

# ChatGPTを活用したプログラミング実習の試み

喜家村 奨  
帝塚山学院大学  
susumu@tezuka-gu.ac.jp  
西野 和典  
太成学院大学  
k-nishino@tgu.ac.jp

稲川 孝司  
大阪公立大学  
c21839@omu.ac.jp  
高橋 参吉  
NPO法人 学習開発研究所  
takahasi-san@u-manabi.org

2022年にOpenAI社が開発したChatGPTが公開され、大きな話題となった。生成AIの登場で、コンピュータを利用したクリエイティブな活動は大きく変わろうとしている。一方、教育環境に生成AIを持ち込むことには、個人・機密情報の流出や著作権侵害などの問題や生成AIに頼りすぎることにより、それが学習者の思考の妨げるのではないかなどの懸念もある。しかし、必修化されたプログラミング学習に、生成AIを有効に活用できれば、そのメリットは大きいと考える。そこで、本稿では、ChatGPTに代表される対話型テキスト生成AIのプログラミング実習への活用について検討する。

キーワード：情報科教育、プログラミング実習、ChatGPT、生成AI、micro:bit

## 1. はじめに

学習指導要領が改訂(小、中学校は平成29年告示、高等学校は平成30年告示)され、将来に向けての人材育成が求められる今日、学習指導要領で示される「プログラミング的思考」、「論理的思考」などを育成するための教育を行うためには、小学校、中学校、高等学校のそれぞれの校種において、教材開発や効果的な指導法が求められている。

筆者らは、大きく変わった学習指導要領の内容を中心に、小学校から中学校への接続、さらに高等学校への接続を考慮したプログラミング的思考および情報の科学的理解を深めるために教材や指導法を検討してきた。また、小学校から高等学校までの接続性、STEAM教育の重要性を念頭に、教育用ワンボードコンピュータmicro:bitを用いたプログラミング教材を開発し<sup>(1)</sup>、教員研修<sup>(2)</sup>を行い、教材は教科書にまとめている<sup>(3)</sup>。特に、状態遷移システムの実装については論文としてまとめた<sup>(4)</sup>。

2022年11月にOpenAI社によって公開されたChatGPTに代表される生成AIは、対話による文章の生成や長文の要約、翻訳などを即座に行うことが可能であり、教育・研究現場での使用のあり方が問われている。ChatGPTは文章を生成するだけでなく、プログラムの自動生成なども可能で、筆者らもmicro:bitのプログラミング教材について、仕様からのプログラム生成やリファクタリングなどを試行してみたが活用性は高いと感じている。

## 2. プログラミング授業での課題

プログラミングが必修化されたことにともないこれからは大多数の学生がプログラミングを学習することになる。これは非常に喜ばしいことであるが、反面、今まで専門教育としてきたプログラミング教育を、あまりその分野に興味を示さない学習者も学ばなければならない状況となった。

大学におけるプログラミングの授業でも、情報を専攻しない学生から「プログラミングが難しい」「プログラミングが苦手だ」という言葉を耳にすることがある。その1つの理由がプログラミングにおける言語習得の難しさにある。とくにテキスト型のプログラミング言語では、カッコの形やピリオドの有り無しなど、細かな文法を理解しなければ、プログラムが思うように動かず学生がつまずく1つの要因になっている。本来、プログラミングを学ぶ意義は、問題解決のためにいかにコンピュータを活用できるかであり、コーディングスキルの向上を目指したものではない。

そこで、生成AIを活用することによって、このコーディングに対する苦手意識を払拭できれば今より多くの学生達がプログラミングを楽しむことができるのではないかと考え、今回のChatGPTの授業での活用を検討することとした。

## 3. ChatGPTを試用するプログラミングの授業

今回、ChatGPTを活用する予定の授業では、大学から学生に付与しているメールアドレスで簡単に利用できるChatGPT (Ver3.5) を活用することとした。

対象とする授業は大学の1回生配当のプログラミング実習科目（共通教養科目）で、授業時間数は週2コマ、うち1コマがメディア授業である。授業の概要は前半（7回）で、Scratchを利用してプログラミングの基礎を学習し、後半（7回）でmicro:bitを用いたセンサなどの制御プログラミングを学習する。

後半のうち4回分は、4章で説明するChatGPTを活用したライトレース・タイムトライアルレースの課題に割り振る予定である。その授業計画は、表1のとおりである。なお、micro:bitの基本的なプログラミングについては、この授業まで（3回分）に習得済みである。

表1 授業計画

| 回数  | 授業の概要   |
|-----|---|
| 1回目 | ロボットカーに搭載されたセンサの制御および無線プログラミング                                |
| 2回目 | 練習用コースを使った基本となるライントレースプログラムの説明とChatGPTをアシスタントとして活用するための注意点の解説 |
| 3回目 | ChatGPTをアシスタントとして、ライントレースプログラムを改造                             |
| 4回目 | 本番コースでのレース開催、レポート作成、フィードバックなど                                 |

#### 4. ロボットカーを用いたプログラミング教材

今回は、ワンボードコンピュータ micro:bit とロボットカー (micro:Maqueen Lite<sup>1</sup>) を用いたライトレース・タイムトライアルレースを題材とすることとした。このレースのルールは以下の通りである。

- ・ マスタ(micro:bit) (図1の①) から無線(Bluetooth)で文字列"start"が送信されたらロボットカーはスタートする。
- ・ ラインをトレースしていき、ゴール地点のゴールセンサ (図1の②) を踏むとゴールとする。
- ・ ゴールラインの5cm先に障害物 (図1の③) が置いてあり、ゴール後、その障害物に当たった場合、失格とする。
- ・ ゴールしたことの検知および、タイム測定はマスタ micro:bit でおこない、ゴール後、タイムが表示される (マスタのプログラムは教員が作成済)。

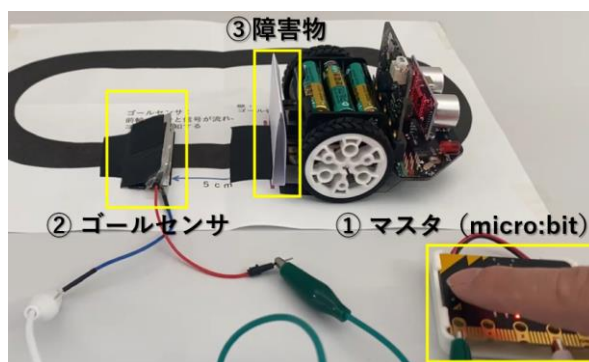


図1 教材のロボットカーと練習用コース

#### 5. ChatGPT を活用したプログラム作成の試行実験

ここでは、ChatGPT を活用して、4章で示したレースのプログラムを実際に作成した試行実験の結果を示す。

##### 5.1 実験1—無線でのスタートメッセージの受信

まず、最初にChatGPTに対してライントレースの基本プログラムを示し、改造点の1つである無線による"start"メッセージの受信後、ロボットカーがスタートするためのプログラムの修正を要求してみた。

以下に、入力プロプトとChatGPTからの回答メッセージを示す。また、図2、図3に入力したライントレースの基本プログラムとChatGPTの生成プログラムを示す。

##### 学生：

つぎのプログラムはmicro:bitのマイクロマクイーンのライントレースのプログラムです。このプログラムに無線で文字列"start"を受け取ってから動き出すようにプログラムを変更してください。

(プログラムは図2に示す)

##### ChatGPT：

以下は、micro:bitのマイクロマクイーンのライントレースプログラムを、無線で文字列"start"を受け取るときに動き出すように変更したものです。基本的なアイデアは、radio.onDataPacketReceived イベントハンドラを使用して、"start"という文字列が受信されたときに動作を開始するようにします。

(プログラムは図3に示す)

このプログラムでは、radio.onDataPacketReceived イベントハンドラが定義され、"start"が受信されたときに無線通信を開始し、その後は以前のプログラムのようにライントレースの動作が続きます。

<sup>1</sup> <https://www.dfrobot.com/product-1783.html>

```

basic.forever(function () {
  if (maqueen.readPatrol(maqueen.Patrol.PatrolLeft) == 0 && maqueen.readPatrol(maqueen.Patrol.PatrolRight) == 0) {
    maqueen.motorRun(maqueen.Motors.All, maqueen.Dir.CW, 50)
  } else if (maqueen.readPatrol(maqueen.Patrol.PatrolLeft) == 1 && maqueen.readPatrol(maqueen.Patrol.PatrolRight) == 0) {
    maqueen.motorRun(maqueen.Motors.M1, maqueen.Dir.CW, 50)
    maqueen.motorRun(maqueen.Motors.M2, maqueen.Dir.CW, 15)
  } else if (maqueen.readPatrol(maqueen.Patrol.PatrolLeft) == 0 && maqueen.readPatrol(maqueen.Patrol.PatrolRight) == 1) {
    maqueen.motorRun(maqueen.Motors.M1, maqueen.Dir.CW, 15)
    maqueen.motorRun(maqueen.Motors.M2, maqueen.Dir.CW, 50)
  }
})

```

図2 入力したライトレースの基本プログラム

```

radio.onDataPacketReceived(({receivedString}) => {
  if (receivedString == "start") {
    basic.forever(function () {
      if (maqueen.readPatrol(maqueen.Patrol.PatrolLeft)==0 && maqueen.readPatrol(maqueen.Patrol.PatrolRight)==0) {
        maqueen.motorRun(maqueen.Motors.All, maqueen.Dir.CW, 50)
      } else if (maqueen.readPatrol(maqueen.Patrol.PatrolLeft)==1 && maqueen.readPatrol(maqueen.Patrol.PatrolRight)==0) {
        maqueen.motorRun(maqueen.Motors.M1, maqueen.Dir.CW, 50)
        maqueen.motorRun(maqueen.Motors.M2, maqueen.Dir.CW, 15)
      } else if (maqueen.readPatrol(maqueen.Patrol.PatrolLeft)==0 && maqueen.readPatrol(maqueen.Patrol.PatrolRight)==1) {
        maqueen.motorRun(maqueen.Motors.M1, maqueen.Dir.CW, 15)
        maqueen.motorRun(maqueen.Motors.M2, maqueen.Dir.CW, 50)
      }
    })
  }
})
// ラジオを初期化
radio.setGroup(1)

```

※下線部分が ChatGPT が新たに挿入したコード

図3 ChatGPTの生成プログラム(1)

図3のChatGPTの生成プログラムをロボットカーの **micro:bit** にダウンロードし、ホストの **micro:bit** から文字列” start” を送信すると正常にロボットカーはスタートしライトレースを開始した。生成されたプログラムを見ると、ライトレースの基本プログラムのコードを無線の受信イベントハンドラである (`radio.onDataPacketReceived`) 内に記述するようにプログラムが修正されていることが分かる。

次に改造点2つ目のゴール後、障害物に当たる前に停止するプログラムの修正を要求するが、これについては、いきなり赤外線センサからの値をもとに停止する

のではなく、無線から文字列” stop” を受信するとロボットカーが停止する改造をまず依頼した。このプロンプトおよびChatGPTが修正したプログラムは割愛するが、これも正常に文字列” stop” を受け取ると停止ように改造することができた。

## 5.2 実験2—超音波距離センサ値の利用

さらに、超音波距離センサから距離を検知し、ゴール後、障害物に衝突せずに停止するようにプログラムを改造するようにプロンプトを与えた。図4に、ChatGPTの生成したプログラムを示す。

**学生：**

では、次に超音波センサで2センチメートル以下の障害物が見つかったら止まるようにしてください。

**ChatGPT：**

了解しました。以下は、超音波センサを使用して、2センチメートル以下の障害物が検知された場合に停止するようにしたプログラムです。

(プログラムは図4に示す)

このプログラムでは、超音波センサからの距離が2センチメートル以下の場合に停止するように追加されています。maqueen.Ultrasonic(PingUnit.Centimeters)を使用して超音波センサの距離を取得し、それを条件に組み込んでいます。

図4のChatGPTが生成したプログラムを見ると、無線の受信イベントハンドラとライントレースの繰り返し処理が並列に実行される構造に変更されている。この生成されたプログラムに関しては超音波距離センサの検知距離の若干の修正が必要ではあったが（条件式「distance < 2」を「distance < 5」に修正）、構造的な問題はなく正常に動作した。

以上のように、ChatGPTのアシストのもと、基本的なライントレースプログラムから、今回のレースのルールに対応する改造を行うことができた。実際の実習では、このような改造の後、さらにライントレースのアルゴリズムのチューニングを行ってもらう予定である。

なお、今回は以下の理由から、ライントレースの基本プログラムをプロンプトとして与え、それを改造する手法をとることにした。

```
let isRunning = false
radio.onDataPacketReceived(({ receivedString }) => {
  if (receivedString == "start") {
    isRunning = true
  }
})
basic.forever(function () {
  if (isRunning) {
    // 超音波センサーから距離を取得
    let distance = maqueen.Ultrasonic(PingUnit.Centimeters)
    // 障害物が2センチメートル以下の場合は停止
    if (distance < 2) {
      isRunning = false
      maqueen.motorStop(maqueen.Motors.All)
    } else {
      // ライントレース動作
      if (maqueen.readPatrol(maqueen.Patrol.PatrolLeft) == 0 && maqueen.readPatrol(maqueen.Patrol.PatrolRight) == 0) {
        maqueen.motorRun(maqueen.Motors.All, maqueen.Dir.CW, 50)
      } else if (maqueen.readPatrol(maqueen.Patrol.PatrolLeft) == 1 && maqueen.readPatrol(maqueen.Patrol.PatrolRight) == 0) {
        maqueen.motorRun(maqueen.Motors.M1, maqueen.Dir.CW, 50)
        maqueen.motorRun(maqueen.Motors.M2, maqueen.Dir.CW, 15)
      } else if (maqueen.readPatrol(maqueen.Patrol.PatrolLeft) == 0 && maqueen.readPatrol(maqueen.Patrol.PatrolRight) == 1) {
        maqueen.motorRun(maqueen.Motors.M1, maqueen.Dir.CW, 15)
        maqueen.motorRun(maqueen.Motors.M2, maqueen.Dir.CW, 50)
      }
    }
  }
})

// ラジオを初期化
radio.setGroup(1)
```

図4 ChatGPTの生成プログラム(2)

- ・ 学習者が基本的なライントレースプログラムの内容を理解した上で、ChatGPTを活用して欲しい。
- ・ プログラムコードをプロンプトとして入力し、そのコードをChatGPTに改造してもらおうという方法を示したい。
- ・ 基本プログラムを与えることで著作権違反となるようなコードが生成されることを抑制したい。
- ・ 学生の中にはChatGPTを利用したことのない学生も存在するため、ある程度、道筋を示し、活用させたい。

将来的には、プログラムコードを一切示さないで、プログラムを自動生成する方法が主流になるかも知れないが、現時点では、限定的な活用方法をとることにした。

## 6. プログラミング授業での生成 AI の活用と留意点

ChatGPT を授業で活用することは、今でも賛否両論あると思うが、今回は以下の理由からプログラミング実習での試行を踏み切ることとした。

### (1) コーディングに対する学習者の負担の軽減

小学校におけるプログラミング学習では、主にScratchのようなブロック型のビジュアルプログラミング言語が利用されるが、高等学校の情報の授業では、Pythonなどのテキスト型プログラミング言語が利用されることが多い。学習がブロック型プログラミング言語からテキスト型プログラミング言語に移行する場合、テキスト型言語の文法理解が問題となる。大学の授業においてもブロック型プログラミング言語は楽しくプログラミングができたが、テキスト型プログラミング言語になると「なぜ動かないか分からない」と悲鳴をあげる学生が多く見受けられる。ChatGPT を用いて、プログラムをデバッグしたり、コードの生成自体を任せることによって、学習者のコーディングにおけるストレスを大いに軽減できるのではないかと考える。

### (2) 必修化に伴う学習意欲の維持の難しさ

高等学校以下のプログラミング必修化にともない大学においても共通科目としてプログラミング関連科目を配置し、情報専攻以外の多くの学生がこのような科目を履修する機会が増えた。エンジニア思考の学生や、プログラミングに興味を持っている学生は積極的にプログラミング言語を習得しようとするが、プログラミングの授業に対して受け身の学生にとっては、プログラミング言語の習得はハードルが高い。教員やチューデントアシスタントが十分サポートできればよいが

実際の授業環境では手薄になるのが現状である。ChatGPTをサポート役として利用することで、このような問題がある程度解決できるのではないかと考える。

### (3) ソフトウェア開発における上流工程の重要性

生成AIの登場で、コーディングの重要度は軽減するように思う。これからは、問題点を明確にし、その解決のためにいかにコンピュータを活用するかという上流工程の学習に力を注ぐべきではないかと考える。

また、生成 AI をプログラミング学習に活用することには、現状、以下のようなことに留意する必要がある。

- ・ AI の学習データによっては著作権問題が起こる可能性があること。
  - ・ 生成されたコードに関して、セキュリティリスク（コードの機密性、生成コード自体の脆弱性など）が存在する可能性があること。
  - ・ 生成 AI はあくまでアシスタント（または copilot）であり、自ら全体を理解し、思考し決定する力が重要性であると学習者が理解すること。
  - ・ 生成 AI との対話において、新しい気付きを意識することが大事であることを学習者が理解すること。
  - ・ AI が生成したコードを含むシステムの動作不全に対する責任問題が発生する可能性があること。
- ※これについては、現在もすべてのコードを自分で作成するという事は少なく、ライブラリなど第三者が作成したコードをシステムに組み込む可能性は既に存在する。よって、コードの全てを理解して動作を保証することよりテスト駆動開発などのコードの検証手法を身に付けることが大切であると考える。

これらを解決するためには、AIに関する社会的な法整備などとともに、AIリテラシー教育、ソフトウェアの検証手法などのソフトウェア工学の教育が必要であると思われる。

## 7. おわりに

大学においても情報を専攻しない学生がプログラミングに関する授業を履修する機会が増えてきている。本稿では、大学のプログラミング実習授業（共通科目）での生成 AI の活用について検討した結果を示した。高等学校の共通教科情報科において、生成 AI をいかに活用するかについては、検定教科書の内容（基本構文、アルゴリズム等）との関係、大学入学共通テストの指導との関係など、さらなる議論が必要のように思うが、高等学校の学習においても、産学連携プロジェ

クトや「総合的な探求の時間」などに身の回りの問題をコンピュータで解決するようなテーマでは、生成AIを活用する価値はあるように思う。

現在、既にビジネスアプリケーションだけでなく、ソフトウェア開発環境についても生成AIを利用した”copilot”機能が実装されてきている。また、各ソフトウェアの機能として実装するのではなく、学習者個人に寄り添ったよりパーソナライズされた生成AIの出現も予想される。プログラミングが必修化された意義は、論理的思考やプログラミング的思考を身に付けるためであり、コーディングスキルを身に付けさせることではない。ならば生成AIを活用することで、コーディングに対するストレスを軽減することは、プログラミング学習において有効ではないか。本稿が生成AIを学習に活用することに対する議論の1つのきっかけになれば幸いである。

### 謝辞

本研究は2020年度よりJSPS科研費JP20K02528（研究代表者：喜家村奨，研究課題名：初等・中等教育の接続を考慮したプログラミング的思考を育成する教材及び指導法の開発）の助成を受けている。また、常日頃から、本研究に対して、的確で心温まる助言をくださる、NPO法人 学習開発研究所の三輪吉和氏に感謝します。

### 参考文献

- (1) 高橋参吉, 喜家村奨, 西野和典:「情報の科学」での「micro:bit」によるプログラミング教育の可能性～小学校から高等学校までの一貫したプログラミング教育～, 日本情報科教育学会第10回研究会報告書, pp.10-15 (2018) .
- (2) 高橋参吉, 喜家村奨, 稲川孝司, 西野和典:「micro:bit」プログラミングで学ぶ情報技術の教材開発および教員研修, 日本情報科教育学会第11回研究会報告書, pp.25-30 (2018) .
- (3) 高橋参吉, 喜家村奨, 稲川孝司:「micro:bit で学ぶプログラミング ブロック型から JavaScript そして Python へ」, コロナ社(2019).
- (4) 喜家村奨, 西野和典, 稲川孝司, 高橋参吉:micro:bit を用いた状態遷移システムの教材開発およびプログラムの実装, 日本情報科教育学会, 日本情報科教育学会誌, vol. 15, No1, pp. 37-46(2022).

### 付録

本稿で示したライントレースの基本プログラムはロ

ボットカー (micro:Maqueen Lite) のメーカーである DFROBOT 社の Wiki 内のチュートリアル ([https://wiki.dfrobot.com/micro\\_Maqueen\\_for\\_micro\\_bit\\_SKU\\_ROB0148-EN](https://wiki.dfrobot.com/micro_Maqueen_for_micro_bit_SKU_ROB0148-EN)) に記載されていたプログラムである。以下、簡単にそのプログラムで使用されている関数等について説明する。

#### ライセンスの値を取得する関数:

maqueen.readPatrol(ライセンスの ID)

#### ライセンスの ID:

maqueen.Patrol.PatrolLeft: 左ラインセンサ

maqueen.Patrol.PatrolRight: 右ラインセンサ

#### 戻り値:

0(黒) or 1(白)

#### モータをドライブする関数:

maqueen.motorRun(モータの ID, 回転方向, 速度)

#### モータの ID:

maqueen.Motors.All: 両方のモータ

maqueen.Motors.M1: 左のモータ

maqueen.Motors.M2: 右のモータ

#### 回転方向:

maqueen.Dir.CW: 正転

maqueen.Dir.CCW: 逆転

#### 速度:

整数 (0~255)

付図1 ライントレースの基本プログラムの説明