

Шерманова М., Тарасенко Е., Кравченко Н., Хрущелева К.

студенты специальности Информатика

Научный руководитель Бегалин А.Ш., ст. преподаватель

КГУ им. А. Байтурсынова

Республика Казахстан, г. Костанай

РАЗРАБОТКА МИНИ-ИГРЫ АЭРО-ХОККЕЙ

Аннотация

В данной статье описываются этапы разработки и использования компьютерной игры Аэро-хоккей. Программа реализована в среде C++.

Ключевые слова: компьютерные игры, мини-игры, интерфейс.

Существуют множество компьютерных игр различных жанров, таких как логические, аркадные, приключения, симуляторы и т.д. Но никто наверно не задумывался, как работают игры, как они устроены внутри, ну, наконец, как они создаются. Сама игра может влиять на координацию и логическое мышление играющего и поэтому относиться к логическим играм.

Разработанная игровая программа «Аэро-хоккей» является мини-игрой.

Цель игры заключается в том, чтобы игрок, управляя с клавиатуры, отбивал от своих ворот движущийся мяч. При каждом попаданий в одни из двух ворот, засчитываются определенные баллы, где воротами является правая и левая стороны игрового поля (см. рис.1)



Рисунок 1. Игровое поле

Управление с клавиатуры возможно только правой ракеткой, следовательно, левой ракеткой управляет сам компьютер. Игрок может перемещаться во всех четырех направлениях, чтобы отбить мяч.

Программой обрабатываются события от нажатия клавиш на клавиатуре, выбора соответствующих пунктов меню, а также считывание забитых голов, для определения победителя.

Игра «Аэро-хоккей» состоит из трех уровней сложности: уровень «Для новичков», «Средний уровень» и «Сложный уровень». Уровни игры «Аэро-хоккей» отличаются по уровням сложности. То есть, каждый уровень отличается, скоростью передвижения мяча и скорость перемещения левой ракетки за мячом.

Интерфейс программы

Интерфейс игры очень прост - состоит из двух окон меню, которые в свою очередь состоят из нескольких разделов. Первое окно - окно меню состоит из следующих разделов: новая игра, справка и выход из игры (см. рис.2). Второе окно меню, следующее при нажатии в главном меню раздела «Новая игра», окно выбора уровней для начала игры и выход. В разделе «Справка» главного меню выявляется окно о программе, правилах игры и др.

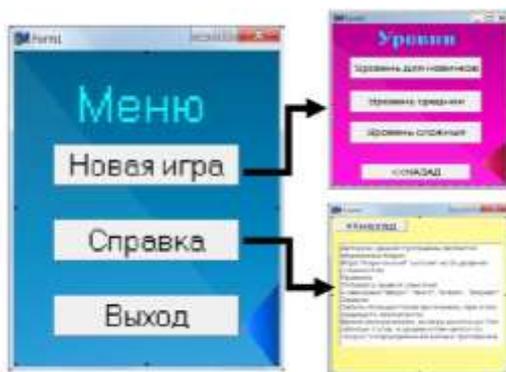


Рисунок 2. Главное меню и переходы в окна справки и новой игры

Начать игру можно открыв окно уровней, а в этом окне открыть игру и уровень по выбору. При этом открывается соответствующее игровое поле, где можно начать игру при нажатии кнопки «START».

В поле помимо кнопки начала игры находятся две ракетки, правая из которых управляется пользователем с клавиатуры, а другая – компьютером, игровой мяч. Кроме всего этого, вы заметите отсчет времени игры, подсчет баллов и переход в меню при завершении.

Описание логической структуры

*/*начальные координаты мяча */*

Shape1->Left =367;

Shape1->Top =216;

Label2->Caption =ball=0;

Label3->Caption =bal2=0; }

Label4->Visible =false;}

Действия в таймере:

void __fastcall TForm3::Timer1Timer(TObject *Sender)

Отскок от правой ракетки:

```
if((Shape1->Top+33>Image2->Top )&&(Shape1->Top<Image2-
>Top+55) &&(Shape1->Left+40>Image2->Left) && (Shape1-
>Left+33<Image2->Left+40 ))dx=-dx;
```

/ перемещение мяча*/*

```
Shape1-> Left+=dx ;
```

```
Shape1-> Top+=dy ;
```

Перемещение левой ракетки за мячом происходит только на уровне этого мяча.

```
/*перемещение левой ракетки за мячом*/
```

```
if(Image3->Top>Shape1-> Top) Image3->Top -= 5;
```

```
else Image3->Top +=5;
```

```
Label1->Caption=Time();
```

```
/*подсчет баллов*/
```

```
Label2->Caption =bal1;
```

```
Label3->Caption =bal2;
```

В данном фрагменте мы видим алгоритм управления для правой ракетки и его пределы в игровом поле

```
void __fastcall TForm3::FormKeyDown(TObject *Sender, WORD &Key,  
TShiftState Shift)
```

```
{ /*управление с клавиатуры правой ракетой */
```

```
if(Key==38)Image2->Top -=3;
```

```
if(Key==40)Image2->Top +=3;
```

```
if(Key==37)Image2->Left -=2;
```

```
if(Key==39)Image2->Left +=2;
```

Так как перемещение мяча задано начальными координатами dx, dy, то при описании этих переменных присваиваются начальные значения в таком виде:

```
int dx=10, dy=10
```

чем больше значения, тем больше скорость передвижения мяча:

```
Shape1-> Left+=dx ;
```

```
Shape1-> Top+=dy ;
```

И при перемещении левой ракетки, чем больше значение Top, тем больше скорость перемещения ракетки за мячом:

```
if(Image3->Top>Shape1-> Top) Image3->Top -= 5;  
else Image3->Top +=5;
```

Этот фрагмент выводит отсчет времени на поле:

```
Label1->Caption=Time();
```

Фрагмент подсчет баллов работает в программе, где описан отскок мяча от стен. То есть когда мяч касается пределов двух ворот в Label2 и Label3 соответственно, подсчитывается количество попаданий (См. рис. 3).

Ниже представлен алгоритм в программе:

```
if(Shape1-> Left<Image1->Left+30) {dx=-dx; bal1 ++; Shape1->Left  
=367;Shape1->Top =216; }
```

```
if(Shape1-> Left>Image1->Left+Image1->Width-70) {dx=-dx; bal2++;  
Shape1->Left =367;Shape1->Top =216;}
```

```
if(((Shape1-> Top<Image1->Top+30)||((Shape1-> Top>Image1-  
>Top+Image1->Height-70)) dy=-dy;
```

```
/*подсчет баллов*/
```

```
Label2->Caption =bal1;
```

```
Label3->Caption =bal2;
```

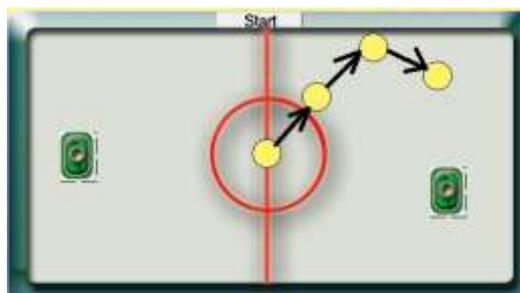


Рисунок 3. Отскок от стен

А при первых пяти набранных баллов, то есть когда bal1 или bal2 принимают значение 5, игра останавливается. При этом начать игру можно перейдя в меню. Конец игры сопровождается окном с текстом о конце игры.