

Scratch の Mesh 機能を用いた双方向性のプログラミング教材の開発 ～中学校技術科のネットワークを用いたコンテンツ制作の導入として～

Development of Teaching Materials for Learning Network-Connected Interactive Programming
with Scratch Programming Language

- As an Introduction of Creating Dynamic Contents for Technology Education in Junior High
Schools -

木下 崇*, 鎌田 敏之**, 本多 満正**

*愛知教育大学大学院生, **愛知教育大学技術教育講座

Takashi Kinoshita*, Toshiyuki Kamada** and Mitsumasa Honda**

*Graduate Student, Aichi University of Education,

**Department of Technology Education, Aichi University of Education

キーワード: 中学校技術科, プログラミング教材, Scratch, Mesh

Keywords: Technology Education, Teaching Materials for Interactive Programs, Scratch, Mesh

1. はじめに

本論文は, ビジュアルプログラミング言語「Scratch」⁴⁾を用いて中学校に設置の PC 間で双方向性のあるコンテンツを動作させるためのシステム構築とその教材開発の報告である。問題の背景としては, 2017 年 3 月告示の中学校学習指導要領技術・家庭科技術分野において教える内容として新しく示された「ネットワークを用いた双方向性のあるコンテンツのプログラミングによって問題解決する活動」を実現する上で, 中学校技術科教員が様々な困難を抱えていることがあげられる。

この問題の解決には, プログラミングの学習経験が乏しい子どもの実態に対して, ①学校教育におけるプログラミング入門期の学習障壁をより少なくするための言語の選定とプログラミングの教材開発, ②インターネットの活用等サーバー利用者側の活動から, 双方向性の実現を手がかりとした, サーバーへの関与が見通しよく捉えられるプログラミング言語の選択が必要である。

そこで本研究では, 上記 2 点を考慮し, 小学校段階での活用事例の豊富なビジュアルプログラミング言語である Scratch を用い, ネットワーク接続された複数の PC 間で Scratch 言語処理系が通信し合う仕組みを活用することで, 生徒及び指導を行う教師双方を対象に, 双方向性のあるコンテンツ

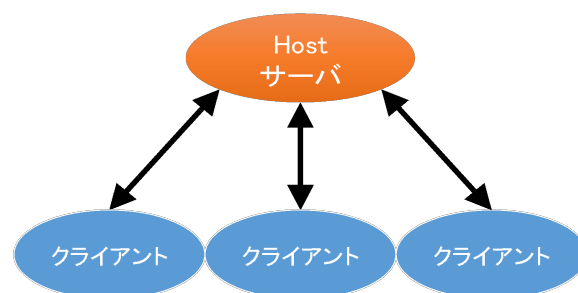


図 1 Scratch のサーバー接続の様子

の考え方と, そこに含まれる情報伝達の仕組みを学ぶためのシステム構築を行い, 教材開発を行った。

筆者らは, 先行研究として教育用プログラミング言語「ドリトル」⁵⁾を用いて, 物流を題材に双方向性のあるコンテンツの仕組みを学習する教材「物流マッチング」の開発を行ってきた¹²⁾。ドリトルは, 日本語を用いたテキストプログラミングを採用しており, 中学校段階の学習者に対し, プログラミングのハードルを一定程度下げることができたと思われる。しかし, 筆者らの現職教員研修受講者から, 中学生のタイピングの速度や正確性を勘案すると, テキストプログラミング言語であるドリトルを授業時間数が少ない技術分野で用いることに不安を覚える旨の意見を聞いた。こうした意見を踏まえ, ドリトルによるプログラミン

グ学習の前に、ビジュアルプログラミング言語である Scratch によって学習者の心的負担の軽減と、以降の学習の効率化をはかる方針とした。

2. 開発したシステム環境

2.1. Mesh 接続下の情報伝達の仕組み

Scratch で複数の PC を接続し相互作用するプログラミングを行うため、Scratch 1.4 が持つ Mesh 機能³⁾を採用した。Scratch でのネットワーク通信は、Host と、それに参加するクライアント群から構成される。Host となる Scratch はサーバー機能を提供し、Host へ参加する Scratch は、Host へクライアントとして接続し、ひとつのネットワークを構成する。こうして接続された複数の Scratch は、ネットワークを通じ、すべての Scratch 言語処理系の間で変数を共有する。すなわち、接続されたすべての Scratch プログラム群が同じ名前の箱に情報を格納することで、互いに他の Scratch プログラムに情報を伝達できるような仕組みとなっている。

2.2. Mesh 機能の有効化手順とその意味づけ

Mesh は Scratch 1.4 の隠し機能であるため、特別なプログラムを新たに用意する必要がない。ただし、Scratch 言語処理系の実体であるイメージファイルを 1 箇所書き換えるため、復元可能なよう、オリジナルのバックアップを取っておく必要がある。環境のバックアップ後、次の 6 ステップにより、Mesh 隠し機能を有効化する。以下、具体的な設定手順とともに、それら手順の意味を解説する。手順だけを説明したものとしては、英語版と一部日本語版がインターネット上に見うけられる。本稿では、システムの仕組みについての理解を進めるために、手順の意味を加え説明する。

1) 環境のバックアップ方法

まず、Scratch を変更前のもとの環境に戻すことができるよう、バックアップをとる。作業は、Scratch をインストールしたフォルダにあるファイル“Scratch.image”を、コピーして別の場所に保存することで完了する。

2) 通信機能の有効化方法と説明

以下の 6 つのステップ（次頁の図 2，番号①～⑥）により設定する。図 2 は、Scratch 開発者モードに切り替えて Mesh 機能を有効化するまでの操作に対応した画面遷移を示したものである。

表 1 通信機能追加方法の 6 ステップ

①開発者モードへの切り替え
②メニューを開く
③「open」の選択
④「browser」の選択
⑤ソースコードの 1 語変更
⑥変更した環境の保存

① 開発者モードへの切り替え

Scratch を立ち上げた状態では、一般ユーザー権限であり、このままでは Mesh 機能を有効化できない。そこで、開発者モードに切り替え、ソースコードに変更を加えられるようにする必要がある。

その具体的手順は以下である。Scratch のウィンドウ左上に表示されている SCRATCH の R 部分にマウスカーソルを合わせ、Shift キーを押しながら左クリック（以後、Shift-Click-R と記す）すると隠されたメニューが開く。そのメニューから「turn fill screen off」の項目を選択することで、Scratch の画面が左上にやや縮小され、右側と下側の余白に背後の開発者モード画面が現れ、操作ができるようになる。

② メニューを開く

図の赤丸に示した右下の薄いグレー部分は開発者モードの画面である。開発者モードでの操作を開始するため、この場所でマウスを左クリックし、「World」とタイトルがつけられたメニューを開く。

③ 「open」の選択

開発者モードに用意されている操作画面を開くため、「World」メニューのなかから、「open...」項目を選択する。タイトルが「open...」となった、新しいメニューが現れる。

④ 「browser」の選択

Mesh 機能は、ソースコードを 1 語書き換えることで有効化できる。そこで、「System Browser」というソースコード閲覧ウィンドウを開くため、「open...」メニューから「browser」項目を選択することで、薄緑色単色の、「System Browser」のウィンドウを開く。このウィンドウは、上段を左から右へ順に、階層的に整理されたソースコードを選択しながら閲覧する機能と、下段に対応するコードを編集する機能の両者を持つ。

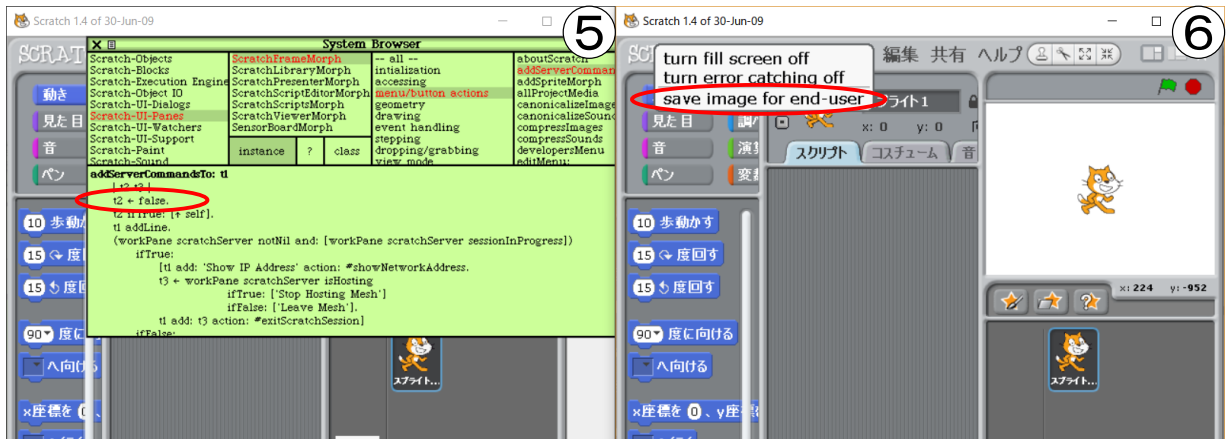
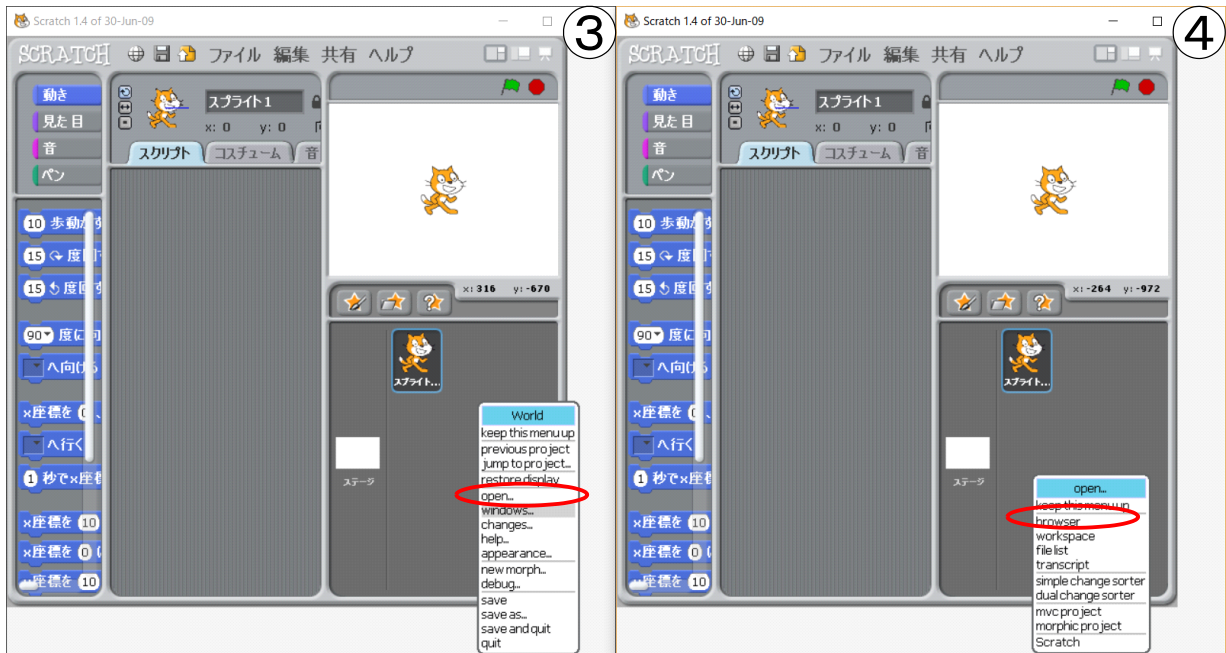
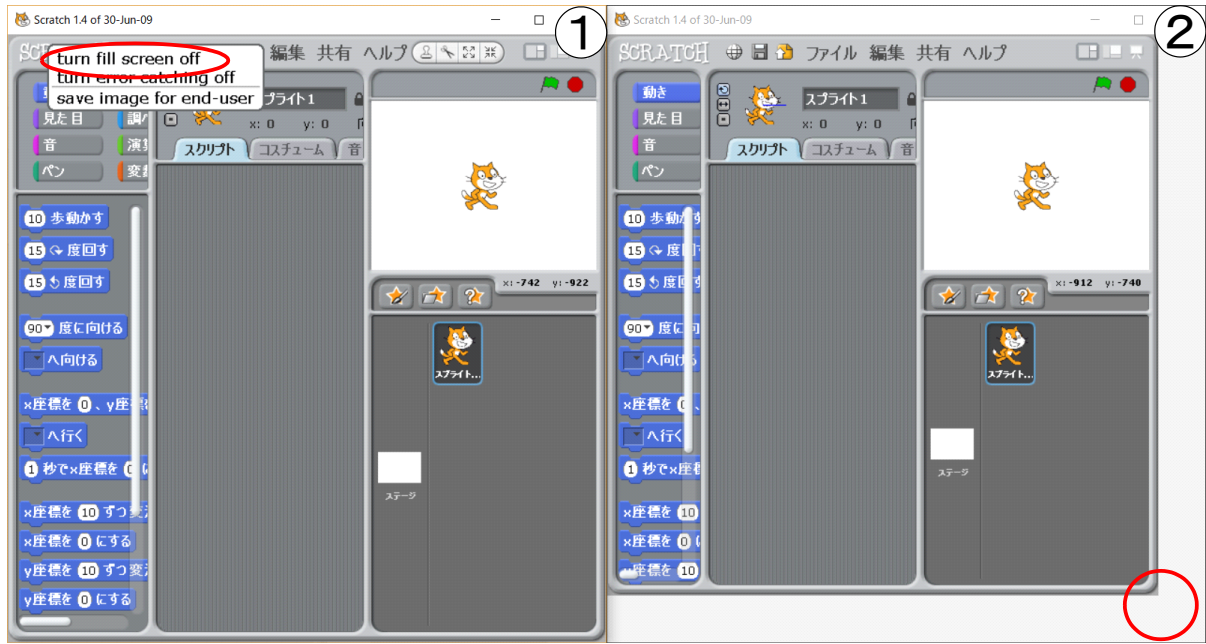


図 2 Scratch が持つ Mesh 機能有効化手順までの画面遷移

⑤ ソースコードの変更

ソースコードのうち、Mesh 機能有効化に関連する部分へたどり着くために、「System Browser」ウィンドウ上段にカテゴリ別に整理されたリストを左から右に向かって、「Scratch-UI-Panes」、「ScratchFrameMorph」「menu/button actions」、「addServerCommandsTo:」の順にクリックして選択する。すると下段に、サーバ機能のメニューを作成する部分に該当するソースコードが表示される。この3行目、「t2 ← true.」を、「t2 ← false.」に書き換える。この意味は、すなわち、t2がメニュー表示禁止の役割を果たしているところをfalse とすることで、禁止を解除することである。ここでの書き換え結果をシステムに反映させるため、Alt+s (Windows の場合、Mac では Command+s) を押す、もしくは、右ボタンメニューから「accept」を選択する。最後に、「System Browser」の左上にある「×」をクリックし、このウィンドウを閉じる。

⑥ 変更した環境の保存

設定した環境を保存するために、再度、Shift-Click-R 操作を行い、メニューから「save image for end-user」を選択する。次に、確認のメニューが開いたら「yes」を選択する。保存が終了すると Scratch は自動的に終了される。

2.3. Mesh 機能によるネットワーク接続の方法

サーバーへの接続は、以下の2ステップで行うことができる。

1) Host となる PC の IP アドレスを調べる

Mesh では、Host となる Scratch がサーバーであり、クライアントとなる Scratch が、Host へ IP アドレスを指定し接続する手順をとる。したがって、クライアントは、Host となる Scratch を動作させる PC の IP アドレスを予め知っている必要がある。そこで、Host とする Scratch で、Shift キーを押しながら「共有」を左クリックし、Mesh 機能に関するメニューを表示させ、Host となるために、メニュー項目「Host Mesh」を選択する。Scratch 内部でサーバ機能が起動すると同時に、この Scratch が動作している PC の IP アドレスが表示されるので、メモをとり、クライアントとして参加する相手に伝え、入力できるようにする。PC によっては

最初にサーバ機能が起動したとき、ファイアーウォールが働き、この PC へ接続許可してよいかを確認する画面が開くことがあるので、「許可」しておく。IP アドレスの表示は、同じポップアップウィンドウの「OK」ボタンにより閉じる。

2) クライアントとなる Scratch の参加

Host となった Scratch 以外の Scratch をクライアントとして Host へ接続することを、Scratch では「Host への参加」と呼ぶ。そこで、クライアントとなる Scratch は、Host の場合と同じ手順、すなわち Shift キーを押しながら「共有」を左クリックし、メニューを表示した後、項目「Join Mesh」を選択する。このとき、IP アドレスを入力する欄が表示されるので、さきにメモした Host となる PC の IP アドレスを入力する。

以上で、複数の Scratch からなる Mesh 通信環境が完成する。

3. 教材と技術科教員の受容

以下、Mesh 機能を用い、ネットワークによる情報共有と通信の仕組みを学ばせることを目的として開発した2つの教材と、技術科教員の受容について述べる。

3.1. ネコと同じ動きをするイヌ

この教材は、子どもの居場所検索のコンテンツをモデルに、その仕組みをシンプルに示すことで、ネットワーク経由の情報共有を直感的に学ぶ教材として開発した。ある Host が自身の PC のマウスの動きを読み取り、他の PC がそのマウスの x 座標と y 座標の値を常に読み取ることで、クライアント側のイヌが Host のネコと同じ動作をするように作成した。登場するキャラクターをネコだけでなく、イヌも登場させた理由は、例えば、Host である PC の画面をクライアントに動画配信しているといった生徒の誤解を避け、位置情報 (x, y) が異なる PC で共有されているイメージを強調する意味を込めた。

図3に本教材の実行画面を示す。①は、Host がマウスでネコを操作していない状態である。①左上に表示されている値は、現在ネコがいる位置情報であり、「host-x」「host-y」のように、Host 上の x と y の値に対応している。ネコを矢印のように

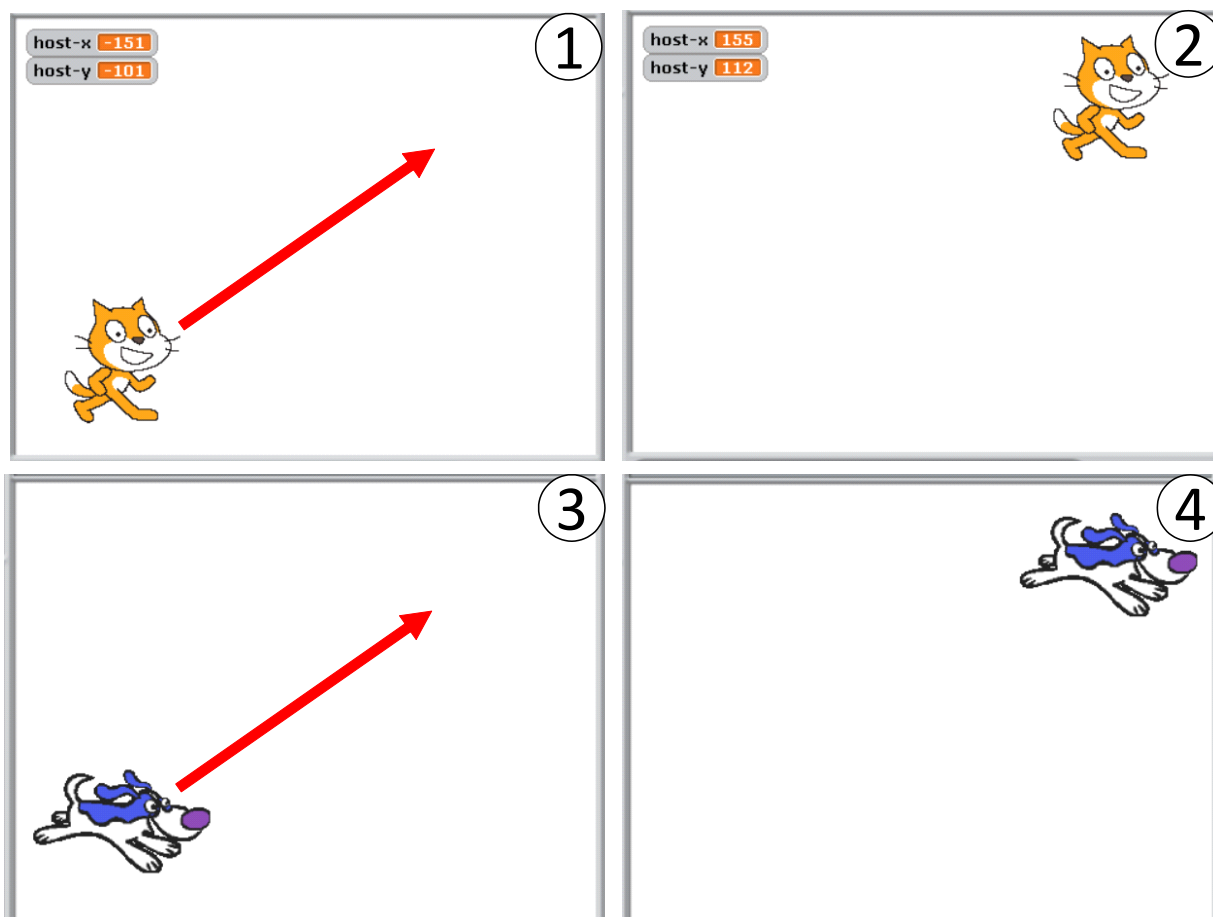


図3「猫と同じ動きをするイヌ」の画面遷移

移動させると、②に示すように、「host-x」「host-y」の値表示も変化する。③、④はクライアント PC の実行画面である。ネコの位置情報「host-x」「host-y」は両者の通信により共有されており、クライアント PC 側でも自動的に同じ値に変化するため、犬の絵が Host のネコと同じ位置に移動している。

3.2. メッセージの送受信

「メッセージの送受信」教材は、チャットに繋がる文字の送受信の仕組みを理解するために開発した。教材のプログラムの仕組みについて説明する。Scratch には、文字を入力するブロックが用意されていることを利用し、簡単にメッセージの送受信の実習を行う。今回の教材設計にあたっては、Host 系の動作を送信のみとし、クライアント系の動作を受信のみに限定した。その理由は、送受信の権限の階層性を含めての動作の仕組みを理解させることを意図したためである。

「メッセージの送受信」教材の具体的な動作の説明については、図4を板書例として以降のよう

な内容をおさえた説明を中学校実践で想定している。図4左の実行画面は、Host 係が下のフィールドに「Hello World!」を入力し、Enter キーを押すことで、message と名付けた変数の格納庫に「Hello World!」という値を送信する。図4右の実行画面は、Host 係の message の変数を常時受信した結果である。そのため、クライアント係は、受信した値の変化に応じて、クライアント係の持つ「メッセージ」という変数の格納庫に受信した変数をコピーしている。

3.3 技術科教員の受容

筆者らは、Scratch の Mesh 機能追加方法と、作成した教材を現職技術科教員に説明したところ、Scratch での Mesh 導入に対して、実践可能である等の高い評価を受けた。

4. まとめにかえて

本研究は、Scratch を用いて PC 間で通信の実習を可能とするために、①通信機能 Mesh を追加す

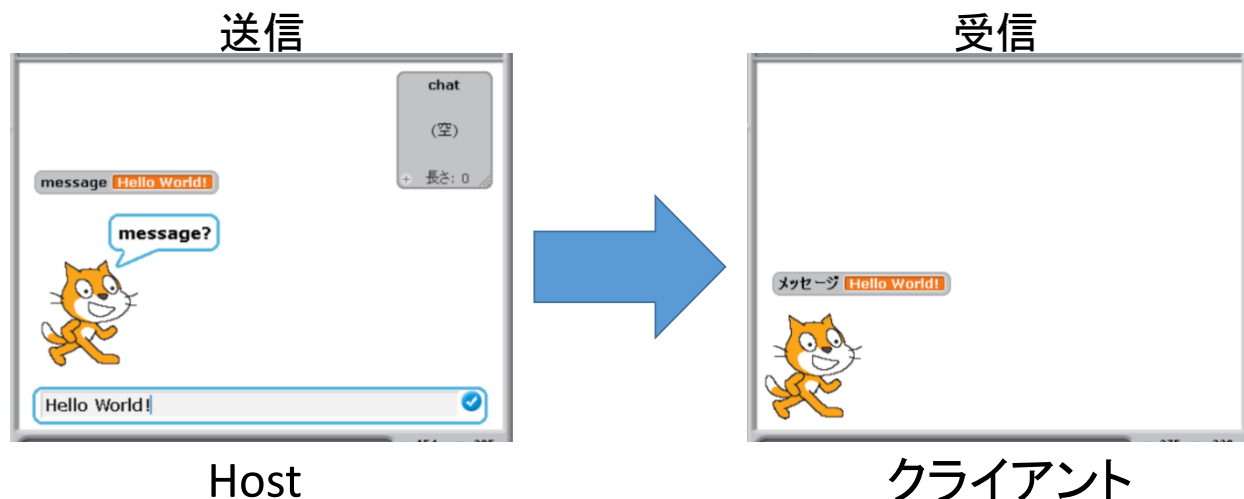


図4 「メッセージ送受信」の実行画面

る方法と手順を意味づけた教師用解説の開発、② Mesh 接続下で通信可能であり、中学生が作成可能なプログラミング教材の開発を報告した。

研究の成果としては、現職技術科教員から双方向性のあるコンテンツのプログラミングについての実践の難易度を下げた内容である旨の評価を受けた。

今後の課題としては、本研究で開発した Mesh 接続下でのプログラミング教材の動作台数の向上である。現在ノート PC で6台の接続を確認している。中学校現場の PC 室で10台前後の同時接続が実現できれば4人に1台のグループ学習が展開できる。課題としたい。

謝辞

本研究にご協力いただきました本学大学院生松村貴広氏に対して、記して感謝いたします。

参考文献

- 1) 鎌田敏之, 本多満正, 木下崇, 秋山大翼: ネットワークを用いた双方向性のあるプログラミングの授業へ向けた教員支援の試みとその評価, 愛知教育大学研究報告, 芸術・保健体育・家政・技術科学・創作編, p. 31-36 (2018)
- 2) 秋山大翼, 木下崇, 本多満正: ネットワークを用いた双方向性のあるプログラミングの教材開発, 愛知教育大学研究報告, 芸術・保健体育・家政・技術科学・創作編, p. 15-19 (2018)
- 3) Mesh 機能の追加について, 例えば, 阿部和広, 簡単だけど奥深い! scratch プログ

ラミングの魅力 (2011),

<http://tech.nikkeibp.co.jp/it/article/CO LUMN/20111019/371078/>, (2018年5月21日閲覧)

- 4) SCRATCH,

<https://scratch.mit.edu/>

- 5) プログラミング学習言語「ドリトル」,

<http://dolittle.eplang.jp/>

(2018年5月28日受理)