Using the Amigo: An Introduction to English-programing	
Getting Started	
RUN	Executes the program currently stored in program memory.
<Esc>	Halts the program currently running.
LIST	Copies the current program to the screen. Each program line must begin with a number, and program lines will be listed and executed in numerical order. To delete a program line, type just its number at the flashing prompt and press <Enter>.
DIR	Lists a directory of files on the SD card. Only shows those files in the root directory.
SAVE “name.bas”	Saves the program currently in program memory to the SD card, with the name and extension inside the quotes.
LOAD “name.bas”	Copies the program “name.bas” from the SD card to program memory.
DELETE “name.bas”	Deletes the file “name.bas” from the SD card.
NEW	Clears program memory; sets the 26 Color Eng-Prog variables to 0.
HOLD	Saves the program currently in program memory to the SD card, with the name of DESKTOP.TMP. Color Eng-Prog and the editor are separate programs and pass this DESKTOP.TMP file back and forth to open and close the editor.
REBOOT	Reboots the Amigo, just like pressing the reset button or interrupting power. If there is a file named DESKTOP.TMP in the root directory of the SD card, loads and runs that file. Otherwise boots up to the READY Color Eng-Prog prompt.
<F1>	Opens the Color Eng-Prog editor, which allows use of the arrow, Home, End, Insert, and Delete keys to edit programs. Returns to Color Eng-Prog and runs the program when <F1> is pressed again.
Screen Commands	
CLS	Clears the screen with the current background color; moves cursor to top left corner.
COLOR f,b	Sets the foreground and background colors for subsequent screen

	commands. Color Eng-Prog has 64 colors (0-63).
LOCATE c,r	Moves the cursor to the specified text column (0-49) and row (0-36).
PRINT	Prints the value of a variable, or a string enclosed in quotes, at the current cursor location, in the current foreground and background colors. Multiple variables and strings can be included in one PRINT statement, separated by commas and/or semicolons.
DISPLAY n	Displays the glyph currently assigned to the character with ASCII value n, at the current cursor location. Multiple values can be included in the same DISPLAY command, separated by commas. The Color Eng-Prog character set includes some graphics characters, and the glyph of any printable character can be customized using the REDEFINE command.
REDEFINE n, a,b,c,d,e,f,g,h	Changes the glyph of ASCII character n to the 8x8 pixel map defined by a-h, which are 8-bit integers. Characters on the screen assume their new glyph immediately upon redefinition.
PLOT x,y,c	Places a small square (one quarter of a text cell, or 4x4 pixels) in color c at graphics column x (0-99) and graphics row y (0-74). Only one foreground color is allowed in any text cell, so plotting a square of a different color in the same text cell will change the color of any other squares in that text cell.
BOX u,v,x,y,c	Places a box on the screen in color c with the top left corner at graphics coordinates u,v and the bottom right corner at x,y.
LINE u,v,x,y,c	Draws a line in color c between graphics coordinates u,v and x,y.
Keyboard Commands	
INKEY	Checks the keyboard buffer for a keystroke and returns the assigned value of any key pressed, or 0 otherwise. Almost always used to the right of the assignment operator, as in A=INKEY.
INPUT “prompt”; a,b,c	Prints the “prompt”, if any, at the current cursor location, then waits for the user to input the prompted number of integers, separated by commas, then press <Enter>. Generates a syntax error if the number of variables input by the user does not equal the number expected. Note that Color Eng-Prog does not handle strings, and this capability is for integers only.

Using the Amigo: An Introduction to English-programing	
Program Control Commands	
GOTO xxx	Directs program flow to jump to the instruction in Line xxx.
GOSUB xxx … RETURN	Directs program flow to the subroutine at Line xxx, then back to the calling routine on the RETURN command in the subroutine. All 26 variables in Color Eng-Prog are shared, so arguments are not explicitly passed and subroutine variables are not private.
IF expression THEN command	Checks to see whether the expression is TRUE (non-zero), and if it is executes the command(s) following THEN. Color Eng-Prog does not include ELSE or ENDIF constructs. Multiple commands can be concatenated after the THEN by separating them with a colon.
PAUSE nnnn	Pauses program execution for nnnn milliseconds.
REM	Allows insertion of a textual remark after the REM for internal documentation. May be used on its own program line, or (generally, but not always) at the end of program line containing other commands.
END	Terminates program execution and returns to the flashing cursor prompt to await user command. END is not required, but is useful to separate the main body of a program from following subroutines.
Audio Commands	
NOTEON nn	Sends a continuing tone of value nn to the audio connector; useful values seem to range from 0 to 100. A powered speaker works best with the Amigo, but earbuds will do in a pinch.
NOTEOFF	Stops any tone currently playing on the audio connector.
PLUCK nn	Sends a note (not a continuing tone) to the audio channel. PLUCK 40 is close to middle C, and consecutive integers will play the notes of adjacent white and black keys on a piano keyboard.
Wii Classic Controller Commands	
JOY	Returns the value of any button pressed on a Wii® Classic Controller (not the nunchuck) attached to the Amigo, or zero if no button is pressed. If two or more buttons are pressed at the same time, JOY returns the sum of their assigned values. The **a** and **b** buttons always override any other buttons, with **a** always taking priority over **b**. Color Eng-Prog does not recognize the joysticks on the Wii controller, just the buttons, and the controller must be connected to the Amigo before the system is started. Use on the right

	side of the assignment operator, as in A=JOY.
File System Commands	
OPEN “file-name”, mode	Opens a file on the SD card in the desired mode (either R, W, or A — for Read, Write, or Append). Files opened in Read mode will not be modified by any program actions. Files opened in Write mode may have any or all of their content modified. Files opened in Append mode may be modified only by adding content at the end of their current content.
WRITE a,b,c	Stores the values of the specified variables into a file opened for writing or appending. You should not include other Color Eng-Prog commands (including REM) on the same program line as WRITE.
READ a,b,c	Places data from a file opened for reading into the specified variables. The number of variables specified must not exceed the number of values in that line of the data file. (Otherwise Color Eng-Prog returns a syntax error.) Note that the the numerical values to be retrieved by READ from the file cannot be negative; Color Eng-Prog will interpret this as an End of File condition. Also note that you should not include other Color Eng-Prog commands (including REM) on the same program line as READ.
CLOSE	Closes the file currently open, if any. Color Eng-Prog can have only one file open at any given time.
GPIO Commands	
INA <pin>	Reads the value of the voltage level (either a logic 1 or 0) on the specified input/output <pin> (0 through 31) of the Propeller chip. Used for interfacing Color Eng-Prog programs to external circuits through the Amigo experimenter’s section.

OUTA[<pin>] = n	Changes the specified input/output <pin> of the Propeller chip (0 through 31) to either a logic 1 or 0. Used for interfacing programs to external circuits through the Amigo experimenter's section.
Operators and Other Commands	
Arithmetic Operators	Color Eng-Prog supports 32-bit integer arithmetic, including operators for addition (+), subtraction (-), multiplication (*), integer division (/), and modulo or remainder (//). Arithmetic operations return a 32-bit integer value.
Comparison Operators	Comparison operators include greater than (>), less than (<), equals (=), greater than or equal to (>=), less than or equal to (<=), and not equal to (<>). Comparison operations return either a -1 (the comparison is TRUE) or 0 (the comparison is FALSE).
Logical Operators	Color Eng-Prog logical operators include NOT, AND, and OR. Logical operations also return either a -1 or a 0 value.
BRUN "filename"	Loads a binary file from the SD card to Amigo main memory and runs it. The loaded file completely replaces Color Eng-Prog (which is itself a Propeller binary). Color Eng-Prog is reloaded and run after "filename" terminates. Several games and other programs are available for the Amigo as .BIN files.
BYTE [nnnnn]	Reads or writes an 8-bit value from / to the Amigo main memory location specified, where nnnnn is between 0 and 32767. The command can be used on either side of the "equals sign" assignment operator, as in A=BYTE[1921] (reading from main memory) and BYTE[1921]=65 (writing to it). Caution should be used when writing to main memory because this could corrupt the Color Eng-Prog image and lead to unexpected results. If this happens, just reboot.
MEM	Returns the amount of unused program memory, in bytes. The Amigo has 4K of program memory with no program loaded.
RND (n)	Returns a random number between 0 and n - 1.
SERIAL r,t,m,b	Initializes the Amigo serial communication channel, where r is the receive Propeller pin (default 31), t is the transmit pin (default 30), m is the mode (default 0), and b is the baud rate (between 300 and 115200). Generally placed at the beginning of a program, as in SERIAL 31,30,0,115200. On the Amigo, Propeller pins 30 and 31 are available via the PropPlug.

Using the Amigo: An Introduction to English-programing	
RX	Fetches a byte from the previously initialized serial receive pin and passes it to a variable, as in A=RX.
TX <byte>	Pushes a byte to the previously initialized serial transmit pin, as in TX A.
VER	Returns the version of Color Eng-Prog loaded in the EEPROM of the Amigo, used on each boot up.