

# 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

稲川 孝司

大阪公立大学

c21839@omu.ac.jp

西野 和典

太成学院大学

k-nishino@tgu.ac.jp

高橋 参吉

NPO法人 学習開発研究所

takahasi-san@u-manabi.org

# 背景

- 2022年にChatGPTが公開され、文章の生成だけでなく、プログラムの生成も可能
- micro:bitのプログラミング教材に関して、仕様から自動販売機のプログラムを生成、プログラムのリファクタリングなどを試行してみた
- 高等学校以下でのプログラミング必修化に伴い、大学でも教養としてプログラミングの授業が増えてきたが、情報を専攻しない学生からは不評

⇒

ChatGPTをプログラミングの授業で活用することで、多くの学生が楽しくプログラミングを学習できるのではないかと...

# ChatGPTを試用するプログラミングの授業

- 対象とする授業は、大学1回生配当科目のプログラミング実習科目（共通教養科目）
- 全14回（週2コマうち1コマがメディア授業）
  - 前半7回：Scratchでプログラミングの基礎を学習
  - 後半7回：micro:bitを用いた制御プログラミングのうち4回をChatGPTの授業にあてる予定
- 教材：micro:bitとロボットカーを用いたライントレース・タイムトライアルレース
- 使用する生成AI：ChatGPT (Ver.3.5) の予定
- 2024年度から実施予定

# ChatGPTを活用する4回の授業計画

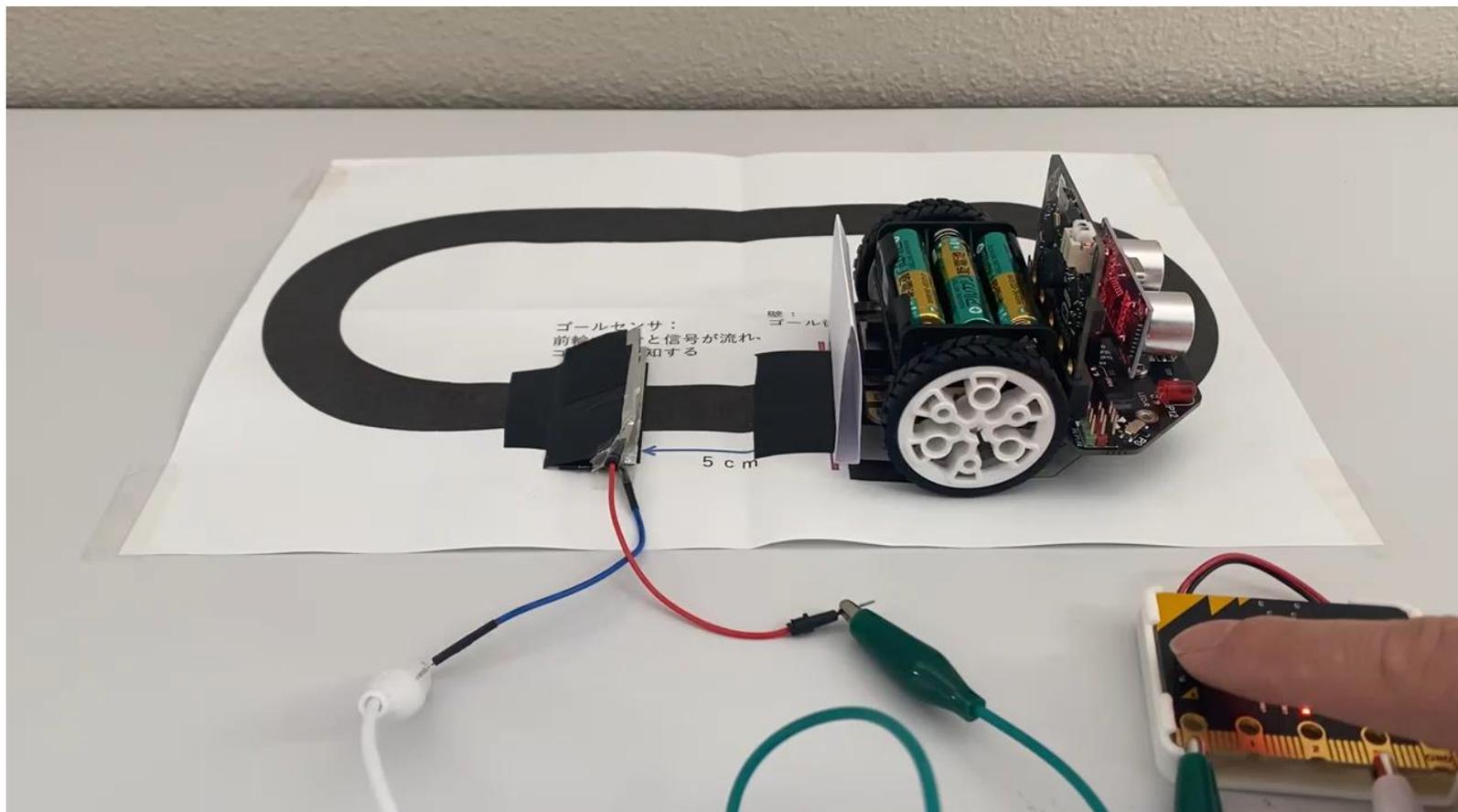
回数	授業の概要
1回目	ロボットカーに搭載されたセンサの制御および無線プログラミング
2回目	練習用コースを使った基本となるライントレースプログラムの説明とChatGPTをアシスタントとして活用するための注意点の解説
3回目	ChatGPTをアシスタントとして、ライントレースプログラムを改造
4回目	本番コースでのレース開催, レポート作成, フィードバックなど

※micro:bitの基本的なプログラミングはこの授業の前に習得済み

# レースのルール

- マスタmicro:bitから無線で文字列”start”が送信されたらスタート
- ラインをトレースしていき、ゴール地点のゴールラインを踏むことでゴール
- ただし、ゴールラインの5cm先に障害物が置いてあり、ゴール後、その障害物に当たってしまったら、失格。障害物に当たる前に停止する
- ゴールの検知、タイムの測定はマスタmicro:bitでおこない、ゴール後、タイムが表示されるようになっている（マスタmicro:bitのプログラムは先生が作成済）

# テストコースでの走行の様子



# ChatGPTを活用したプログラム作成の試行実験

- ChatGPTに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、以下のプログラムの修正を実験的に行ってみた
- 実験1：無線で文字列“start”を受け取ってから、スタートするようにプログラムを変更する
- 実験2：超音波距離センサで障害物を検知して、障害物に当たる前にストップするようにプログラムを変更する

# 実験Ⅰ

## 無線でのスタートメッセージの受信

- ライントレースの基本プログラムを示し、無線でメッセージを受け取ってからスタートするようにプログラムを修正するようにChatGPTに依頼

# 実験1:入力したプロンプト

## 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.

# 実験1:与えたプログラム

```
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)  
  }  
})
```

# 実験 I : ChatGPTからの返答

## ChatGPT:

以下は, micro:bitのマイクロマクイーンのライ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.

# 実験 I: ChatGPTの生成したプログラム

```
radio.onDataPacketReceived(({receivedString}) => {  
  if (receivedString == "start") {  
    basic.forever( function() {  
      //      :  
      //      与えたプログラム  
      //      :  
    })  
  }  
})  
// ラジオを初期化  
radio.setGroup(1)
```

# 実験Ⅱ

## 超音波距離センサ値の利用

- 超音波距離センサで障害物との距離を測定し、衝突する前にストップするように修正する

# 実験Ⅱ:入力したプロンプト

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.

# 実験Ⅱ: ChatGPTからの返答

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.

# 実験Ⅱ： ChatGPTの生成したプログラム

```
let isRunning = false
radio.onDataPacketReceived(({ receivedString }) => {
  if (receivedString == "start") {
    isRunning = true
  }
})
```

```
basic.forever(function () {
  if (isRunning) {
    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
    }
  }
})
```

障害物との距離を修正することで  
問題なく動作した

# プロンプトとしてライントレースの 基本プログラムを与えた理由

- 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.

# プログラミング授業での生成AI の活用について

- コーディングに対する学習者の負担の軽減
- 必修化に伴う学習意欲の維持の難しさ
- ソフトウェア開発における上流工程の重要性

# プログラミング授業で生成AIを活用する 上での留意点

- (AIの学習データによっては) 著作権問題がおこる可能性
- 生成されたコードの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) の可能性
- 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大事
- 生成AIとの対話において新しい気づきを意識することが大事
- 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

**※自分が作ろうとするプログラムの正しい振る舞い理解し、  
検証できる力が重要**

# まとめ

- 大学の教養科目のプログラミング実習でChatGPTを活用することを検討
- ビジネスアプリケーションだけでなく開発環境にも、生成AIを用いたcopilot機能は導入されていく
- 高校の共通教科情報科で、ChatGPTのようなテキスト生成AIをどう活用するかは、検定教科書の内容や大学入試共通テストの指導との関係など、さらなる議論が必要

※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。

# 参考:仕様を与えてプログラムを生成した例

User :

micro:bit で自動販売機のシミュレータを作成します。手伝ってくれますか？

ChatGPT :

もちろんです！Micro:bit で自動販売機のシミュレータを作成する際に、どのような機能を実装したいか、具体的な要件や機能について教えていただけますか？それに基づいて、具体的な手順やコードを提供させていただきます。

User :

自動販売機の仕様は以下の通りです。

- ボタン A が 1 回、押されるとコインが 1 枚投入される。1 枚投入されたら、文字 C を LED に表示する。
- コインが 2 枚、投入されたら商品を出力する。商品の出量は文字 I を LED に表示して表現する。
- 商品が購入されたら、コインの投入枚数はゼロになり、初期状態に戻る。

以上です。言語は JavaScript で作成してください。

## ChatGPT:

理解しました。以下は、Micro:bit で自動販売機のシミュレータを JavaScript で作成するサンプルコードです。このコードは、指定された仕様に基づいてボタン A の操作でコインの投入と商品の出力をシミュレートします。

```
// 初期状態
let coinCount = 0; // コインの投入枚数

// ボタン A が押された時の処理
input.onButtonPressed(Button.A, function () {
  coinCount++; // コインを 1 枚投入
  basic.showString("C"); // 文字 C を LED に表示
  basic.pause(1000); // 表示を 1 秒間維持
});

// メインループ
basic.forever(function () {
  // コインが 2 枚投入されたら商品を出力する
  if (coinCount >= 2) {
    basic.showString("I"); // 文字 I を LED に表示
    basic.pause(1000); // 表示を 1 秒間維持
    coinCount = 0; // コインの投入枚数を初期化
    basic.clearScreen(); // LED 表示をクリア
  }
});
```

このコードでは、ボタン A が押されるたびに coinCount 変数が増加し、文字 C が LED に表示されます。また、メインループでは coinCount が 2 枚以上になると文字 I が表示され、その後に coinCount がリセットされ LED 表示がクリアされます。

このコードを Micro:bit に転送し、ボタン A を押して動作を確認してみてください。必要に応じて、仕様に合わせてコードを調整することができます。

## User:

次に仕様を以下のように変更したいと思います。

- 商品は、コイン 2 枚で購入できるが、ボタン B が押されないと出力されない。つまりボタン B が購入ボタンとなる。
  - コインは自動販売機に 3 枚まで投入できるが、4 枚目は自動販売機に保持されず、コイン返却口から返却される。コインが返却されたことは、文字 R を LED に表示して知らせる。
- 以上の修正をお願いします。

## ChatGPT:

理解しました。以下は、仕様変更に基づいた Micro:bit での自動販売機のシミュレータのサンプルコードです。

```
// 初期状態
let coinCount = 0; // コインの投入枚数

// ボタン A が押された時の処理
input.onButtonPressed(Button.A, function () {
  // コインが 3 枚まで投入された場合
  if (coinCount < 3) {
    coinCount++; // コインを 1 枚投入
    basic.showString("C"); // 文字 C を LED に表示
    basic.pause(1000); // 表示を 1 秒間維持
  } else {
    // 4 枚目のコインは返却され、文字 R を表示
    basic.showString("R");
    basic.pause(1000);
    coinCount = 0; // コインの投入枚数を初期化
    basic.clearScreen(); // LED 表示をクリア
  }
});
```

```
// ボタン B が押された時の処理(商品購入)
input.onButtonPressed(Button.B, function () {
  // コインが 2 枚以上投入されている場合
  if (coinCount >= 2) {
    basic.showString("I"); // 文字 I を LED に表示
    basic.pause(1000); // 表示を 1 秒間維持
    coinCount = 0; // コインの投入枚数を初期化
    basic.clearScreen(); // LED 表示をクリア
  }
});
```

このコードでは、ボタン A が押された際に coinCount を増加させ、コインが 3 枚未満の場合は通常通り投入され、3 枚以上の場合は返却される仕様となっています。ボタン B が押さ