

Textures
Texture Atlas
Sprite Sheets
Fonts

Textures

(Review)

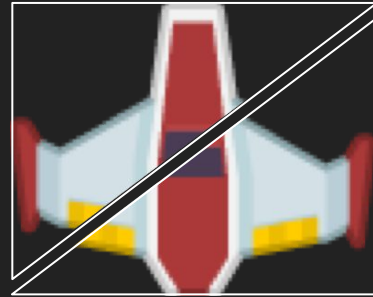
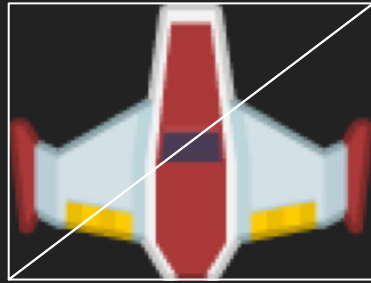
Texture Coordinates



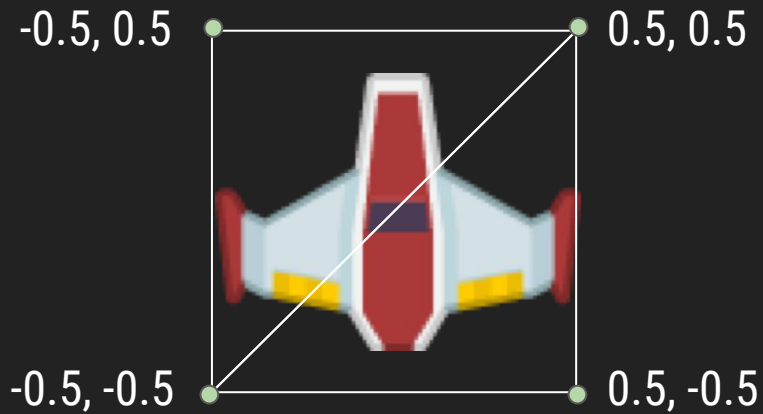
Texture coordinates are referred to as **UV** coordinates (X, Y and Z were already taken) :)

Notice the range from 0.0 to 1.0 and not by pixels.

2D Sprite Made of 2 Triangles

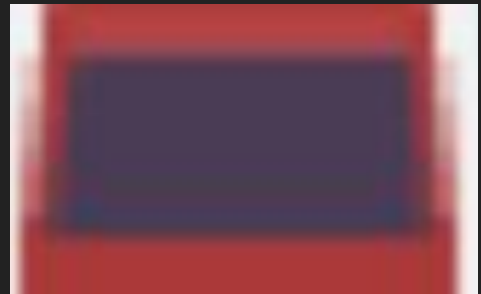


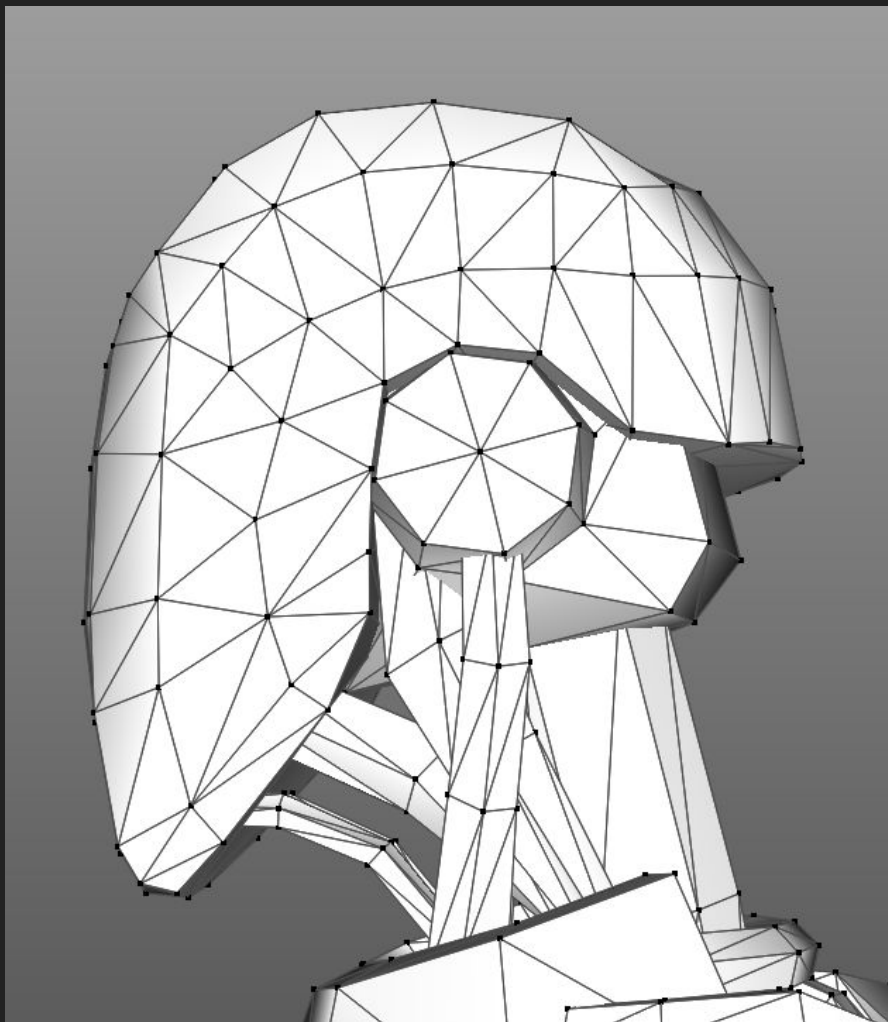
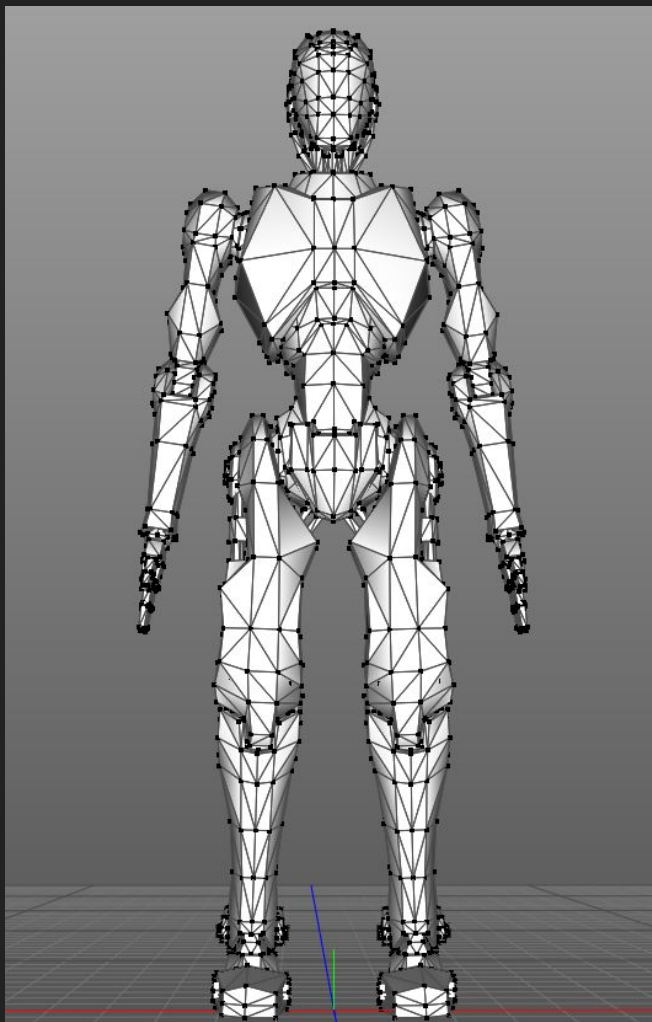
Need to match vertices to UV coordinates

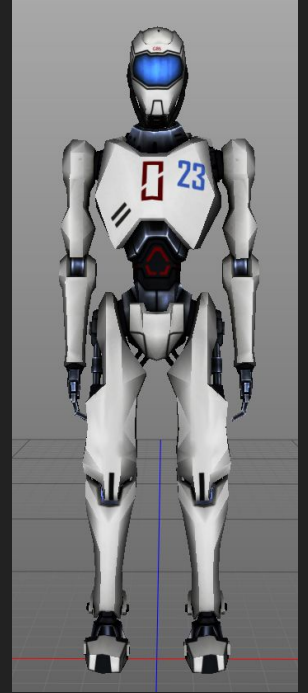
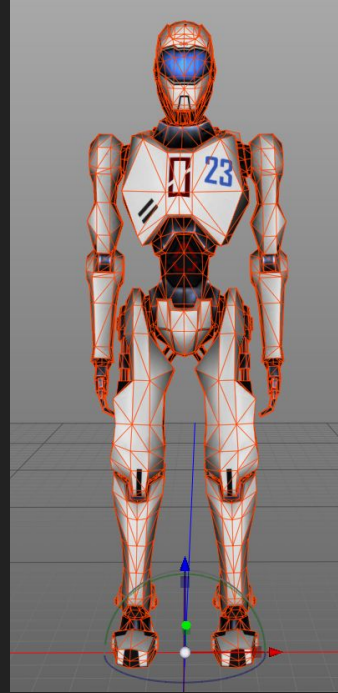
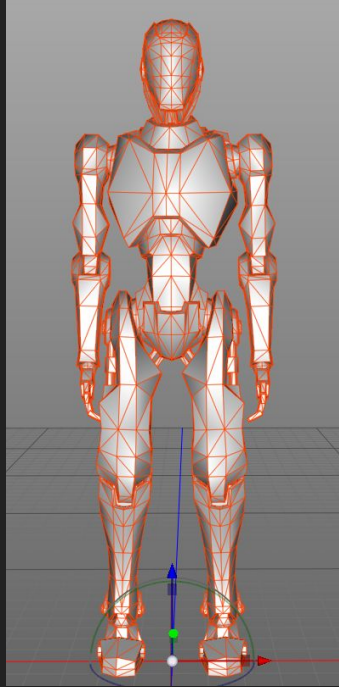


```
float vertices[] = {-0.5, -0.5, 0.5, -0.5, 0.5, 0.5, -0.5, -0.5, 0.5, 0.5, -0.5, 0.5};  
float texCoords[] = {0.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0};
```

Portion of Texture





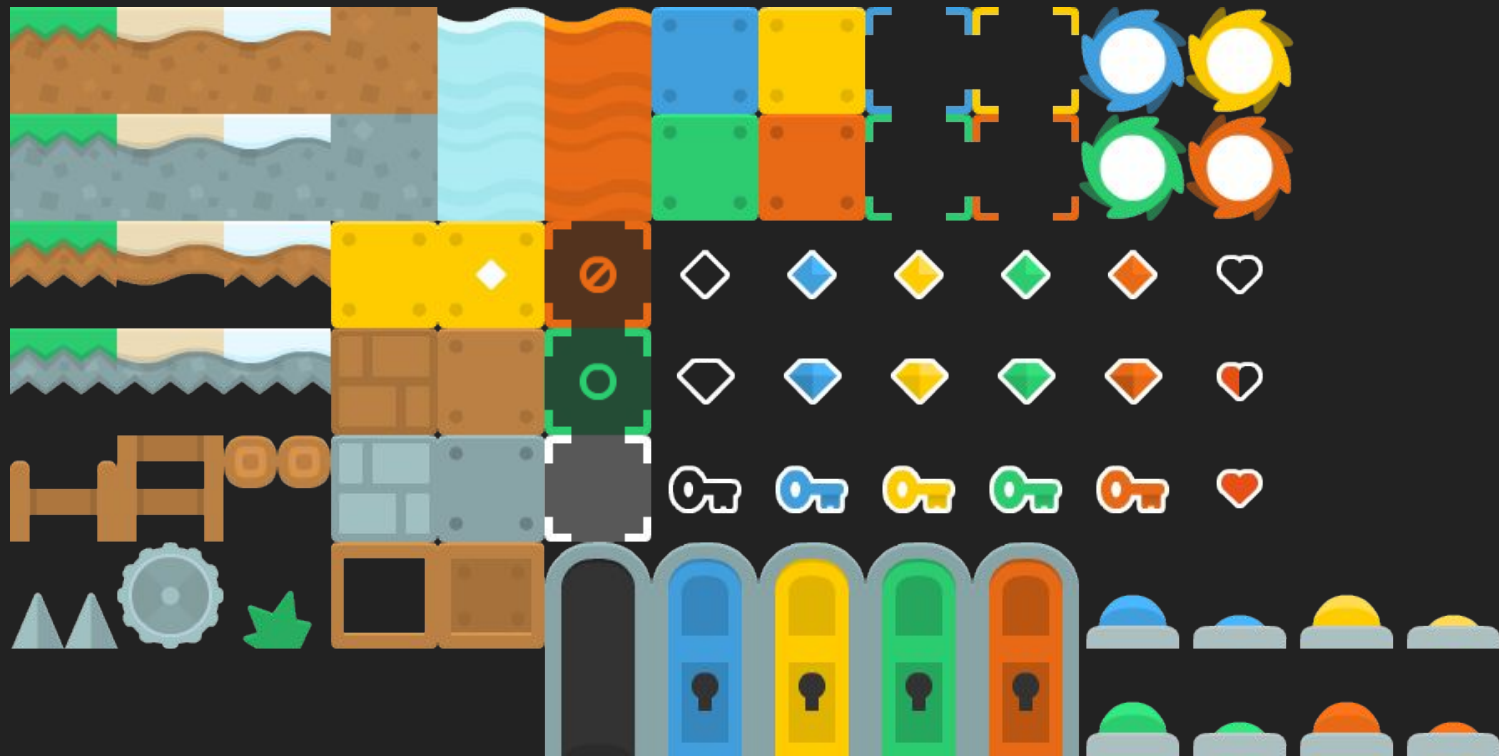


Texture Atlases

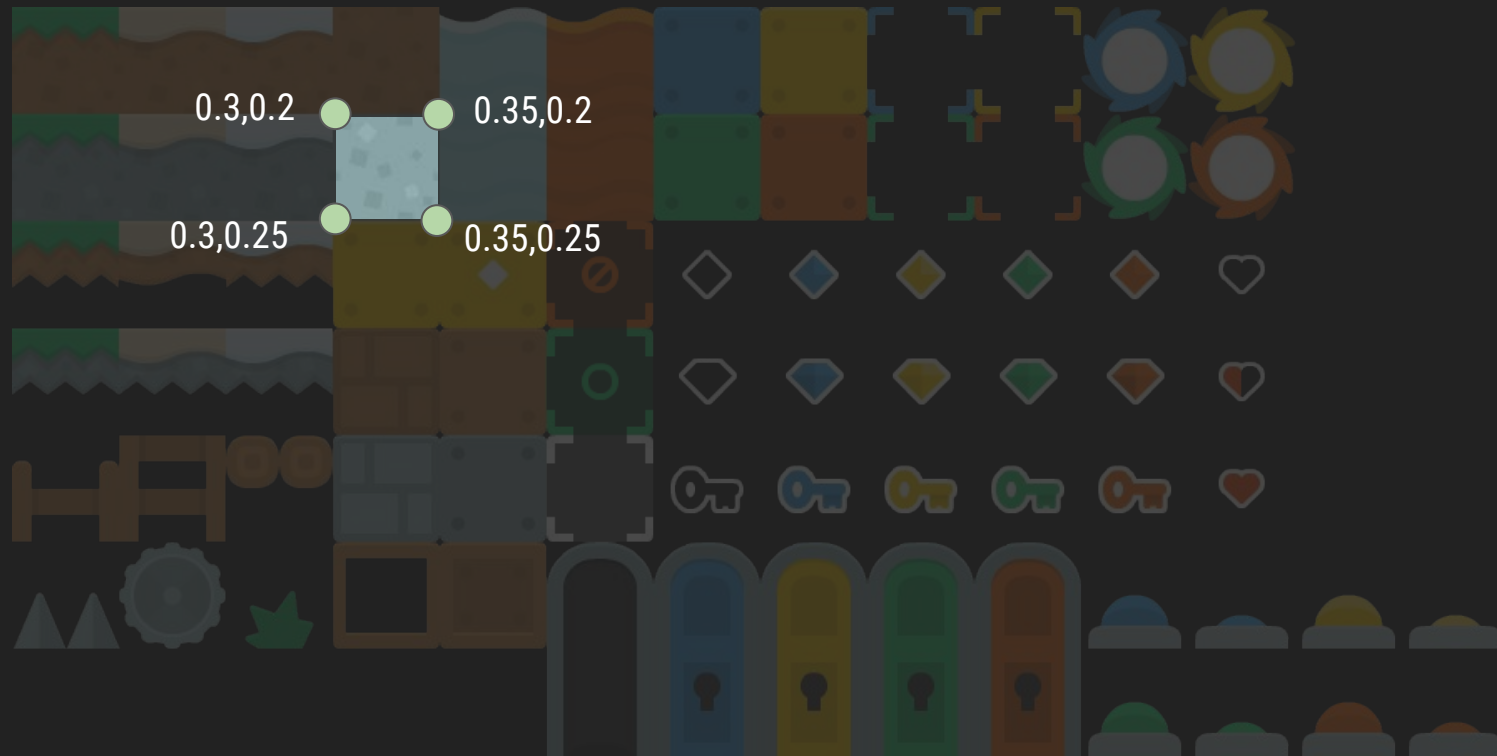
(Multiple Sprites in a Single Texture)

A large grid of Angry Birds characters and objects, including various types of pigs, birds, eggs, and power-ups, arranged in a structured layout. The grid is composed of many small, identical icons of different characters and items, such as green pigs, red birds, yellow birds, and various power-ups like the bomb, the bomb bird, and the bomb pig. The icons are arranged in a grid that is approximately 10 columns wide and 10 rows high. The background is a solid black color. The icons are arranged in a way that suggests a collection or a catalog of the game's characters and items. The icons are arranged in a grid that is approximately 10 columns wide and 10 rows high. The background is a solid black color. The icons are arranged in a way that suggests a collection or a catalog of the game's characters and items.

Tileset



Tileset



3D Game Example



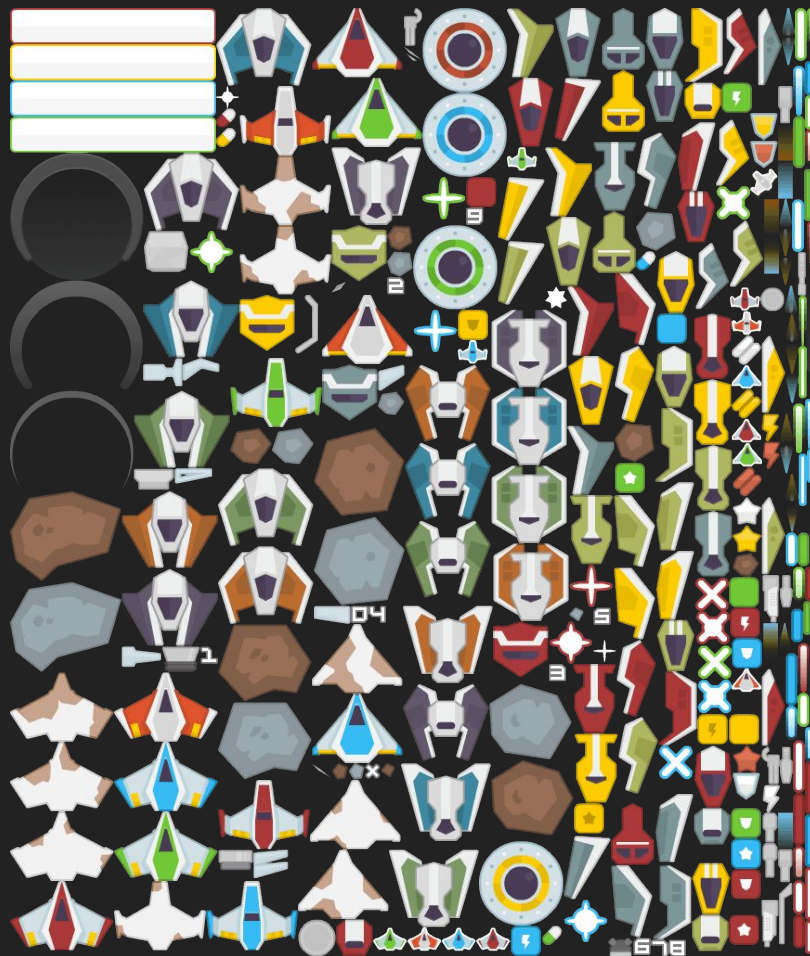
Fonts!

! " # \$ % & ' () * + , - . /
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [\] ^ _
` a b c d e f g h i j k l m n o
p q r s t u v w x y z { | } ~
€ ¤ , f „ … † ‡ ^ ‰ Š ‹ Œ ¤ Ž ¤
¤ ‘ ’ “ ” • – — - ™ š › œ ¤ ž Ÿ
ı ¢ £ ¤ ¥ ¦ § ¨ © ª « ¬ ® ¯
° ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½ ¾ ¿
À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï
Ð Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ ß
à á â ã ä å æ ç è é ê ë ì í î ï
đ ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ

We are going to work
with evenly spaced
texture atlases.

Not Evenly Spaced

(you can not make a
uniform grid on this)



Evenly Spaced



Evenly Spaced



Evenly Spaced



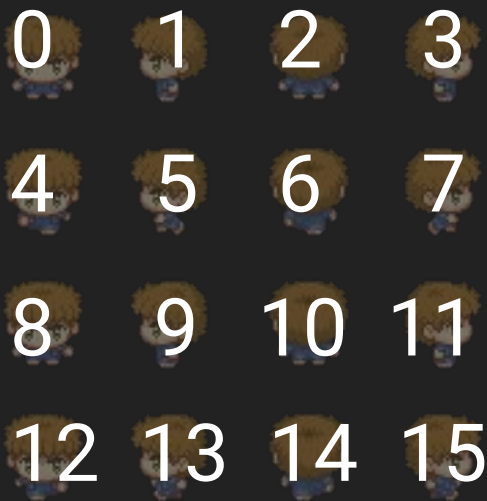
! " # \$ % & ' () * + , - . /
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [\] ^ _
` a b c d e f g h i j k l m n o
p q r s t u v w x y z { | } ~

Drawing a Single Sprite

(From a Texture Atlas)



george_0.png



We need the UV coordinates of the individual sprite.



A 2x4 grid of cartoon characters. Each character has a large white number on its chest. The numbers are 0, 1, 2, 3 in the top row and 4, 5, 6, 7 in the bottom row. The characters are arranged in a regular grid pattern.

u,v

7

```
float u = (float)(index % cols) / (float)cols;
float v = (float)(index / cols) / (float)rows;

float width = 1.0f / (float)cols;
float height = 1.0f / (float)rows;

float texCoords[] = { u, v + height, u + width, v + height, u + width, v,
                      u, v + height, u + width, v, u, v };

float vertices[] = { -0.5, -0.5, 0.5, -0.5, 0.5, 0.5,
                    -0.5, -0.5, 0.5, 0.5, -0.5, 0.5 };
```

```

void Entity::DrawSpriteFromTextureAtlas(ShaderProgram *program, int index)
{
    float u = (float)(index % cols) / (float)cols;
    float v = (float)(index / cols) / (float)rows;

    float width = 1.0f / (float)cols;
    float height = 1.0f / (float)rows;

    float texCoords[] = { u, v + height, u + width, v + height, u + width, v,
                          u, v + height, u + width, v, u, v };

    float vertices[] = { -0.5, -0.5,  0.5, -0.5,  0.5, 0.5,
                        -0.5, -0.5,  0.5, 0.5,  -0.5, 0.5 };

    glBindTexture(GL_TEXTURE_2D, textureID);

    glVertexAttribPointer(program->positionAttribute, 2, GL_FLOAT, false, 0, vertices);
    glEnableVertexAttribArray(program->positionAttribute);

    glVertexAttribPointer(program->texCoordAttribute, 2, GL_FLOAT, false, 0, texCoords);
    glEnableVertexAttribArray(program->texCoordAttribute);

    glDrawArrays(GL_TRIANGLES, 0, 6);

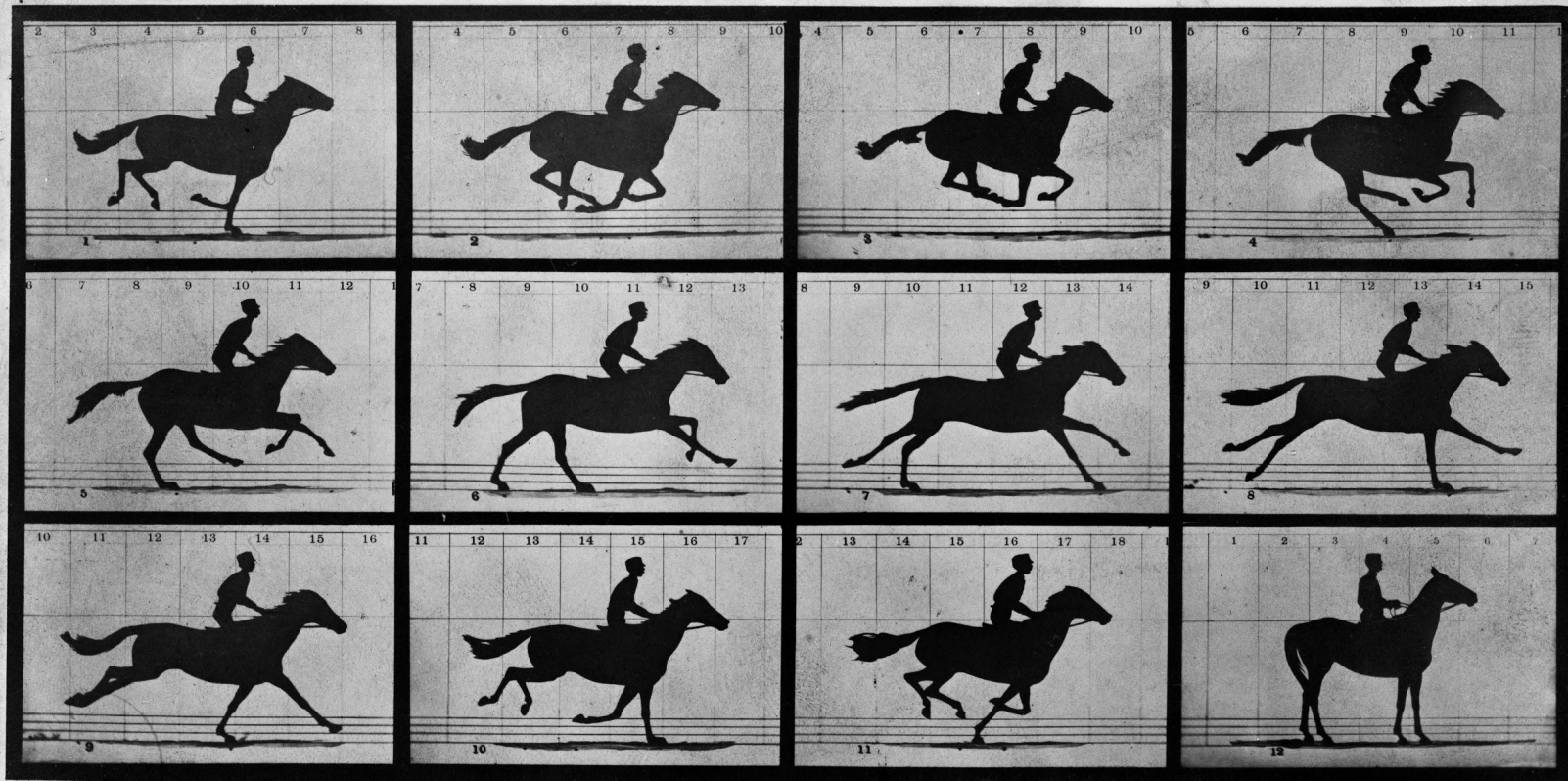
    glDisableVertexAttribArray(program->positionAttribute);
    glDisableVertexAttribArray(program->texCoordAttribute);
}

void Entity::Render(ShaderProgram *program) {
    glm::mat4 modelMatrix = glm::mat4(1.0f);
    modelMatrix = glm::translate(modelMatrix, position);
    program->SetModelMatrix(modelMatrix);

    DrawSpriteFromTextureAtlas(program, 7);
}

```

Animation!



Copyright, 1878, by MUYBRIDGE.

MORSE'S Gallery, 417 Montgomery St., San Francisco.

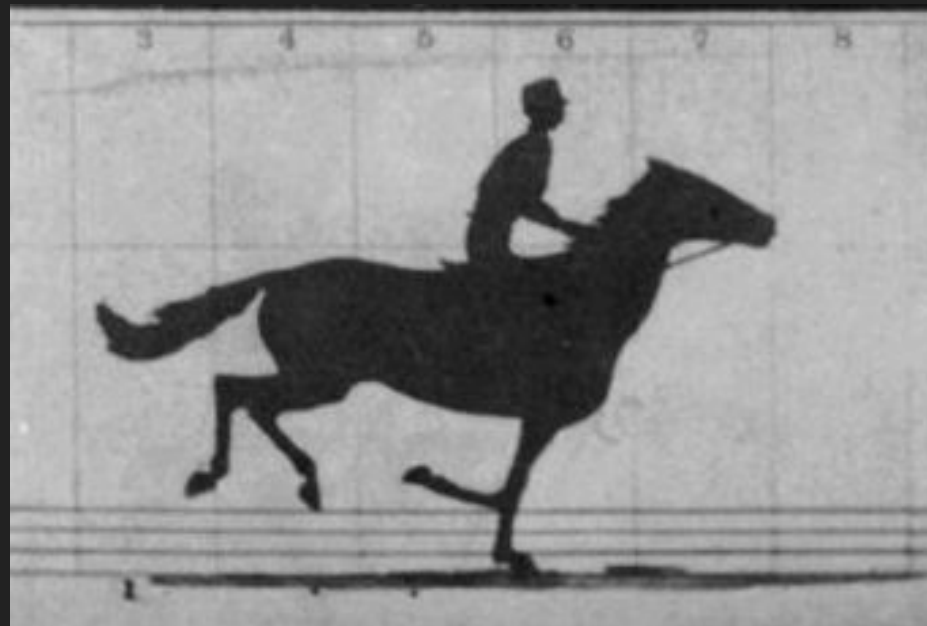
THE HORSE IN MOTION.

Illustrated by
MUYBRIDGE.

AUTOMATIC ELECTRO-PHOTOGRAPH.

"SALLIE GARDNER," owned by LELAND STANFORD; running at a 1.40 gait over the Palo Alto track, 19th June, 1878.

The negatives of these photographs were made at intervals of twenty-seven inches of distance, and about the twenty-fifth part of a second of time; they illustrate consecutive positions assumed in each twenty-seven inches of progress during a single stride of the mare. The vertical lines were twenty-seven inches apart; the horizontal lines represent elevations of four inches each. The exposure of each negative was less than the two-thousandth part of a second.





Define indices of animation: (3, 7, 11, 15)

Have a timer.

Go to next frame when timer hits value.

If last frame (go to first) - looping.

```
player.textureID = LoadTexture("george_0.png");  
player.cols = 4;  
player.rows = 4;  
player.animIndices = new int[4] {3, 7, 11, 15};  
player.animFrames = 4;
```

```
void Entity::Update(float deltaTime)
{
    position += movement * speed * deltaTime;

    animTime += deltaTime;
    if (animTime >= 0.25f)
    {
        animTime = 0.0f;
        animIndex++;
        if (animIndex >= animFrames)
        {
            animIndex = 0;
        }
    }
}
```

```
void Entity::Render(ShaderProgram *program) {
    glm::mat4 modelMatrix = glm::mat4(1.0f);
    modelMatrix = glm::translate(modelMatrix, position);
    program->SetModelMatrix(modelMatrix);

    DrawSpriteFromTextureAtlas(program, animIndices[animIndex]);
}
```

Monospaced Font Rendering

	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
o	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y	z	{		}	~	□
€	□	,	f	„	...	†	‡	^	‰	Š	<	Œ	□	Ž	□
□	‘	’	“	”	•	—	—	~	™	š	>	œ	□	ž	ÿ
	i	¢	£	¤	¥		§	¨	©	ª	«	¬	–	®	¯
°	±	²	³	´	µ	¶	·	,	¹	º	»	¼	½	¾	¿
À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
đ	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

! " # \$ % & ' () * + , - . /
 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 @ A B C D E F G H I J K L M N O
 P Q R S T U V W X Y Z [\] ^ _
 ` a b c d e f g h i j k l m n o
 p q r s t u v w x y z { | } ~
 € ¤ , f „ … † ‡ ^ % ¯ Š ‹ Œ º Ž
 º ‘ ’ “ ” • – — ~ ™ š › œ º ž Ÿ
 ÿ i ¢ £ ¤ ¥ ¦ § ¨ © ª « ¬ ® ¯
 ° ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½ ¾ ¿
 À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï
 Ð Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ ß
 à á â ã ä å æ ç è é ê ë ì í î ï
 ð ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ

For each character in a string

- Draw 2 Triangles
- Use UV coordinates for character



Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

[illegible]

	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
0	1	2	3	4	5	6	7	8	9	:	:	<	=	>	?

@ A B C D E F G H I J K L M N O

P O R S T U V W X Y Z [\] ^

g b c d e f g h i j k l m n o

p q r s t u v w x y z { | } ~ □

€ ¤ £ ¢ ¥ ¦ § ¨ © ª « ¬ ® ¯ ° ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½ ¾

□ ‘ ’ “ ” ● — — ~ ™ § > ∞ □ † √

© 1994 by The McGraw-Hill Companies, Inc.

$$0 \quad + \quad 2 \quad 3 \quad ' \quad || \quad 9 \quad : \quad 1 \quad 0 \quad \infty \quad 1/2 \quad 1/3 \quad 3/4 \quad ;$$

À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï

Đ Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ü Ý Þ ß

2' 2' 2^ 2^ 2: 2° 3 4 5 5' 5^ 5: 6 6' 6^ 6: 7 7' 7^ 7: 8 8' 8^ 8: 9 9' 9^ 9: 10 10' 10^ 10: 11 11' 11^ 11: 12 12' 12^ 12: 13 13' 13^ 13: 14 14' 14^ 14: 15 15' 15^ 15: 16 16' 16^ 16: 17 17' 17^ 17: 18 18' 18^ 18: 19 19' 19^ 19: 20 20' 20^ 20: 21 21' 21^ 21: 22 22' 22^ 22: 23 23' 23^ 23: 24 24' 24^ 24: 25 25' 25^ 25: 26 26' 26^ 26: 27 27' 27^ 27: 28 28' 28^ 28: 29 29' 29^ 29: 30 30' 30^ 30: 31 31' 31^ 31: 32 32' 32^ 32: 33 33' 33^ 33: 34 34' 34^ 34: 35 35' 35^ 35: 36 36' 36^ 36: 37 37' 37^ 37: 38 38' 38^ 38: 39 39' 39^ 39: 40 40' 40^ 40: 41 41' 41^ 41: 42 42' 42^ 42: 43 43' 43^ 43: 44 44' 44^ 44: 45 45' 45^ 45: 46 46' 46^ 46: 47 47' 47^ 47: 48 48' 48^ 48: 49 49' 49^ 49: 50 50' 50^ 50: 51 51' 51^ 51: 52 52' 52^ 52: 53 53' 53^ 53: 54 54' 54^ 54: 55 55' 55^ 55: 56 56' 56^ 56: 57 57' 57^ 57: 58 58' 58^ 58: 59 59' 59^ 59: 60 60' 60^ 60: 61 61' 61^ 61: 62 62' 62^ 62: 63 63' 63^ 63: 64 64' 64^ 64: 65 65' 65^ 65: 66 66' 66^ 66: 67 67' 67^ 67: 68 68' 68^ 68: 69 69' 69^ 69: 70 70' 70^ 70: 71 71' 71^ 71: 72 72' 72^ 72: 73 73' 73^ 73: 74 74' 74^ 74: 75 75' 75^ 75: 76 76' 76^ 76: 77 77' 77^ 77: 78 78' 78^ 78: 79 79' 79^ 79: 80 80' 80^ 80: 81 81' 81^ 81: 82 82' 82^ 82: 83 83' 83^ 83: 84 84' 84^ 84: 85 85' 85^ 85: 86 86' 86^ 86: 87 87' 87^ 87: 88 88' 88^ 88: 89 89' 89^ 89: 90 90' 90^ 90: 91 91' 91^ 91: 92 92' 92^ 92: 93 93' 93^ 93: 94 94' 94^ 94: 95 95' 95^ 95: 96 96' 96^ 96: 97 97' 97^ 97: 98 98' 98^ 98: 99 99' 99^ 99: 100 100' 100^ 100: 101 101' 101^ 101: 102 102' 102^ 102: 103 103' 103^ 103: 104 104' 104^ 104: 105 105' 105^ 105: 106 106' 106^ 106: 107 107' 107^ 107: 108 108' 108^ 108: 109 109' 109^ 109: 110 110' 110^ 110: 111 111' 111^ 111: 112 112' 112^ 112: 113 113' 113^ 113: 114 114' 114^ 114: 115 115' 115^ 115: 116 116' 116^ 116: 117 117' 117^ 117: 118 118' 118^ 118: 119 119' 119^ 119: 120 120' 120^ 120: 121 121' 121^ 121: 122 122' 122^ 122: 123 123' 123^ 123: 124 124' 124^ 124: 125 125' 125^ 125: 126 126' 126^ 126: 127 127' 127^ 127: 128 128' 128^ 128: 129 129' 129^ 129: 130 130' 130^ 130: 131 131' 131^ 131: 132 132' 132^ 132: 133 133' 133^ 133: 134 134' 134^ 134: 135 135' 135^ 135: 136 136' 136^ 136: 137 137' 137^ 137: 138 138' 138^ 138: 139 139' 139^ 139: 140 140' 140^ 140: 141 141' 141^ 141: 142 142' 142^ 142: 143 143' 143^ 143: 144 144' 144^ 144: 145 145' 145^ 145: 146 146' 146^ 146: 147 147' 147^ 147: 148 148' 148^ 148: 149 149' 149^ 149: 150 150' 150^ 150: 151 151' 151^ 151: 152 152' 152^ 152: 153 153' 153^ 153: 154 154' 154^ 154: 155 155' 155^ 155: 156 156' 156^ 156: 157 157' 157^ 157: 158 158' 158^ 158: 159 159' 159^ 159: 160 160' 160^ 160: 161 161' 161^ 161: 162 162' 162^ 162: 163 163' 163^ 163: 164 164' 164^ 164: 165 165' 165^ 165: 166 166' 166^ 166: 167 167' 167^ 167: 168 168' 168^ 168: 169 169' 169^ 169: 170 170' 170^ 170: 171 171' 171^ 171: 172 172' 172^ 172: 173 173' 173^ 173: 174 174' 174^ 174: 175 175' 175^ 175: 176 176' 176^ 176: 177 177' 177^ 177: 178 178' 178^ 178: 179 179' 179^ 179: 180 180' 180^ 180: 181 181' 181^ 181: 182 182' 182^ 182: 183 183' 183^ 183: 184 184' 184^ 184: 185 185' 185^ 185: 186 186' 186^ 186: 187 187' 187^ 187: 188 188' 188^ 188: 189 189' 189^ 189: 190 190' 190^ 190: 191 191' 191^ 191: 192 192' 192^ 192: 193 193' 193^ 193: 194 194' 194^ 194: 195 195' 195^ 195: 196 196' 196^ 196: 197 197' 197^ 197: 198 198' 198^ 198: 199 199' 199^ 199: 200 200' 200^ 200: 201 201' 201^ 201: 202 202' 202^ 202: 203 203' 203^ 203: 204 204' 204^ 204: 205 205' 205^ 205: 206 206' 206^ 206: 207 207' 207^ 207: 208 208' 208^ 208: 209 209' 209^ 209: 210 210' 210^ 210: 211 211' 211^ 211: 212 212' 212^ 212: 213 213' 213^ 213: 214 214' 214^ 214: 215 215' 215^ 215: 216 216' 216^ 216: 217 217' 217^ 217: 218 218' 218^ 218: 219 219' 219^ 219: 220 220' 220^ 220: 221 221' 221^ 221: 222 222' 222^ 222: 223 223' 223^ 223: 224 224' 224^ 224: 225 225' 225^ 225: 226 226' 226^ 226: 227 227' 227^ 227: 228 228' 228^ 228: 229 229' 229^ 229: 230 230' 230^ 230: 231 231' 231^ 231: 232 232' 232^ 232: 233 233' 233^ 233: 234 234' 234^ 234: 235 235' 235^ 235: 236 236' 236^ 236: 237 237' 237^ 237: 238 238' 238^ 238: 239 239' 239^ 239: 240 240

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 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 10

00J'' 0057012 1*

01234567891011121314151617181920212223242526272829303132333435363738394041424344454647484950515253545556575859606162636465666768697071727374757677787980818283848586878889909192939495969798991001011021031041051061071081091101111121131141151161171181191201211221231241251261271281291301311321331341351361371381391401411421431441451461471481491501511521531541551561571581591601611621631641651661671681691701711721731741751761771781791801811821831841851861871881891901911921931941951961971981992002012022032042052062072082092102112122132142152162172182192202212222232242252262272282292302312322332342352362372382392402412422432442452462472482492502512522532542552562572582592602612622632642652662672682692702712722732742752762772782792802812822832842852862872882892902912922932942952962972982993003013023033043053063073083093103113123133143153163173183193203213223233243253263273283293303313323333343353363373383393403413423433443453463473483493503513523533543553563573583593603613623633643653663673683693703713723733743753763773783793803813823833843853863873883893903913923933943953963973983994004014024034044054064074084094104114124134144154164174184194204214224234244254264274284294304314324334344354364374384394404414424434444454464474484494504514524534544554564574584594604614624634644654664674684694704714724734744754764774784794804814824834844854864874884894904914924934944954964974984995005015025035045055065075085095105115125135145155165175185195205215225235245255265275285295305315325335345355365375385395405415425435445455465475485495505515525535545555565575585595605615625635645655665675685695705715725735745755765775785795805815825835845855865875885895905915925935945955965975985996006016026036046056066076086096106116126136146156166176186196206216226236246256266276286296306316326336346356366376386396406416426436446456466476486496506516526536546556566576586596606616626636646656666676686696706716726736746756766776786796806816826836846856866876886896906916926936946956966976986997007017027037047057067077087097107117127137147157167177187197207217227237247257267277287297307317327337347357367377387397407417427437447457467477487497507517527537547557567577587597607617627637647657667677687697707717727737747757767777787797807817827837847857867877887897907917927937947957967977987998008018028038048058068078088098108118128138148158168178188198208218228238248258268278288298308318328338348358368378388398408418428438448458468478488498508518528538548558568578588598608618628638648658668678688698708718728738748758768778788798808818828838848858868878888898908918928938948958968978988999009019029039049059069079089099109119129139149159169179189199209219229239249259269279289299309319329339349359369379389399409419429439449459469479489499509519529539549559569579589599609619629639649659669679689699709719729739749759769779789799809819829839849859869879889899909919929939949959969979989991000100110021003100410051006100710081009101010111012101310141015101610171018101910201021102210231024102510261027102810291030103110321033103410351036103710381039104010411042104310441045104610471048104910501051105210531054105510561057105810591060106110621063106410651066106710681069107010711072107310741075107610771078107910801081108210831084108510861087108810891090109110921093109410951096109710981099110011011102110311041105110611071108110911101111111211131114111511161117111811191120112111221123112411251126112711281129113011311132113311341135113611371138113911401141114211431144114511461147114811491150115111521153115411551156115711581159116011611162116311641165116611671168116911701171117211731174117511761177117811791180118111821183118411851186118711881189119011911192119311941195119611971198119912001201120212031204120512061207120812091210121112121213121412151216121712181219122012211222122312241225122612271228122912301231123212331234123512361237123812391240124112421243124412451246124712481249125012511252125312541255125612571258125912601261126212631264126512661267126812691270127112721273127412751276127712781279128012811282128312841285128612871288128912901291129212931294129512961297129812991300

NO	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	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
----	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

530557 *See* 530556

2000

Abstract

Downloaded from ascelibrary.org by University of California, San Diego on 06/01/15. Copyright ASCE, For All Rights Reserved, No part of this document may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or by any information storage or retrieval system, without permission in writing from ASCE.

[illegible]

7-10-20

Depending on the font texture, you may have to shift the character value.

```

void DrawText(ShaderProgram *program, GLuint fontTextureID, std::string text, float size, float spacing, glm::vec3 position) {

    float width = 1.0f / 16.0f;
    float height = 1.0f / 16.0f;

    std::vector<float> vertices;
    std::vector<float> texCoords;

    for (int i = 0; i < text.size(); i++)
    {
        int index = (int)text[i];

        float u = (float)(index % 16) / 16.0f;
        float v = (float)(index / 16) / 16.0f;

        texCoords.insert(texCoords.end(), { u, v + height, u + width, v + height, u + width, v,
                                             u, v + height, u + width, v, u, v } );

        float offset = (size + spacing) * i;
        vertices.insert(vertices.end(), { offset + (-0.5f * size), (-0.5f * size),
                                          offset + (0.5f * size), (-0.5f * size),
                                          offset + (0.5f * size), (0.5f * size),
                                          offset + (-0.5f * size), (-0.5f * size),
                                          offset + (0.5f * size), (0.5f * size),
                                          offset + (-0.5f * size), (0.5f * size) });
    }
}

```

After the vertices and texCoords are setup, we can draw using familiar code.

```
glm::mat4 modelMatrix = glm::mat4(1.0f);
modelMatrix = glm::translate(modelMatrix, position);
program->SetModelMatrix(modelMatrix);

glBindTexture(GL_TEXTURE_2D, fontTextureID);
glVertexAttribPointer(program->positionAttribute, 2, GL_FLOAT, false, 0, vertices.data());
glEnableVertexAttribArray(program->positionAttribute);

glVertexAttribPointer(program->texCoordAttribute, 2, GL_FLOAT, false, 0, texCoords.data());
glEnableVertexAttribArray(program->texCoordAttribute);

glDrawArrays(GL_TRIANGLES, 0, vertices.size() / 2.0f);
glDisableVertexAttribArray(program->positionAttribute);
glDisableVertexAttribArray(program->texCoordAttribute);
}
```

Let's Animate George!

(grab code from the
“Entity” example in github)