How to represent data?

- Cards
  - Each card is represented by a number (0 to 39)

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39

- Arrays hold information for each card
  - values[40]: value of the card (0-9)
  - matched[40]: whether the card was matched (0,1)

Variables to keep in Registers

- Game state
  - Marked card (card with red box)
  - Number of cards showing
  - First selected card
  - Second selected card
  - Golden card
  - Number of unmatched cards
- Game clock
  - Minutes left in game clock
  - Tens of seconds left in game clock
  - Second left in game clock
- Timer Events (in milliseconds)
  - Time of last update of game clock
  - Time of selection of second card
  - Time of last selection of golden card

How to use registers?

- Each variable is stored in a register
  - Valid throughout the program (global)
  - Even inside function calls

- One possible mapping
  - $t0 - $t5: temp registers
  - $s0 - $s1: saved registers
  - $t6 - $t9, $s2 - $s7 and $k0, $k1: global variables

Main Loop

```
#initialize everything
timer:
    #check for key press
    #check for clock timer
    #check for pair card timer
    #check golden card timer
    #check exit conditions
j time
```

How do timers work?

- System call 30 returns time in milliseconds

- Initialize saved time to current time
- Inside the main loop
  - read current time
  - if current time > saved time + 1000

    timer expired

    update saved time

Useful functions

- *drawCard(int card, int font)*
  - *card* is number between 0 and 39
  - *font* is index into array of fonts (0 – 15)
  - use array to store the location to draw each card
  - can be extended to draw clock (*card* >= 40)
- *drawMark(int card, int color)*
  - draws or clears box around card
  - call after redrawing the marked card
- *rand40()*
  - return random number between 0 and 39
  - used for selecting golden card