```
(in-package :scratch)

(defparameter *scale-names* '(c c. cis des des. d d. dis es es. e e. eis f f. fis ges ges. g g. g
is as as. a a. ais bes bes. b b. bis))

(defparameter *dict-name-pitch* (loop for name in *scale-names*
                                      for i from 1
                                      collect (cons name i)))

(defparameter *dict-pitch-key* '((0 . 0) (1 . 1) (2 . 2) (3 . 3) (4 . 4) (5 . 6) (6 . 7) (7 . 8)
(8 . 9) (9 . 11) (10 . 12) (11 . 13) (12 . 14) (13 . 15) (14 . 16) (15 . 17) (16 . 18) (17 . 20)
(18 . 21) (19 . 22) (20 . 23) (21 . 24) (22 . 26) (23 . 27) (24 . 28) (25 . 29) (26 . 30) (27 . 3
2) (28 . 33) (29 . 34) (30 . 35) (31 . 0)))

(defparameter *dict-interval-pitch* '((unisono . 0) (diesis . 1) (diesis-minore . 1) (diesis-magg
iore . 2) (semitono-minore . 2) (semitono-maggiore . 3) (tono-minore . 4) (tono . 5) (tono-maggio
re . 6) (terza-minima . 7) (terza-minore . 8) (terza-piu-di-minore . 9) (terza-maggiore . 10) (te
rza-piu-di-maggiore . 11) (quarta-minima . 12) (quarta . 13) (piu-di-quarta . 14) (tritono . 15)
(quinta-imperfetta . 16) (quinta-piu-di-imperfetta . 17) (quinta . 18) (piu-di-quinta . 19) (sett
ima-naturale . 26) (sesta-minore . 21) (sesta-maggiore . 23) (ottava . 31)))

(defun unify-pitch (pitch)
  (cond ((< pitch 1) (unify-pitch (+ pitch 31)))
        ((> pitch 31) (unify-pitch (- pitch 31)))
        (t pitch)))

(defun name->pitch (name)
  (cdr (assoc name *dict-name-pitch*)))

(defun pitch->name (pitch)
  (car (find (unify-pitch pitch) *dict-name-pitch* :key #'cdr)))

(defun name->key (name)
  (cdr (assoc (name->pitch name) *dict-pitch-key*)))

(defun pitch->key (pitch)
  (multiple-value-bind (octave pitch-class) (floor pitch 31)
    (+ (* 36 octave) (cdr (assoc pitch-class *dict-pitch-key*)))))

(defun interval->pitch (interval)
  (cdr (assoc interval *dict-interval-pitch*)))

(defun pitch->interval (pitch)
  (car (find (unify-pitch pitch) *dict-interval-pitch* :key #'cdr)))




(defun without-last (lst)
    (reverse (cdr (reverse lst))))

(defun permutate (lst)
  (cons (car (last lst)) (without-last lst))
  ;(append (rest lst) (list (first lst)))
  )

(defun rearrange-list (lst selector)
  (let ((index (position selector lst)))
    (if index
        (append (nthcdr index lst) (subseq lst 0 index))
        lst)))

(defmacro make-rotator (data)
  `(let ((lst ,data))
     #'(lambda (&optional selector)
         (if selector
             (setf lst (rearrange-list lst selector))
             (setf lst (permutate lst)))
         (first lst))))

(defparameter *limit-rotator* (make-rotator '(limit-2 limit-3 limit-5 limit-7)))
```

```lisp
(defparameter *limit* (funcall *limit-rotator*))

(defun next-limit (&optional selector)
  (setf *limit* (funcall *limit-rotator* selector)))

(defparameter *quality-rotator* (make-rotator '(consonant dissonant)))

(defparameter *quality* (funcall *quality-rotator*))

(defun next-quality (&optional selector)
  (setf *quality* (funcall *quality-rotator* selector)))

(defparameter *model-generator*
  (let ((counter 0))
    #'(lambda (&optional next)
        (let* ((models '((limit-2
                          (consonant
                           ((ottava)
                            (ottava)
                            (ottava)
                            (ottava)))
                          (dissonant
                           ((tritono ottava))))
                         (limit-3
                          (consonant
                           ((quinta ottava)
                            (quinta ottava)
                            (quinta ottava)
                            (quinta ottava))
                           ((quarta ottava)
                            (quarta ottava)
                            (quarta ottava)
                            (quarta ottava))
                           ((quinta ottava)
                            (quarta ottava)
                            (quinta ottava)
                            (quarta ottava)))
                          (dissonant
                           ((tritono settima-naturale)
                            (tritono settima-naturale)
                            (tritono settima-naturale)
                            (tritono settima-naturale))))
                         (limit-5
                          (consonant
                           ((terza-minore quinta ottava)
                            (terza-maggiore sesta-maggiore ottava)
                            (terza-maggiore sesta-maggiore)
                            (terza-maggiore quinta ottava))
                           ((terza-maggiore quinta ottava)
                            (terza-minore sesta-minore ottava)
                            (terza-maggiore sesta-maggiore ottava)
                            (terza-minore quinta ottava))
                           ((terza-minore quinta ottava)
                            (terza-minore sesta-minore ottava)
                            (terza-minore sesta-maggiore ottava)
                            (terza-maggiore quinta ottava))
                           ((terza-minore sesta-minore ottava)
                            (terza-maggiore sesta-maggiore ottava)
                            (terza-maggiore sesta-maggiore ottava)
                            (terza-maggiore quinta ottava)))
                          (dissonant
                           ((tono-maggiore terza-piu-di-minore quinta-imperfetta)
                            (tono-maggiore terza-piu-di-minore quinta-imperfetta)
                            (tono-maggiore terza-piu-di-minore quinta-imperfetta)
                            (tono-maggiore terza-piu-di-minore quinta-imperfetta))))
                         (limit-7
                          (consonant
                           ((terza-maggiore quinta settima-naturale ottava)
                            (terza-maggiore quinta settima-naturale ottava)
                            (terza-maggiore quinta settima-naturale ottava)
                            (terza-maggiore quinta settima-naturale ottava))
                           ((terza-minore quinta ottava)
```

```
                          (terza-maggiore quinta settima-naturale)
                          (terza-maggiore quinta settima-naturale)
                          (terza-maggiore quinta ottava))
                         ((terza-maggiore quinta ottava)
                          (terza-minore sesta-maggiore ottava)
                          (terza-maggiore sesta-maggiore ottava)
                          (terza-maggiore quinta settima-naturale ottava)))
                        (dissonant
                         ((tono-minore tono-maggiore quarta-minima)
                          (tono-minore tono-maggiore quarta-minima)
                          (tono-minore tono-maggiore quarta-minima)
                          (tono-minore tono-maggiore quarta-minima))))))
              (pick (cdr (assoc *quality* (cdr (assoc *limit* models)))))
              (len (length pick)))
          (when next (incf counter))
          (when (>= counter len) (setf counter 0))
          (nth counter pick)))))

(defun next-model ()
  (funcall *model-generator* t))

(defparameter *genere-rotator* (make-rotator '(diatonico cromatico enarmonico)))

(defparameter *quarta-rotator* (make-rotator '(prima seconda terza)))

(defun next-genus (&optional selector)
  (setf *genere* (funcall *genere-rotator* selector)))

(defun next-quarta (&optional selector)
  (setf *quarta* (funcall *quarta-rotator* selector)))

(defparameter *genere* (next-genus))

(defparameter *quarta* (next-quarta))


(defparameter *tetrachord-generator*
  #'(lambda ()
      (let ((quarte '((diatonico
                       (prima . (tono semitono-maggiore tono))
                       (seconda . (tono tono semitono-maggiore))
                       (terza . (semitono-maggiore tono tono)))
                      (cromatico
                       (prima . (semitono-minore terza-minore semitono-maggiore))
                       (seconda . (semitono-maggiore semitono-minore terza-minore))
                       (terza . (terza-minore semitono-minore semitono-maggiore)))
                      (enarmonico
                       (prima . (diesis-maggiore terza-maggiore diesis-minore))
                       (seconda . (diesis-minore terza-maggiore diesis-maggiore))
                       (terza . (terza-maggiore diesis-maggiore diesis-minore))))))
        (cdr (assoc *quarta* (cdr (assoc *genere* quarte)))))))
```