

生成 AI を利用した教材の研究

愛媛県立新居浜東高等学校

本田 知仁

1 主題設定の理由

教材を使用する場合、自作するか業者などの作成したものを使用するかの選択となる。極力自分で作ったものを使用したいが、作成するための時間を確保するのも難しい。そこで、生成 AI を利用して教材のアイデア出しや教材の作成をさせることができないかと考えた。

2 実践内容

ChatGPT を使用し、授業で使用する教材を作成した。今回は、JavaScript を html に埋め込み、ブラウザ上で動かすことができる教材を試した。あらかじめどのようなものを作るか想定し、それを生成 AI に作らせる場合と、アイデアの段階から ChatGPT を使用してそこから改良していく場合の 2 種類の方法で試した。なお、ChatGPT については無料版を利用しており、有料版よりも制度が落ち、画像生成についても制限がある。

3 結果・考察

(1) 小テスト

ア 進数変換問題

「2 進数と 10 進数の変換の小テストを 10 問分作るプログラムを JavaScript で埋め込んで作って。」と指示したところ、問題が作られ、回答を確認するボタンが作られた (図 1)。その後、問題数を整理、解答を隣に出すように指示し直した (図 2)。ここまで 3 回のやり取りで作ることができた。

2進数と10進数の変換クイズ

2進数: 1100100 を10進数に変換すると?	
2進数: 1001100 を10進数に変換すると?	
2進数: 100111 を10進数に変換すると?	
2進数: 11000111 を10進数に変換すると?	
2進数: 101101 を10進数に変換すると?	
2進数: 10010111 を10進数に変換すると?	
2進数: 1101010 を10進数に変換すると?	
2進数: 1000010 を10進数に変換すると?	
2進数: 1000111 を10進数に変換すると?	
2進数: 10110110 を10進数に変換すると?	

10問中0問正解しました！

図 1 2 進数の変換

進数変換クイズ

2進数: 111 → 10進数?		不正解 (7)
2進数: 101001 → 10進数?		不正解 (41)
2進数: 1100101 → 10進数?		不正解 (101)
10進数: 128 → 2進数?		不正解 (10000000)
10進数: 38 → 2進数?		不正解 (100110)
10進数: 213 → 2進数?		不正解 (1010101)
2進数: 1001111 → 16進数?		不正解 (4F)
2進数: 1001001 → 16進数?		不正解 (49)
16進数: EF → 2進数?		不正解 (11101111)
16進数: B3 → 2進数?		不正解 (10110011)

10問中0問正解しました！

図 2 各進数変換

イ 論理回路問題

画像出力ができず回路図を表示できなかった。

(2) 文字のデジタル化

ASCII のコード表から文字を選ぶとその文字に対応した 2 進数（コード）が表示される教材を作ろうと試みた。まず、「2 進数と文字コードの関係を説明する JavaScript を作って。Script は html に埋め込んで。」と指示したところ、変換ボタンを押せば各進数に変換されるものができた（図 3）。「ASCII コードのエリアマップ上の文字をクリックしたら、2 進数が出てくる JavaScript を作って。」としたところ、想定に近いものができた（図 4）。その後は以下のように調整した。

- ・上位下位それぞれ 4 ビットを上と左に表示させたところ、2 進数に間違いがあったので訂正させた。
 - ・対応する文字の上位下位ビットが黄色く表示されるように指示した。
 - ・文字をクリックすると 2、10、16 進数を表示するようにした。
- 計 6 回のやり取りで想定していたようなものにすることができた（図 5）。

2進数と文字コードの関係

任意の1文字を入力してください:

文字: A
Unicodeコードポイント: 65 (10進数)
2進数: 1000001
16進数: 41

ASCIIコード エリアマップ

文字をクリックすると、その文字の2進数が表示されます。

	!	"	#	\$	%	&	'
(	)	*	+	,	-	.	/
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	{		}	~	

図 3 文字コード生成（1 回目）

図 4 文字コード生成（2 回目）

ASCIIコード エリアマップ

文字をクリックすると、その文字の2進数と16進数が表示され、対応する行・列とセルがハイライトされます。

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000																
0001																
0010		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
0011	0	1	2	3	4	5	6	7	8	9	:	;	=	>	?]	
0100	[	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

文字: 4
ASCIIコード: 52 (10進数)
2進数: 00110100
16進数: 34

図 5 文字コード生成（6 回目）

(3) 音のデジタル化

音（波）を標本化、量子化、符号化する作業を実際に動かして体験できる教材を作ろうと試みた。初めに「サイン波を標本化、量子化、符号化してデジタル化

する過程を学ぶ JavaScript を作って。Script は html に埋め込んで。」と指示した（図 6）。思っていたものとは全く違うものができ、一度にたくさんの工程を指示するとうまく処理できないと感じた。そこで大まかにそれぞれの段階を作り、細かなところは最後に調整することにした。6 回ほど指示をし直したところ想定していたものに近いものができた（図 7）。量子化については、それらしいものに近づいたが、何回試しても数値が正しく表示されなかった。符号化については膨大な 0 と 1 の羅列になるので負荷が高くなった。量子化の数値が正確に取得できないため音のデジタル化についてはここで終了させた（図 8）。

サイン波のデジタル化

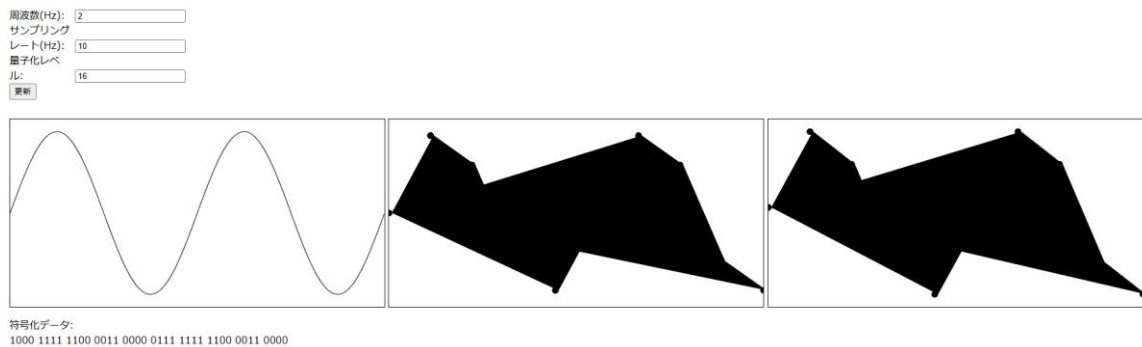


図 6 音のデジタル化（1 回目）

サイン波の標本化

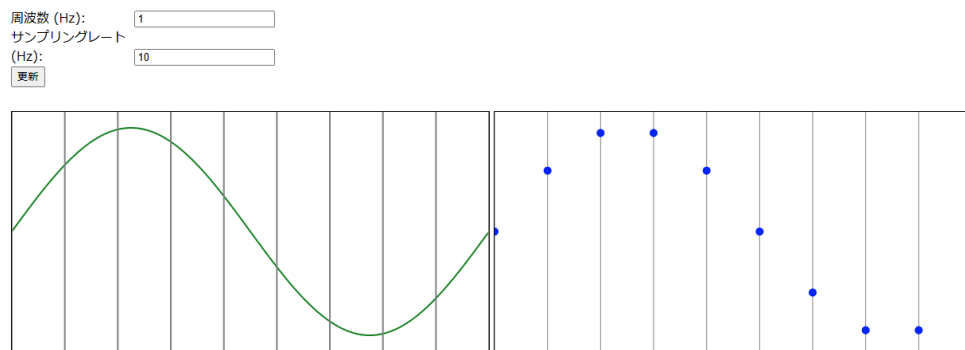


図 7 音のデジタル化（6 回目）

サイン波の標本化と量子化

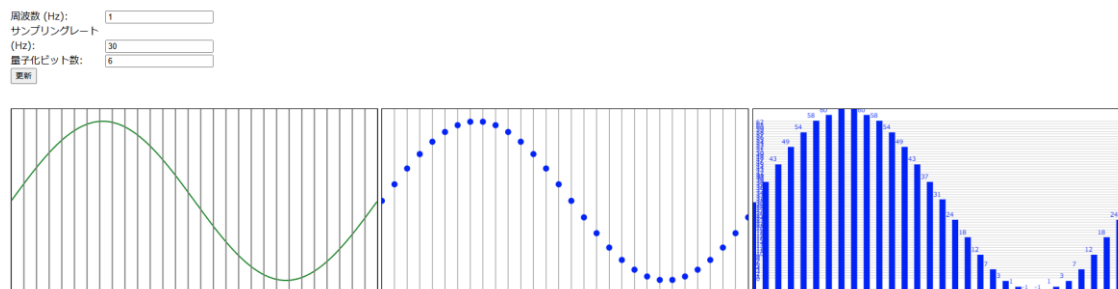


図 8 音のデジタル化（6 回目）

(4) 動画

10fps、20fps、30fps、60fps のアニメーションを作り、違いを比べる教材を作ろうと試みた。まず、「人が歩いている動画をフレームごとに表示するものを JavaScript で埋め込んで作ってください。」と指示をしたが画像が無いため何も表示されなかった。画像も生成してもらえるか尋ねたところ、可能だったため、「人が歩いている様子の動画をループで、10FPS、20FPS、30FPS、60FPS の 4 種類作ってください。」と指示したところ 4 種類に分かれたものが生成された（図 9）。ただし、画像生成の精度が悪く、無料版での回数制限のためにうまく生成できなかった。他の画像生成 AI も試したものの、アニメーション用の大量のフレーム生成は苦手なようで、作ることができなかった。ソースを見るとファイル名が分かるので別途画像を用意すると教材として使用することはできるように思える。

人が歩いているアニメーション

10 FPS



再生

停止

図 9 動画の説明（10FPS のみ抜粋）

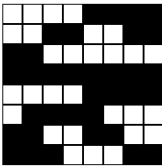
(5) ランレングス圧縮

ランレングス圧縮の過程を視覚化できる教材を作ろうと試みた。「ランレングス圧縮を理解するための教材を JavaScript の埋め込みで作って。例として 8 × 8 の白黒のビットマップを使って。」という指示から開始し、圧縮結果と圧縮率の表示を行った。しかし、圧縮結果や圧縮率が間違っており、例題としての画像も圧縮率があまり変化しないものだったため、何回も修正が必要であった。合計 13 回の手直しの結果、表示は正しくなったが、過程を分かりやすくいう視覚化できるものにはほど遠いものであった（図 10）。

ランレングス圧縮 (RLE) の理解

8×8の白黒ビットマップを使ったランレングス圧縮の例です。

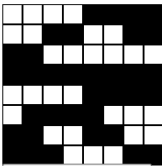
元のビットマップ



圧縮結果 (RLE)

0:4 1:4 0:2 1:2 0:2 1:2 1:2 0:6 1:8 0:4 1:4 0:1 1:4 0:3 1:2 0:2 1:2 0:2 1:3 0:3 1:2

圧縮されたデータの復元

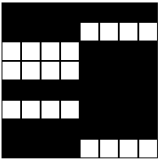


ランレングス圧縮を実行

ランレングス圧縮 (RLE) の理解

8×8の白黒ビットマップを使ったランレングス圧縮の例です。

元のビットマップ



圧縮前のビット数と列

11111111 11110000 00001111 00001111 11111111 00001111 11111111 11110000

圧縮結果 (RLE)

1:12 0:8 1:4 0:4 1:12 0:4 1:16 0:4

圧縮後のビット数と列

01100 01000 00100 00100 01100 00100 10000 00100

圧縮前のビット数: 64, 圧縮後のビット数: 40, 圧縮後の割合: 62.50%

ランレングス圧縮を実行

図 10 ランレングス圧縮（1回目）

図 11 ランレングス圧縮（13回目）

(6) 情報デザイン

情報デザインの分野で、アイデアから生成 AI に出してもらおうことを試した。「情報デザインを高校生に教えるためにどのような教材が効果的だと思いますか。」と質問したところ 7 つのアイデアを提案された。その中から「5. データ可視化の体験型教材。視覚的なデータの取り扱い（例：run-length encoding や圧縮、グラフ作成）など、データがどのように表現され、受け手に影響を与えるかを学ぶ教材を作ると良いでしょう。実際のデータを扱うと、データのパターンが視覚的にわかりやすくなり、デザインの意味を実感できます。」を参考にし、グラフの違いを考える授業を行うことにした。「架空の売り上げデータを作って、見やすいグラフと見にくいグラフを比較するものを JavaScript で作って html に埋め込んでください。」と指示したところ棒グラフと折れ線グラフが作られた。どちらのグラフもメリットデメリットがあるので、それを生徒に考えさせることにした。また、商品数を増やした場合の棒グラフと折れ線グラフを作らせることで、データ数が増えたときの違いについても考えさせることにした。

授業実践をしたところ、どちらのグラフが効果的なのかさまざまな意見が出て活発な活動ができた。ただし、メリットデメリットがはっきりしておらず、どちらを選択すれば良いかということをはっきりと伝えにくい生徒には違いが分かりにくかったようにも思える。

売上データの比較：棒グラフ vs 折れ線グラフ

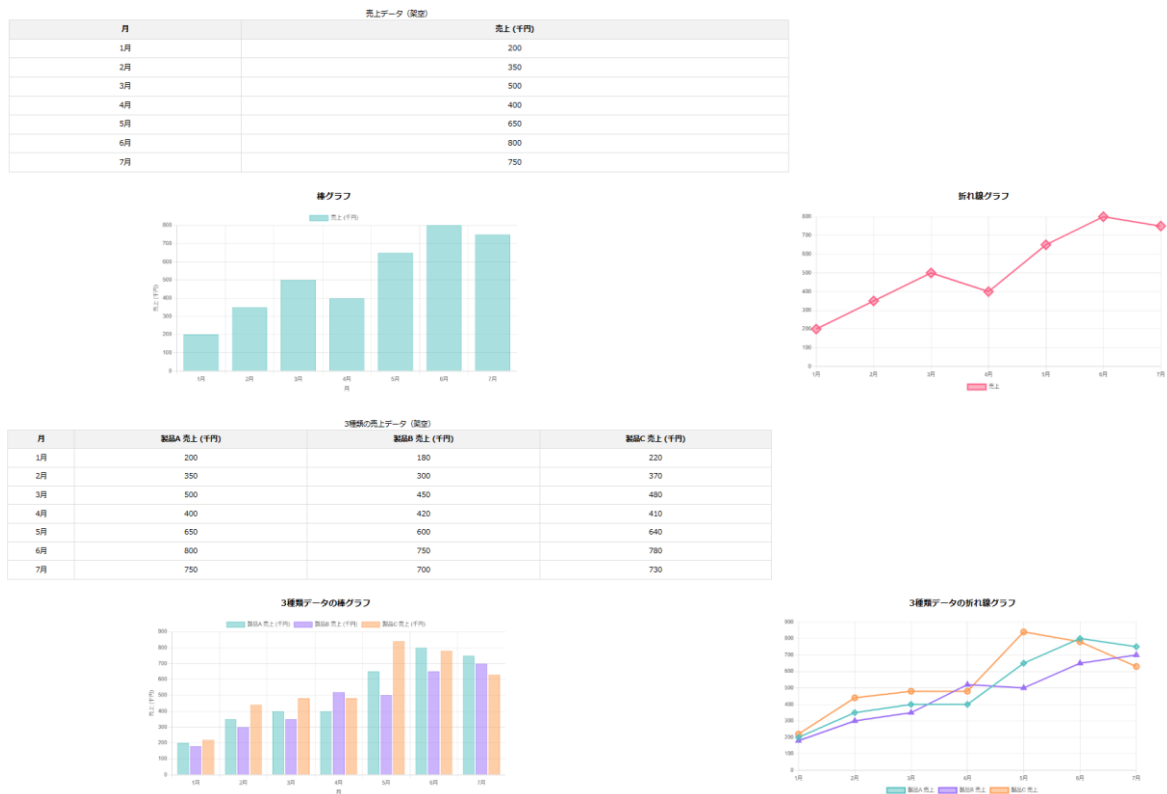


図 12 棒グラフと折れ線グラフ

4 まとめ

今回は、無料版の ChatGPT を利用して、html に JavaScript を埋め込む教材の作成を試みたが、複雑なものでなければ十分に活用できるように感じた。プロンプトについては、初めから具体的に指示しても想定通りのものはできにくいことから、大雑把に指示してそこから調整していくほうがやりやすいように感じた。しかし、一度形が確定したものを別の形に作り直すのは難しいので、自分の中でのイメージがある場合は、一度目の指示から自分のイメージに近いものになるまで何回もやり直した方がよいように思える。また、数値の間違ひは直すのが難しかった。最後に手動でソースを直すか、諦めて切り捨てることも必要だと思う。

画像を生成しなければならない教材や問題の作成は難しい。そのような教材や問題を作りたい場合は自分で描く必要がある。論理回路などは、生成 AI で論理式を生成してもらって自分で問題を作り変えることはできる。生成 AI の得意な部分で手助けしてもらうことが効果的である。

JavaScript で動的なものを作りたかったが、最終的には実装しきれないことも多かった。徐々に出力されるソースが大きくなっていくことから CSS 等で代用できる部分もあるように思える。どのようなフォーマットで教材を作れば使い勝手がよく、改良しやすくなるのかは、今後研究を進めていく必要がある。