


§14. Создаем свою первую игру. Блоха.

Задача: создать простенькую игру, основанную на принципе случайного распределения.

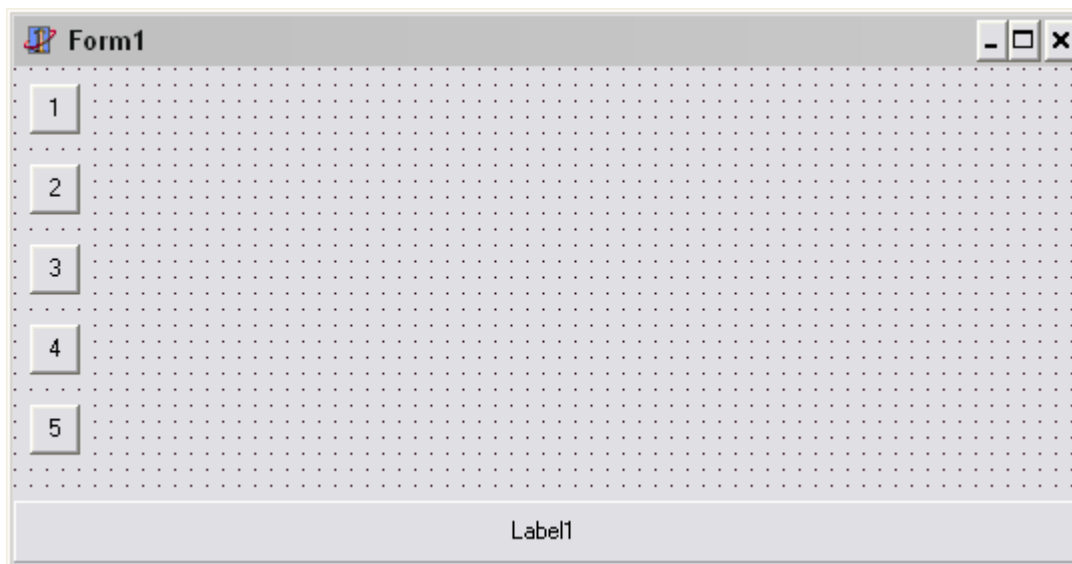
Придумаем сюжет.

Итак, что собой представляет игра? При нажатии на кнопку(блоху), она сдвинется на случайное количество пикселей по горизонтали. То есть прыгнет. Так как прыжки блохи мало предсказуемы, то и мы сделаем так, чтобы, после щелчка по блохе, она перескакивала на любое место по горизонтали. Если блоха (кнопка) пересечет середину экрана, то будем считать, что блоха (кнопка) выполнила свою задачу и может отдыхать (отключается). Ваша задача: за наименьшее количество щелчков перетащить всех блох на вторую половину формы. Как только все пять кнопок пересекут "экватор", игра заканчивается.

Проектируем форму.

1. Откройте **Delphi**.
2. Высота формы нам не важна, а вот ширину формы сделайте равной 675 (Width= 675). Свойство **Color** сделайте красивым. У меня, например, **clAqua**
3. На форму поместите 5 кнопок – **Button**. Размеры кнопок сделайте равными 25*25 (Width= 25, Height= 25). Расположите их как на рисунке, который находится ниже.
4. Свойства **Caption** у кнопок сделайте соответственно 1,2,3,4 и 5.
5. Поместите на форму элемент **Panel** – кнопка  на панели инструментов **Standart**. С помощью панели мы будем отражать текущее состояние игры.
6. Свойству **Align** присвойте значение **alBottom** (по нижнему краю формы). Вы увидите, как панель «приклеится» к нижнему краю формы.
7. Свойство **Caption** очистите. Панель должны быть чистая.
8. На панель поместите элемент **Label**. На нем будут выдаваться пояснения.

Если Вы все выполнили, то Ваша форма будет иметь вид, похожий на эту:



Приступим к программированию.

Программируем игру:

Все действия в игре происходит при нажатии на кнопку.

Так как, главное событие-это щелчок по кнопке, то с этого и начнём:

9. Сделайте двойной щелчок по **кнопке 1**, чтобы попасть в обработчик события.

При щелчке кнопка должна переместиться на случайное количество пиксель по горизонтали. Следовательно, менять будет свойство **left**. Но как сделать случайность? В **Delphi** для таких случаев используют функцию **Random (X)**. Данная функция генерирует случайное число в диапазоне от **0** до **X**. Но, чтобы запустить генератор случайных чисел, вначале необходимо выполнить процедуру **Randomize**. Учтем это позже, а сейчас нам нужно указать, какое максимальное число использовать в функции. Ширина формы – 675 пиксель. Первые 25 пиксель заняты под кнопку (Помните, размер кнопки- 25*25), а $675-25= 650$.

Значит наш оператор примет вид: **button1.Left:= Random (650)+25**; Почему +25 догадайтесь сами.

10. Далее, считаем количество щелчков. Как? Самый простой вариант - закрепить за этим делом специальную переменную, например, **k**. А в обработчике события напишем: **k:= k+1**; (k- количество щелчков)

Если кнопка пересечет середину экрана, то:

- Отключаем эту кнопку;
- Проверяем, все ли кнопки пересекли середину экрана.

11. Отключаем кнопку: **button1.enabled:= false**;

А как проверить, все ли кнопки пересекли середину? Как в 10 пункте, введем ещё один счетчик, который будет отсчитывать количество кнопок, пересекающих середину, **n**. Теперь необходимо написать оператор: **n:= n+1**; При пересечении кнопкой середины формы, **n** автоматически увеличиться на единицу. И теперь понятно, как указать программе: все ли кнопки пересекли границу: достаточно проверить переменную **n**. Если **n = 5**, то цель достигнута.

12. А теперь посмотрите внимательно на код **кнопки 1**. Вам все должно быть понятно:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  button1.Left := random (650)+25;
  k:=k+1;
  if button1.Left > 325 then
    begin
      button1.enabled:=false;
      n:=n+1;
      if n=5 then
        begin
          form1.Color:=clred;
          ShowMessage (' Победа ');
        end;
    end;
  label1.Caption:= 'Вы сделали '+ IntToStr(k)+' ходов';
end;
```

13. Далее необходимо описать переменные. После раздела **implementation** в окне кода запишите переменные, как мы делали в предыдущем уроке:

```
var  
  Form1: TForm1;  
  
Implementation  
var n,k:integer;
```

14. Запустите программу (**F9**). Проверьте, работает ли кнопка?
15. Закройте программу - то есть вернитесь в режим проектирования и сохраните свою первую игру под именем **bloha**.
16. Запустите игру еще раз. Проверьте, как работает кнопка. Закройте программу. И запустите еще раз. Внимательно понаблюдайте за работой кнопки. У Вас не возникает ощущения, что ваша кнопка вместо того, чтобы случайно прыгать по экрану, все время прыгает по одинаковым местам? Дело в том, что мы не запустили генератор случайных чисел.
17. Перейдите в режим проектирования. Щелкните кнопкой мыши по форме, чтобы она выделилась, и перейдите на вкладку **Events** (События) в Инспекторе объектов.
18. Дважды щелкните рядом с событием **OnCreate**, чтобы создать обработчик события.
19. Введите команду: **randomize**;
20. Здесь же введем начальные значения переменных (помните, как в прошлом уроке):

```
procedure TForm1.FormCreate(Sender: TObject);  
begin  
  randomize;  
  k:=0;  
  n:=0;  
end;
```

Программируем другие кнопки.

21. Тот же самый обработчик нужно записать для **2-ой кнопки**. Только везде вместо **button1** - Вы будете писать **button2**.
22. Аналогично расправьтесь с 3-ей, 4-ой, 5-ой кнопкой.
23. Запустите программу. Теперь, когда все обработчики прописаны, игра должна заканчиваться правильно: форма окрасится в красный цвет, в **Label1** будет красоваться количество сделанных ходов, а Вы увидите **сообщение о победе**.
24. Обязательно сохраните вашу игру, на следующем уроке мы ее исправим (доведем до ума, модифицируем, сделаем ее для двоих, разберемся, что такое красивый и некрасивый код).

Самостоятельно:

1. Очень внимательно разберитесь с каждой строчкой кода. Поймите, зачем нужно каждое действие.
2. Постарайтесь самостоятельно еще раз написать данную игру, не подсматривая в материал урока.
3. Усовершенствуйте программу по своему вкусу (например, измените вид курсора мыши, измените цвета, шрифты).
4. Поставьте на панель еще одну метку (**Label2**) и сделайте так, чтобы она отражала текущую информацию (комментировала игру - например, при достижении 1-ой кнопки заветной середины, выдавала бы надпись: «1-ая кнопка финишировала»).
5. Сделайте кнопки более информативными (например, чтобы свойство **Caption** кнопки отражало координату **Left** этой кнопки).
6. Разберитесь внимательно код **кнопки 1**. Сделайте пояснения на русском языке, что делает

каждая строчка в этой процедуре.

Вместо вопросиков Вам нужно сделать краткие пояснения

```
procedure TForm1.Button1Click(Sender: TObject); // ?
Begin // ?
  button1.Left := random (650)+25; // ?
  k:=k+1; // ?
  if button1.Left > 325 then // ?
    Begin // ?
      button1.enabled:=false; // ?
      n:=n+1; // ?
      if n=5 then // ?
        Begin // ?
          form1.Color:=clred; // ?
          ShowMessage (' Победа '); // ?
        end; // ?
      end; // ?
    end; // ?
  label1.Caption:= 'Вы сделали '+ IntToStr(k)+' ходов'; // ?
end; // ?
```

7. Придумайте сами какой-либо сюжет, основанный на принципе случайного распределения... и попробуйте воплотить его в жизнь.