```
struct Bike1 {
  uint32_t x;      // x coordinate
  uint32_t y;      // y coordinate
  const unsigned short *image; //
ptr->image
  char life;      // 0=dead, 1=alive};
typedef struct Bike1 P1;
P1 Player1[4];
void Player1_Init(void){  int i;
  for(i=0;i<4;i++){
    Player1[i].x = 0;
    Player1[i].y = 70;
    Player1[i].image = &cycle1[80];
    Player1[i].life = 1;}
             P1direction = east;
             boost1count = 3;
ST7735_DrawBitmap(Player1[0].x
, Player1[1].y, cycle1, 40, 15);   }
int MoveP1(int h, int v){
             if(boost1>0){
                          h = h*2;
                          v = v*2;
                          boost1--;}
if((Player1[0].x)> 159 ){ //check x
position
Player1[3].life = 0;
return 1;//error, do nothing}
if((Player1[1].y)> 254 ){ //check y
position
Player1[3].life = 0;
return 1;  //error, do nothing}
while(h !=0 || v !=0){
                          if(h<0){
      Player1[0].x -= 1;
                                       h++;}
                          if(0<h){
                          Player1[0].x += 1;
                          h--;}
             if(v<0){
                          Player1[1].y -= 1;
                          v++;}
                          if(0<v){
                          Player1[1].y += 1;
                                       v--;}
P1pos[P1Index].x = Player1[0].x;
P1pos[P1Index].y = Player1[1].y;
checkIndex = P1Index;
while(checkIndex !=0){
if((P1pos[P1Index].x ==
P2pos[checkIndex].x)&&(P1pos[P
1Index].y ==
P2pos[checkIndex].y)){
Player1[3].life = 0;}
checkIndex--;}
checkIndex = P1Index-1;
if(checkIndex>0){
while(checkIndex !=0){
if((P1pos[P1Index].x ==
P1pos[checkIndex].x)&&(P1pos[P
1Index].y ==
P1pos[checkIndex].y)){
Player1[3].life = 0;}
checkIndex--;}}
P1Index++;
if(Player1[1].y <128){
ST7735_DrawBitmap(Player1[0].x
, Player1[1].y, pixel1, 1, 1);}}
UART_OutChar(0x02);//STX
UART_OutChar(0x0E);//current
screen
UART_OutChar(Player1[0].x);//p1
data
UART_OutChar(Player1[1].y);
UART_OutChar(Player2[0].x);//p2
data
UART_OutChar(Player2[1].y);
UART_OutChar(0x00);//extra
UART_OutChar(0x03);//ETX
return 0; }
/////////////////////////////////////////
int P1direction = east;
int P2direction = west;
int arrH[4] = {0, 1, 0, -1};
int arrV[4] = {1, 0, -1, 0};
void Set_Flag(){
Arena_Flag = 1;
}
int Arena_Map(int R1, int L1, int
R2, int L2){
if(R1 == 1 && L1 ==0){
             if(P1direction == 3){
                          P1direction = 0;}
             else{
             P1direction++;}}
if(R1 == 0 && L1 ==1){
             if(P1direction == 0){
                          P1direction = 3;}
             else{
             P1direction--;}
h1 = arrH[P1direction];
v1 = arrV[P1direction];
h2 = arrH[P2direction];
v2 = arrV[P2direction];
while((Player1[3].life) == 1
             && (Player2[3].life )==
1){
while(Arena_Flag == 0){}
Arena_Map(rightb1, leftb1,
rightb2, leftb2);
MoveP1(h1, v1);
MoveP2(h2, v2);
             Arena_Flag = 0;
```