Upsets

A challenging, yet very easy-to-learn educational brain-game based on the concepts of set theory.

Set theory is important in a structured approach to mathematics, and has consequently become an integral part of many computer programming languages (e.g., Pascal, C, etc.). The *Upsets* program is an educational game that offers you an opportunity to learn about and practice the basics of set theory, and then apply your understanding in either a solitaire or two-person game. If you are new to the concepts, terms, and symbols used in set theory, first see our sidebar "Getting Set-tled For Upsets"

The game is designed around a deck of cards with four different colored dots. The four colors on the cards are the elements used to describe which cards are in the set. For example, a set may comprise all cards with a blue dot and be identified by the single element **BLUE**. In Figure 1, this would include cards 1, 3, and 4. Another set could be all cards which have either blue dot or a red dot using the union function. The set expression would be:

BLUE ∪ RED

This set would include cards 1, 3, 4, 5, and 6 in Figure 1.

Another set could be described using the intersection function to include only cards having both a blue dot and a red dot. The expression for this set would be:

BLUE ∩ RED

This set would include cards 1 and 3 in Figure 1.

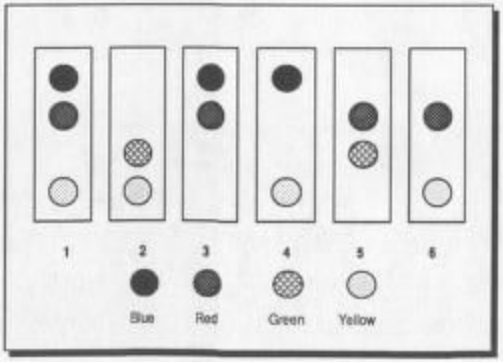
When the minus sign appears in a set expression, it will indicate the exclusion of one or more colors from the set being defined. An example using the difference connective is:

BLUE – RED

In this example, only card 4 would be selected from Figure 1. This means that you take the set of all cards with a blue dot, then remove any cards having a red dot.

Upsets displays up to four colors on each of the cards. You may use any or all of these four colors with any of the three connectives to form a set expression or set name. The set you describe could include all, some, or none of the cards—depending on the colors, connectives, and position of the parentheses.

Figure 1.
A sample set of "cards" in a game of Upsets. Note that the colors of the cards in your computer's version may be different from those indicated below.



Get Ready, Get Set . . .

The first menu in the program allows you to select one of three different games:

```
(1) Practice
(2) Solitaire
(3) Challenge
```

When you select option 1 for a Practice session, the computer does not keep score, or limit the game length. In this mode, you are able to create set names and watch the results. This provides an excellent way to discover the logic rules that govern set names. The computer places the cards on the screen randomly from a "deck" of fifteen cards. Note that a blank card is never one of the cards in the game, because it would represent the "null" set. According to the rules of set theory, the null set is included in *all* sets and, because a blank card would *always* be selected, it is not used in the game of *Upsets*.

Option 2, Solitaire, lets you play *Upsets* by yourself. Here, you express sets that include as many cards as possible on your side of the screen (the left), while selecting as few cards as possible from the right side of the screen. You make all the plays in Solitaire, but the computer keeps score, giving you 10 points for each card included in the set on your side of the screen, and giving your non-playing "opponent" 10 points for each card that your expression selects on the right. You can win the game in two ways: (1) by selecting *all* of your cards and *fewer* than all of your opponent's cards in any one turn, or (2) by having the highest score after 10 turns.

In Option 3, Challenge, two players take turns defining the set name. One player uses the cards on the left side of the screen, while the other uses the cards on the right side. After each player takes a turn, the computer evaluates the new set, and each player receives 10 points for each card in the set.

After you select a game option, a prompt asks you to enter the players' names (one or two names, depending on the game type selected). Next, you select the number of cards you would like to work with. You can have from 3 to 6 cards displayed on each side of the screen. The more cards you select, the more complicated the game, and the greater the possible score.

Finally, a prompt asks how many connectives (from 1 to 3) you would like to use in the set name.

```
Select Level Of Expression:
1) One Connective
2) Two Connectives
3) Three Connectives
```

Getting Set-tled For Upsets

Set theory is the branch of mathematics concerned with the definition and relationships between the grouping of objects, numbers . . . and well, *anything*. We say that a particular set "contains elements," very much like a box contains items. For example, a standard deck of 52 playing cards is a set of all the cards after you discard the "set" consisting of jokers and any other non-face or number cards. We can define a grouping or set that contains all the hearts. We say this set contains all cards with red hearts on them. We may do the same for cards having the red diamond, the black spade, and the black club.

We can define other sets by using "connectives" to form relationships between different sets. The basic connectives (those used in our *Upsets* program) are: **union**, **intersection**, and **difference**. Figures A, B, and C show three diagrams (called Venn diagrams) that graphically demonstrate how these functions work. The two circles, X and Y, represent two different sets. The black areas show the set created by the function.

The **union** function (see Figure A) is used to form a larger set containing elements of two smaller sets. This function is expressed by the ∩ symbol and is similar to the OR function used in BASIC. Taking our deck of cards again, we can define a set containing all red cards as being the union of hearts and diamonds, or expressed symbolically:

red cards = hearts ∪ diamonds

Obviously, this function combines two of our first four sets into another set of *all* red cards.

The **intersection** function is expressed by the ∩ symbol and is similar to the BASIC **AND** function (see Figure B). It creates a set by selecting elements that are contained in both sets. The fewer elements that are *shared*, the more *exclusive* the set becomes; e.g., the set of cards formed by the intersection of all aces and all red cards is a set of only two cards—the ace of hearts and the ace of diamonds.

red aces=(red cards) ∩ (aces)

The **difference** function represented by a minus sign allows for the exclusion of certain elements from a set (see Figure C). Using the deck of 52 cards, we can form a selection of all cards not having numbers—the set of all face cards and aces. Symbolically we write:

(face cards ∪ aces) = (all cards) – (cards with numbers)

These functions can be combined in set expressions, just like mathematical expressions, and parentheses are used to designate the order of evaluation.

Venn Diagrams

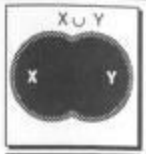


Figure A.
This diagram represents the union function—similar to the BASIC OR function.

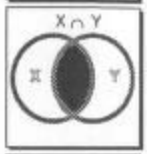


Figure B
The intersection function is like the BASIC AND statement.

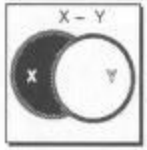


Figure C.
Here, the difference function excludes elements from set X that also occur in set Y.

This will determine the length and complexity of the set name. With only one connective, you are limited to simple set expressions like those in the examples above (e.g., BLUE ∩ RED). And with only one connective you may use any two (but *only* two) of the four colors in any expression.

Two connectives allow a more complex name such as: (BLUE ∪ RED) ∩ GREEN. Selecting three connectives expands the set name to the most complex level available in *Upsets*. It also allows the switching of parentheses to change the order in which the expression is evaluated—expanding the number of strategic tools available.

The Playing Screen

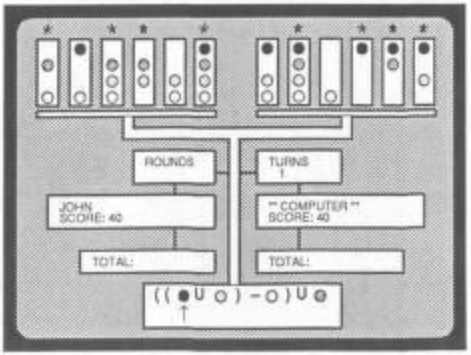
After selecting all of the options described, the playing screen appears (see Figure 2). Depending upon which game you've chosen, 3 to 12 cards are displayed across the top of the screen. No two cards are ever alike. When the computer evaluates a set expression, it indicates which cards are in the designated set by placing a mark above each of the cards that satisfies the set expression.

Below the cards are 2 small boxes labeled **Round** and **Turns**. The box labeled **Round** is activated only when you play the Challenge option. In this option, you play 4 rounds. A round is over when one player has created a set that includes all of the cards on either one or both sides of the screen, or after 20 turns. The player with the highest total score after 4 rounds wins the game. In rounds 1 and 3, the player on the left side of the screen goes first; in rounds 2 and 4, the player on the right side goes first.

The **Turns** box indicates the current turn, and is active only during the Solitaire or Challenge options. In Solitaire, the length of a game is 10 turns. In Challenge, the length of each round is 20 turns. A turn is over after *both* players have made a change to the current set expression.

Directly below the **Round** and **Turns** boxes are the 2 **Score** boxes each labeled with the player's name. In Solitaire, the score box on the left side always holds the player's score; the score box on the right contains the score of your non-playing opponent. In a Challenge game, the name of each player appears in the box with that player's score. The score displayed here is only for the current round. At the start of each round, each **Score** is added to the associated **Total** box, and the **Score** box is reset to zero.

Figure 2.
This simulated screen depicts a solitaire game where the set expression describes 4 cards from each side.



Express Yourself

In the bottom center portion of the screen is the current set name. At the beginning of a game or round, the name is empty except for parentheses:

((. . .) . .) . .

Here, we are using periods to indicate what appears on the screen as empty character positions. Below this empty set name is a cursor. The current player moves the cursor to any position in the set name and, by pressing the [SPACE BAR], selects *one* legal connective symbol or color dot for that position in the name. To toggle through all of the legal symbols for that position, simply continue pressing the [SPACE BAR]. Once you have displayed the desired symbol, both your selection and your turn are completed by pressing [ENTER] or [RETURN]. If you move away from a position without pressing [ENTER] or [RETURN], the original symbol appearing at that location will return.

Before a set name can be evaluated by the computer, all parts of it must be filled in. It needs to contain the number of connectives you chose at the beginning of the game *and* enough colored dots to evaluate the connectives. It may take several turns before the set name is completed for evaluation. A valid name with 1 connective may look like this:

((• ∩ •) . .) . .

(Periods represent spaces, the • character represents a colored dot, and ∩ ∪ – are the connectives.)

A set name with 2 connectives may look like this:

((• ∩ •) – •) . .

A set name with all 3 connectives would look like this:

((• ∩ •) – •) ∪ •

A set name can be filled out in any order, with each player limited to setting either a single dot color or a connective during a given turn. But until the set name is completely filled out and evaluated at least once, players can't *change* any of the connectives or dots. If all 3 connectives are being used, players also have the option of changing the order of parentheses.

Order Of Evaluation

The parentheses used in a set name, as described earlier, indicate the order the connectives are to be evaluated. The computer always evaluates that part of the expression within the innermost parentheses *first*, working its way outward to the outermost connective not contained within parentheses; or, it *begins* at the left and evaluates towards the right. The default position for parentheses will always appear as in the first example shown above, which causes connectives to be evaluated from left to right. The following examples will illustrate the effect parentheses have on a 3-connective set name:

((B ∩ R) – G) ∪ Y

(Here, the letters B, R, G, and Y represent the colors Blue, Red, Green, and Yellow.) This set would include any cards containing either (1) a yellow dot, or (2) both a blue dot and a red dot, but not a green dot.

(B ∩ R) – (G ∪ Y)

This set name is the same as the one above—*except* for the position of the parentheses. It now includes cards that contain both blue and red, but not green or yellow. Because there are only four colors, and no two cards can be identical, the set named above can include only one card.

To alter parentheses, just press **P**, and the parentheses change. This allows you to switch the parentheses to any of three possible settings; the set name at the bottom of the screen will be rewritten showing the new arrangement. Once you have the parentheses set to the desired arrangement, press [ENTER] or [RETURN] to fix the setting. This completes a player's turn and begins reevaluation of the set name and an updating of the score accordingly. If you decide that you don't want to change the position of the parentheses before you press [RETURN] or [ENTER], you can return the name to its original form by simply moving the cursor. This allows you to continue your turn to make a different change to the set name. You may make only *one* permanent change to the set name during your turn.

To quit, press [ESC] (or ← on the C-64 and [FCTN] 9 on the TI-99/4A). Whether you wish to simply introduce yourself to set theory, or use your talents of logic to develop effective game strategies, *Upsets* is definitely a real setup for challenging fun.

CONTROL CAPSULE

Upsets

KEY	FUNCTION
SPACE BAR	Change symbol
ENTER or RETURN	Accept symbol in expression
P	Alter parentheses
ESC*	Return to Main Menu

*FCTN 9 on TI-99/4A, ← on C-64