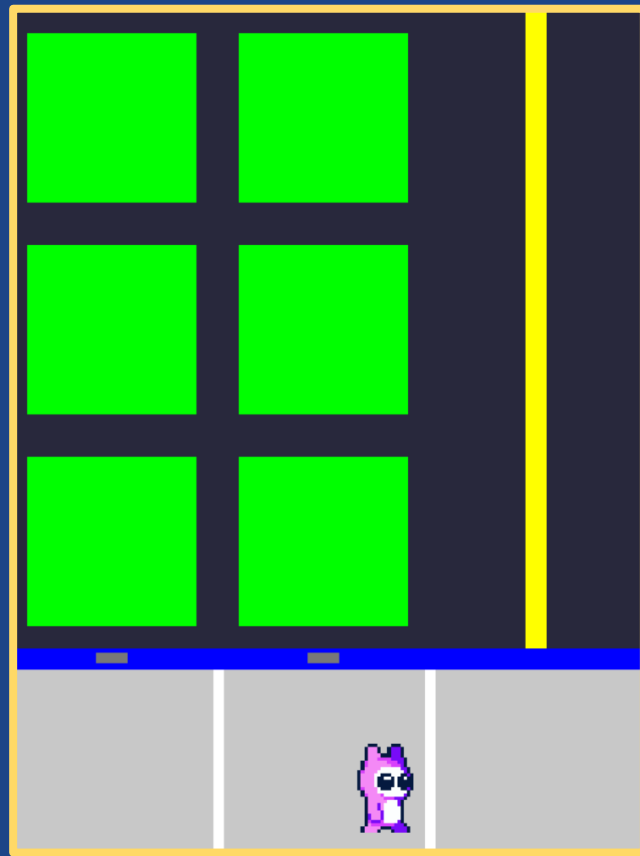
操縱陷阱阻止跑者



躲避殺手操縱的陷阱



終點線

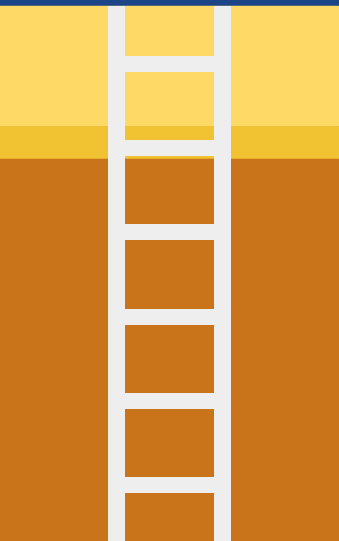




03

遭遇的困難、挑戰

卡爛、耍笨





遇到的困難與解決辦法



封包亂掉

因為client端可能會同時接收多個封包，變一個大封包，所以運作的時候會出事

我固定了封包的長度，所以如果長度大於一般的大小那只取前13個字
後面的資料再繼續處理下去這樣簡單來說就是在client端切封包資料



GUI的顯示區域

我是第一次寫pygame，我以為可以直接加個鏡頭之類的控制顯示的畫面區域，結果他根本沒這個東西

後來我使用玩家的座標來計算鏡頭偏移量，讓渲染畫面座標整個移過去



讓人物和爆炸動起來

人物我原本想要用的是一個方格，陷阱也是，我原本想用單純的色塊，但這樣太無趣了，我就想加GIF讓他可以動起來，結果它不能放GIF

後來我把動畫的每個影格用進去，一個圖片一個圖片跑，而且因為陷阱爆炸會需要獨立計算，所以又開了一個線程給它獨立運行



04

重點程式碼說明

網路實作與遊戲設計



Server code – network pt1

TCP連線區

```
240 // 連線區
241 memset(str,0,100);
242 n = accept(sd, (struct sockaddr *) &clnt,&clnt_len);
243 if(n!=-1)
244 {
245     for(i=0;i<5;i++)
246     {
247         if(clnt_sd[i]==-1)
248         {
249             clnt_sd[i]=n;
250             player_state[i]=1;
251             printf("Client[%d] is connected.\n",i);
252             playing++;
253             printf("playing: %d\n",playing);
254             sprintf(str,"%d",i);
255             n = send(clnt_sd[i],str, strlen(str)+1,0);
256             if(n == SOCKET_ERROR)
257             {
258                 clnt_sd[i]=-1;
259                 player_state[i]=0;
260                 printf("Client[%d] is disconnected.\n",i);
261                 playing--;
262             }
263             else
264             {
265                 //printf("(server send to client[%d]): %s\n",i,str);
266             }
267             HANDLE Thread1 = CreateThread(NULL, 0, (LPTHREAD_START_ROUTINE)broadcast_all, NULL, 0, NULL);
268             break;
269         }
270     }
271 }
272 }
```

接收client封包

```
273 // 接收TCP並區捕區
274 for(i=0;i<5;i++)
275 {
276     if(clnt_sd[i]!=-1)
277     {
278         memset(str,0,100);
279         n = recv(clnt_sd[i],str, MAXLINE,0);
280         if(n!=-1)
281         {
282             //printf("Client[%d] send:%s\n",i,str);
283             if(str[0]=='0')
284             {
285                 int ID = str[4]-'0';
286                 int x = (str[5]-'0') *1000 + (str[6]-'0') *100 + (str[7]-'0') *10 + (str[8]-'0');
287                 int y = (str[9]-'0') *1000 + (str[10]-'0') *100 + (str[11]-'0') *10 + (str[12]-'0');
288                 player_x[ID]=x;
289                 player_y[ID]=y;
290                 sprintf(str,"0x04d%04d%04d",ID,player_x[ID],player_y[ID]);
291                 broadcast(str);
292                 if(!game_over)
293                 {
294                     check_player_alive();
295                 }
296                 if(!game_over && ID!=0 && player_x[ID]>=2250)
297                 {
298                     // 到終點
299                     printf("player[%d] win\n",ID);
300                     sprintf(str,"6000100000000");
301                     game_over = 1;
302                     broadcast(str);
303                 }
304             }
305             if(str[0]=='1')
306             {
307                 int ID = (str[3]-'0') *10 + str[4]-'0';
308                 t_Args[ID]->unopen_num = (str[5]-'0') *1000 + (str[6]-'0') *100 + (str[7]-'0') *10 + (str[8]-'0');
309                 HANDLE Thread2 = CreateThread(NULL, 0, (LPTHREAD_START_ROUTINE)trap_action, t_Args[ID], 0, NULL);
310             }
311         }
312     }
313 }
```


TCP連線區

```
240 // 連線區
241 memset(str,0,100);
242 n = accept(sd, (struct sockaddr *) &clnt,&clnt_len );
243 if(n!=-1)
244 {
245     for(i=0;i<5;i++)
246     {
247         if(clnt_sd[i]==-1)
248         {
249             clnt_sd[i]=n;
250             player_state[i]=1;
251             printf("Client[%d] is connected.\n",i);
252             playing++;
253             printf("playing: %d\n",playing);
254             sprintf(str,"%d",i);
255             n = send(clnt_sd[i],str, strlen(str)+1,0);
256             if(n == SOCKET_ERROR)
257             {
258                 clnt_sd[i]=-1;
259                 player_state[i]=0;
260                 printf("Client[%d] is disconnected.\n",i);
261                 playing--;
262             }
263             else
264             {
265                 //printf("(server send to client[%d]): %s\n",i,str);
266             }
267         }
268         HANDLE Thread1 = CreateThread(NULL, 0, (LPTHREAD_START_ROUTINE)boardcast_all, NULL, 0, NULL);
269         break;
270     }
271 }
272 }
```

接收client封包

```
273 // 接收TCP 並廣播區
274 for(i=0;i<5;i++)
275 {
276     if(clnt_sd[i]!=-1)
277     {
278         memset(str,0,100);
279         n = recv(clnt_sd[i],str, MAXLINE,0);
280         if(n!=-1)
281         {
282             //printf("Client[%d] send:%s\n",i,str);
283             if(str[0]=='0')
284             {
285                 int ID = str[4]-'0';
286                 int x = (str[5]-'0') *1000 + (str[6]-'0') *100 + (str[7]-'0') *10 + (str[8]-'0');
287                 int y = (str[9]-'0') *1000 + (str[10]-'0') *100 + (str[11]-'0') *10 + (str[12]-'0');
288                 player_x[ID]=x;
289                 player_y[ID]=y;
290                 sprintf(str,"0%04d%04d%04d",ID,player_x[ID],player_y[ID]);
291                 broadcast(str);
292                 if(!game_over)
293                 {
294                     check_player_alive();
295                 }
296                 if(!game_over && ID!=0 && player_x[ID]>=2250)
297                 {
298                     //到終點
299                     printf("player%d win\n",ID);
300                     sprintf(str,"6000100000000");
301                     game_over = 1;
302                     broadcast(str);
303                 }
304             }
305             if(str[0]=='1')
306             {
307                 int ID = (str[3]-'0') *10 + str[4]-'0';
308                 t_Args[ID]->unopen_num = (str[5]-'0') *1000 + (str[6]-'0') *100 + (str[7]-'0') *10 + (str[8]-'0');
309                 HANDLE Thread2 = CreateThread(NULL, 0, (LPTHREAD_START_ROUTINE)trap_action, t_Args[ID], 0, NULL);
310             }
311         }
312     }
313 }
```

Server code – network pt2

確認玩家TCP連線(thread)

```
140 void* check_socket_alive()
141 {
142     int n,i;
143     char tmp[100];
144     while(1)
145     {
146         for(i=0;i<5;i++)
147         {
148             if(clnt_sd[i]!=-1)
149             {
150                 n = send(clnt_sd[i],"000", 4,0);
151                 if(n!=-1)
152                 {
153                     clnt_sd[i]=-1;
154                     player_state[i]=0;
155                     printf("Client[%d] is disconnected.\n",i);
156                     playing--;
157                     sprintf(tmp,"1%04d00000000",i);
158                     player_x[i]=9999;
159                     player_y[i]=9999;
160                     broadcast(tmp);
161                 }
162             }
163         }
164         Sleep(5000);
165     }
166 }
```

廣播、廣播所有玩家的位置(thread)

```
76 void broadcast(char *str)
77 {
78     int j;
79     for(j=0;j<5;j++)
80     {
81         cli.sin_port = htons(BroadcastPort +j);
82         sendto(Broadcast_sd, str, strlen(str), 0,(LPSOCKADDR)&cli,cli_len);
83         //printf("server broadcast on port:%d: %s\n",BroadcastPort+j ,str);
84     }
85 }
86
87 void* broadcast_all()
88 {
89     char str[100];
90     int j;
91     Sleep(200);
92     for(j=0;j<player_num;j++)
93     {
94         sprintf(str,"0%04d%04d%04d",j,player_x[j],player_y[j]);
95         //printf("%s\n",str);
96         broadcast(str);
97     }
98 }
```

確認玩家TCP連線

```
140 void* check_socket_alive()
141 {
142     int n,i;
143     char tmp[100];
144     while(1)
145     {
146         for(i=0;i<5;i++)
147         {
148             if(clnt_sd[i]!=-1)
149             {
150                 n = send(clnt_sd[i],"000", 4,0);
151                 if(n==-1)
152                 {
153                     clnt_sd[i]=-1;
154                     player_state[i]=0;
155                     printf("Client[%d] is disconnected.\n",i);
156                     playing--;
157                     sprintf(tmp,"1%04d00000000",i);
158                     player_x[i]=9999;
159                     player_y[i]=9999;
160                     boardcast(tmp);
161                 }
162             }
163         }
164         Sleep(5000);
165     }
166 }
```

廣播、廣播所有玩家的位置

```
76 void broadcast(char *str)
77 {
78     int j;
79     for(j=0;j<5;j++)
80     {
81         cli.sin_port = htons(BraodcastPort +j);
82         sendto(Braodcast_sd, str, strlen(str), 0,(LPSOCKADDR)&cli,cli_len);
83         //printf("server broadcast on port:%d: %s\n",BraodcastPort+j ,str);
84     }
85 }
86
87 void* boardcast_all()
88 {
89     char str[100];
90     int j;
91     Sleep(200);
92     for(j=0;j<player_num;j++)
93     {
94         sprintf(str,"0%04d%04d%04d",j,player_x[j],player_y[j]);
95         //printf("%s\n",str);
96         boardcast(str);
97     }
98 }
```



Server code - game



陷阱啟動(thread)

```
100 void* trap_action(void* t_Args)
101 {
102     int trap_ID = ((struct T_Args *) t_Args)->trap_ID;
103     int unopen_num = ((struct T_Args *) t_Args)->unopen_num;
104     //printf("%d,%d\n", trap_ID, unopen_num);
105     int i;
106     char str[100];
107     for(i=0; i<3; i++)
108     {
109         if(i != unopen_num)
110         {
111             trap[trap_ID][i]=0;
112         }
113     }
114     sprintf(str, "3%04d%04d0000", trap_ID, unopen_num);
115     boardcast(str);
116     Sleep(1000);
117     for(i=0; i<3; i++)
118     {
119         if(i != unopen_num)
120         {
121             trap[trap_ID][i]=1;
122         }
123     }
124     sprintf(str, "4%04d%04d0000", trap_ID, unopen_num);
125     boardcast(str);
126     check_player_alive();
127     Sleep(2000);
128     for(i=0; i<3; i++)
129     {
130         if(i != unopen_num)
131         {
132             trap[trap_ID][i]=0;
133         }
134     }
135     sprintf(str, "5%04d%04d0000", trap_ID, unopen_num);
136     boardcast(str);
137 }
138 }
```

確認玩家存活

```
46 void check_player_alive()
47 {
48     int i;
49     char str[100];
50     for(i=0; i<player_num; i++)
51     {
52         if(player_state[i]==1 && traped(i))
53         {
54             sprintf(str, "2%04d00000000", i);
55             player_state[i]=-1;
56             boardcast(str);
57         }
58     }
59     for(i=1; playing > 1 && i<player_num; i++)
60     {
61         if(player_state[i]==1)
62         {
63             break;
64         }
65     }
66     if(i==player_num)
67     {
68         //殺光光
69         printf("player0 win\n");
70         sprintf(str, "60000000000000", i);
71         game_over = 1;
72         boardcast(str);
73     }
74 }
```

陷阱啟動

```
100 void* trap_action(void* t_Args)
101 {
102     int trap_ID = ((struct T_Args *) t_Args)->trap_ID;
103     int unopen_num = ((struct T_Args *) t_Args)->unopen_num;
104     //printf("%d,%d\n", trap_ID, unopen_num);
105     int i;
106     char str[100];
107     for(i=0; i<3; i++)
108     {
109         if(i != unopen_num)
110         {
111             trap[trap_ID][i]=0;
112         }
113     }
114     sprintf(str, "3%04d%04d0000", trap_ID, unopen_num);
115     boardcast(str);
116     Sleep(1000);
117     for(i=0; i<3; i++)
118     {
119         if(i != unopen_num)
120         {
121             trap[trap_ID][i]=1;
122         }
123     }
124     sprintf(str, "4%04d%04d0000", trap_ID, unopen_num);
125     boardcast(str);
126     check_player_alive();
127     Sleep(2000);
128     for(i=0; i<3; i++)
129     {
130         if(i != unopen_num)
131         {
132             trap[trap_ID][i]=0;
133         }
134     }
135     sprintf(str, "5%04d%04d0000", trap_ID, unopen_num);
136     boardcast(str);
137 }
138 }
```

確認玩家存活

```
46 void check_player_alive()
47 {
48     int i;
49     char str[100];
50     for(i=0;i<player_num;i++)
51     {
52         if(player_state[i]==1 && trapped(i))
53         {
54             sprintf(str,"2%04d00000000",i);
55             player_state[i]=-1;
56             boardcast(str);
57         }
58     }
59     for(i=1;playing > 1 && i<player_num;i++)
60     {
61         if(player_state[i]==1)
62         {
63             break;
64         }
65     }
66     if(i==player_num)
67     {
68         //殺光光
69         printf("player0 win\n");
70         sprintf(str,"600000000000",i);
71         game_over = 1;
72         boardcast(str);
73     }
74 }
```


Client code – 接收廣播並處理

```
113 def recvData():
114     global screen_dx
115     global Broadcast_Port
116     global Broadcast_sd
117     global game_over
118     while True:
119         DISPLAY.fill((40,40,60))
120         data, addr = Broadcast_sd.recvfrom(1024)
121         data=data.decode('big5')
122         #print(data)
123         while len(data):
124             if data[0]=='0':
125                 if player_x[int(data[1:5])] > int(data[5:9]):
126                     player_dir[int(data[1:5])] = 1
127                 if player_x[int(data[1:5])] < int(data[5:9]):
128                     player_dir[int(data[1:5])] = 0
129                 player_x[int(data[1:5])]=int(data[5:9])
130                 player_y[int(data[1:5])]=int(data[9:13])
131                 player_state[int(data[1:5])] = 1
132                 if player_ID == int(data[1:5]):
133                     screen_dx = player_x[player_ID]-400
134                     if screen_dx < 0:
135                         screen_dx = 0
136                     elif screen_dx > 1600:
137                         screen_dx = 1600
138                 player_frame[int(data[1:5])] = player_next_frame(player_frame[int(data[1:5])])
139
140             if data[0]=='1':
141                 player_x[int(data[1:5])]=9999
142                 player_y[int(data[1:5])]=9999
143                 #print(f'player{int(data[1:5])} disconnect')
144             if data[0]=='2':
145                 if int(data[1:5]) != player_ID :
146                     player_x[int(data[1:5])]=9999
147                     player_y[int(data[1:5])]=9999
148                 player_state[int(data[1:5])] = 2
149             if data[0]=='3':
150                 for i in range(3):
151                     if i != int(data[5:9]):
152                         trap[int(data[1:5])][i] = 1
153             if data[0]=='4':
154                 for i in range(3):
155                     if i != int(data[5:9]):
156                         trap[int(data[1:5])][i] = 2
157                 threading.Thread(target=BOOM,args=(int(data[1:5]),int(data[5:9]))).start()
158             if data[0]=='5':
159                 for i in range(3):
160                     if i != int(data[5:9]):
161                         trap[int(data[1:5])][i] = 0
162             if data[0]=='6':
163                 print("game over")
164                 if int(data[1:5]) == 0 :
165                     game_over = 0
166                     print("murder win")
167                 else :
168                     game_over = 1
169                     print("runner win")
170             data = data[13:]
171             update_screen()
```

(這是單一線程運作的)

Client code – 鍵盤控制與送TCP

```
204 while True:
205
206
207 # 偵測事件
208 for event in pygame.event.get():
209     # 按叉叉就退出
210     if event.type == QUIT:
211         server_sd.close()
212         pygame.quit()
213         sys.exit()
214     keys = pygame.key.get_pressed()
215
216     if keys[pygame.K_a] or keys[pygame.K_LEFT]:
217         player_x[player_ID] -= 10
218     if keys[pygame.K_d] or keys[pygame.K_RIGHT]:
219         player_x[player_ID] += 10
220     if player_ID and (keys[pygame.K_w] or keys[pygame.K_UP]):
221         player_y[player_ID] -= 10
222     if player_ID and (keys[pygame.K_s] or keys[pygame.K_DOWN]):
223         player_y[player_ID] += 10
224
225     if player_x[player_ID] > XUB:
226         player_x[player_ID] = XUB
227     if player_x[player_ID] < XLB:
228         player_x[player_ID] = XLB
229     if player_ID != 0:
230         if player_y[player_ID] < YLB:
231             player_y[player_ID] = YLB
232         if player_y[player_ID] > YUB:
233             player_y[player_ID] = YUB
234
235
236     if player_state[player_ID] == 1:
237         if keys[pygame.K_w] or keys[pygame.K_a] or keys[pygame.K_s] or keys[pygame.K_d] or keys[pygame.K_LEFT] or keys[pygame.K_RIGHT] or keys[pygame.K_UP] or keys[pygame.K_DOWN]:
238             DATA = "0" + str(player_ID).zfill(4) + str(player_x[player_ID]).zfill(4) + str(player_y[player_ID]).zfill(4)
239             server_sd.send(DATA.encode())
240     if player_state[player_ID] == 2:
241         if keys[pygame.K_w] or keys[pygame.K_a] or keys[pygame.K_s] or keys[pygame.K_d] or keys[pygame.K_LEFT] or keys[pygame.K_RIGHT] or keys[pygame.K_UP] or keys[pygame.K_DOWN]:
242             screen_dx = player_x[player_ID] - 400
243             if screen_dx < 0:
244                 screen_dx = 0
245             elif screen_dx > 1600:
246                 screen_dx = 1600
247             update_screen()
248
249     if keys[pygame.K_1] or keys[pygame.K_2] or keys[pygame.K_3] or keys[pygame.K_KP1] or keys[pygame.K_KP2] or keys[pygame.K_KP3]:
250         if player_ID == 0:
251             button = player_x[0] // 200 - 1
252             if 10 > button >= 0:
253                 if not trap_opened[button]:
254                     if keys[pygame.K_1] or keys[pygame.K_KP1]:
255                         push = 0
256                     if keys[pygame.K_2] or keys[pygame.K_KP2]:
257                         push = 1
258                     if keys[pygame.K_3] or keys[pygame.K_KP3]:
259                         push = 2
260                     DATA = "1" + str(button).zfill(4) + str(push).zfill(4) + "0000"
261                     server_sd.send(DATA.encode())
262                     trap_opened[button] = 1
263
264     fpsClock.tick(FPS)
```



05

心得

一點點自己的想法



我當初是想說，我要用pygame做GUI，但server端要用C寫，算是個人執著吧(?)我覺得用C當server比較快。

我花了一個六日完成這個專案，花了大概26小時，我也是第一次寫pygame，成果還算滿意，我只覺得我以後碰遊戲設計要先學美工。

網路部分我覺得我規劃的還不錯，我是先寫好整個的規劃再開始做的，速度就會快一點，像是封包大小、內容，都是在寫之前就定好的，當然中途一定有小出錯，但有先寫好文本就會很方便。

BTW我是寫完才知道這東西叫做death run，然後再上平台去改名稱的，這東西不叫gmod，這只是gmod的一個遊玩小分支。





玩家人物素材

<https://craftpix.net/freebies/free-pixel-art-tiny-hero-sprites/?num=1&count=228&sq=cute&pos=5>

爆炸素材

<https://craftpix.net/freebies/free-flame-effects-sprite-pack/>





THANKS!

感謝觀看

