


```
64         return;
65     }
66     Sukima2.Left += 4;
67     if( Sukima2.Left > Width )
68     {
69         Sukima2.Left = -Sukima2.Width;
70     }
71 }
72
73 private void timer1_Tick(object sender, EventArgs e)
74 {
75     Cpos = PointToClient(Cursor.Position);
76     moveSukima(); //すき間の横移動
77
78     if( shotFlg )
79     {
80         movePlayer(); //プレイヤー上移動
81         return; // timer1_Tick から抜ける
82     }
83
84     if (Cpos.X < 0)
85     {
86         Cpos.X = 0;
87     }
88     if( Cpos.X > Width - Player.Width )
89     {
90         Cpos.X = Width - Player.Width;
91     }
92     Player.Left = Cpos.X;
93 }
94
95 private void Form1_Load(object sender, EventArgs e)
96 {
97     initGame();
98 }
99
100 // ゲームの初期化
101 private void initGame()
102 {
103     hitFlg = false; //false:当たっていない
104     Gameover.Hide(); // ゲームオーバーを隠す
105     button1.Enabled = false; // 開始ボタンを無効にする
106     score = 0; // スコアのクリア
107     scoreLabel.Text = "0";
108     Sukima2.Width = 80; //すき間の幅
109     initPlayer();
110 }
111
112 // プレイヤーの初期化
113 private void initPlayer()
114 {
115     Player.Top = Height - (Player.Height * 2);
116     Player.Left = Cpos.X;
117     shotFlg = false; //False:発射していない
118 }
119
120 private void Form1_MouseDown(object sender, MouseEventArgs e)
121 {
122     shotFlg = true; // true:発射移動中
123 }
124
125 private void button1_Click(object sender, EventArgs e)
126 {
127     initGame();
```

```
128     }
129
130     private void Form1_Load_1(object sender, EventArgs e)
131     {
132
133     }
134 }
135 }
136
```