```
NASLOV    PUSH BC
          CALL PIXEL_ADD
          POP BC
          RET
```

NASLOV nam bo posredoval naslov določenega položaja, koordinate pa bodo pri tem ostale nedotaknjene.

Zdaj pa k delu. Lotimo se najprej glavne zanke:

```
URA       LD  A, (23672)     ;vrednost "n", ki jo bomo določili
          CP  n              ;kasneje, uravnava hitrost ure.
          JR  C, TEST        ;Če še ni presežena vrednost n,
          PUSH  BC           ;skoči v zanko TEST. Sicer shrani
          LD  BC, (VISURE)   ;koordinate Krpana.
```

Z imeni bomo označevali naslove rezerviranih celic. Zloga na naslovu VISURE bosta vsebovala koordinate stolpca ure. Ker bo stolpec na levi strani zaslona, bo njegova oddaljenost od levega roba (koordinata x) v registru C vedno 0, koordinata y, shranjena v registru B pa bo pravzaprav višina stolpca. Ta bo na začetku 175 in se bo zmanjševala. Ko bo dosegla 0, bo igre konec.

```
          CALL  NASLOV       ;Izračunaj naslov.
          LD  (HL), 129      ;Izbriši gornjo vrstico, vendar naj
                             ;stranici (bita 0 in 7) ostaneta.
          LD  (23672), A     ;Uri daj vrednost 0 (register A bo
                             ;imel po klicu NASLOVa vrednost 0,
                             ;ker se bodo koordinate vedno na-
                             ;našale na levi gornji bit likov).
          DEC  B             ;Znižaj višino stolpca ure, ter
          LD  (VISURE), BC   ;shrani njegove koordinate.
          POP  BC            ;S sklada vzemi podatke o položaju
                             ;junaka. Zadnji ukaz, ki je vplival
                             ;na zastavice, je bil DEC B:
          JR  NZ,'TEST       ;če višina stolpca še ni 0, pojdi v
          LD  BC, (TOCKE)    ;zanko TEST, sicer naloži v BC do-
          RET                ;sežene točke in se vrni v basic.
TEST      PUSH  BC           ;Shrani koordinate.
          CALL  KEY_SCAN     ;Preleti tipkovnico. KEY_SCAN vrne
          POP  BC            ;v registru E kod pritisnjene tipke
          LD  A, E           ;(glej poglavje o rabi tipkovnice).
```


Zdaj bomo preverjali, ali je bila pritisnjena katera od tipk za premik. Naj služita za premik levo in desno tipki O in P, za gor in dol pa Q in A. Seveda lahko izberete tudi druge tipke, le spremeniti boste morali vrednosti, ki slede.

```
          CP  #25            ;Če je pritisnjena tipka Q...
          CALL  Z, GOR       ;       ...kliči podprogram GOR.
          CP  #22            ;    -"-                P...
          CALL  Z, DESNO     ;       ...    -"-         DESNO.
          CP  #1A            ;    -"-                O...
          CALL  Z, LEVO      ;       ...    -"-         LEVO.
          CP  #26            ;    -"-                A...
          CALL  Z, DOL       ;       ...    -"-         DOL.
          CALL  Z, METULJ    ;Kliči podprogram za premik metulja
          LD  A, (KLJUC)     ;V zlogu KLJUČ shranimo število že
          CP  n              ;pobranih ključev. Če so pobrani že
          CALL  Z, OBRAC     ;vsi (na začetku je na zaslonu n
                             ;ključev) kliči podprogram za obra-
                             ;čun točk.
          JR  URA            ;Ponovi glavno zanko.
```

Če bo igralcu uspelo pobrati vse ključe, preden se bo čas iztekel, se bo vrednost doseženih točk za vse ključe pomnožila s preostankom časa. To bo opravil podprogram OBRAC.

Glavna zanka je tako pripravljena. Nadaljujmo kar po vrsti s podprogrami, ki jih bomo potrebovali. Najprej GOR:

```
GOR       LD  A, 175         ;Če je lik na zgornjem robu zaslo-
          CP  B              ;na, ne more več navzgor. Zato se v
          RET  Z             ;tem primeru vrni v glavno zanko.
```

Zdaj moramo preveriti, ali ni nad junakovo glavo metulj ali ključ. V ta namen je najprimernejši tretji (oz. četrti, če štejemo od roba klobuka) zlog nad junakovo glavo. Če je tam metulj, bo imel vrednost 1 četrti bit zloga, če bo tam ključ, bo "prižgan" bit 0. To lahko strnemo v kratek podprogram, ki ga bomo dodali na koncu. Večkrat nam bo prišel prav. Imenujmo ga BIT40:

```
BIT40     BIT  4, (HL)       ;Če je tu metulj, kliči podprogram,
          CALL  NZ, KONEC    ;ki bo zaključil igro;
          BIT  0, (HL)       ;če je tu ključ, kliči podprogram,
          CALL  NZ, ZADET    ;ki bo obračunal zadetek
          RET                ;in se vrni.
```