SNAKE

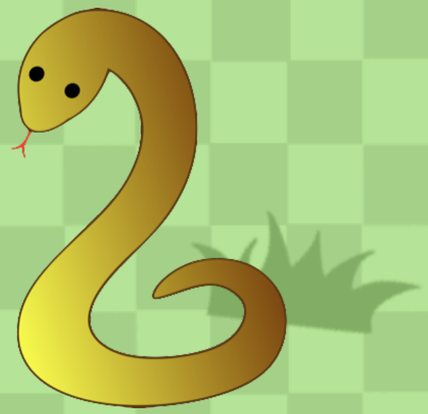
alexander eckert
kerolos khalil
jessica ortega
benen kim

Four Main Parts

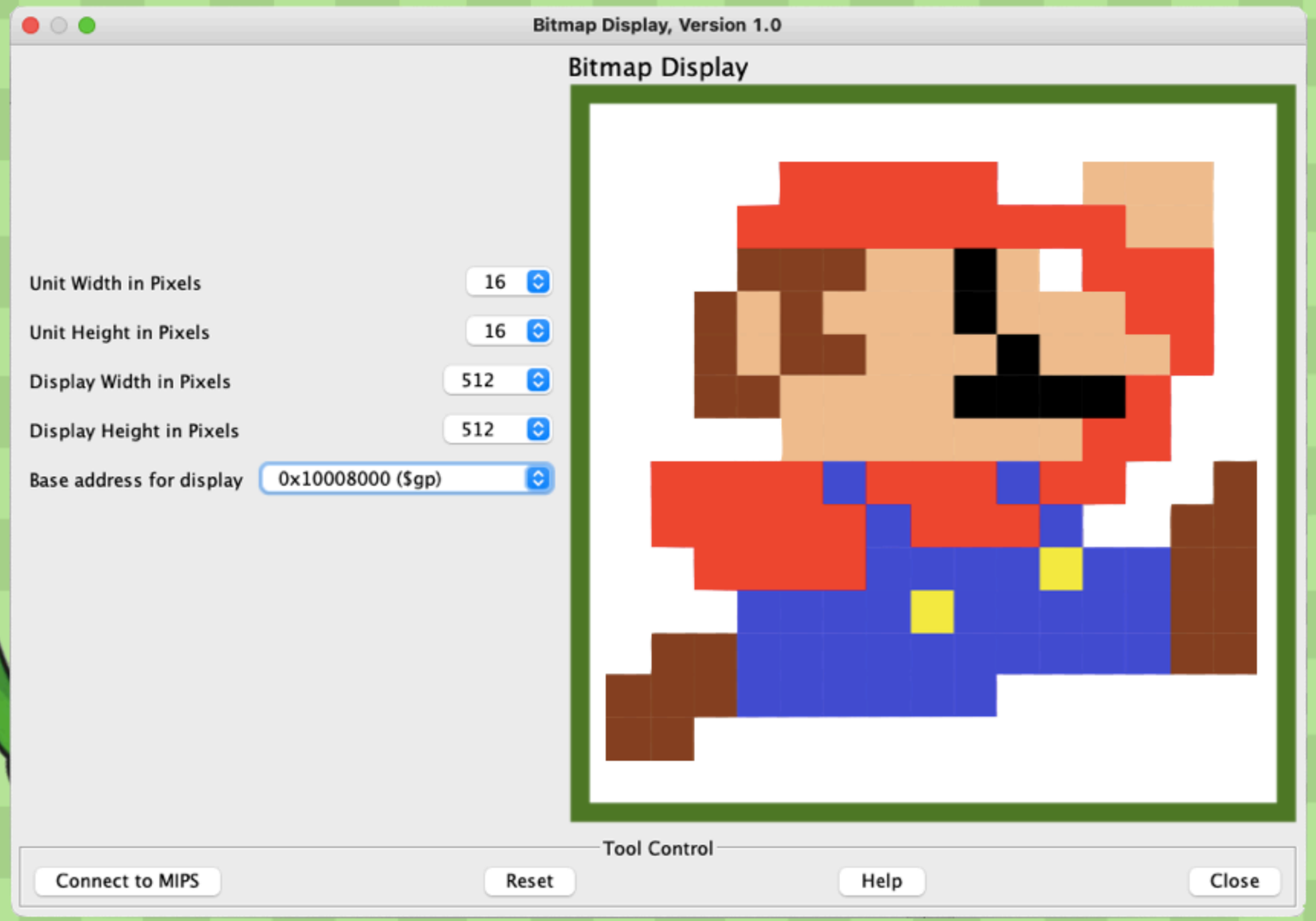
- drawing gameboard
- updating the gameboard
- receiving player input
- additional features

Drawing Game Board

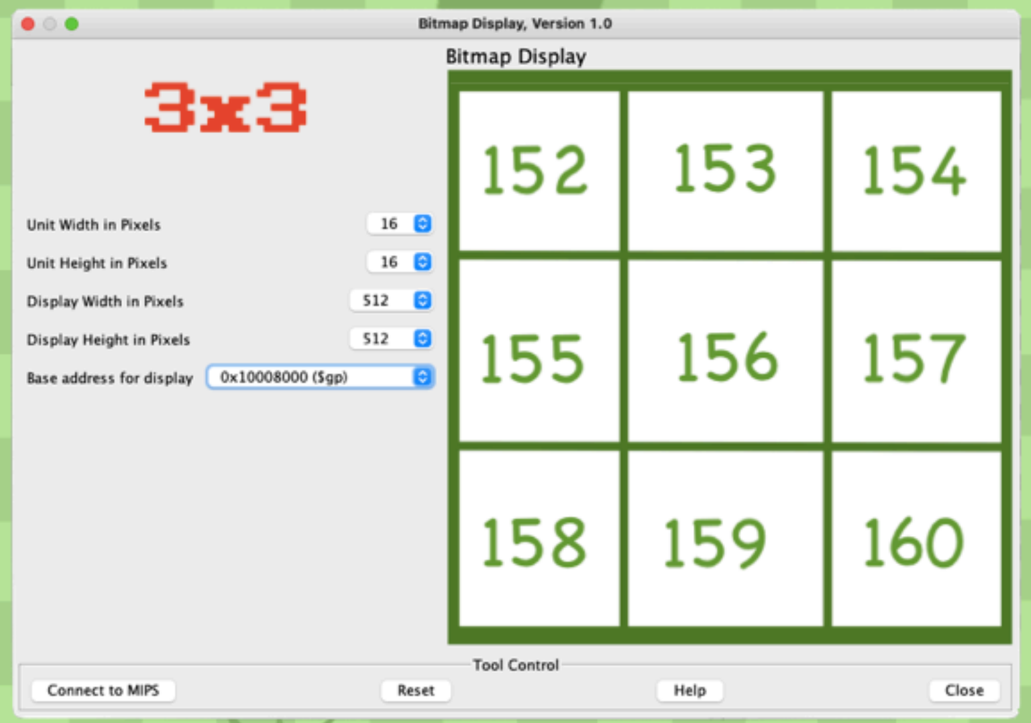
- background
- borders
- snake
- apples
- everything the player sees



Bitmap Display



Display in Memory

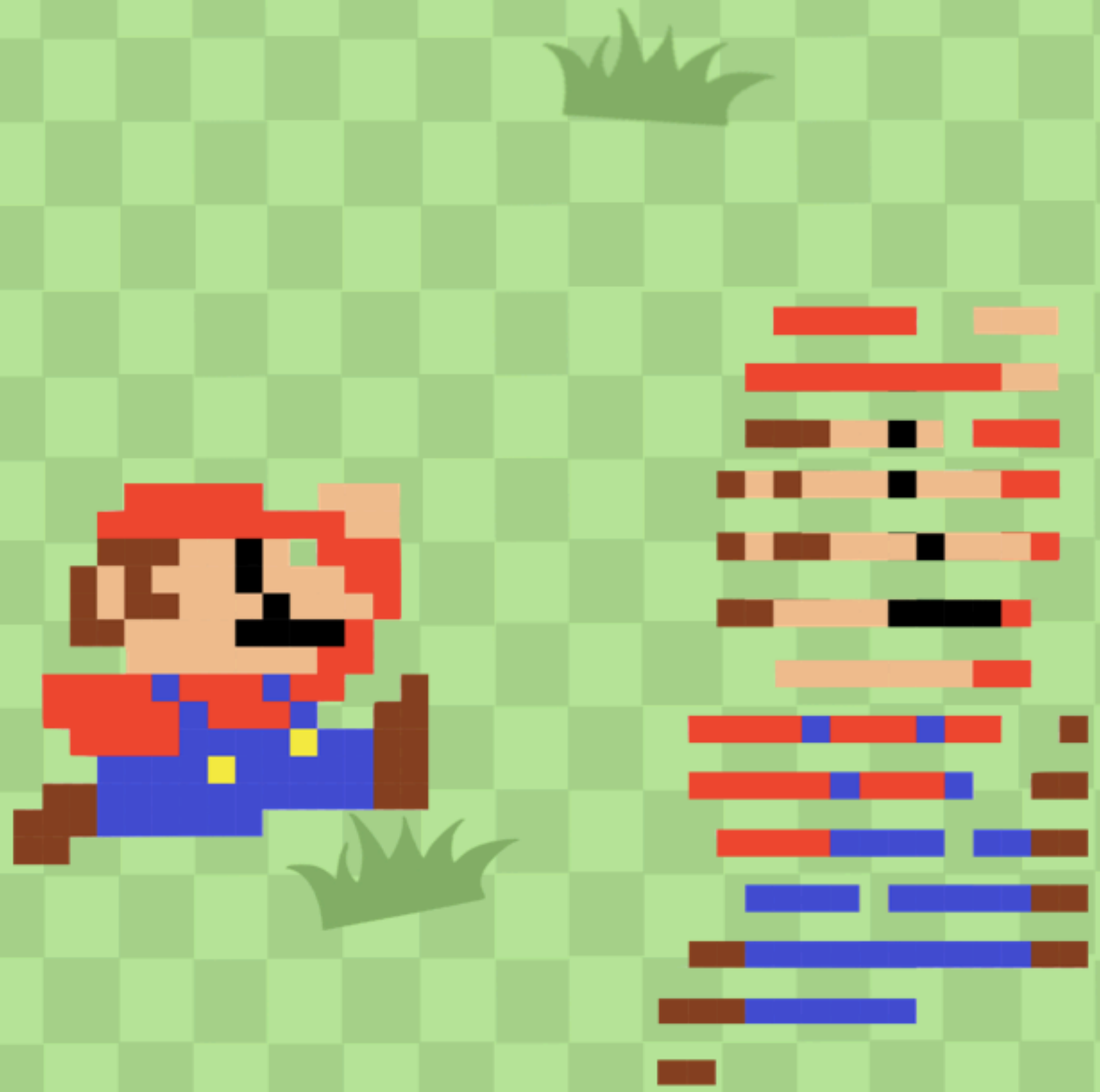


- $gp = global pointer
- refers to base address
- 152 in our example
- each pixel represented by one word in memory

$gp

152	153	154	155	156	157	158	159	160

Mario in Memory



Drawing a Pixel

$a0 = x $a1 = y

```
GetPixelAddress:
        lw $v0, screenWidth     #Store screen width into $v0
        mul $v0, $v0, $a1       #multiply by y position
        add $v0, $v0, $a0       #add the x position
        mul $v0, $v0, 4         #multiply by 4
        add $v0, $v0, $gp       #add global pointer from bitmap display
        jr $ra                  # return $v0

DrawPixel:
        sw $a1, ($a0)           #$a0 is the address of the pixel to color, $a1 is the color to draw the pixel
        jr $ra                  #return
```

- Every pixel is 1 word in memory
- Every word is 4 bytes
- An address refers to 1 byte
- The first byte is the pixel's color

Updating Game Board

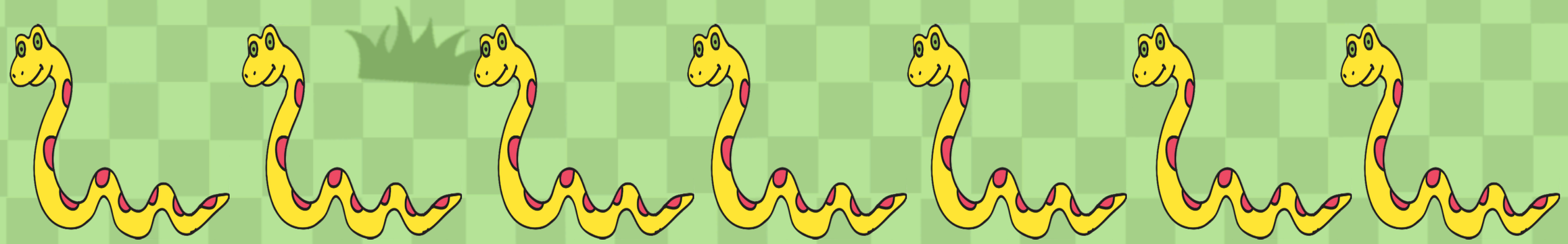
- moving the snake
- turning the snake
- drawing new food
- checking for collisions

system call 32 for sleep

```
li $v0, 32                    #syscall value for sleep
la $a0, 60                    #sleep for 60 milliseconds
syscall                       #program will sleep for whatever $a0 is set to
```

- call after every game board update
- if $a0 is 60, the program will sleep for 60 milliseconds between updates
- 1000 milliseconds in a second, so this will give 16 frames per second
- our snake moves 1 tile per frame
- the shorter the program sleeps between frames, the faster the snake moves

Moving the Snake

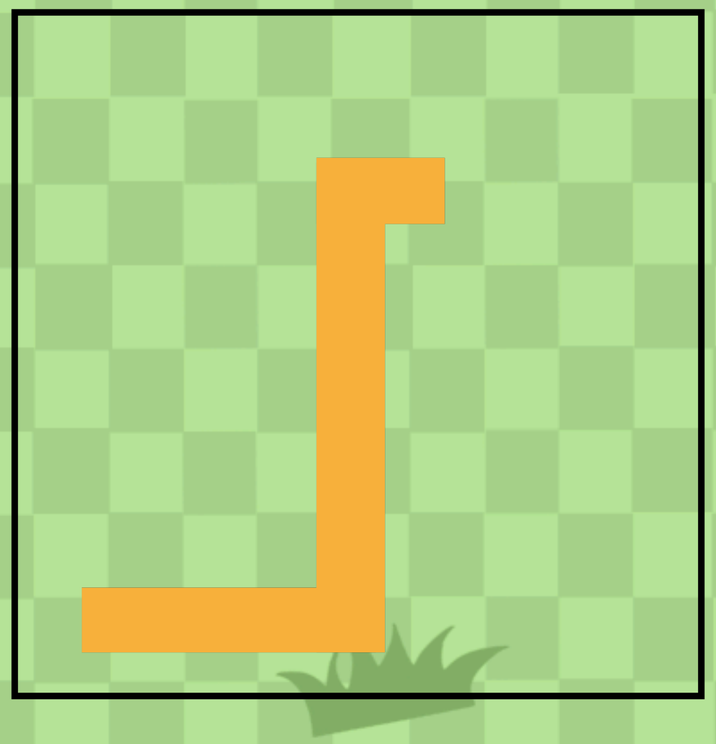


-the snake's body and tail must follow the same path taken by the head

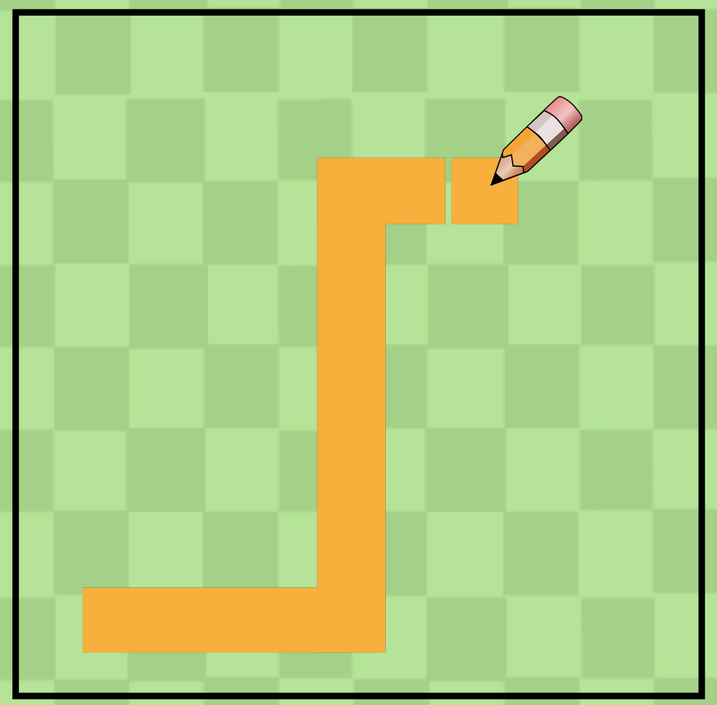
-we use two arrays; one keeps track of the head's direction changes and the other keeps track of the coordinates those changes happened

Moving the Snake

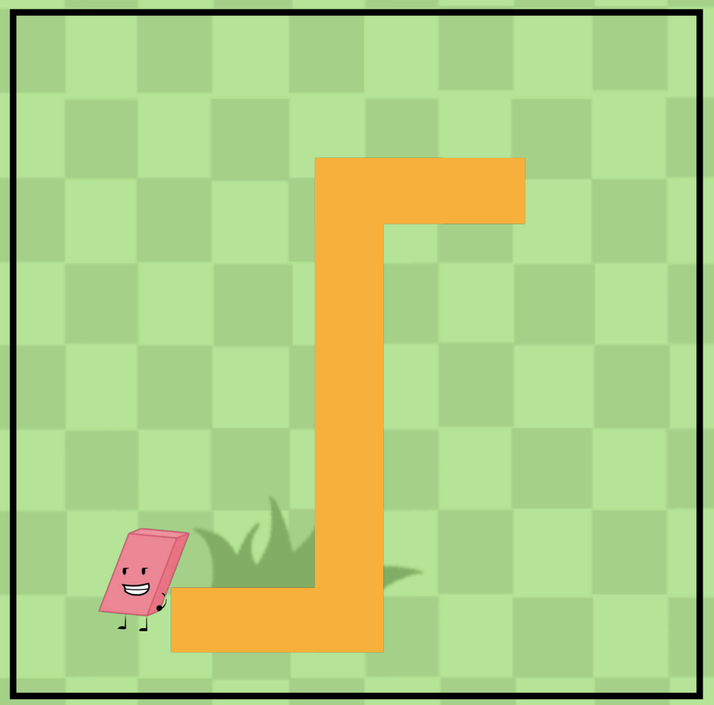
1. Snake



2. Draw head



3. Erase Tail

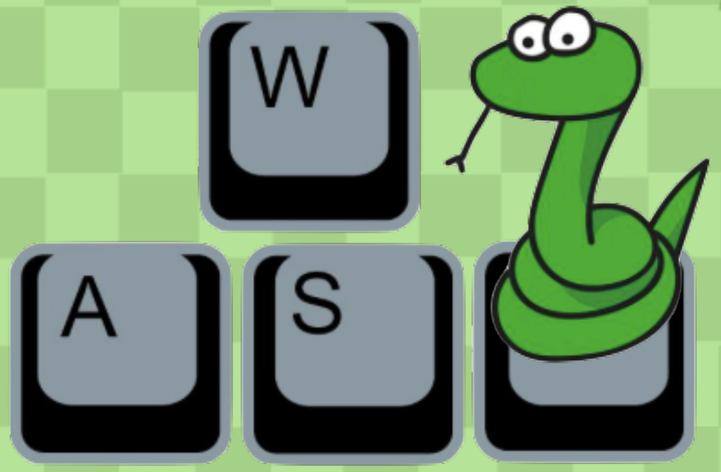


-if the snake consumed an apple, we simply don't erase the tail

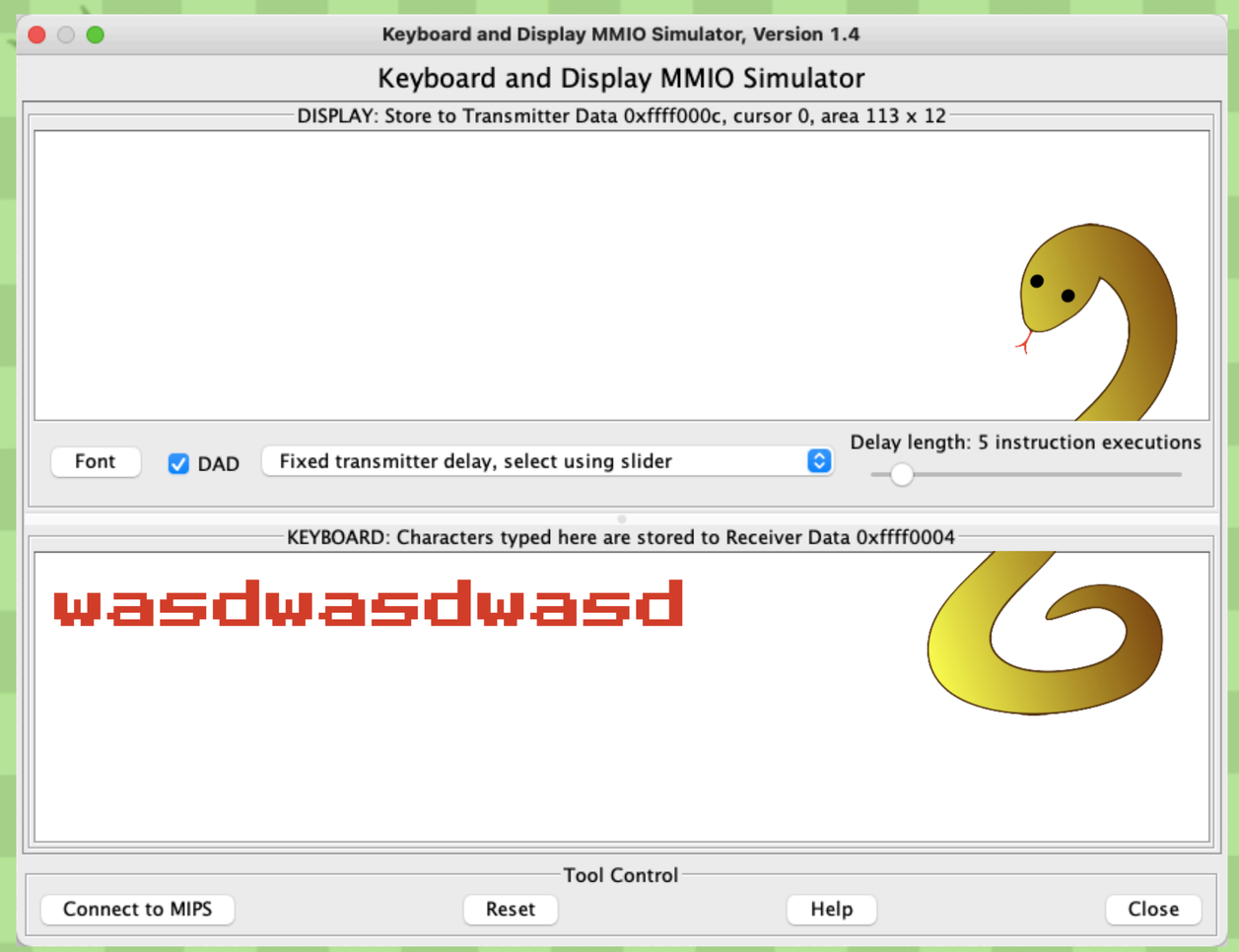
Receiving Player Input

-this is how the player interacts with the game

-'w' 'a' 's' and 'd' for up left down and right



Keyboard and Display MMIO Simulator



ALL character input is saved to address 0xFFFF0004

Check For Input

w = 100

a = 97

s = 115

d = 100

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Change Direction

Additional Features

- score
- cheat codes
- sounds
- colors and aesthetic

Score

-made a score counter to update whenever snake consumes an apple

-did not have time to make live score counter

-used system call 56 to display score after game

```
li $v0, 56                  #syscall value for java prompt
la $a0, gameOver            #get message
lw $a1, score               #get score
syscall
```

Cheat Codes

-wanted to add a way to change the snake's speed

-used '+' to increase speed, '-' to decrease speed, and '=' to reset speed

-ASCII values

+	=	43
-	=	45
=	=	61

Sound

Using System Call 31, we added sounds for eating food, game over, and game restart

```
#game restart sound
li $v0, 31                  #system call for sound
li $a0, 45                  #type of noise
li $a1, 400                 #DURATION
li $a2, 13                  #instrument number
li $a3, 127                 #volume
syscall
```

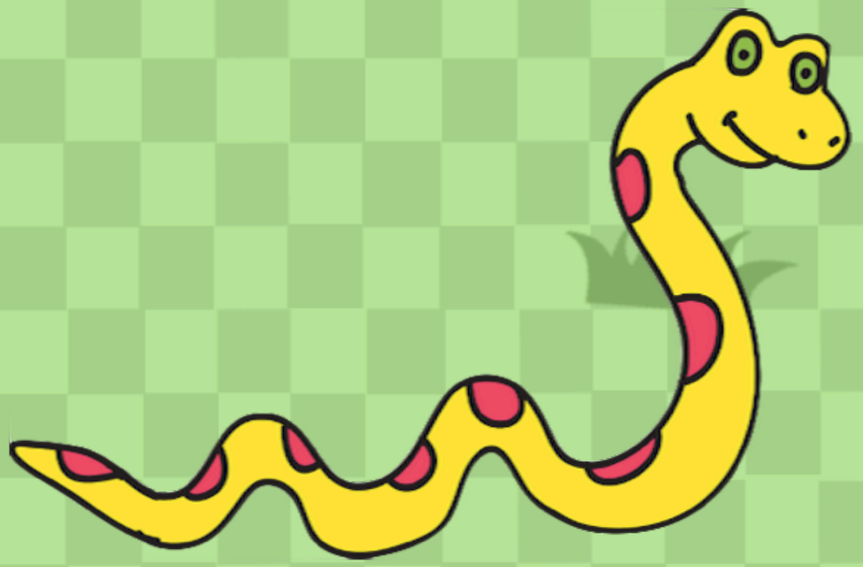
Design

-used hex values for colors

```
#Colors
snakeColor:     .word 0xC29979      #Set snake color
bgColor:        .word 0xCDEAC0      #Set background color
gridColor:      .word 0xB6E697      #set grid color
borderColor:    .word 0x50723C      #Set wall color
appleColor:     .word 0xE7471D      #Set apple color
saveColor:      .word 0xCDEAC0      #mutable color
```

-created a checkered background (huge headache)

-game size 32 x 32 pixels



Demo!