CK-002

スマートフォンを活用した高校生向け AI プログラミング教育システムの開発と評価
Development and Evaluation of a Smartphone-Based AI Programming Education System for High School Students

加納 徹†
Toru Kano

1. はじめに

近年、AI 技術は大きく発展し、個人でも容易に活用できるようになったことから、急速に社会に浸透しつつある。一方で、内閣府の AI 戦略 2022 では、AI を使うだけでなく、自ら設計・開発できる人材の育成が重要な課題としてあげられている [1]。特に、将来の技術者や創造的思考力を持つ人材を育てるうえで、初等中等教育の段階から AI やプログラミングへの理解と関心を高める教育が重要視されている。

しかし、現在の高校教育では、AI の基礎的な学習や AI 技術を「使ってみる」ことにとどまっており、技術の仕組みの理解や、プログラミングを通じた実践的な学習には至っていない。また、多くの高校生にとっては、パソコンを用いたプログラミングはハードルが高いものとなっている。一方、高校生が日常的に使い慣れているスマートフォンを活用することができれば、より身近で実感の持てる AI 教育が可能となることが考えられる。

本研究では、高校生がスマートフォンを用いて AI アプリケーションの開発を体験できる教育システムを開発し、その効果を検証した。開発したシステムでは、Web ブラウザ上で動作する AI ライブラリをベースとした独自のラッパーライブラリにより、数行の JavaScript コードで AI アプリを作成できる環境を提供している。具体的には、顔認識技術を活用してカメラ画像上にサングラスや涙などのエフェクトを重ねて表示する Web アプリを制作する教材を作成し、スマートフォンで動作する Web プログラミング環境上で実習を行った。

本稿では、本 AI プログラミング教育システムの設計と授業実践の概要について述べ、アンケート結果に基づいてその教育効果と課題について考察する。

2. 関連研究

2020 年度からの学習指導要領の改訂[2]により、プログラミング教育が小学校から高等学校にかけて段階的に必修化されている。特に、高校生を対象としたプログラミング教育では、複数の言語を段階的に学習するアプローチが提案されている。間辺ら[3]は、ドリトル、JavaScript、PHP の 3 言語を用いた高等学校でのプログラミング教育を実践し、情報システムの仕組みを理解させる効果を報告している。10 時間の授業実践をとおして、情報システムやプログラミングに対する生徒の理解が深まることを示した。大井ら[4]は、高等学校情報科において Python を用いたプログラミング教育を実践し、既存プログラムの改造を通じて生徒独自のコードを作成させる手法の有効性を報告している。また、プログラミング学習における可視化技術の重要性も指摘されており、Mladenović ら[5]は、小学校 5 年生 98 名を対象とした研究において、プログラム実行の可視化がプログラミング概念の理解に与える効果を検証し、学習効果の向上を確認した。しかし、これらの取り組みの多くは従来のパソコン環境を前提としており、多くの高校生にとって身近なスマートフォンを活用したアプローチや、AI 技術との連携については十分に検討されていない。

情報教育においては、AI 技術の普及に伴い、初等中等教育段階での AI 教育の必要性も高まっている。Yim ら[6]は、初等中等教育における AI 学習ツールに関する調査を実施し、46 の研究を分析して AI リテラシー教育の現状を明らかにした。この研究では、Google の Teachable Machine [7]、Learning ML [8]、Machine Learning for Kids [9]などが、初等中等教育段階での AI リテラシー教育に適したツールとして報告されている。また、PBL（Project Based Learning）、人間・コンピュータ協調学習、遊び・ゲームベースのアプローチが、構成主義に基づく方法論とともに頻繁に適用されていることが明らかになった。しかし、現在の AI 教育は主に「AI を理解する」「AI ツールを使う」段階にとどまっており、「AI を作る」体験を提供する教材の開発には至っていない。

一方、スマートフォンを教育に活用する取り組みは、学習への親しみやすさと継続性の向上が期待され、発展してきた。プログラミング教育においても、CodePen [10]や JSFiddle [11]などの Web ベース開発環境や、SoloLearn [12]、Mimo [13]、Programming Hub [14]、Progate [15]などのアプリケーションが、モバイルデバイス向けのプログラミング学習環境を提供している。しかし、これらの多くは基本的なプログラミング構文の学習に焦点を当てており、AI 技術を組み込んだ実用的なアプリケーション開発までは扱っていない。

本研究は、スマートフォンを活用し、AI 技術を組み込んだ Web アプリケーションの開発を体験できる点において、従来にない新しいアプローチを提示している。既存の研究では、プログラミング教育と AI 教育、スマートフォン活用がそれぞれ独立して扱われることが多く、これらを統合したアプローチは限られている。また、複雑な AI 処理を数行のコードで実現できる独自のラッパーライブラリにより、初学者でも短時間で AI アプリを完成させることができる。さらに、顔認識という身近で理解しやすい AI 技術を題材とし、視覚的に魅力的なエフェクトを作成することで、学習者の興味と動機を効果的に喚起する教材設計を行った。

† 茨城大学　Ibaraki University

3. AI プログラミング教育システム

本研究で開発した教育システムは、スマートフォン上で AI アプリを実装・動作確認できる Web ベースのプログラミング環境と、それに対応する学習教材で構成される。

本システムの主な特徴は、スマートフォンの小さな画面でも快適に操作できるユーザインタフェースを備え、従来はパソコンの使用が前提とされていたプログラミング学習を、スマートフォン上で可能にした点にある。また、顔認識技術を活用した Web アプリ開発を通じて、AI の仕組みを実践的に理解できる体験型学習環境を提供している。

システムの画面はコード編集部と実行結果確認部に分かれており、編集したコードの実行結果を即座に確認できる構成となっている。さらに、作成したアプリケーションを一般公開できるデプロイ機能も搭載し、学習成果を他者と共有できる環境を提供している。

3.1 システム構成

システムのアーキテクチャを図 1 に示す。フロントエンドでは HTML、CSS、JavaScript を用いてユーザインタフェースを構築しており、スマートフォンでの操作性を重視した縦長のデザインを採用している。また、Google が開発する AI ライブラリである TensorFlow.js をベースとした独自ライブラリにより、ブラウザ上で直接 AI 処理を実行できる環境を提供している。バックエンドでは、Web サーバ、アプリケーションサーバ、データサーバが、システムの各機能を分担している。Web サーバは静的ファイル配信と HTTPS 処理を担当し、アプリケーションサーバは PHP による動的コンテンツ生成と IP アドレスによるセッション管理を実行している。データサーバはユーザが作成したコードファイルの保存・管理を行っている。この構成により、ログイン不要でユーザごとに個別の作業環境を提供し、複数の学習者が同時に利用できる仕組みを実現している。

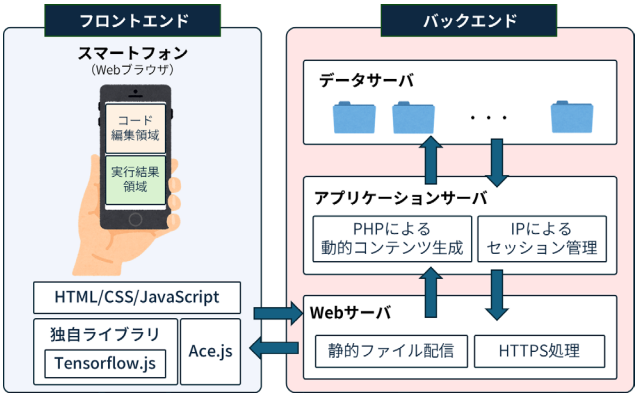

図 1　システムのアーキテクチャ

3.1.1 プログラミング環境の実装

システムの画面を図 2 に示す。画面構成は大きく上部の「コード編集領域」と、下部の「実行結果領域」に分かれている。画面右上には実行ボタンを配置し、ワンタップでコードを実行できるよう設計した。また、アプリ開発演習に必要なコードに関連する情報を表示する「ヒント」ボタンと、アプリケーションを実際に公開（デプロイ）できる「アプリ公開」ボタンを配置している。

画面上部のコード編集領域には、ブラウザ上で動作する高機能エディタライブラリである Ace.js を採用した。Ace.js はタッチデバイスでの操作に対応しており、スマートフォンにおいても比較的快適なコード編集が可能となっている。また、JavaScript を含む多数のプログラミング言語の構文ハイライト機能に加え、軽量かつ高速に動作する特徴を持つため、本研究のスマートフォン向けプログラミング環境に適していると考えられる。

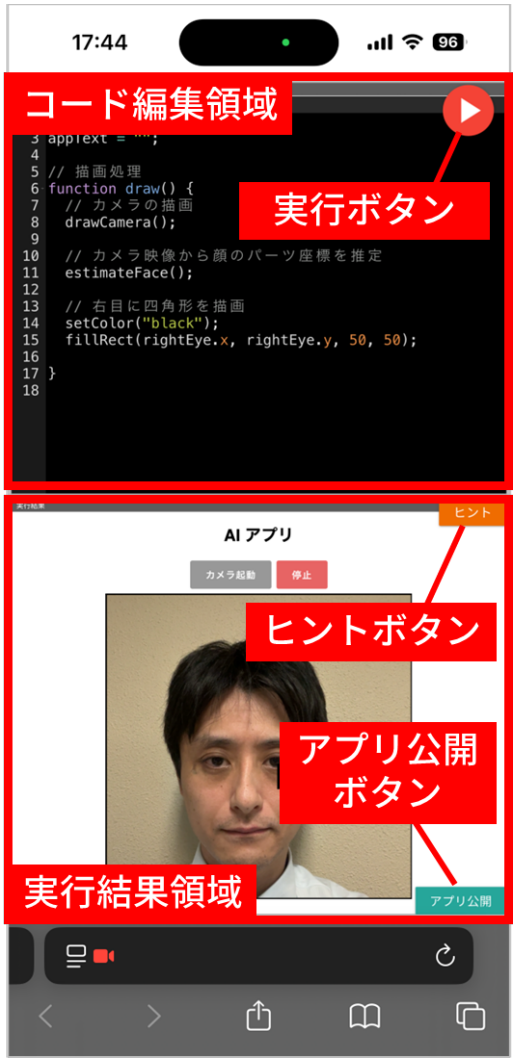

図 2　プログラミング教育システムの画面構成

3.1.2 AI 処理ライブラリの開発

顔認識アプリのコーディングにあたり、TensorFlow.js をベースとした独自のラッパーライブラリを開発した。このライブラリの設計において最も重視した点は、コードの原型を残しつつ全体の流れの理解の弊害となる情報を極力排除することである。従来は数十行のコードが必要であった顔認識 AI アプリケーションを、数行の JavaScript で実装できるよう設計した。

具体的には、カメラの描画（カメラの起動・停止ボタンを含む）は「drawCamera()」、顔のパーツ座標推定は「estimateFace()」、描画は「fillRect(x, y, width, height)」のように、直感的で理解しやすい関数名とシンプルな引数構

成を採用した。これにより、学習者は複雑な技術的詳細に惑わされることなく、プログラムの本質的な処理の流れを学ぶことができる。実際、現在のライブラリ開発はこのようにコードを簡略化する方向に進んでおり、将来的にこのような簡潔な API が一般的になることも予想される。また、AI の処理過程が完全にブラックボックスにならないよう、検出結果として顔の位置座標を数値で出力する機能を提供し、学習者が AI の動作原理を具体的に理解できるよう工夫している。

3.2 学習教材の設計

本システムの学習目標として、最小限の JavaScript プログラミング知識で AI アプリケーションを開発できる能力の習得を掲げている。従来の段階的なプログラミング学習とは異なり、数行のコードで AI 技術を活用したアプリケーションを作成できる体験を通じて、プログラミングへの興味と自信を引き出すことを重視している。

教材構成は、基本的なプログラミング概念の説明を最小限にとどめ、簡単な JavaScript 文法（変数の役割、関数の呼び出し）を理解した後、すぐに AI 機能を活用したアプリケーション開発に進む構成とした。具体的には、カメラ映像の表示、顔認識 API の呼び出し、検出結果の可視化といった処理を数行のコードで実現し、学習者が短時間で動作する AI アプリケーションを作成できるよう設計している。また、「ヒント」ボタンを押すことで、顔認識を行った際に得られるキーポイントの定義（図 3）や、描画関数における座標系の定義（図 4）、サンプルコード（図 5）などを、随時確認できるようになっている。

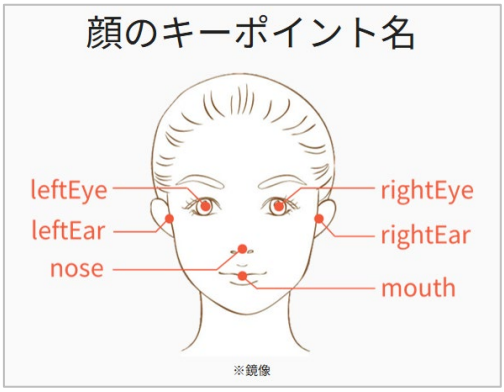

図 3　顔認識におけるキーポイント名の定義

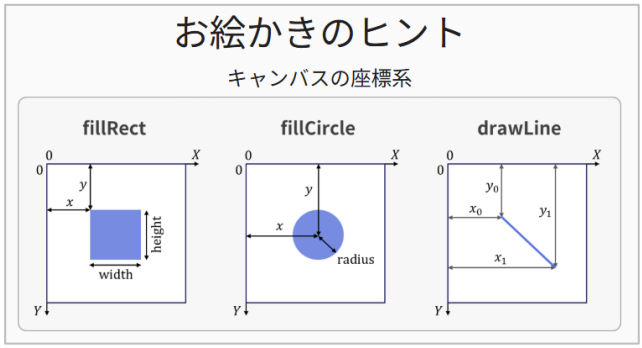

図 4　描画関数の座標系定義

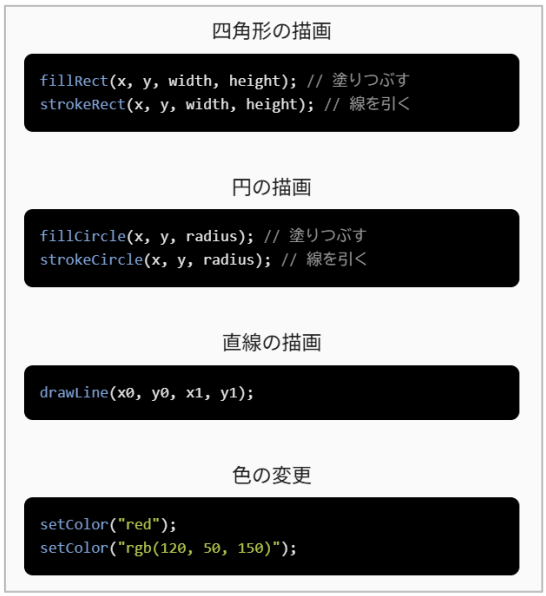

図 5　描画関数のサンプルコード

3.2.1 演習課題

実践的な課題として、以下に示す 3 つの段階的な演習を設計した。すべての演習において、図 6 に示すソースコードの雛形が用意されている。冒頭部分でアプリの名前（appName）や説明（appText）を設定すると、それが実行結果の HTML に埋め込まれる仕組みとなっている。

```
// アプリの名前と説明
appName = "AI アプリ";
appText = "";

// 描画処理
function draw() {
  // カメラの描画
  drawCamera();

  // カメラ映像から顔のパーツ座標を推定
  estimateFace();

  // 右目に四角形を描画
  setColor("black");
  fillRect(rightEye.x, rightEye.y, 50, 50);
}
```

図 6　演習におけるソースコードの雛形

【演習 1】四角いサングラスを装着してみよう。
基本的な顔認識とオブジェクト描画の理解を目的とした導入課題である。座標系に注意しながら、用意された雛形の「draw()」関数内を拡張して、「fillRect(x, y, width, height)」で両目に四角形を描画する。さらに、「drawLine(x0, y0, x1, y1)」で線を引くことにより、サングラスを模した図形の描画を行う。この演習では、顔のパーツ座標の取得方法と基本的な描画関数の使い方を習得する。演習 1 の解答例を図 7 に示す。

【演習 2】丸いサングラスを装着してみよう。
描画関数のバリエーションを学習する課題である。演習 1 の応用として、「fillCircle(x, y, radius)」による円の描画を行い、丸いサングラスを実装する。四角形から円形への描画方法の違いを学習し、異なる形状のオブジェクト描画方法を習得する。

【演習 3】涙を流してみよう。
より創造的なアプリケーション開発を体験する発展課題である。「setColor("cyan")」などで色の変更を行い、「fillRect(x, y, width, height)」による四角形の形を変形させることで、涙のエフェクトを作成する。この演習では、色の設定や形状の調整により、動的で表現豊かなビジュアルエフェクトの実装方法を学習する。演習 3 の解答例を図 8 に示す。

これらの演習は、10〜15 行程度の JavaScript コードで実現でき、学習者は複雑なプログラミング技術を習得することなく、短時間で多様な AI アプリケーションを完成させることができる。さらに、演習で作成したアプリは友人や家族と共有することができるため、学習へのモチベーション向上が期待される。

図 7　「サングラスを装着しよう」の解答例

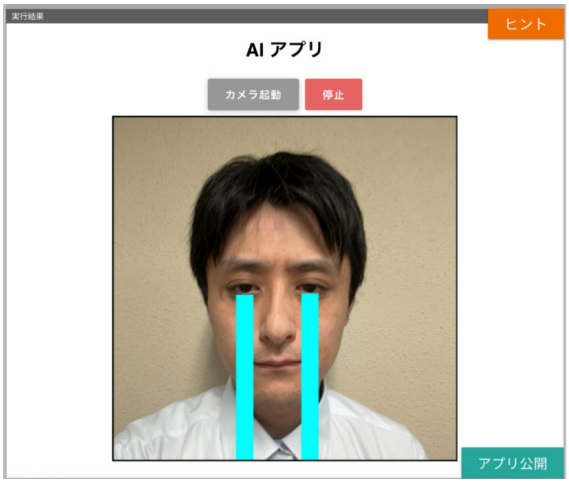

図 8　「涙を流そう」の解答例

4. 模擬講義における実践

開発した AI プログラミング教育システムの有効性を検証するため、2024 年 7 月に開催された茨城大学オープンキャンパスにおいて、高校生を対象とした模擬講義「AI アプリを作ってみよう」を実施した。実施時間は 50 分（AI の概説とシステム説明 20 分、演習 25 分、アンケート 5 分）、参加者は高校生 58 名であり、全員が個人所有のスマートフォンを使用してプログラミングを体験した。

4.1 実施内容

模擬講義ではまず、AI 技術の基礎概念とプログラミングの必要性、および開発したシステムの操作方法について、20 分間で説明を行った。その後、25 分間で 3 つの演習課題（四角いサングラス、丸いサングラス、涙エフェクト）を順次実施した。各演習では、参加者が自分のスマートフォンを用いてコーディングを行い、完成したアプリケーションを友人同士で共有・体験する時間も設けた。最後に、学習体験に関するアンケート調査を 5 分間で実施した。

4.2 アンケート結果

演習終了後、参加者 58 名を対象にアンケート調査を実施し、46 名から有効回答を得た（有効回答率 79.3%）。アンケート項目は、参加者属性（学年、性別、文系・理系の別、使用機材）と学習体験評価に分かれている。学習体験評価は、「楽しさ」「わかりやすさ」「プログラミングへの興味」「エディタの使いやすさ」「スマートフォンプログラミングの簡単さ」「AI アプリ開発の簡単さ」「全体的な満足度」の 7 項目について 5 件法で回答を求めた。ただし、各項目の尺度は設問に応じて調整しており、統計分析では 1（最も否定的）から 5（最も肯定的）に数値変換して扱った。また、「授業時間の長さ」について、「長い／やや長い／ちょうど良い／少し短い／短い」の 5 段階で評価を求めた。参加者属性の集計結果を図 9 に、学習体験評価の集計結果を図 10 に示す。

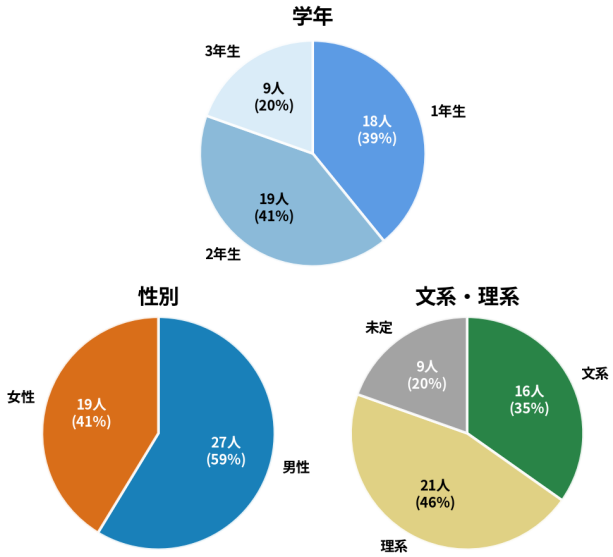

図 9　参加者の属性（$n = 46$）

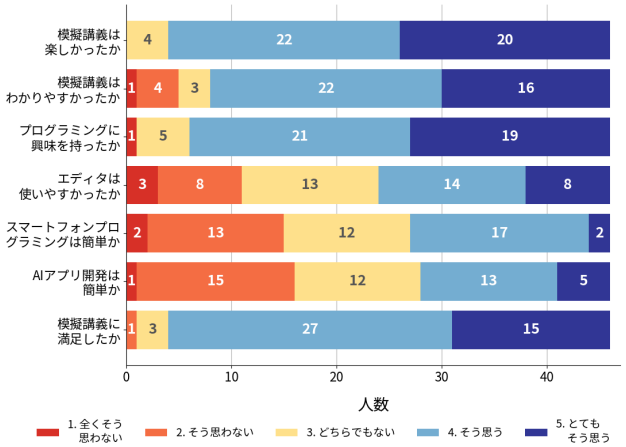

図 10　学習体験の評価（$n = 46$）

4.2.1 参加者の属性

参加者の内訳は、1 年生が 18 名（39.1%）、2 年生が 19 名（41.3%）、3 年生が 9 名（19.6%）であった。性別は男性が 27 名（58.7%）、女性が 19 名（41.3%）、文系・理系の別は、文系が 16 名（34.8%）、理系が 21 名（45.7%）、未定が 9 名（19.6%）であった。

また、使用機材について、スマートフォンの OS は iOS が 25 名（54.3%）、Android が 21 名（45.7%）、ブラウザは Google Chrome が 26 名（56.5%）、Safari が 17 名（37.0%）、その他が 3 名（6.5%）であった。

4.2.2 学習体験の満足度

「模擬講義は楽しかったか」という問いに対しては、「とても楽しかった」が 20 名（43.5%）、「やや楽しかった」が 22 名（47.8%）と、回答者の 91.3%が肯定的な評価を示した（$M = 4.35, SD = 0.67$）。「どちらでもない」は 4 名（8.7%）であり、否定的な回答は 0 名であった。

「模擬講義はわかりやすかったか」という問いについては、「とてもわかりやすかった」が 16 名（34.8%）、「少しわかりやすかった」が 22 名（47.8%）と、82.6%が理解しやすいと回答した（$M = 4.04, SD = 0.95$）。

「プログラミングに興味を持ったか」という問いには、「とても興味を持った」が 19 名（41.3%）、「少し興味を持った」が 21 名（45.7%）と、87.0%が興味を示した（$M = 4.26, SD = 0.80$）。

4.2.3 システムの操作性

「ソースコードエディタは使いやすかったか」という問いに対しては、「とても使いやすかった」が 8 名（17.4%）、「少し使いやすかった」が 14 名（30.4%）と、47.8%が使いやすいと評価した（$M = 3.35, SD = 1.15$）。一方、「やや使いにくかった」が 8 名（17.4%）、「とても使いにくかった」が 3 名（6.5%）と、23.9%が使いにくさを感じていた。

「スマートフォンによるプログラミングは簡単であると思ったか」については、「とてもそう思う」が 2 名（4.3%）、「ややそう思う」が 17 名（37.0%）と、41.3%が簡単であると感じていた（$M = 3.09, SD = 1.00$）。一方、「あまりそう思わない」が 13 名（28.3%）、「全くそう思わない」が 2 名（4.3%）と、32.6%が困難に感じていた。

「AI アプリ開発は簡単であると思ったか」については、「とてもそう思う」が 5 名（10.9%）、「ややそう思う」が 12 名（26.1%）と、37.0%が簡単と回答した（$M = 3.13, SD = 1.06$）。一方、「あまりそう思わない」が 15 名（32.6%）、「全くそう思わない」が 1 名（2.2%）と、34.8%が困難に感じていた。

4.2.4 全体的な満足度

「模擬講義に満足したか」については、「とても満足した」が 15 名（32.6%）、「やや満足した」が 27 名（58.7%）と、91.3%が満足したと回答した（$M = 4.22, SD = 0.66$）。「あまり満足しなかった」は 1 名（2.2%）のみであり、高い満足度が確認された。

また、「授業の時間の長さ」については、「ちょうどよい」が 31 名（67.4%）と最も多く、「少し短い」が 10 名（21.7%）、「やや長い」が 3 名（6.5%）、「短い」が 1 名（2.2%）、「長い」が 1 名（2.2%）であった。

4.3 考察

4.3.1 開発システムの教育効果

アンケート結果から、開発したシステムは高校生のプログラミング学習に対する興味・関心の向上に大きく貢献していることが確認された。特に、楽しいと感じた学生が 91.3%、プログラミングへ興味を持った学生が 87.0%、全体満足度が 91.3%という高い評価は、従来の講義形式では得ることが難しい成果であると考えられる。

この効果の要因として、数行のコードで実際に動作する AI アプリケーションを作成できるという体験が、プログラミングに対する心理的ハードルを大幅に下げたことが考えられる。また、完成したアプリを友人と共有・体験できる仕組みが、学習の楽しさと達成感を高めたと推察される。

4.3.2 文系・理系による差異の分析

参加者を文系・理系別にクロス集計した結果、プログラミングへの興味について、文系学生は 16 名中 12 名（75.0%）、理系学生は 21 名中 19 名（90.5%）が興味を示した。また、未定の学生は 9 名全員（100.0%）が興味を示していた。文系と理系の間には 15.5 ポイントの差が見られたが、χ^2検定の結果（$\chi^2 = 3.60, df = 2, p = 0.166$）、統計的な有意差は認められなかった。これは標本数の制約によるものと考えられるが、注目すべきは文系学生でも 75.0%という高い興味を示したことである。

スマートフォンによるプログラミングの簡単さについては、文系学生が 16 名中 6 名（37.5%）、理系学生が 21 名中 8 名（38.1%）、未定の学生が 9 名中 5 名（55.6%）、簡単であると回答しており、文系・理系間で大きな差はみられなかった。χ^2検定の結果（$\chi^2 = 0.94, df = 2, p = 0.625$）においても、有意差は認められなかった。

一方、AI アプリ開発の簡単さについては、文系学生が 16 名中 4 名（25.0%）、理系学生が 21 名中 7 名（33.3%）、未定の学生が 9 名中 7 名（77.8%）、簡単であると回答し、χ^2検定の結果（$\chi^2 = 7.28, df = 2, p = 0.026$）から、3 群間に統計的な有意差が認められた。事後検定では、文系と未定の間で統計的な有意差が認められた（$\chi^2 = 6.51, df = 1, p = 0.011$）。この$p$値は Bonferroni 補正後の有意水準（$\alpha = 0.0167$）も下回っており、補正後においても有意な結果が得られた。

特筆すべきは、未定の学生群が全項目で高い評価を示したことである（興味 100.0%、プログラミングが簡単 55.6%、AI アプリ開発が簡単 77.8%）。これは、進路選択に迷って

いる学生に対して、本システムが新たな興味や可能性を発見する機会を提供できている可能性を示唆している。

4.3.3 性別による差異の分析

性別によるクロス集計の結果、プログラミングへの興味については、男性 27 名中 23 名（85.2%）が、女性 19 名中 17 名（89.5%）が肯定的な回答をし、女性の方がわずかに高い興味を示した。χ^2検定の結果（$\chi^2 = 0.18, df = 1, p = 0.67$）から、統計的な有意差は認められなかったが、AI というテーマや、スマートフォンを活用した親しみやすいアプローチが、女性学生の関心を効果的に引いた可能性を示唆している。

また、エディタの使いやすさについては、男性 27 名中 11 名（40.7%）が、女性 19 名中 11 名（57.9%）が使いやすいと回答し、女性の方が操作性を高く評価する傾向がみられた。こちらもχ^2検定の結果（$\chi^2 = 1.32, df = 1, p = 0.25$）において統計的な有意差は認められなかったものの、女性の方がスマートフォンの操作に慣れ親しんでいることや、直感的なユーザインタフェースを好む傾向が影響している可能性がある。

4.3.4 操作性に関する課題と改善

システムの操作性については、改善の余地があることが明らかになった。エディタの使いやすさについて約 24%が使いにくいと感じ、スマートフォンでのプログラミングを困難と感じた参加者が約 33%、AI アプリ開発を困難と感じた参加者が約 35%存在した。

特筆すべきは、操作の困難さを感じながらも学習体験全体への満足度は高いという点である。これは、アプリケーションを完成させた達成感が、操作性の課題を上回る学習効果をもたらしていることを示唆している。

今回の実践結果を踏まえ、以下の改善が必要であると考えられる。まず、スマートフォン用エディタのユーザビリティ向上があげられる。具体的には、コード入力インタフェースの最適化や、タッチ操作に適した UI 設計の見直しが重要である。また、エラー発生時のサポート機能の充実や、段階的なヒント表示システムの導入により、学習者の自律的な問題解決能力を支援することも求められる。これらの改善により、高い学習効果を維持しながら、操作性の課題にも対応できると考えられる。

5. まとめと今後の課題

本研究では、スマートフォンを用いて AI プログラミングを体験できる高校生向けの教育システムを開発し、その効果を検証した。開発したシステムは、数行の JavaScript コードで AI アプリケーションを作成できるよう設計し、TensorFlow.js ベースの独自ライブラリにより直感的な API を提供した。茨城大学のオープンキャンパスにおける 46 名の高校生を対象とした実践では、模擬講義の楽しさ（91.3%）、プログラミングへの興味（87.0%）、全体満足度（91.3%）のいずれにおいても高い肯定評価を得た。特に、文系学生の 75.0%がプログラミングに興味を示し、女性学生が男性学生を上回る興味を示すなど、従来のプログラミング教育では十分にアプローチすることができていなかった学習者層への効果が確認された。これは、「数行のコードで AI 開発」という設計思想が、プログラミング経験の有無や性別に関わらず効果的であることを実証している。

一方で、システムの操作性については改善の余地があることも明らかになった。エディタの使いやすさについて約 24%が使いにくさを感じ、スマートフォンでのプログラミングや AI アプリ開発について約 35%の参加者が困難さを感じていた。また、本研究では 50 分間の短期的な効果のみを検証したため、継続的な学習による長期的な効果については未検証である。さらに、46 名という限られた標本による検証であったため、統計的な一般化可能性についてはより大規模な実証研究が必要である。

今後の課題として、スマートフォン用プログラミングインタフェースの最適化、エラー表示の改善、段階的なヒント表示システムの導入など、操作性と学習支援機能の向上があげられる。また、長期的な学習効果の測定や、より客観的な評価指標の開発、大規模な実証研究の実施による、教育効果の多角的な検証が求められる。

参考文献

[1] 内閣府：AI 戦略 2022（オンライン），入手先〈https://www8.cao.go.jp/cstp/ai/aistrategy2022_honbun.pdf〉（参照 2025-06-12）.

[2] 文部科学省：小学校学習指導要領（平成 29 年告示）解説 総則編，東洋館出版社 (2017).

[3] 間辺広樹, 並木美太郎, 長慎也, “高等学校における複数言語によるプログラミング教育の提案”, 情報処理学会論文誌教育とコンピュータ (TCE), Vol.3, No.3, pp.29-41 (2017).

[4] 大井良知, 堀江光代, “高等学校情報科におけるプログラミング教育の方法-プログラミング言語 Python を用いて”, 大阪教育大学附属高等学校池田校舎研究紀要, Vol.55 (2023).

[5] Mladenović M., Žanko Ž., Aglić Čuvić M., “The impact of using program visualization techniques on learning basic programming concepts at the K–12 level”, Computer Applications in Engineering Education, Vol.29, No.1, pp.145-159 (2021).

[6] Yim I.H.Y., Su J., “Artificial intelligence (AI) learning tools in K-12 education: A scoping review”, Journal of Computers in Education, Vol.12, pp.93-131 (2025).

[7] Google Creative Lab: Teachable Machine, Experiments with Google (online), available from〈https://teachablemachine.withgoogle.com/〉(accessed 2025-06-13).

[8] Google for Developers: Machine Learning Crash Course (online), available from〈https://developers.google.com/machine-learning/crash-course〉(accessed 2025-06-13).

[9] Dale Lane: Machine Learning for Kids (online), available from〈https://machinelearningforkids.co.uk/〉(accessed 2025-06-13).

[10] CodePen, “Online Code Editor and Front End Web Developer Community”, https://codepen.io/（参照 2025-06-12）.

[11] JSFiddle, “Test your JavaScript, CSS, HTML or CoffeeScript online with JSFiddle code editor”, https://jsfiddle.net/（参照 2025-06-12）.

[12] SoloLearn, "Learn to Code for Free", https://www.sololearn.com/（参照 2025-06-12）.

[13] Mimo: Learn Coding/Programming, Mimo (online), available from〈https://mimo.org/〉(accessed 2025-06-13).

[14] Rightsol Private Limited: Programming Hub: Learn to Code, Programming Hub (online), available from〈https://programminghub.io/〉(accessed 2025-06-13).

[15] Progate：プログラミング学習アプリ，入手先〈https://prog-8.com/〉（参照 2025-06-13）.

[16] Google: TensorFlow.js, Machine Learning for JavaScript Developers (online), available from〈https://www.tensorflow.org/js〉(accessed 2025-06-13).

[17] Ajax.org: Ace - The High Performance Code Editor for the Web (online), available from〈https://ace.c9.io/〉(accessed 2025-06-13).