The information contained herein is for the use of employees of Bell Laboratories and is not for publication. (See GEI 13.9-3)

Title- **An Object Editor for a Real Time Animation Processor**

Date- **October 1, 1980**

TM- **81-11356-13**

Other Keywords- **Computer Graphics**
Graphic Input

Author
S.P. Ressler

Location
MH 7A-304

Extension
4377

Charging Case- **011170-0101**
Filing Case- **39394**

ABSTRACT

The object editor allows one to use the real time capabilities of the animation processor to create and/or modify graphic objects. These objects can subsequently be used in games, or other application programs. The editor allows one to change the definition of objects which are being processed and displayed by the animation processor in an interactive manner. One is presented with a "rubber solid", which is completely manipulable. Objects, or pieces thereof, which appear to be quite solid can be stretched and/or squashed in real time.

The animation processor does hidden surface calculations in real time while the object editor modifies the description of the object, resulting in a dynamically changing object. This editor is not meant to be a production line system for interactive object design, rather it is a research tool for exploring the possibilities of interactive object manipulation.

Pages Text 14	Other 5	Total 19
No. Figures 4	No. Tables 0	No. Refs. 2

SEE REVERSE SIDE FOR DISTRIBUTION LIST

DISTRIBUTION
(REFER GEI 13.9-3)

COMPLETE MEMORANDUM TO

CORRESPONDENCE FILES

OFFICIAL FILE COPY
PLUS ONE COPY FOR
EACH ADDITIONAL FILING
CASE REFERENCED

DATE FILE COPY
(FORM E-1328)

10 REFERENCE COPIES

<ALLES,H G
<BARON,ROBERT V
BERGLAND,G D
<CHAMBERS,J M
<CHRISTENSEN,C
CLOGSTON,ALBERT M
<GIORDANO,PHILIP P
+HANNAY,N B
<KEESE,W M
KOGELNIK,H
LIEN,Y EDMUND
LIMB,JOHN O
LUCKY,R W
<MARKY,GERALDINE A
<MC DONALD,B S
+MC DONNELL,JOHN P
MILLER,STEWART E
<MORGAN,SAMUEL P
NETRAVALI,A N
NINKE,WILLIAM H
PENZIAS,A A
REUDINK,D O
<ROBERTS,CHARLES S
SLICHTER,W P
<TEWKSBURY,S K
+VYSSOTSKY,V A
WALDEN,JOHN S
27 NAMES

COVER SHEET ONLY TO

CORRESPONDENCE FILES

4 COPIES PLUS ONE
COPY FOR EACH FILING
CASE

AAGESEN,JOHN
ABOWD,MARYPAT
ABRAHAM,S A
ACKERMAN,A FRANK
AHO,ALFRED V
AHRENS,RAINER B
<AHUJA,SUDHIR R
ALBRECHT,PAUL
ALCALAY,D
ALDEBOL,ANGELA L
ALGER,M J

COVER SHEET ONLY TO

ALLEN,R C
ALLISON,C E,JR
AMRON,IRVING
ANDERSON,DAVID J
ANDERSON,KATHRYN J
ANDERSON,MILTON M
ARNOLD,GEORGE W
ARNOLD,PHYLLIS A
ARNOLD,SAMUEL L
ARNOLD,THOMAS F
ASELTINE,EDWARD G
ASTHANA,ABHAYA
ATAL,BISHNU S
AUSTIN,B J
BAKER,BRENDA S
BAKER,DONN
BAKER,GREGORY
BAKER,MITCHELL B
BALENSON,CHRISTINE M
BALLARD,E D,JR
BARBATO,ROBERT R
BARNES,CYNTHIA A
BAROFSKY,ALLEN
BARZ,H
BAUER,HELEN A
BAUMANN,JAMES L
<BAXTER,LESLIE A
BEACHY,MILTON
BEBLO,WILLIAM
BECKER,RICHARD A
BEHRMAN,J S
BENCO,DAVID S
BENISCH,JEAN
BENJAMIN,PHILIP M
BENNETT,RAYMOND W
BENNETT,RICHARD L
BENNETT,WILLIAM C
BERGH,A A
BERK,DONALD A
BERLINER,EDWARD F
BERNOSKE,BEVERLY G
BERNSTEIN,L
BERZINS,ALEXANDER H
<BETHEA,C G
BICKFORD,NEIL B
BILLINGTON,MARJORIE J
BILOWOS,R M
BISHOP,J DANIEL
BISHOP,THOMAS P
BISHOP,VERONICA L
BITTNER,B B
BLAKE,BARRY A
BLAKE,GARY D
BLAZIER,S D
BLINN,J C
BLOSSER,PATRICK A
BLUM,MARION
BODEN,F J
BOGART,MILDRED
BOGART,THOMAS G
BOIVIE,RICHARD H
BONANNI,L E

COVER SHEET ONLY TO

BORGEL,JEAN-CLAUDE
BORKAR,SURESH R
BOULIN,D M
BOURNE,STEPHEN R
BOWYER,L RAY
BOYCE,W M
BOYER,PHYLLIS J
BOZA,L B
BRADLEY,M HELEN
BRAUNE,DAVID P
BRAUN,DAVID A
BRESLER,RENEE A
BRIGGS,GLORIA A
BROAD,MARTHA M
BROSS,JEFFREY D
BROVMAN,INNA
BROWN,ELLINGTON L
BROWN,H GORDON
BROWN,LAURENCE MC FEE
BROWN,W R
BROWN,W STANLEY
BROZ,G C
BRYANT,DAVID J
BUCHE,CHARLES M
BULLEY,R M
BURG,F M
BURIC,MILORAD R
BURKE,MICHAEL E
BUROFF,STEVEN J
BUSTAD,ERIC KENNETH
BUTLETT,DARRELL L
<BUTZIEN,PAUL E
BYRNE,EDWARD R
CAMPBELL,JERRY H
<CANADAY,RUDD H
CARAM,BRUCE
CARROLL,J DOUGLAS
CARTER,DONALD H
CARUSO,R E
CASPERS,BARBARA E
CASS,JAMES L
CASWELL,PETER F
CAVINESS,JOHN D
CHAI,D T
CHAMBERS,B C
CHANG,JIH JIE V
CHARLESON,LAWRENCE M
CHENG-QUISPE,ENRIQUE
CHENG,Y
CHEN,E
CHEN,HAN-TEE
CHEN,PING C
CHEN,STEPHEN
CHERRY,LORINDA L
CHEUNG,ROGER C
CHE,HER-DAW
CHILDS,CAROLYN
CHIN,AUGUSTIN Y
CHODROW,M M
CHONG,PHEE
CHRIST,C W,JR
CHUNG,MICHAEL

COVER SHEET ONLY TO

CHU,ALLAN Y
CHU,JEAN
CHU,SAI-CHEONG
CLAYTON,D P
CLINE,LAUREL M I
CLINE,RUSSELL R
CLONINGER,JOHN M,JR
COATES,KAREN E
COBEN,ROBERT M
COFFMAN,JAMES E
COHEN,AARON B
COHEN,H M
COHEN,RICHARD L
<COLE,LOUIS M
COLE,MARILYN O
COLLICOTT,R B
CONKLIN,DANIEL L
CONNERS,RONALD R
COOPER,ARTHUR E
COOPER,MICHAEL H
COREY,D N
COSTELLO,PETER E
COTTRELL,JENNIE L
COUGHRAN,W M,JR
<COURTNEY-PRATT,J S
COVINGTON,RALPH L
CRAGUN,JOAN
CRAIG,JOHN R
CRAIG,N E
CRISTOFOR,EUGENE
<CRUME,L L
CRUPI,JOSEPH A
CUMMINGS,MICHAEL
DALRYMPLE,FREDERICK L
DARDEN,GERALDINE C
DARYANANI,CAROL C
DATTISMAN,WILLIAM G,JR
DAVEY,DOUGLAS A
DAVISON,JOSEPH W
DAVIS,JEFFREY B
DAVIS,R DREW
DE BRULER,DENNIS L
DE FAZIO,M J
DE GRAAF,D A
DE TREVILLE,JOHN D
DEEM,G S
DENNIS,T M
DERR,MARCIA A
DEUTSCH,DAVID N
DICKMAN,BERNARD N
DIMARCO,ROBERT T
DINEEN,THOMAS J
DINSMORE,PETER T
DIPAOLO,MARICATHERINE R
DOLOTTA,T A
DONOFRIO,LOUIS,J
DOUGHTY,DENNIS O
DOWDEN,DOUGLAS C
DOWD,PATRICK G
DRAKE,LILLIAN
DRECKMAN,GEORGE T
DRUSEIKIS,FREDERICK C

COVER SHEET ONLY TO

D'ANDREA,LOUISE A
D'ANGELO,GREGORY J
DUCHARME,ROBERT LAWRENCE
DUFFY,F P
DUGGER,DONALD D
DUMAIS,VALERIE
DUNCANSON,ROBERT L
DUNKIN,PATRICIA
DVORAK,JOSEPH
DWYER,T J
EDELSON,DAVID
EIGEN,D J
<EILENBERGER,ROBERT L
<EITELBACH,DAVID L
ELDRIDGE,JOHN
ELLIS,DAVID J
ELY,T C
EMMOTT,J T
ENGLAR,BRUCE WYATT
EPLEY,ROBERT V
EPSTEIN,J M
ESSERMAN,ALAN R
ETHINGTON,RICHARD D
EVANS,MARK S
EVERHART,J R
FABISCH,M P
FABRICIUS,WAYNE N
FAIRCHILD,DAVID L
FASCIANO,V
FECHTER,ROBERT
FEDER,J
FERIDUN,K K
FETTERMAN,C H
FEUER,ALAN R
FEUSTER,I REED
FINKEL,LAWRENCE D
FISANICK,GEORGIA J
FISCHER,HERBERT B
FISHER,ALAN S
FISHER,DANA G
FISHMAN,DANIEL H
<FLANAGAN,J L
FLEMING,JAMES R
FORAN,LINDA L
FOSS,J W
FOUNTOUKIDIS,A
FOWLER,GLENN D
FOWLER,H EUGENE
FOWLKES,EDWARD B
FOX,PHYLLIS A
FOY,J C
FRANK,AMALIE J
FRASER,A G
FRASER,D L,JR
FREEMAN,K G
FREEMAN,MARTIN
FREEMAN,R DON
FRENCH,A F,JR
FRETWELL,L J,JR
FROST,H BONNELL
GABBE,JOHN D
GADENZ,RENATO N
...
857 TOTAL

+ NAMED BY AUTHOR > CITED AS REFERENCE < REQUESTED BY READER (NAMES WITHOUT PREFIX WERE SELECTED USING THE AUTHOR'S SUBJECT OR ORGANIZATIONAL SPECIFICATION AS GIVEN BELOW)

MERCURY SPECIFICATION...

COMPLETE MEMO TO
1135-DPH   113-DIR   111-EXD   113-EXD   115-EXD

COVER SHEET TO
COPLGR = COMPUTING/PROGRAMMING LANGUAGES/GRAPHICS

TM-81-11356-13   19 TOTAL PAGES

TO GET A COMPLETE COPY:

1. BE SURE YOUR CORRECT ADDRESS IS GIVEN ON THE OTHER SIDE.
2. FOLD THIS SHEET IN HALF WITH THIS SIDE OUT AND STAPLE.
3. CHECK THE ADDRESS OF YOUR LOCAL CORRESPONDENCE FILES
IF LISTED, OTHERWISE HO 1A-127. USE NO ENVELOPE.
4. INDICATE WHETHER MICROFICHE OR PAPER COPY IS DESIRED.

CORRESPONDENCE FILES

( ) HO 1A-127 ( ) AL 2A-150
( ) IH 7K-101 ( ) MV 1D-40
( ) WH 3E-204

PLEASE SEND ME A COMPLETE ( ) MICROFICHE ( ) PAPER COPY.

DISTRIBUTION CONTINUED
(REFER GEI 13.9-3)

COVER SHEET ONLY TO

GALE,WILLIAM A
GALLANT,R J
GAMORAN,S H
GANA,JORGE L
GARDNER,GLENN
GARST,BLAINE,JR
GEERS,I J,JR
GENTER,C ROY,II
<GENTLEMAN,KIRSTE L
GEORGEN,MICHAEL R
GEPNER,JAMES R
GEYLING,F T
GIBB,KENNETH R
GIBSON,J C
GIFFORD,WARREN S
<GILLETTE,DEAN
<GITHENS,J A
GITHENS,JAY L
<GLASSER,ALAN L
GLOVER,FREDERICK S
GLUCK,F G
GOEDDE,CURTIS P
GOGUEN,N H
GOLASEK,RUTH T
GOLDENBERG,H R
GOLDSTEIN,A JAY
GOMEZ,ANGEL
<GOODMAN,DAVID J
GOODMAN,JESSE D
GOODWIN,HEATHER M
GOPIN,ALAN
GORDON,GLEN W
GORDON,MOSHE B
<GORMAN,J E
GORTON,D R
GRAHAM,R L
GREENBAUM,HOWARD J
GREENLAW,R L
GREER,RICHARD LEE
GRIPPO,RONALD V
GROFF,R H
GROSS,ARTHUR G
GRUENWALD,JOHN
GRZELAKOWSKI,MAUREEN E
GUBITOSI,LOUIS H
GUERRA,JOSE A
GUEST,RONALD A
GUETA,ROBERT J
GUIDI,PIER V
GUMAER,H K
GUNDERMAN,R E
GUTTMAN,NEWMAN
GUYTON,ARTHUR
HAFER,E H
HAISCH,H F,JR
HALKIOTIS,PENELOPE D
HALLIN,THOMAS G
HALL,ANDREW D,JR
HALL,MILTON S,JR
HALL,WILLIAM G
HALPIN,T
HAMILTON,LINDA S
HAMILTON,PATRICIA A
HAMMOND,MARK
HAMMOND,RICHARD A
HANCOCK,J DAVID
<HANSELL,P E
HARRINGTON,J J
HARRINGTON,T DORSEY
HARRIS,F L
HARTMANN,R H
HARTMAN,DON W
HARTNETT,RONALD J
HARTWELL,WALTER T
HARUTA,K
<HASKELL,BARRY G
HAUSE,A DICKSON
HAWKINS,DONALD T
HAWKINS,RICHARD B
HAYDEN,CHARLES C
HAYWARD,HENRIETTA M
HEIDER,BRUCE H
HERGENHAN,C B
HEJMAN,ARIEL
HICKEY,JOHN W

COVER SHEET ONLY TO

HILBERT,ROSS J
HILL,DWIGHT D
HITZEMAN,STANLEY P
HOCHBERG,GLENN A
HOLT,ROBERT H
HORNEMAN,GARY A
HORWEDEL,J H
HOUGHTON,THOMAS F
HOWARD,L A,JR
HOWARD,PHYLLIS A
HOYT,WILLIAM F
HO,DANIEL K Y
HO,DON I
HO,JENNY
HO,TIEN-LIN
HUANG,ALAN
<HUBER,RICHARD V
HUME,ANDREW G
HUNTER,ERWIN L
HWANG,SHENG Y
IPPOLITI,C D
IRVINE,M M
ISENMAN,M E
ISMAN,MARSHALL A
JAASMA,E G
JACKSON,H M,2ND
JACKSON,J H
JACOBS,H S
JAKUBOVITS,JERRY
JANTZ,R C
JARVIS,JOHN F
JARVIS,PATRICIA A
JAVELINE,MITCHELL G
JESSUP,R F
JOHNSON,STEPHEN C
JOHNSON,STEPHEN P
JOHNSTONE,I
JOHNSTON,EUGENE E
<JOHNSTON,W DEXTER,JR
JONES,JACQUELYN H
<JUDICE,C N
JULESZ,BELA
JULKE,ROBERT I
KACHURAK,J J
KAHN,MORRIS S
KAPLAN,FRANK
KAPLAN,MICHAEL M
KARR,M A
KASHDAN,SAMUEL D
KASNY,STEVEN R
KASTEN,ROBERT A
KATAMURA,WILLIAM
KATSEFF,HOWARD P
KAUFMAN,ANN S
KAUFMAN,MATTHEW E
KAYEL,R G
KEENE,BRIAN
KELLY,L J
KEMPINSKI,RICHARD V
KENNEDY,BARBARA A
<KERNIGHAN,BRIAN W
KERTZ,D R
KESSELL,D L
KHO,SAMUEL
KINAL,IHOR J
KINCAID,B M
<KINIALA,CHANDRA M R
KITA,RONALD A
KLAHN,R
KLAPMAN,RICHARD N
KLEIMAN,STEVEN
KLEINER,BEAT
<KLIEGMAN,JAY R
KNAG,RALPH H
KNOWLTON,KEN
KNUDSEN,DONALD B
KOENIG,ANDREW R
KOETHER,MARY E
KOHN,MELINDA C
KOLWICZ,KEVIN D
KOO,HAENG S
KORNEGAY,R L
KOSMAN,ROBERT A
KOWALSKI,THADDEUS J
KRAUSE,CHARLES E

COVER SHEET ONLY TO

KRETZMER,E R
KREVITT,JAMES STEPHEN
KRUNNFUSZ,DAVID A
KUMAR,J
<KUTSCH,JAMES A,JR
KWONG,TONY C W,JR
LAIRD,LINDA M
LAMSON,VICTORIA
LANCASTER,JAMES R,II
LANDIS,T J
LANDRY,MARK D
LANDWEHR,JAMES M
LANGER,JEFF
LANGE,W P
LANG,ALLEN L
LAPSLEY,LORRAINE
LATIMER,ELIZABETH A
LECHLEIDER,JOSEPH W
LEE,RICHARD T
LEGGETT,ERNEST W,JR
LEON,RAMON V
LERNER,LOUIS I
LETH,JAMES W
LEVIE,IRMELIN G
LEVY,ROGER PHILIP
LEWART,RUTH
LICWINKO,J S
LIEBERT,T A
LIGHT,D R
LIMPERT,R A
LINCOLN,CHARLES E
LINDERMAN,JOHN P
LINDGREN,GARY L
LIND,R O
LING,MOSES M
LINNELL,MELVIN GAIL
LIODICE,L J
LIU,HUEI-CHI RICHARD
LONDON,SAUL AARON
<LUDERER,GOTTFRIED W R
LUKAS,FRANCIS X J
LUTZ,KENNETH J
LYFORD,WILLIAM C
<LYONS,TERRY G
MAC DONALD,RICHARD W
MAC GREGOR,T E
MAC RAE,A U
MACALUSO,ROSANNE M
MACLACHLAN,GEORGE F
MACLENNAN,CAROL G
MACRI,P P
MAHLER,GLENN F
MAHROUS,SERENELLA F
MALLOWS,COLIN L
MALYJ,MARK
MANCUSI,M D
MANSFIELD,W H,JR
MAN,F
MARANZANO,J F
MARIONE,JOHN P
MARKSON,ROBERT N
<MARSHALL,WILLIAM T
MARSH,MERRILL
MARTIN,JOHN L
MASHEY,J R
MASTERS,CAROLINE E
MATHEWS,WILLIAM
MATTHEWS,R L
MAUNSELL,H I
MAUZEY,P
MC BURNETT,NEAL D
MC CABE,P S
MC CUNE,R J
MC CUSKER,DEBRA L
MC DONALD,L J
MC EOWEN,JAMES R
MC GILL,R
MC GONEGAL,CAROL A
MC GUIRE,J A
MC MULLEN,E C
MC NEES,GARY K
MC RAE,JEAN E
MCADAMS,HARLEY H
<MCCOY,ROBERT J
MCCULLOUGH,RICHARD H

COVER SHEET ONLY TO

MCILROY,M DOUGLAS
MCLEAN,BRUCE A
MCLEOD,BRUCE A
MEAD,JERUD
MEARS,BRIAN ROGER
MEE,C,III
MELCHNER,MELVIN J
MENDELL,HARRY B
MENIST,DAVID B
<MENNINGER,R E
MERRICK,THOMAS B
METZ,R F
MEYER,BRIAN C
MEYER,JERRI-ANN
MEYER,W C
MIHALAKIS,JAMES M
MILANO,MICHAEL L
MILLER,ALAN H
MILLER,JO ANNE H
MILLER,L D
MILLS,ARLINE D
MINNIHAN,BRIAN A
MINTZ,NANCY L
MITCHELL-STEVENS,V E
MITCHELL,J C
<MITRA,D
MOLINELLI,JOHN J
MOLTA,J W
MONAGHEN,IRENE M
MONGE,DARYL L
MOORE,HARRY H,IV
MOORE,JAMES L
MORGAN,DENNIS J
MORIARTY,PATRICIA D
MORRIS,ROBERT
MORRIS,ROBERT J T
MORR,PHILLIP L
MORTENSON,ROBERT F,JR
MROZ,WAYNE F
MUELLER,MARK B
MULLER,R ALLAN
MULROONEY,TIMOTHY J
MUNK,PAUL R
MURAKAMI,GARY J
MURCIA,JEAN
NEAL,M CATHERINE
NELSON,NILS-PETER
NELSON,WINSTON L
NEWBERRY,ROSEMARIE C
NEWMAN,BERNARR A
NOESKE,BRYON R
NORTH,STEPHEN C
NORTON,DEBORAH M
NORTON,DIANA M
NORTON,H O
NOWITZ,D A
NYLUND,RICHARD W
OBERER,ERIC
OCHS,CHRISTOPHER D
OH,RICHARD YOUNG
OLSEN,DONALD R
OLSEN,RONALD G
OPFERMAN,D C
OPPENHEIMER,PETER E
<ORCHARD,R A
ORCUTT,S E
ORSHAN,SCOTT
ORTEL,WILLIAM C G
O'CONNOR,JOHN J
O'SULLIVAN,J A
OSAJIMA,Y R
OTTO,GEORGE V E
OTT,PETER A
OZIMEK,LISA C
PACELEY,SUSAN EKSTROM
PANYKO,S F
PAPPAS,A L
PASHWA,W S
PASTERNACK,G
PAVLIDIS,THEODOSIOS
<PAYNE,RICHARD A
PEKARICH,S P
PELLEGRIN,J F
PENNINO,THOMAS P
PETERSON,DALE R

COVER SHEET ONLY TO

PETERSON,RALPH W
PETILLO,MICHAEL S
PFISTER,R G
PHALEN,GEORGE B
PHILLIPS,S J
PIASECKI,RICHARD L
PICKLE,GREGORY A
PICUS,SUSAN J
PIETROCOLA,RICHARD M
PILCH,J P
PILLA,M A
PINNES,EDWARD L
<PINSON,ELLIOT N
PIOTROWSKI,ROBERT T
PITZ,FRANK C
PLAIT,ROBERT L
POLLAK,H O
POLLI,PHILIP V
POPLAWSKI,DAVID A
PORTER,ERIC D
POST,DAVID E
PRESLEY,JOSEPH H
PRICE,JOHN D
PRIEVE,BARTON G
PRINS,G C
PROCHASKA,RICHARD F
PRUITT,JOHN W
PUCCI,M F
PUTTRESS,JOHN J
PYRCE,JOHN D
QUINN,MARGARET E
RAACK,G A
RAGUCCI,JOSEPH D
RAWISZER,JAY A
REED,S C
REESE,RANDALL D
REHERT,ALLEN F
REISER,JOHN F
REMDE,J R
REPSHER,W G
RESSLER,MICHAEL P
RESSLER,SANFORD
REYNOLDS,P E
RHODES,STEVE
RICHARDSON,JEROME C
RICHARDSON,RALPH ERIC
RICHARDS,H A
<RIDDLE,GUY G
RIGAL,E X
RINALDY,E A
RINGWOOD,STEVE A
RITZMAN,STEPHEN J
ROBERTS,HAROLD G
ROCHKIND,MARC J
RODRIGUEZ,ERNESTO J
ROHRLICH,JOHN A
RONKIN,GEORGE
ROOME,W D
ROSENBLUM,ED
ROSENTHAL,CHARLES W
ROSE,C M,JR
<ROSIN,ROBERT FISHER
<ROSLER,LAWRENCE
ROTH,ERIC A
ROVEGNO,HELEN D
RUBINSTEIN,ROY S
RUSSO,JACK
RUSSO,SALVATORE
RYAN,DORIS R
RYDER,JONATHAN L
RYPKA,DAVID J
SAKAYAMA,LARRY
SALMON,RUTH L
SALTZ,CYNTHIA A
SAND,DOUGLAS S
SANTILLO,JAMIE ANNETTE
SAPHIER,DAVID
SATHE,SHARAD C
SATZ,LAWRENCE R
SCHAEF,NANCY ANN
SCHAPIRA,ALEX
SCHATZ,BRUCE RAYMOND
SCHEIDERMAN,CAROLE L
SCHILLING,JUDITH M
SCHINDLER,G E,JR

DISTRIBUTION CONTINUED
(REFER GEI 13.9-3)

COVER SHEET ONLY TO

SCHLEGEL,C T
SCHMELZ,LANCE P
SCHMIDT,FREDERICK A
SCHONBRUN,STANLEY
SCHUBERT,RICHARD A
SCHWEBEL,JOHN C
<SCHWENKER,JOHN E
SCRIBNER,N M
SEAGREN,M K
SEARS,E R
SEARS,W D
SEGLIO,PATRICIA A
SEIDE,S B
SEKINO,W I
SELBO,JOAN C
SEMAN,JOHN A
SEMMELMAN,C L
SEREDA,BOHDAN
SEREN,R J
SETHI,RAVI
SHANK,JERE C
SHANK,R A
SHARMA,D R
SHARMA,RAMAUTAR
SHAW,KENNETH A
SHEEHAN,J C
SHERIDAN,HUGH J
SHERMAN,J W
SHICHMAN,HAROLD
SHIFLETT,CURTIS C
SHIH,KO-JEN
SHIPLEY,EDWARD A
SHLIFERSTEIN,ABE
SHOMO,ROBERT E
SHORTER,J W
SHPIZ,TERRY
SHUGARD,WILLIAM J
SHULMAN,SHERYL
SHUR,ALLAN I
SILBERG,ELLEN
SILVERBERG,MICHAEL A
SILVERBERG,S M
SINGER,H R
SINGH,R P
SINOWITZ,NORMAN R
SIPPLE,WILLIAM A
SLAUGH,JOHN H
SMITH,DALE W
SMITH,GERALD H
SMITH,LESLIE T
SMITH,NANCY W
SMITH,RAMLEY D
SNOW,BRUCE F
SNYDER,THOMAS S
SO,H C
SPACER,THOMAS R
SPIRES,R J
STANZIONE,DAN C
STARK,MARTIN A
STARNER,GUY H
STEFFEN,JOSEPH L
STEIGERWALT,R A
STEPNOSKI,R AARON
STREETER,LYNN A
STROHECKER,CARL A
STROM,ROBERT C
<STUCK,E W
STUK,LAURALU V
STURMAN,DAVID JOEL
STURMAN,JOEL N
SUMMERS,CHARLES R
SURKO,PAMELA
SURRATT,GROVER T
SWANSON,G K
SWANSON,RONALD H
SZYMANSKI,THOMAS G
TAGUE,BERKLEY A
TAMMARO,ENN
TANENBAUM,WILLIAM M
TARTARONE,STUART A
TASCO,AGNES M
TATEM,R J
TAYLOR,WILLIAM D,JR
TEETOR,PAUL R
THILLMAN,STEVEN A

COVER SHEET ONLY TO

THOMAS,A C
THOMAS,R B
<THOMPSON,J S
THOMPSON,K
THOMSON,MICHAEL G R
TODD,BYRON E
TOOLE,PATRICK J
TOTH,STEPHEN W
TRAN,ROGER T
TROESTER,ROBERT
TSE,STEPHEN KAM HONG
*TUKEY,JOHN W
TURNER,JONATHAN S
TUTELMAN,DAVID M
ULLRICH,RITA T
UNDERWOOD,R W
VADEN,LAURY
VAN WIK,CHRISTOPHER J
VANANDEL,JOSEPH H
VEDDER,D
VERBINSKI,CYNTHIA A
VERNON,F D
VIZCARRONDO,J
VOGEL,DENNIS R
VOGEL,GERALD C
VOLLHERBST,KEITH
WADSACK,R L
WAGNER,BRUCE D
WAGNER,MARY R
WAHL,ERNST J
WAHL,JAMES E
WAINWRIGHT,PAUL F
WALKER,J D
WALLACE,JOHN J
WALSH,EDWARD D
WALTERS,R W H,JR
WANDZILAK,PHILIP D
WANG,DA-LUN
WANG,T L
WARD,CHARLES E W
WARNER,JACK L
WATSON,D S
WEBER,SUSAN S
WEBER,THOMAS A
WEIGAND,W F
WEINBERGER,PETER J
WEISMANN,CAROL E
WELLNER,G J,JR
WETHERELL,CHARLES S
WEYTHMAN,JAMES E
WHIPPLE,JOHN H
WHITE,E WARNER
WHITE,P E
WHITTEN,DOUGLAS E
WILKEN,DONALD E
WILHOITE,W L
WILLIAMS,R D
WILSON,CYNTHIA L
WILSON,ELIZABETH A
WILSON,WALTER H
WINCHELL,D F
WISNIEWSKI,STANLEY C,II
WITTENBERG,LEE
WOLFE,ROBERT M
WONG,PAUL
WOODRUFF,JOHN L
WOOD,JOSEPH L,III
<WOOLEY,BRUCE A
WRIGHT,DENNIS E
<WRIGHT,WILLIAM FREDRICK
WURTH,RICHARD T
WU,SU-YU
WYSE,NEAL C
YACOBELLIS,ROBERT H
YAFFEY,C L
YAMIN,ELAINE E
YAMIN,M
YAMONACO,RANDALL V
YATES,GREGORY S
YEAGER,DAVID M
YOSTPILLE,J J
<YUE,ON-CHING
YUREK,EUGENE
ZEBO,T J
ZELMS,C M

COVER SHEET ONLY TO

ZISLIS,PAUL M
ZOLNOWSKI,JEAN C
ZYSMAN,G I
857 NAMES

Bell Laboratories

Subject: An Object Editor for a Real Time Animation Processor
Case- 011170-0101 -- File- 39394

date: October 1, 1980

from: S.P. Ressler

TM: 81-11356-13

MEMORANDUM FOR FILE

1. Introduction

The function of the object editor is to allow a user to manipulate the description of an object while it is being displayed. Display of the object is performed by the animation processor *(AP)*[1]. The term *object* as used in this paper is a collection of x,y,z coordinates which define a group of polygons that can be manipulated and referred to as a single entity. An object in space has a single location and orientation (attitude), which form part of its description. The object editor as it exists is not meant to be a production line design tool, rather it is a research tool for exploring the possibilities of real-time interaction with computer generated objects.

2. User Interaction

The user interacts with the system by using the joysticks, buttons and sliders present on the *AP (see figs 1 & 2)*. Many of the operations used, have been multiplexed onto the buttons located on the base of the joysticks. One button has a select mode function which changes the functions of the other buttons. Feedback for the current mode is seen immediately on the graphics screen via a blinking menu, and on the terminal which presents auxiliary text information *(see fig 3)*. The two circular images on the terminal correspond to the two joysticks, as viewed from the top. The surrounding words are labels which correspond to the buttons located on the joysticks. The words are dynamic labels and as the functions of the buttons change so do the labels on the terminal.

The decision to move most of the functions onto the buttons was made because the buttons are very discrete controls. which make them easy to use. The sliders which can also be used, give one a means of quickly doing something, (i.e. positioning the object) but they can get clumsy for accurate activities. One can effectively use the buttons, to single step actions if desired. The speed at which the buttons affect things is controlled by a slider. One can use the combined effect of a speed slider and buttons to make the button controls very usable, for most of the system's functions.

The joystick is used to control the orientation of the object. Since we are dealing with a real time environment the need to precisely specify the orientation of the object, is eliminated. One simply moves the joystick that is controlling the orientation of the object, to the desired view. The object can also be made to rotate continuously while other operations are performed.

3. System Capabilities

The user is given the choice of several modes which follow a logical order, smaller operations to larger. These modes are selected via the mode select button. The possible modes are: point, polygon, volume and addition. Selecting one or another mode affects the functions of the other buttons in a very straightforward manner. The x,y,z moving buttons will move a single point when in point mode, a polygon when in polygon mode, and a volume when in volume mode. The mechanism one uses to select or step through the data description is via the stepping button. The stepping button will step you along points when in point mode, polygons when in polygon mode. It makes no sense to step along volumes when in volume mode so the x,y,z moving buttons do nothing in volume mode. The entire system functions in this way. If the system is in polygon mode, the copy button copies polygons. If it is in point mode, the delete button deletes a point. etc..etc... If a capability makes no sense then the button will do nothing.

Addition mode is a little different then the others. It enables you to add pieces to the object. It is different from copying because it adds completely new points. When in addition mode four of the joystick buttons get redefined to enable the addition of a dot, vector, rectangle. or cube.

4. Implementation Environment

The object editor exists as one part of a complete animation/games system currently under development. The *AP* is run together with an LSI 11/23 and share a common memory and bus. The system uses a Micropolis winchester disk for storage and runs completely stand alone. The *AP* software is an animation language which is written in C under UNIX V7, using YACC and Lex. This new animation language was used to write the object editor.

5. Getting Hard Copy

Another facility in the object editor is the ability to take a snapshot for creating Unix type plot files. The images created by this method are simple one color line drawings. Moving slider 46 causes the system to ask for a file name. The display list of the animation processor is then place into that file. Next you must transfer that file to the VAX. You can then do one of several things. If you are on a Tektronix terminal simply say *draw filename* and the display list will be converted and drawn on the screen. If you want to create a plot file say *toplot file* and a file will be created with the name you gave appended with a .f . This is a Unix device independent plot file and can be used to produce plots on any of the devices supported by plot.

6. Object Editor Users Manual

6.1. Starting Up

This section is basically a users manual for the objects editor and contains information which is quite specific to the operation of the editor. The editor was written in ap2 code which is shown in some explanations, however an understanding of this code is not necessary for the use of the directions.

The editor is started up via the command *ap2 Obedit.*

6.2. Selecting an Object for Editing

The first 7 sliders, (pot numbers 32-38, see figure 1 for the location of all the pot numbers) control the visibility of 7 different objects to be edited. Move the sliders down and up to select or deselect the objects. The object initially displayed is attached to the first slider. If more than one object is being displayed only the last object selected can be edited the other objects will be rotated by the joystick but cannot be edited. The 7 objects attached to the object select sliders are default examples of objects and may be changed by editing the file Obed. Modify the code with the following form:

```
name:  obj       { mtx_fin; loc(0,0,0); invis;
#include "x1obj"            /* replace file name x1obj with your own */
                 }
```

These object files must contain only the point descriptions not the *obj* declaration. This will force the system to recompile the next time you run the editor, which does take a while (about 10 min.).

6.3. Function Selection

The first and last buttons on joystick 2, (pots 11 & 15) are the function select buttons. The first button (pot 11) steps you forward, (Funkpos) and the second steps you backwards (pot 15). Both buttons will continually cycle through all possible modes, "wrapping around", however they step in opposite directions. When the system starts up, you are in point mode, and I would advise you to cycle through all the modes upon starting up to check the system out and make sure all is well. When in point mode the dot in the menu *(see fig 4)* at the top of the screen should be blinking. The order of modes and their visual feedback, in parenthesis is as follows:

1 Point *(blinking dot, in menu)*

2 Polygon *(blinking solid square, in menu)*

3 Volume

 3.1 Volume Select *(volume select cube blinks blue)*

 3.2 Volume Scale *(volume select cube blinks green)*

 3.3 Volume Points Movement *(blinking open square, in menu)*

 3.4 Volume Points Scaling *(blinking line above open square, in menu)*

4 Addition *(blinking dot, vector, rectangle, cube in menu)*

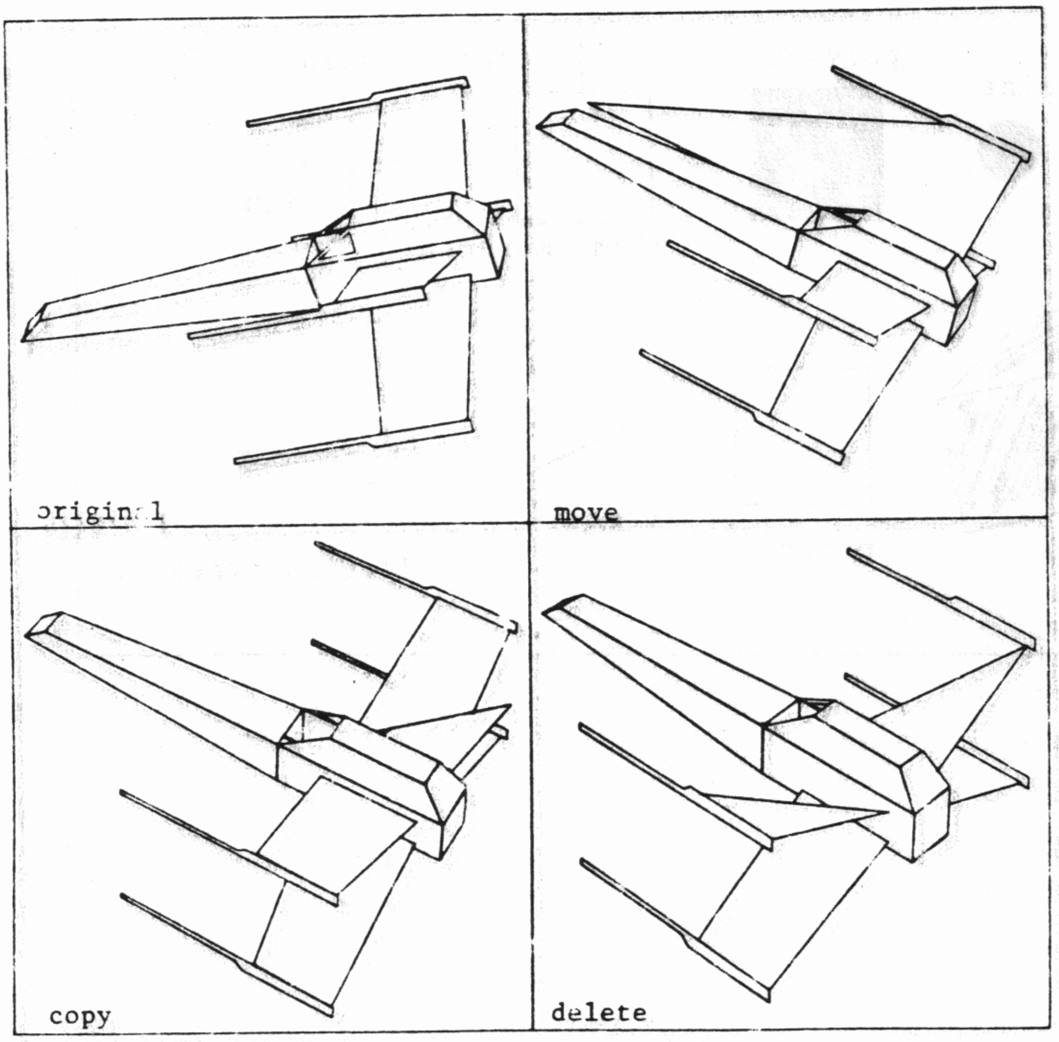
Following is a detailed explanation of each mode:

Mode: Point

Feedback Blinking dot in the menu. Blinking dot at the current point.

Description: Enables the manipulation of any *single* point in the object being edited.

Usage: The x,y,z buttons on the joystick move the current point, in the x,y,z directions. The speed slider affects the speed of x,y,z movement. The sign button is a toggle between plus and minus. Point movement is in the sign direction for each axis. The sign is displayed in the menu and on the terminal. The copy and delete buttons will copy and delete a single point. The point is selected via the step button which will step forwards if the sign is plus and backwards if the sign is minus. Point stepping will "wrap around" through the entire object description.

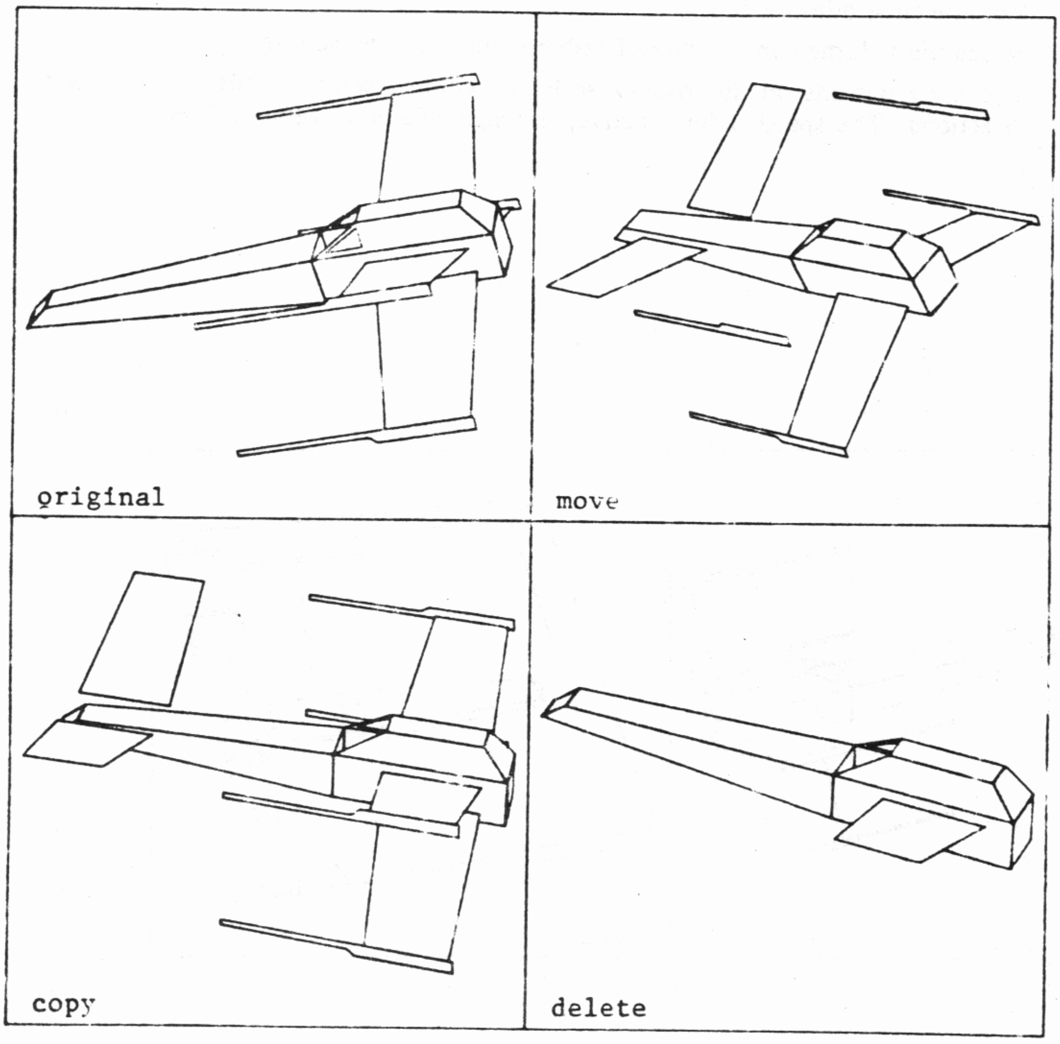


Mode: Polygon

Feedback Blinking solid square in menu. Show poly button blinks current polygon.

Description: Enables the manipulation of any polygon in the object being edited.

Usage: The x,y,z buttons on the joystick move the current polygon in the x,y,z directions. The speed slider is active, and sign affects movement direction. The copy and delete buttons will copy and delete a single polygon. The current polygon is selected via the step button which will step through the object description a polygon at a time in the sign direction. The show poly button is a toggle to blink and unblink the current polygon. (Note: Make sure to unblink the polygon before moving on to another polygon, as color information can get lost)



Mode: Volume Select

Feedback Volume cube blinks blue.

Description: Moves the volume cube around to enable the selection of any point inside the cube, for further manipulation, in volume pts move or volume pts scale modes.

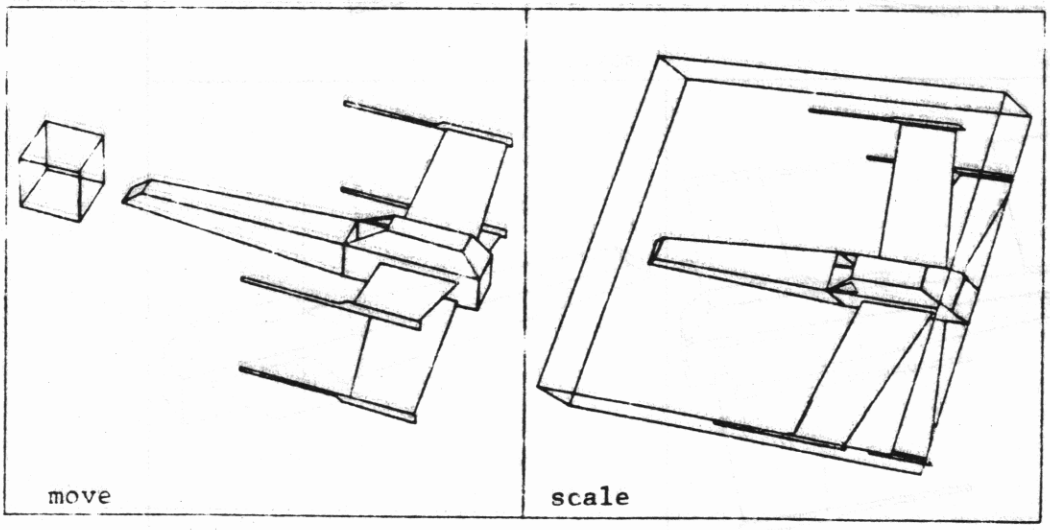
Usage: The x,y,z buttons on the joystick move the volume selection cube in the x,y,z directions. The speed slider is active, and sign affects movement direction.

Mode Volume Scale

Feedback Volume cube blinks green.

Description: Scales the volume cube so more flexible volume can be selected.

Usage: The x,y,z buttons on the joystick scale the volume selection cube in the x,y,z directions. The speed slider is active, and sign affects movement direction.

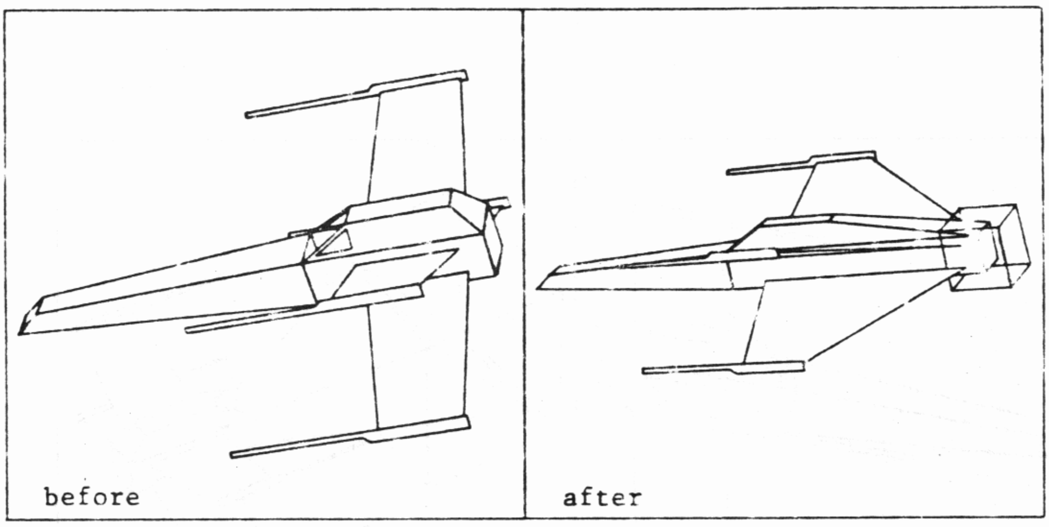


Mode: Volume Points Movement

Feedback Open square in menu blinks.

Description: Points inside the volume cube are activated for movement.

Usage: The x,y,z buttons on the joystick move all points inside the volume cube in the x,y,z directions. The speed slider is active, and sign affects movement direction. All points will move regardless of whether or not only part of a polygon is in the volume cube.

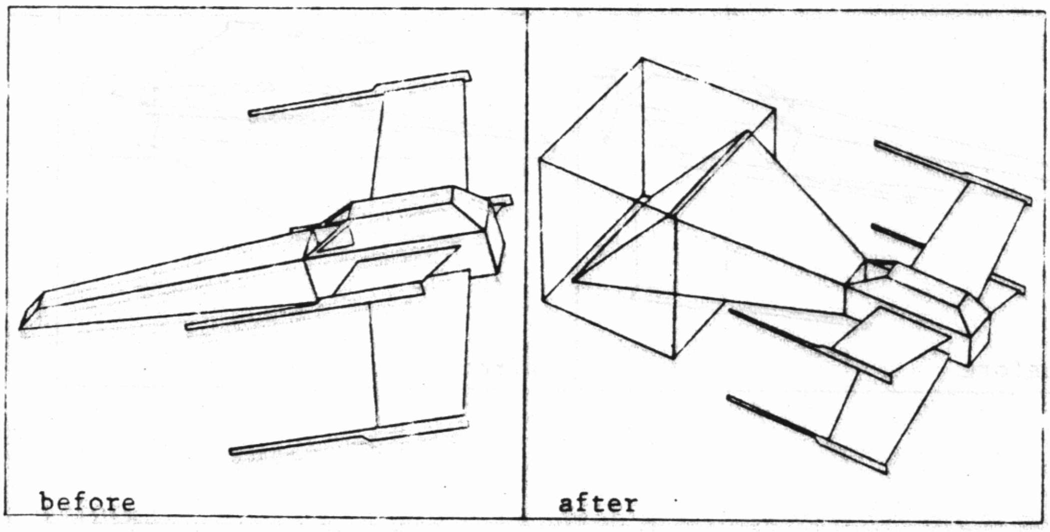


Mode: Volume Points Scale

Feedback Line above open square in menu blinks.

Description: Points inside the volume cube are activated for scaling.

Usage: The x,y,z buttons on the joystick scale all points inside the volume cube in the x,y,z directions. The scaling is always towards or away from the center of the volume cube. The speed slider is active, and sign affects the direction of the scaling. All points will scale regardless of whether or not only part of a polygon is in the volume cube.

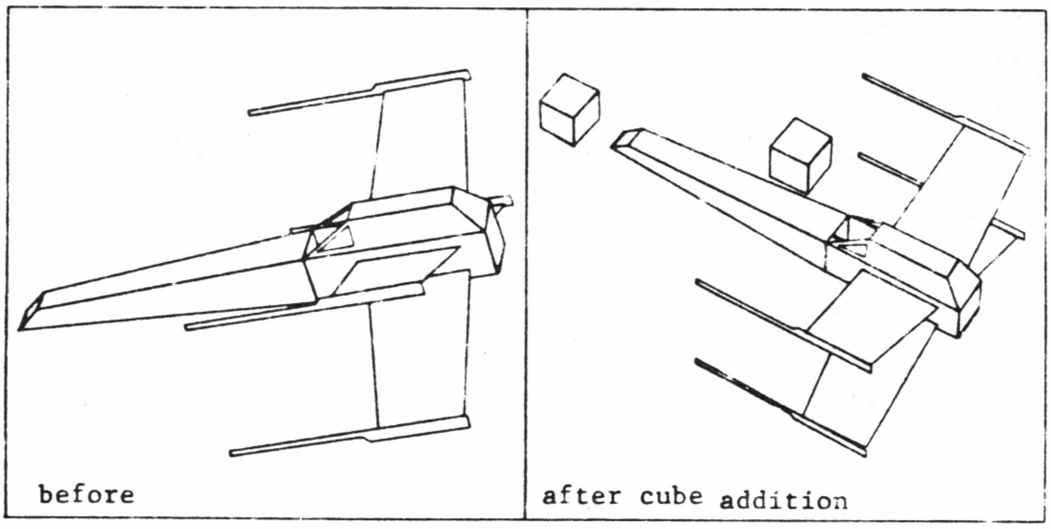


Mode: Addition

Feedback Dot, vector, rectangle and cube in menu blink. Terminal display the dot, vector, rectangle and cube button.

Description: Allows the addition of a dot, vector, rectangle or cube to the object description.

Usage: The buttons on joystick 1 get redefined when this mode is entered, to be dot, vector, rectangle or cube addition buttons. Press one of these buttons once and the corresponding shape will appear at the center of the object (location 0,0,0). Enter one of the movement modes to subsequently move this new shape to its desired location.



before

after cube addition

References

[1] H.G. Alles, W.C. Fischer, *An Animation Processor for Action Oriented Three Dimensional Color Graphics* TM 81-11356-1.

[2] Kernighan and Ritchie, *The C Programming Language.* Prentice Hall 1978.

7. Getting Hard Copy

Another facility in the object editor is the ability to take a snapshot for creating Unix type plot files. The images created by this method are simple one color line drawings. Moving slider 46 causes the system to ask for a file name. The display list of the animation processor is then place into that file. Next you must transfer that file to the VAX. You can then do one of several things. If you are on a Tektronix terminal simply say *draw filename* and the display list will be converted and drawn on the screen. If you want to create a plot file say *toplot file* and a file will be created with the name you gave appended with a .f . This is a Unix device independent plot file and can be used to produce plots on any of the devices supported by plot.

8. Acknowledgements

Thanks to Carl Christensen who wrote the animation processor language and was always willing to answer sticky questions. Much thanks also to David Baraff who wrote the hidden line routines which were used to draw the pictures in this paper.

MH-11356-SPR

S.P. Ressler

Atts.

References(1-2)
Appendix

Appendix: Object Manipulation Techniques and Subroutines

In the course of creating the object editor a number of techniques for dealing with the object descriptions were developed. This section is a discussion of these techniques and an explanation of some subroutines used in the editor.

In order to manipulate the objects one must first understand little of how they are represented. Objects exist in part as a collection of polygons. A polygon consists of a collection of x,y,z, coordinates, i.e. points. All polygons end with a color description point. All points have the following C[2] structure:

```
typedef    struct {
           unsigned size: 9;
           unsigned type: 7;      /* type of point */
           unsigned uu2:  4;
           unsigned x:   12;      /* x coord. */
           unsigned uu3:  4.
           unsigned y:   12;      /* y coord. */
           unsigned uu4:  4;
           unsigned z:   12;/* z coord. */
} Pts_s, *Pts;
```

Each point in the display can be modified via the PTS macro which accesses the memory where the points are located in the 11/23. This PTS macro points to the Pts_s structure. The type field in above structure is the identifying op code of the point, which can be one of the following:

DOP_COLOR	color of sub object
DOP_ENDOBJ	end of object
DOP_DOT	small dot
DOP_LRGDOT	big dot
DOP_PVSRT	start of polygon or vector
DOP_POLY	perimeter continue
DOP_VEC	vector continue
DOP_END	end of picture

A typical polygon would start with a point which has a **DOP_PVSRT** as its type, followed by a collection of points with **DOP_POLY** as their types. The last point for each polygon would be a **DOP_COLOR**, and the z field of the point structure will contain the color information, which is 4 bits for the intensity, blue, green, and red components. The last polygon of the object would contain a **DOP_ENDOBJ** instead of a **DOP_COLOR**. Polygons must have a matching first and last point for closure, whereas vectors and dots need not. Vectors and dots also are always followed by a **DOP_COLOR** as the last point in their data descriptions.

Each point in the display can be modified via the PTS macro which accesses the memory where the points are located in the 11/23. This PTS macro points to the Pts_s structure. The macro takes as an argument the point number. Each point in the display is consecutively number. The trick with object manipulation is to find the desired point at the right time. The statement:

```
spt = PTR(obj_name)->pts;
```

will assign to spt the number of the first point for the object obj_name. Objects like points are also consecutively numbered, and the argument to the macro PTR is the object number. To find the last point for the particular object:

```
ept = spt + PTS(spt)->size;
```

the volume selection cube.

```
comb_obj(obj1,obj2,buf,fbuf)
int obj1,obj2;   /* two object names*/
Pts_s *buf, *fbuf;     /* pointers to point buffers */
```

Comb_obj combines two objects into a single one. Obj1 is appended to the end of obj2.

Fpoly determines the data bounds of a polygon given a point in that polygon. It sets two global variables, Polypt and Pept, to the beginning and ending of the polygon.

```
movpts(objname,startpt,endpt,xd,yd,zd)
int objname;    /* object name */
int startpt,endpt; /* the starting and ending points */
                /* to be moved */
int xd,yd,zd;   /* the amount of x,y and z to move the points*/
```

Movpts physically moves the points from startpt to endpt by xd, yd, and zd. Makes sure to match up the beginning and ending points of polygons if only part of the polygon is to be moved. Fmovpts is a faster version which doesn't do the polygon beginning and ending point checking.

```
cop_pt(objname,buf,fbuf)
int objname;    /* object name */
Pts_s *buf, *fbuf; /* point buffers */

del_pt(objname,buf,fbuf)
int objname;
Pts_s *buf, *fbuf;
```

These routines copy and delete a single point from the point description of the object. The point to be deleted or copied is in the global variable Obpt.

```
cop_poly(objname,buf,fbuf)
int objname;            /* object name */
Pts_s *buf, *fbuf;      /* two temporary point buffers */

del_poly(objname,buf,fbuf)
int objname;
Pts_s *buf, *fbuf;
```

These routines copy and delete the current polygon, (can be determined by fpoly). The point buffers can be written to and read from using the _rd_obj and _wt_obj subroutines. The display is immediately changed and updated to show the result.

```
get_volume(objname)
int objname;    /* determines the points located inside */
                /* the volume selection cube */
```

This routine determines those points in object objname, that are inside the selection cube. The point numbers are places in an array called Volume, and an index to the last point in the array is set in the int Vi.

```
mov_volume(objname,xd,yd,zd)
int objname;    /* object name */
int xd,yd,zd;   /* distance in x y and z to move */
```

The mov_volume subroutine uses the array Volume to move all the points inside of the volume selection cube a distance xd,yd and zd. It simply calls the fmovpts routine with the point numbers in Volume.

```
scl_pts(objname,xs,ys,zs)
int objname;    /* object name */
int xs,ys,zs;   /* x y and z scaling factors */
```

This routine scales all the points in Volume independently in the x,y and z dimensions by the factors xs,ys and zs. The scaling is towards or away (depending on Sign) from the center of

The field. size, is only valid at the first point of the object and contains the number of points for that particular object. *Do not write to this field unless you know what your doing!!!* The integer ept now contains 1 + the last point of the object. One now has the point boundaries of the object to be dealt with. This technique is object independent and does not depend on any particular point names, and is used frequently.

Another useful technique is to combine the capability of processes, with mvrs to create *thruster* controls. The X, Y, Z buttons of the system are controls which cause constant action while depressed and stop the action when let go. For example to move a polygon in the x direction you press the X button (if in polygon mode) and the polygon will keep on moving as long as the button is pushed. A mvr is a statement in the AP language, which allows one to define the function of a particular physical pot. A process is a piece of C or AP code which executes periodically, (up to 30 times a second), and can be started or stopped any time with a high degree of flexibility. To create a thruster one must do the following:

Detect when a button has been pushed.
Start a processes which performs the desired function.
Detect when the button is released.
Stop the process.

Buttons which function as thrusters are very useful control devices which are very easy to deal with.

Some useful functions for dealing with objects are:

```
_rd_obj(objname,buf)
int objname;    /* the name of an object */
Pts_s *buf:     /* pointer to a section of point data*/
```

The subroutine _rd_obj reads the points which make up the description of an object from the portion of 11/23 memory which is DMA'ing out to the animation processor, and places it into a buffer which can be manipulated without affecting the display:

```
_wt_obj(objname,buf,size)
int objname;
Pts_s *buf;
int size;          /* the size of the object in points */

initbufs()         /* initialized point buffers */
```

The subroutine _wt_obj will take the contents of the buffer and put the points back into the section of 11/23 memory which DMA's out to the animation processor and reallocates the space in that memory to accommodate objects that have changed in size. It is the users responsibility to know what the size of the object is for reallocation into the displayable portion of memory.

```
obj_write(fname,objname)
char *fname;    /* pointer to name of file*/
int objname;    /* object name*/
```

The routine obj_write takes an object that is currently being displayed and writes a file (fname) which contains AP source code for the description of that object. This new file can then be used as a valid AP object for any sort of application program on the animation processor.

```
fpoly(objname,pt)
int objname;    /* object name */
int pt;         /* point number */
```

Figure 1

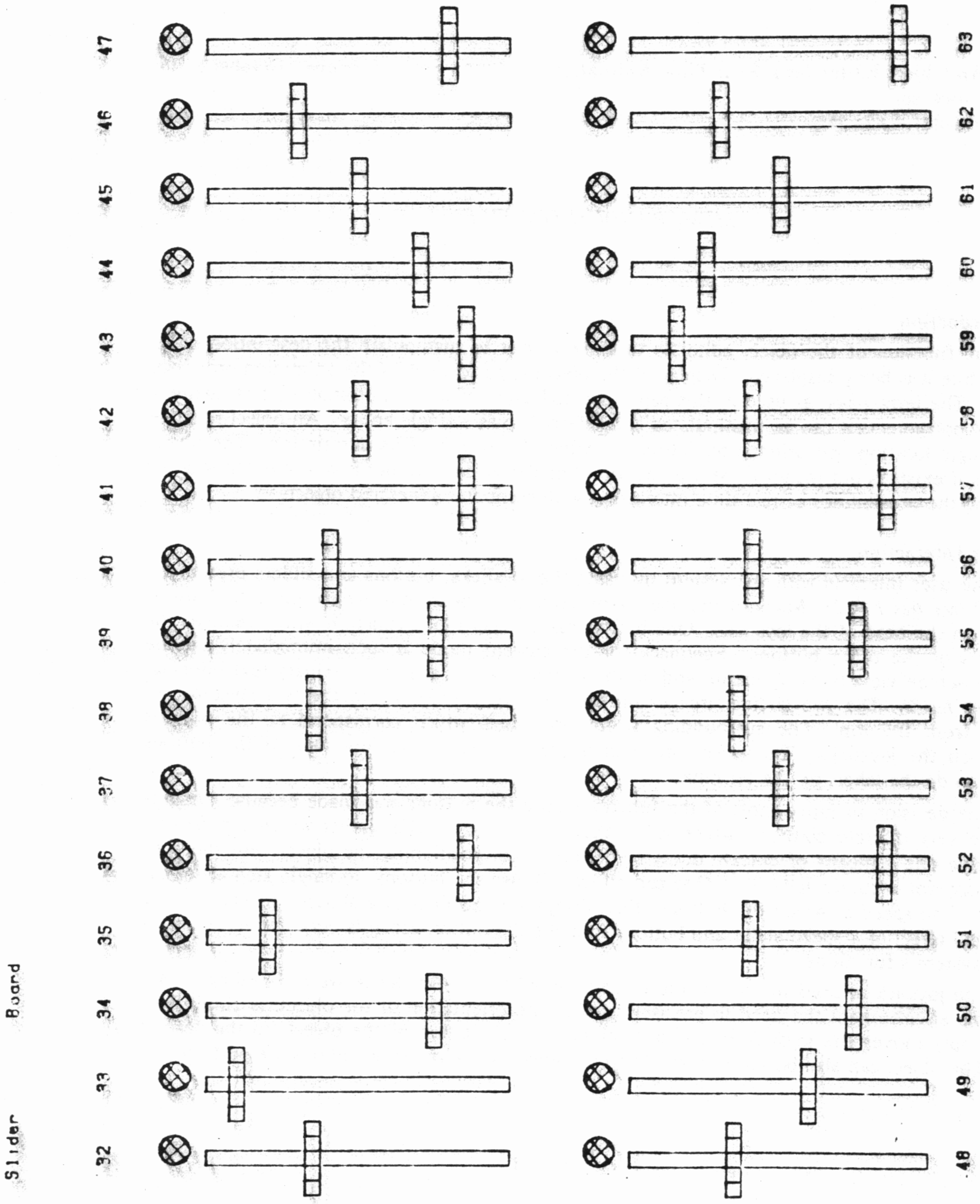

Slider Board
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

Figure 3

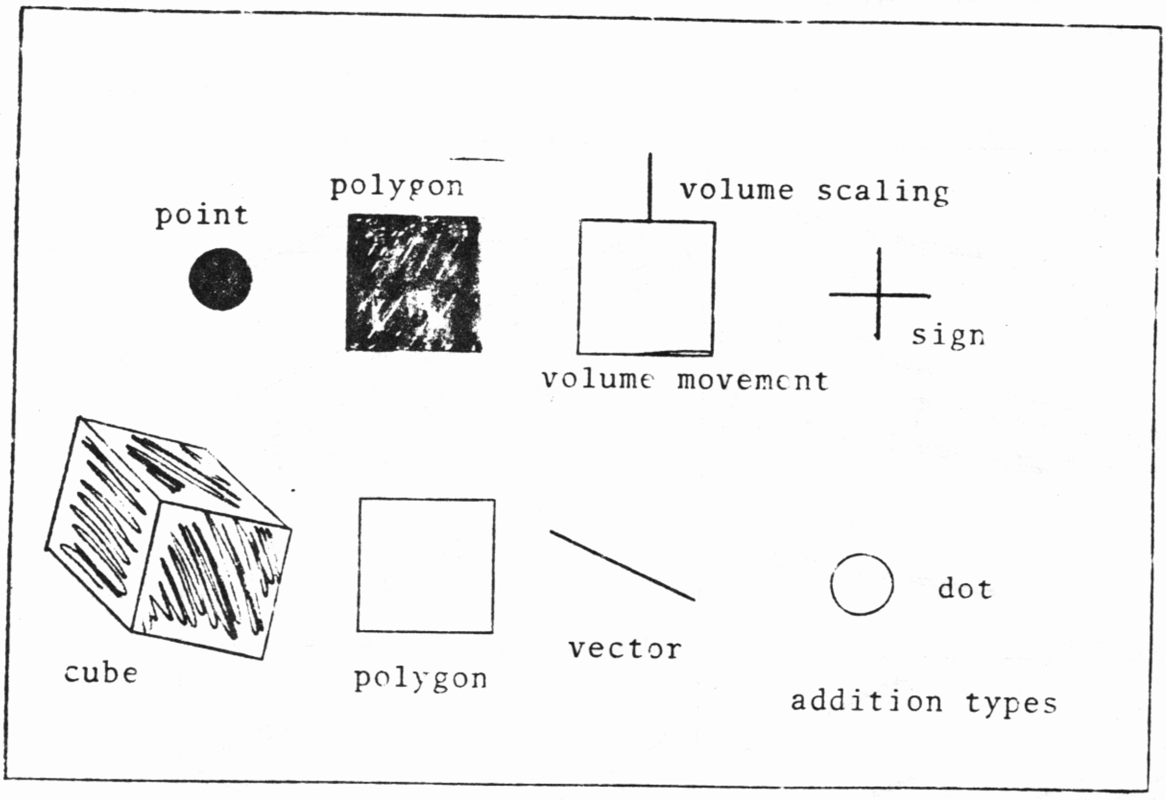
```
           Animation Processor 3D Object Editor

                                         Slider Functions
                                         47 ... save obj into file
     Mode: Area Select                   48 ... x y z speed
     Sign: +                             51 - 54 active polygon color
     Enter file name:                    55 - 58 background color

         Show Poly                                Y
        ------------                         ------------
       /            \                       /            \
Copy  /              \Delete           X   /              \Z
     /     ----       \                   /     ----       \
    |     /    \       |                 |     /    \       |
Step|     \____/       | Sign    Modepos |     \____/       | Modeneg
     \                /                   \                /
      \              /                     \              /
       \____________/                       \____________/
         Joystick 1                           Joystick 2
```

Figure 4

OBJECT EDITOR MENU SYMBOLS


point
polygon
volume scaling
sign
volume movement
cube
polygon
vector
dot
addition types