VIMUI (**VI**rtual **MU**sic **I**nstrument) on **jdummy2d tutorial**
Напишем на jdummy2d вирутуальный музыкальный инструмент
Играть можно будет на PC-клавиатуре, мышью и пальцами по тач-скрину (с экраном >9”, на меньших будет неудобно, так как клавиш у нас будет много).
В этом уроке используется jdummy2d alpha-23aug2012

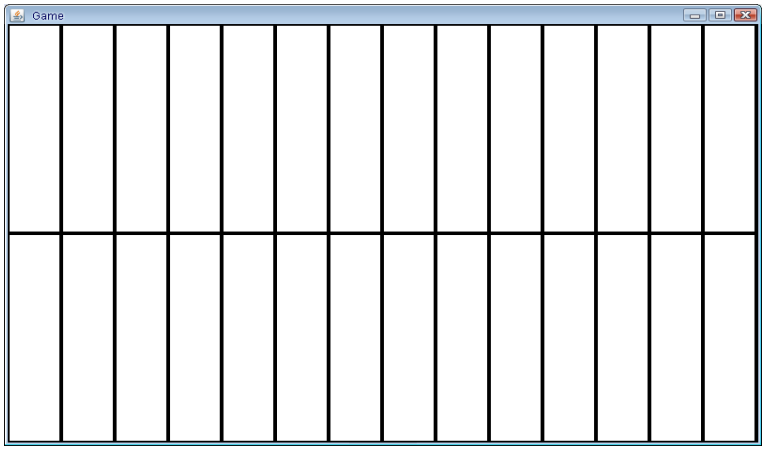
Разметка клавиатуры.
Виртуальных миди-клавиатур, пытающихся имитировать фортепианную клавиатуру с разметками типа q2w3er5t6y7u и двумя неполными октавами уже хватает. Сделаем немножко по-другому и задействуем больше клавиш.

```
Первая октава: ctrl  z x spc c v b n m , . /
Вторая октава: shift a s d f g h j k l ; Enter
Третья октава:   tab q w e r t y u i o p -
Четвертая октава:esc 1 2 3 4 5 6 7 8 9 0 F9
                   C C#D D#E F F#G G#A A#B
```

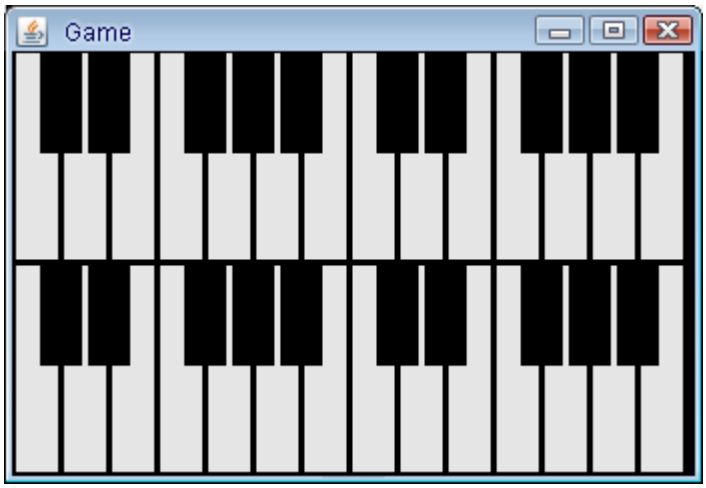
Нестандартно, но привыкнуть можно, зато вдвое больше октав, благодаря чему можно хоть как-то «развернуться».

Итак, экран у нас будет делиться на 14 частей по горизонтали и на 2 части по вертикали. При нажатии будем чуть затемнять клавиши.
Нарисуем белые клавиши:

```
FUNC RENDER()
    CLS(BLACK)
    SHAPE_FRECTS()
    SETCOLOR(WHITE)
    INT S = GETW()/14
    FOR (INT I=0;I<14;I++)
        FRECT(I*S-GETW()/2+2, 0+2,          S-4, GETH()/2-4)
        FRECT(I*S-GETW()/2+2, -GETH()/2+2, S-4, GETH()/2-4)
    NEXT
    END_SHAPE()
ENDFUNC
```



Теперь черные клавиши:




```
FUNC RENDER()
  CLS(BLACK)
  SHAPE_FRECTS()
  INT S = GETW()/14
  SETCOLOR(SMOKE) /* сменил цвет белых клавиш */
  FOR (INT I=0;I<14;I++)
    FRECT(I*S-GETW()/2+2, 0+2,          S-3, GETH()/2-3)
    FRECT(I*S-GETW()/2+2, -GETH()/2+2, S-3, GETH()/2-3)
  NEXT
  SETCOLOR(BLACK)
  FOR (INT I=0;I<14;I++)
    IF (I!=2 AND I!=6 AND I!=9 AND I!=13) THEN
      FRECT(I*S-GETW()/2+2+S/2, 0+2+GETH()/4,          S-3, GETH()/4-3)
      FRECT(I*S-GETW()/2+2+S/2, -GETH()/2+2+GETH()/4, S-3, GETH()/4-3)
    ENDIF
  NEXT
  END_SHAPE()
ENDFUNC
```

Создадим звуки. Audacity → Создание → Волна, квадратичная, 130,81 Гц (До), 2 секунды. Затем Эффекты → Плавное затухание. Выделить волну двойным щелчком, Файл → Экспортировать выделенное. Сохраним в data/assets/ как 1.wav. Повторим: 261,63 Гц, 523,25 Гц, 1046,5 Гц (2.wav, 3.wav, 4.wav). Загружаем все 4 файла:

```
SOUND[] O = NEW SOUND[4]

FUNC INIT()
  FOR (INT I=0;I<4;I++)
    O[I] = SOUND_LOAD(TXT(I+1)+".wav")
  NEXT
ENDFUNC
```

Почему квадратичная волна, а не сэмпл пиано: потому что на мобильных устройствах такие сэмплы звучат либо слишком тихо, либо как говно. А у square-волны звук яркий, сочный, то что надо, в общем.

Клавишные константы и ID звука для каждой клавиши:
Откуда взялся следующий список: я сделал DEBUG(TXT(KEYCODE)) в функции KEYUP, запустил через run-console.bat и нажал по очереди нужные клавиши. Скопировал вывод консоли, заменил \r на запятые с пробелом, а \n на пустоту.

```
INT[] KEYMAP = {129, 54, 52, 62, 31, 50, 30, 42, 41, 55, 56, 76, 59, 29, 47, 32,
34, 35, 36, 38, 39, 40, 74, 66, 61, 45, 51, 33, 46, 48, 53, 49, 37, 43, 44, 69,
131, 8, 9, 10, 11, 12, 13, 14, 15, 16, 7, 252}

LONG[] ID = NEW LONG[48]
```

Добавим обработчик нажатия клавиши PC-клавиатуры:

```
FUNC KEYDOWN(INT KEYCODE) RETURNS BOOL
  INT N = -1
  // ищем позицию нажатой клавиши
  FOR (INT I=0;I<KEYMAP.LENGTH;I++)
    IF (KEYMAP[I] == KEYCODE) THEN
        N = I
        BREAK
      ENDIF
  NEXT
  IF (N != -1) PIANODOWN(N)
  RETURN TRUE
ENDFUNC

FUNC KEYUP(INT KEYCODE) RETURNS BOOL
  INT N = -1
  FOR (INT I=0;I<KEYMAP.LENGTH;I++)
    IF (KEYMAP[I] == KEYCODE) THEN
        N = I
        BREAK
      ENDIF
  NEXT
  IF (N != -1) PIANOUP(N)
  RETURN TRUE
ENDFUNC
```

Добавим в самое начало программы новую глобальную переменную:

```
BOOL[] ISPLAYING = NEW BOOL[48]
```

Она нужна, чтобы не повторять звук, пока не отпущена клавиша.
Функции проигрывания и остановки звуков по их ID:

```
FUNC PIANODOWN(INT I)
  IF (ISPLAYING[I]) RETURN
  INT On = (int)Math.floor(I/12)
  INT Nn = I — On*12
  /* http://ru.wikipedia.org/wiki/Равномерно_темперированный_строй */
  FLOAT Pt = (float)Math.pow(Math.pow(2,Nn), 1.0f/12.0f)
  ID[I] = O[On].PLAY(1,Pt,0)
  ISPLAYING[I] = TRUE
ENDFUNC

FUNC PIANOUP(INT I)
  ISPLAYING[I] = FALSE
  INT On = (int)Math.floor(I/12)
  INT Nn = I - On*12
  O[On].STOP(ID[I])
ENDFUNC
```

Компилируем, запускаем, пробуем играть. Где-то можно играть одновременно 4 ноты, где-то 3, где-то только две. Например, есть такое ограничение у PC-клавиатуры: нельзя одновременно нажать E+6+U или 4+5+6.

Без надписей на клавишах играть сложновато. Готовим надписи для белых и черных клавиш отдельно:

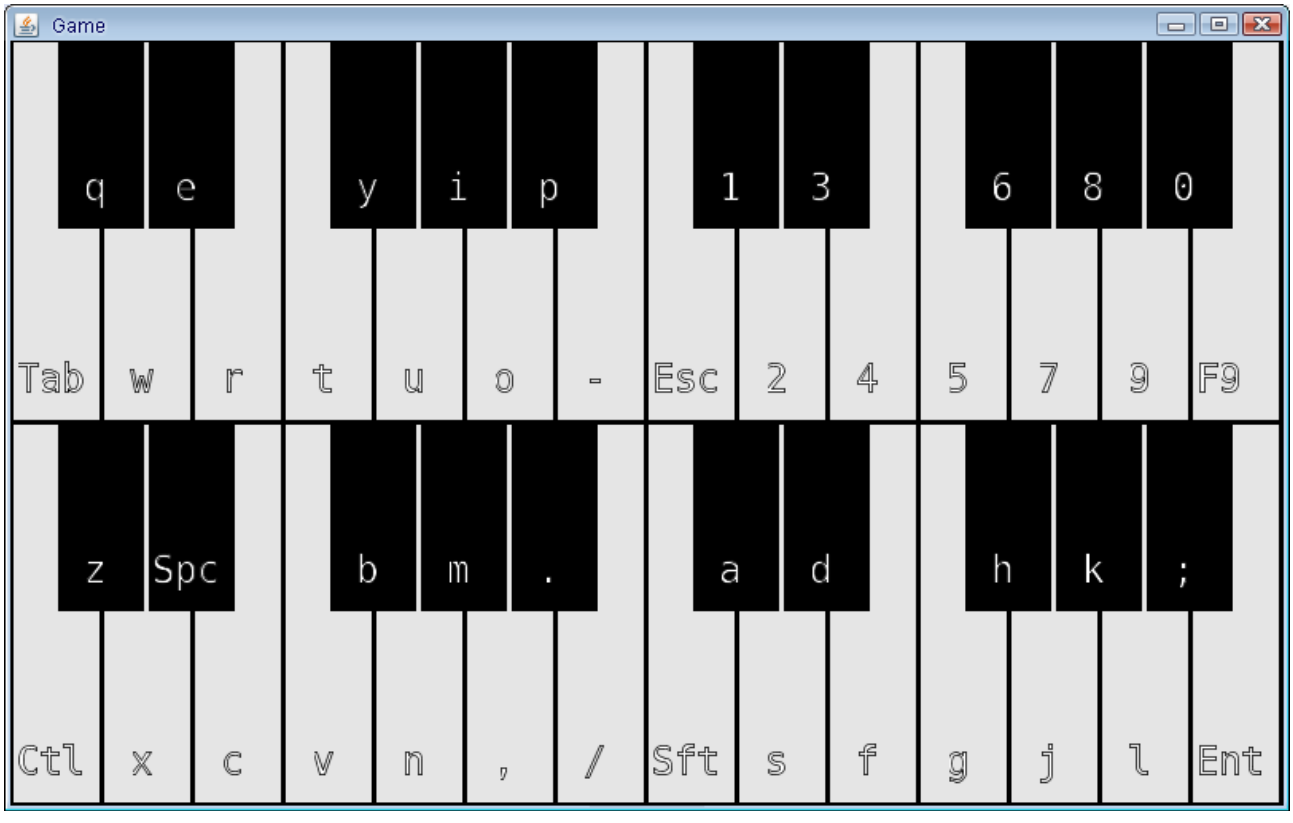
```
STR[] w1 = {"Ctl" ," x" ," c" ," v" ," n" ," ," ," /" ,"Sft" ," s" ," f" ," g" ," j" ," l" ,"Ent" }
STR[] w2 = {"Tab" ," w" ," r" ," t" ," u" ," o" ," -" ,"Esc" ," 2" ," 4" ," 5" ," 7" ," 9" ," F9" }
STR[] b1 = {" z" ,"Spc" ,"SKIP" ," b" ," m" ," ." ,"SKIP" ," a" ," d" ,"SKIP" ," h" ," k" ," ;" }
STR[] b2 = {" q" ," e" ,"SKIP" ," y" ," i" ," p" ,"SKIP" ," 1" ," 3" ,"SKIP" ," 6" ," 8" ," 0" }
```

Я сгенерировал более разборчивый в этой ситуации шрифт (DejaVu Sans Mono).

```
...
FUNC INIT()
  SET_FONT("myfont")
  FOR (INT I=0;I<4;I++)
    O[I] = SOUND_LOAD(TXT(I+1)+".wav" )
  NEXT
  FOR (INT I=0;I<ISPLAYING.LENGTH;I++)
    ISPLAYING[I] = FALSE
  NEXT
ENDFUNC

FUNC CYCLE()
  RENDER()
  RENDER_TEXT()
  CAPFPS(30) /* для экономии энергии и разгрузки CPU */
ENDFUNC

FUNC RENDER_TEXT()
  INT S = GETW()/14
  BEGIN_SPRITES()
  FOR (INT I=0;I<14;I++)
    PRINT(w2[I],I*S-GETW()/2+4, 0+40)
    PRINT(w1[I],I*S-GETW()/2+4, -GETH()/2+40)
    IF (I!=2 AND I!=6 AND I!=9 AND I!=13) THEN
      PRINT(b2[I],I*S-GETW()/2+4+S/2, 0+40+GETH()/4)
      PRINT(b1[I],I*S-GETW()/2+4+S/2, -GETH()/2+40+GETH()/4)
    ENDIF
  NEXT
  END_SPRITES()
ENDFUNC
...
```



Теперь хорошо бы подсвечивать нажимаемые кнопки.

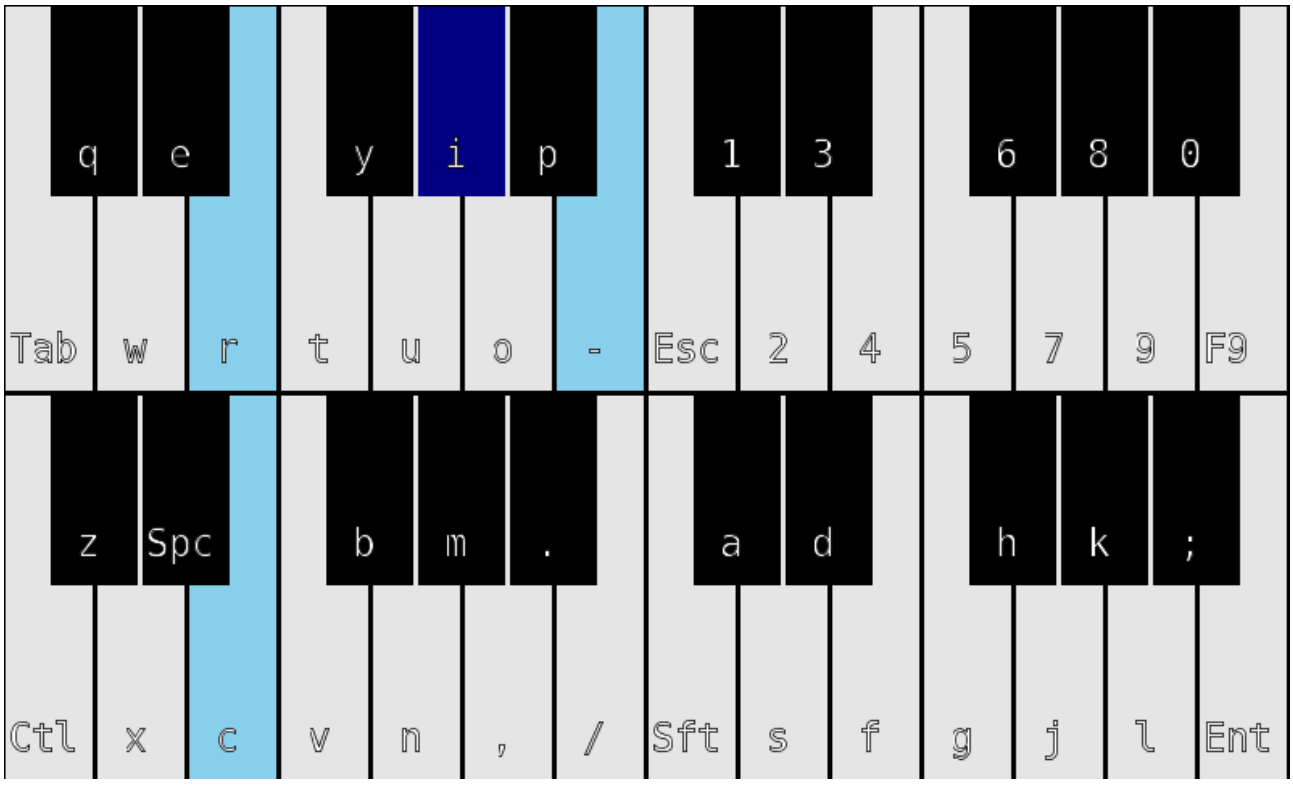
Записываем соотношение позиций и индексов:

```
INT[] posW1 = { 0, 2, 4, 5, 7, 9, 11, 12, 14, 16, 17, 19, 21, 23 }
INT[] posW2 = { 24, 26, 28, 29, 31, 33, 35, 36, 38, 40, 41, 43, 45, 47 }
INT[] posB1 = { 1, 3, -1, 6, 8, 10, -1, 13, 15, -1, 18, 20, 22 }
INT[] posB2 = { 1+24, 3+24, -1, 6+24, 8+24, 10+24, -1, 13+24, 15+24, -1, 18+24, 20+24, 22+24 }
```

Новая функция RENDER выглядит так:

```
FUNC RENDER()
  CLS(BLACK)
  SHAPE_FRECTS()
  INT S = GETW()/14
  FOR (INT I=0;I<14;I++)
    SETCOLOR(SMOKE)
    IF (ISPLAYING[posW2[I]]) SETCOLOR(SKY)
    FRECT(I*S-GETW()/2+2, 0+2,          S-3, GETH()/2-3)
    SETCOLOR(SMOKE)
    IF (ISPLAYING[posW1[I]]) SETCOLOR(SKY)
    FRECT(I*S-GETW()/2+2, -GETH()/2+2, S-3, GETH()/2-3)
  NEXT
  FOR (INT I=0;I<14;I++)
    IF (I!=2 AND I!=6 AND I!=9 AND I!=13) THEN
    SETCOLOR(BLACK)
    IF (ISPLAYING[posB2[I]]) SETCOLOR(NAVY)
    FRECT(I*S-GETW()/2+2+S/2, 0+2+GETH()/4,          S-3, GETH()/4-3)
    SETCOLOR(BLACK)
    IF (ISPLAYING[posB1[I]]) SETCOLOR(NAVY)
    FRECT(I*S-GETW()/2+2+S/2, -GETH()/2+2+GETH()/4, S-3, GETH()/4-3)
    ENDIF
  NEXT
  END_SHAPE()
ENDFUNC
```

Готово:



С клавиатурой покончено. Перейдем к мыши/пальцам:

```
FUNC TOUCHDOWN(INT x, INT y, INT ptr, INT btn) RETURNS BOOL
  INT IND = DETERM_IND(x,y)
  IF (IND!=-1) PIANOUP(IND)
  IF (IND!=-1) PIANODOWN(IND)
  RETURN TRUE
ENDFUNC

FUNC TOUCHUP(INT x, INT y, INT ptr, INT btn) RETURNS BOOL
  INT IND = DETERM_IND(x,y)
  IF (IND!=-1) PIANOUP(IND)
  RETURN TRUE
ENDFUNC
```

Функция DETERM_IND использует алгоритм, похожий на алгоритм затенения нажатых клавиш из функции RENDER:

```
FUNC DETERM_IND(INT x, INT y) RETURNS INT
  //преобразовываем координаты курсора в координаты графики (0,0 в центре окна):
  x -= GETW()/2
  y -= GETH()/2
  y = -y
  INT S = GETW()/14
  INT IND = -1
  FOR (INT I=0;I<14;I++)
    IF (x>I*S-GETW()/2+2 AND y>0+2 AND _
_       x<I*S-GETW()/2+2 + S-3 AND y<0+2+GETH()/2-3) THEN
      IND = posW2[I]
    ENDIF
      IF (x>I*S-GETW()/2+2 AND y>-GETH()/2+2 AND _
_       x<I*S-GETW()/2+2 + S-3 AND y<-GETH()/2+2+GETH()/2-3) THEN
      IND = posW1[I]
    ENDIF

      IF (I==2 OR I==6 OR I==9 OR I==13) CONTINUE

      IF (x>I*S-GETW()/2+2+S/2 AND y>0+2+GETH()/4 AND _
_       x<I*S-GETW()/2+2+S/2 + S-3 AND y<0+2+GETH()/4+GETH()/4-3) THEN
      IND = posB2[I]
    ENDIF
    IF (x>I*S-GETW()/2+2+S/2 AND y>-GETH()/2+2+GETH()/4 AND _
_       x<I*S-GETW()/2+2+S/2 + S-3 AND y<-GETH()/2+2+GETH()/4+GETH()/4-3) THEN
      IND = posB1[I]
    ENDIF

      IF (IND!=-1) BREAK
  NEXT
  RETURN IND
ENDFUNC
```

Проверяем: на PC и на Android работает превосходно. Можно зажать и не отпуская сдвинуть мышь/палец, чтобы нота звучала дольше. Но тут и минус — клавиша остаётся закрашенной пока её не нажмёшь снова. Одним из вариантов решения является запись времени нажатия и последующие проверки. Создадим для этого глобальный массив:

```
LONG[] STARTED = NEW LONG[48]
```

Изменим PIANODOWN:

```
FUNC PIANODOWN(INT I)
  IF (ISPLAYING[I]) RETURN
  INT On = (int)Math.floor(I/12)
  INT Nn = I - On*12
  FLOAT Pt = (float)Math.pow(Math.pow(2,Nn), 1.0f/12.0f)
  ID[I] = O[On].PLAY(1,Pt,0)
  ISPLAYING[I] = TRUE
  STARTED[I] = GETTIME() /* время в миллисекундах */
ENDFUNC
```

Изменим функцию RENDER:

Все
IF (ISPLAYING[pos[I]]) SETCOLOR(*)*
заменим на:
IF (ISPLAYING[pos[I]] **AND GETTIME()-STARTED[pos*[I]]<NOTELEN**) SETCOLOR(*)*

NOTELEN у меня = 1750, хоть звук и имеет длину 2 секунды, но в конце его почти не слышно.

Полный проект вы найдете в *C:/jdummy2d/tutorials/VIMUI.zip*