

# オンライン上でのコンピューティング実習支援システム 「CreCoSpeak」の開発と授業実践

松澤 芳昭<sup>1,a)</sup> 落合 祥希<sup>1,b)</sup> 齋藤 敦輝<sup>1,c)</sup> 村田 和義<sup>1,d)</sup>

## 概要：

2020年度は新型コロナウイルス感染拡大防止の観点から、教育現場でICTを利用したオンラインでの授業実施が求められる世情となった。本研究では大学でのプログラミング導入教育において、これまで対面演習で行ってきた教員(TA)–学生間での密なコミュニケーションを損なうことなく授業を展開することを目的に、WebRTC技術を用いたオンライン上でのコンピューティング実習支援システム「CreCoSpeak」の開発と授業実践を行った。本システムの設計目標は、(1)教員が学生達の作業状況を概観し演習進行の雰囲気を感じとることができる、(2)学生の質問や課題レビューに際し画面共有と顔の見えるビデオチャットによる迅速な対応ができる、(3)学生同士が課題を一緒に解いたり、教え合ったり、グループワークを行うことができる、である。文科系学部生約230名を対象とした、教員4名、TA14名によるプログラミング導入授業で本システムを半年間を運用した。光回線を使用している学生約70%を含むすべての受講生がPCを利用したリモート環境で受講した。アンケート分析の結果、「CreCoSpeak」を用いた演習は、約8割の学生にとって、質問がしやすく満足のいく環境であるという評価が得られた。本授業の良かった点として「先生や先輩方が優しく教えてくださった」や「困ったときにその場で対応を求められること」などもあげられており、本システムにより、オンライン下でも密なコミュニケーションを伴うプログラミング演習を展開できることが示唆された。

キーワード：プログラミング教育、遠隔教育、実習支援、WebRTC、TA( Teaching Assistant)

## 1. はじめに

2020年度は新型コロナウイルス感染拡大防止の観点から、教育現場でICTを利用したオンラインでの授業実施が求められる世情となった。著者らが所属する青山学院大学も、2020年度は最終的に通年で一部の授業を除く全ての授業がフル・オンライン(学生、教員ともに対面での参加者はいない授業形態)で実施されることとなった。各授業の運営方法は各教員に一任されることになり、各教員がこの状況下でも対面授業と同等以上の質を担保した授業を実施できるよう、ICTを駆使して授業の方法を模索し、授業方法に関するアクションリサーチを実践した一年となった。

フル・オンライン授業の実施に際し、講義部分に関しては、事前に収録・公開し学生が一定の期間に視聴するオンデマンド型であれ、対面授業と同様に時間を共有し学生の

反応も見ながら講義をすすめるリアルタイム型であれ、基本的には教員側からの映像を学生に配信するという設備を利用するという意味で技術は安定している。しかし、実際に学生が活動する演習を伴う授業に関しては、知見が不足していると思われる。教員やTA( Teaching Assitant)による演習支援や、学生間の交流も通した協調学習の場をどのように提供するべきか、という課題がある。

本研究では大学でのプログラミング導入教育において、これまで対面演習で行ってきた教員(TA)–学生間での密なコミュニケーションを損なうことなく授業を展開することを目的に、WebRTC技術を用いたオンライン上でのコンピューティング実習支援システム「CreCoSpeak」の開発と授業実践を行った。

## 2. 原要求

青山学院大学社会情報学部では、文理融合系の総合学部で、1年次の後期(秋学期)にプログラミング入門教育が週2コマ、2単位の演習必修授業として行われる。2020年度はコロナ禍の影響で、フル・オンライン下での実施が求

<sup>1</sup> 青山学院大学 社会情報学部  
School of Social Informatics, Aoyama Gakuin University  
a) matsuzawa@si.aoyama.ac.jp  
b) a8117047@gmail.com  
c) a8118062@gsuite.si.aoyama.ac.jp  
d) kmurata@si.aoyama.ac.jp



図 1 2018 年度 (対面) 時代の当該授業風景

められる状況となり、実施方法の検討を行った。

プログラミング授業のオンライン化に当たっては、オンデマンド方式と自学自習方式を組み合わせる設計が考えられた。実際に、プログラミング学習の自習教材は、例えば米 Khan Academy を筆頭として無料のものも広く配布されており、学术界でも報告のある Track [1] などのように、演習や論理動作テストまで自動で行うようなツールもある。ただし、これは職業訓練などの場で、十分に自律した学習者を対象として、専門教育としてのプログラミング学習の学習効率を考える場の最適解であり、大学でのプログラミング導入教育の実施に際しては以下の懸念があった。

## 2.1 TA (Teaching Assistant) の支援環境

プログラミング実習は、テキストや講義では全ては説明しきれない多様な問題がつきまとう。文法の問題だけではなく、アルゴリズムの問題、論理エラーの問題などがある。デバッグには迅速な画面共有が必要不可欠である。初学者では解決できないこれらの問題を解決するだけでなく、解決の過程を学習者に見せることによる教育効果も大きい [2]。

講師・TA・学生同士が学生の雰囲気を感じ取ることができることも重要である。大学の授業では基本的には学生からの質問依頼 (いわゆる挙手) に対応していけばよいという考え方もある。しかし、オンライン授業では学生が挙手しづらいという事情がある。対面授業でも行っているように TA が机間巡視をして、支援が必要そうな学生を見つけたら TA のほうから積極的に働きかけることができるような環境が望ましい。オンライン授業では、学生が何をやっているのか、進捗が見えないという問題がある。したがって、オンライン下でも対面の教室のようにデスクトップ画面を随時共有することで情報を集め、指導にフィードバックすることができないかを考えた。

## 2.2 学習者どうしの学び合いの環境整備

2018 年度に対面で実施された我々のプログラミング教育の様子を図 1 に示す。この場面は、学生が個別に与えられた練習問題を問いている場面であり、学生同士がグループワークを行っているわけではない。にもかかわらず、隣に並んだ学習者どうしが教え合いながら問題に取り組んでいる。対面の実習環境では、他の学習者とのコミュニケーションを通じた協調学習が自然発生する。プログラミングは個人作業になりがちな作業と思われているが、プログラミング学習での学習コミュニティの重要性については、パートラ [3] も指摘している古くからの課題である。

学習の場を共有することのもう一つの機能に、学習者同士が直接話をしていなくても、他の学習者の挙手の状況であったり、各学習者と TA との質問のやり取りなどを横目で見たりすることで、雰囲気を感じ取ることができることがある。オンライン授業で挙手しづらいのは、他の学習者の動向がよくわからないことが一因と考えられる。学習者と TA の質問のやり取りを横で聞くことで学ぶ効果は、解決の過程を学習者に共有するという目的で教育効果も高く、TA の優良な個別指導を積極的に公開するという意味で効率的である。学生にとって「TA さんが忙しいなら控えるが、空いているならちょっと聞いてみたい」という状況で、後ろを通りがかったときに聞いてみる、という状況もあるであろう。このような雰囲気づくりとその学習環境の良さをできる限りオンラインでも活かしたい。

## 2.3 課題の提出に関連する問題

本論文の実践対象であるプログラミング導入教育では、毎週 3~5 間のプログラミング練習課題を課している。昨年度まで対面で行ってきた授業では、TA が目視で課題をチェックし、紙のチェックシートにサインをするという方式で課題チェックを行っていた。(同様の方式をとっている文献も見受けられる [4])

我々が目視で課題をチェックする意味はいくつかある。第 1 に、プログラムは仕様どおりに動作すればよいというものではない、ということをお教えるためである。動作チェックだけでなく、ソースコードを通読することで、無駄な変数や無駄な処理、冗長な処理を指摘する必要がある。繰り返しや関数の作成など、プログラムの書き方の指示がある問題もあるため、現代技術で一定水準の評価を展開するにはまだ人手でのチェックが必要である。

第 2 に、課題が解けることが目的ではなく、理解することが目的ということをお教えるためである。学生はどうしても課題を解くことだけを目的として、理解しないで課題を提出しようとする。TA には課題チェックの際に、適宜「どうしてこれで動くのか」を口頭試問するように指示している。TA は個々のレベルに合わせた形成的評価を行い、学習者が自分のプログラムを説明したり、解説を聞いたりす

るコミュニケーションからプログラミングを学習すると考えているためである。

第3に、剽窃を防ぐためである。これは、フル・オンライン授業では顕著な問題となる。週越しに課題を提出できるので、技術的には答えをコピーして提出することは容易である。TAには、とくに週越しの課題チェックの際には口頭試問を強化して、ソースコードの意味を聞くように指示し、学生にも周知した。このことで不正への抑止力になり、正当に課題を問いている人への不信感を払拭し、安心して課題創作に取り組めるようにしている。

## 2.4 オンラインでも「顔の見えるコミュニケーション」

これらに加えて、フル・オンライン授業となった2020年度の授業では、TA・教員と学習者で作る教室の雰囲気作りで、「顔の見えるコミュニケーション」を行うためという第4の意味が生まれたと考えている。課題チェックをビデオ通話で行わず、質問もしない、となると、全く顔を見ない授業の展開となり、孤独感が増す。最低限、課題チェックではビデオ通話を行い、TAは顔出して元気よく挨拶を行い、コミュニケーションを行うような環境を整えることを考えた。

## 3. 先行研究

### 3.1 学術研究

対面でのプログラミング教育をサポートを目的に、学生の開発環境をモニタリングする環境を提案する研究がここ十年ほどで行われてきた。2010年前後よりWebクラウドシステムが技術的な主流となったことから、ブラウザで動作するWeb開発環境を開発し、クラウドシステムに操作ログを蓄積することが比較的安価にできるようになり、それらのシステムの実装とログの視覚化の研究が進んでいる。井垣らのC3PV [5]は進捗を一覧して出力するインターフェイスを提案している。市村ら [6]、加藤ら [7]は、学習者のつまづき予測を組み込んだ情報の提示を提案している。いずれの研究も、対面でのプログラミング教育に際して、講師やTAが受講者の状況を把握することを目的としているところに特徴がある。

TAの支援に際する研究も行われている。野口ら [4]は質問・課題チェックや集計作業など、TAの事務処理を効率化するシステムについて考案している。安田ら [2]は、学生とTA間で問題解決過程を動画によって記録して、共有するシステムを提案している。業務の効率化が目的ではなく、積極的にTAによる良質な個別指導を配布していこうという試みである。この考え方はオンライン上での授業遂行にあたって、できる限り整備したいモデルであり、オンラインのコミュニケーションは記録を残すことが比較的容易なため、更に発展して利用できる可能性がある。横山ら [8]は教室でのTA支援デバイスを提案している。HMD(Head

Mount Display)を利用して、実地でTAに学習者の情報を提供しようという試みである。

本ソフトウェアが提案するデスクトップ監視機能は、対面教室でよく用いられている仕組みである。大森ら [9]は、単にデスクトップ監視をするのではなく、ヒートマップを用いた巡視方法を提案している。

又吉ら [10]は、コロナ禍でフル・オンラインでのプログラミングの演習デザインについて、本報告に先に立って発表をしている。オンラインでの会議システムRemoと質問支援システムを組み合わせた環境の提案と実践成果の報告を行っている。

### 3.2 既存のミーティングソフトウェア

提案システムの開発に際しては、既存ソフトウェアの組み合わせによる問題解決の可能性について検討した。具体的には、Zoomのブレイクアウトルーム、[10]でも利用されているRemoである。Zoomは通話が安定しているが、2020年9月時点でブレイクアウトルームを参加者が移動できる機能が追加されていないことが問題であり、後に改善されたが、それでも各ルームに分散した学習者から情報を集めることが難しいことが問題であった。Remoは学生が自由にテーブルを移動でき、アイコンにより雰囲気を感じることができることや複数人での画面共有ができることなど高機能であるが、前述の問題は同様であることと、価格が高価であることは懸念材料であった。

## 4. 「CreCoSpeak」設計

### 4.1 設計目標

本研究では、オンライン上でのコンピューティング実習支援システム「CreCoSpeak」を提案する。本システムの設計目標は、プログラミング教育などのコンピューティングを伴う演習授業において、

- (1) 教員が学生達の作業状況を概観し演習進行の雰囲気を感じとる事ができる
- (2) 学生の質問や課題レビューに際し画面共有と顔の見えるビデオチャットによる迅速な対応ができる
- (3) 学生同士が課題を一緒に解いたり、教え合ったり、グループワークを行うことができる

ことを実現することである。

「CreCoSpeak」の名前の由来は、CreCoという研究室の愛称に、少し覗き見するという意味の「Peek」と話すという意味の「Speak」をかけた造語である。「Speak」をつけたものとした。フル・オンライン授業でもよく話をし、対話によるコンピューティングの学習を進めてほしいという願いと、TAが机間巡視により学生の画面を覗き見たり、学生が隣の同僚学生とTAとの会話を覗き見たりすることで対面授業の良さをオンライン授業でも提供する、という意味が込められている。

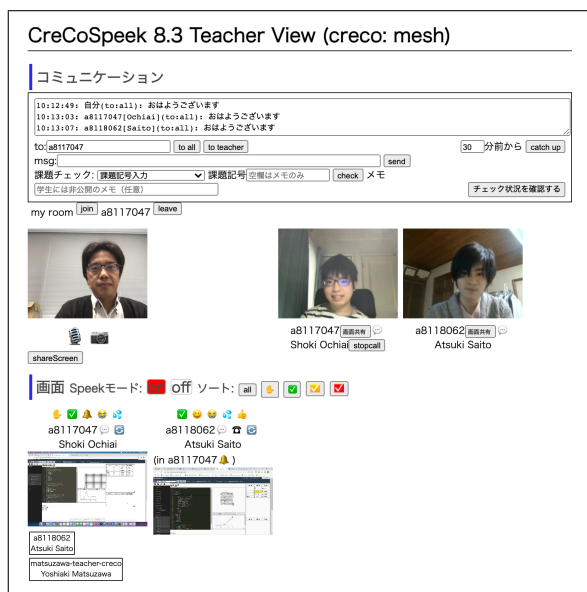


図 2 CreCoSpeak 画面 教師側

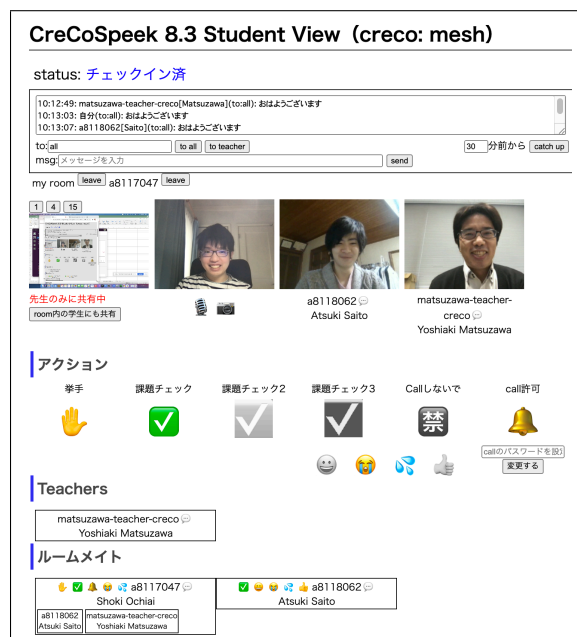


図 3 CreCoSpeak 画面 学生側

## 4.2 機能概要

CreCoSpeak は Web ブラウザで動作する Javascript アプリケーションである。本節では、CreCoSpeak の画面を用いながら機能の概要を説明する。

### 4.2.1 教師側の機能

CreCoSpeak 教師側の画面を図 2 に示す。この画面は教師がシステムの認証を通過し、classroom を指定してあるクラスにチェックインすると表示される。下部の「画面」という区画に、そのクラスにチェックインしている学生が表示される(図 2 では共著者が 2 名の学生役となっている)。学生は名前のほかに、その学生の挙手や状態アイコンの状況表示、その学生とのチャットやビデオコールへのボタン、そしてデスクトップ共有の画面が表示される。デスクトップ共有の画面の更新頻度はデフォルトで 1 秒毎である。著者らの環境(光回線, MacBookPro 2019)では、50 名ほどの学生を共有しても滞りなく動作するが、教師の環境によっては動作不良となるので、常時共有は「Speak モード」の切り替えにより On/Off することができるようになっている。

上部の「コミュニケーション」という区画は、学生とテキストチャットをしたり、学生とビデオチャットしたりすることができるインターフェイスである。教師は、話したい学生を画面区画から選び、電話アイコンを押下することでビデオ通話を始めることができる。ビデオ通話時は学生のデスクトップ画面は拡大表示され、画面を見ながらの対応ができるようになっている。必要に応じて教師の画面も学生に提示できる。テキストチャット区画では対象の学生や全体に向かってテキストチャットをすることができる。テキストチャット下の区画には、TA による指導記録機能がある。TA がチェックした課題や質問対応記録をデータベースに入力することができるインターフェイス

である。

学生側のビデオ通話受信のインターフェイスとして、チェックイン時に画面共有とビデオ通話は常に受け入れるようになっている。したがって、教師が学生に接続した場合には、常に共有されているデスクトップ画面付きのビデオ通話が、学生が「接続許可」という意思表示なしに、即時に始まることになる。この仕様は、学生と円滑に、顔の見えるコミュニケーションを行うという目的のための設計であり、結果で説明するように多くの学生に受け入れられているが、一部賛同しない学生もいる。

### 4.2.2 学生側の機能

CreCoSpeak 学生側の画面を図 3 に示す。先の説明のように、学生はチェックイン時に画面共有とビデオ通話は常に受け入れるようブラウザ、および OS に許可を与えるようになっている。学生による随時デスクトップ共有の更新頻度は 1 秒毎で快適に TA が指示を行える。ネットワーク設計が最適でなかった授業当初は 2 秒毎であったが、最低 2 秒ごとの更新があればプログラミングの指導や課題のチェックは可能である。アニメーション等のプログラムの場合は更新頻度を上げる必要があり、講師の依頼に応じて学生側が更新頻度を変更できる機能も付いている。

学生側のコミュニケーション区画は、教師側と同様である。アクション区画では、挙手や課題チェック、各種の意思表示アイコンを提示するボタンがある。「ルームメイト」区画では、各ルームメイトとアイコンで表現された状態を一覧することができる。ただし、教師側の画面区画のように、常時他の学生のデスクトップ共有画面をみることはできない。

call 機能では、学生同士のビデオ通話やデスクトップ画面共有を行うことができる。グループワークを行いたい学



図 4 待ち行列の表示

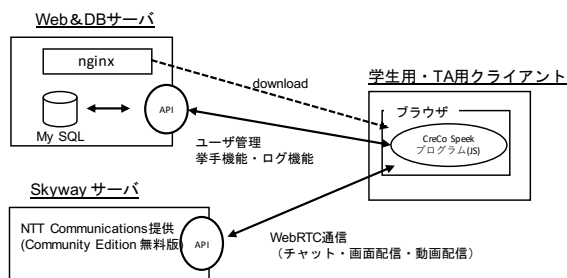


図 5 システム構成

生は、アクション区画の「call 許可」ボタンを押して、他の学生からのビデオ通話呼び出しを受けられることができる。必要ならパスワードを設定する。学生がグループワークを行っている際に、TA がその学生に call した場合には、当該学生とグループワークしている学生全員とのビデオ通話となり、グループに対する指導をすることができる。

図 2, 図 3 では割愛しているが、学生、教師ともに、挙手や課題チェック待ち行列の詳細を確認することができるようになっている。その表示例を図 4 に示す。

### 4.3 実装

本節では、CreCoSpeak の技術的背景を述べる。

本システムのシステム構成図を図 5 に示す。本システムは、Javascript で記述されたクライアントで動作する Web アプリケーションである。テキストチャット、ビデオチャット、画面共有のコミュニケーション部分は、WebRTC (Web Real Time Communication) 技術を利用している。WebRTC の実装に関しては、NTT コミュニケーションズ株式会社が提供する SkyWay サービス、Community Edition を利用している。SkyWay サービスは API が明瞭で利用しやすく、品質も安定している。本研究開発は SkyWay サービス提供の目的である Research & Development の促進と合致しており、Community Edition のサービス範囲内で授業での運用を行った。

大まかに、SkyWay による WebRTC を利用する場合に、3つの接続方法がある。1対1通信の場合の P2P (Peer to Peer) 接続、多対多通信の場合には、Room と呼ばれる多人数が入る「部屋」を作ることによって効率の良いコミュニケーションができる。Room には一つのサーバが中継する SFU (Selective Forwarding Unit) 方式と、全てのノードを P2P で接続する Mesh 方式の2つがある。クラスへの

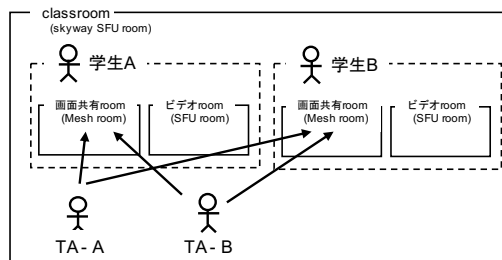


図 6 デスクトップ共有システムの論理構成図

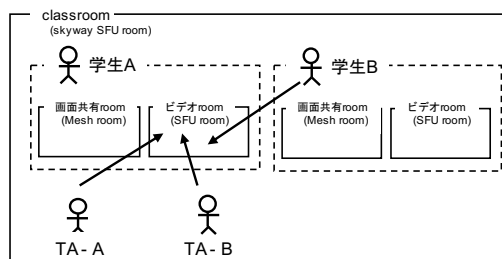


図 7 ビデオ会議システムの論理構成図

チェックイン、デスクトップ共有、TA 指導、グループワークそれぞれの機能に際して、それぞれの技術利用で安定性、接続効率、ネットワークの負荷容量が異なってくる。授業での実践にあたっては数週間の試行錯誤を要したが、現在では以下のような構成で授業での実運用に十分耐えうる実装ができることがわかった。

本システムのデスクトップ共有システムの論理構成図を図 6 に示す。まず、全体として個々のクラス (学生数十名と教員数名が情報を共有する単位) に際して、SFU room を一つ作る。ここでは、チェックイン管理、チャット管理、接続管理を行いここではビデオ通信行わない。デスクトップ共有に関しては、多数の学生のデスクトップについて、少数の教員・TA が閲覧することになる。全て共有してしまう SFU 方式では非効率であり、1クラス 30 名ではサーバを遅滞させることになった。したがって、学生各々が Mesh 方式の画面共有 room を持つことにし、教員、TA は各学生の画面共有 room に接続する。後に示すように、約学生 50 名、教員数名のクラスが安定的に機能する。

本システムのビデオ会議システムの論理構成図を図 7 に示す。TA による指導、グループワークに使用するビデオ会議に関しては、多数の学生や TA がつなぎ相互に考えると、Mesh ルームでは 3 名以上の場合には各端末に負担がかかり動作が不安定になるが、SFU ルームは安定している。したがって、各学生が各自のビデオ room を持ち、call 機能では、TA や他の学生は必要に応じてそのビデオ room に接続する設計となっている。SFU の利用量は、週 2 コマの授業を 4 回、他の授業やゼミの指導で本システムを利用して、月 300GB 程度の利用量であった。

表 1 授業概要

対象学部	社会情報学部 (文理融合系)
対象授業	コンピューティング実習 (全 15 回)
前提条件	前期に Scratch を 4 週間体験
受講者数と学年	約 230 名 (1 年生のみを評価の対象)
授業時間数	90 分×2 コマ×15 週
指導体制	4 クラス, 教員 4 名, TA13 名

表 2 授業内容とスケジュール

授業回	内容
第 1 回	基本的な命令
第 2 回	変数と式
第 3 回	条件分岐
第 4 回	繰り返し
第 5 回	入れ子構造
第 6 回	関数・引数
第 7 回	関数・戻り値
第 8 回	HTML の基礎
第 9 回	CSS の基礎
第 10 回	中間試験と作品制作 1
第 11 回	HTML と JavaScript の連携
第 12 回	Canvas の利用
第 13 回	タイマーとアニメーション
第 14 回	作品制作 2
第 15 回	まとめ

## 5. 演習概要

青山学院大学社会情報学部 2020 年度後期に開講された「コンピューティング実習」にて本システムの開発・運用を行った。授業の概要をまとめて表 1 に示す。前期に 4 週間 Scratch を体験した初学者に、Javascript を利用してプログラミングを教える週 2 コマ、2 単位の演習必修授業である。

学生に提示している学習目標は以下の通りである。

- 基礎的な Computational Thinking を身につけること：基礎的なプログラムの動作を理解し、自力でアルゴリズムを構築して、プログラミング言語を用いてそれを十分に、正確に記述できるようになること
- 基礎的な Computational Thinking を用いた Computing の実践ができること：身の回りの問題を形式的に切り取り、コンピュータを利用した解決方法がデザインできるようになること

文理融合系の学生に対し、苦手意識を植え付けることがないように留意しながら、後に発展可能なプログラミング能力を授けるのが、本授業の教授目標である。

授業内容とスケジュールを表 2 に示す。前半はタートルグラフィックスを利用した基本的なプログラムの書法と考え方を学習し、中盤では、HTML や CSS による Web 制作、後半からは HTML と JavaScript を組み合わせて、インタラクティブな Web 作品を作成する内容となっている。

2020 年度はコロナ禍の影響で、全ての教員・TA・学生が自宅から遠隔で授業をすすめるフル・オンラインでの授業運営を行った。授業は、リアルタイム型、オンデマンド教材そして CreCoSpeak を使った演習の組み合わせにより行った。毎回初めに Zoom に集合し、15 分程度進め方の打ち合わせをリアルタイムで行った後、Web 上でのテキストと、YouTube にアップされた動画を利用して学習を進め、練習問題を解くという流れで実施した。

学生によるプログラミング演習環境 (開発環境) として、我々が独自に開発して 2019 年度より授業で導入している JavaScript の学習環境「Pike Place」を利用した。「Pike Place」はブラウザで動作する Web アプリケーションタイプの開発環境であり、クラウドサーバにログインして利用する。ユーザが作成したファイルや操作記録はすべてクラウドサーバに保管される。学生は、各自のコンピュータを

表 3 学生の受講環境 (n=241)

形式	ノートブック型 88.8%, デスクトップ型 10%
OS	Windows 68.9%, Mac 29.9%
マイク・カメラ	5 名が無し (最終的には 1 名)
ネットワーク	光回線 71%, Mobile Wifi 8.3%, ADSL 6.2%, CATV 2.5%

用い、各作業スペースから Pike Place にアクセスし、プログラミング実習を行った。

学生の受講環境について授業初めにアンケート調査を行った。その結果 241 名より回答があり、結果は表 3 のとおりであった。光回線を使用している学生は約 70%であった。マイク・カメラを利用できない学生は、当初は 5 名であったが、本授業のために購入した学生もおり最終的には 1 名であった。

学生には授業時間中は自学自習や各自の演習の時間も含めて、常に CreCoSpeak にチェックインしていることが求められた。CreCoSpeak の運用上の注意点として、以下を指示した。

- チェックイン中はあなたのデスクトップが TA に送信されます。授業改善・学生支援目的のみ使用します。TA が支援するには情報が必要ですのでご協力下さい。
- 画面内容は評価対象とはなりません。みられたくないものは、デスクトップにおかないでください。
- チェックイン中は TA から通話がある可能性があります。円滑な遂行のため、受理を待たずにつなぎます。ノック音が来たら応対してください。TA からの通話が来次第カメラ・マイクが ON になります。相応の格好でお望みください。
- 顔を出すことで格段にコミュニケーションしやすくなります。TA は笑顔で応対しますので、みんなで明るい雰囲気を作ってください。
- 通話してほしくないときは、「通話しないで」アイコンを ON にしておいてください。カメラを拒否する場合には、カメラにシールを貼ることで対応ください。
- マイクがない人、トラブルありの人は、チャットで対応します

表 4 room 毎の TA による学生の call (ビデオ通話による対応) 状況概要

	学生			週															計	平均					
	学生数	出席率	平均	4	5	6	7	8	9	10	11	12	13	14	15										
room1	28	95%	2.09	40	56	51	43	34	45	27	33	41	24											394	39.4
room2	29	87%	2.50	62	66	39	31	29	47	17	41	46	37		31	18								415	41.5
room3	28	100%	2.28	81	73	59	45	36	50	19	30	58	34			28	24							485	48.5
room4	29	93%	2.09	42	51	53	29	32	40	23	25	37	29											361	36.1
room5	28	91%	1.70	31	41	41	31	27	32	22	24	33	21									30	13	303	30.3
room6	29	91%	1.85	26	43	35	35	32	32	31	30	32	25											321	32.1
room7	27	94%	2.03	40	45	40	48	37	37	32	15	30	37									35	14	361	36.1
room8	28	96%	2.32	50	52	45	38	45	42	36	26	41	31											406	40.6
total	226	93%	2.11	372	427	363	300	272	325	207	224	318	238	124	69									3046	38.1

## 6. 結果

### 6.1 開発と導入の状況

本システムの開発は、コロナ禍でのプログラミング授業運営という状況から、毎週開発と実践を行い、学生の意見を反映して安定した実装方法と機能を練り直すというインクリメンタル方式で開発を行った。そのため、本システムは1週目より導入が試みられたが、当初は安定稼働せず、ログ取得もできなかったため、安定稼働をはじめた4週目からの使用状況の記録になっている。学生同士の call 機能は開発が遅れたため、授業で導入したのは12週目である。

教務上は4クラス(1クラス約60名)に分かれているが、60名の常時デスクトップ監視については技術上の不安もあったため、13週目まではさらに2つの room に分割し、約30名で一つの room という運用とした。後半は技術が安定したため、14週目、15週目は4クラス(約60名1クラス)の運用を行い、滞りなく実施可能であった。

### 6.2 使用状況

CreCoSpeak の使用状況として、room 毎の TA による call 状況概要を表 4 に示す。

出席率は CreCoSpeak へのチェックイン状況で計測し、全授業時間の50%のチェックインが確認された学生を出席としている。概ね90%を超えており良好である。クラス間のばらつきに関しても、誤った room へ入っている学生も考えると誤差の範囲と考えられる。1授業あたり、1学生あたりの平均被 call 回数は2.11である。実験計画のミスにより、課題チェックと質問を区別するログを採取することができなかったため、正確な割合を算出できないが、課題チェックのための1回と、質問1回程度のコミュニケーションを行うことができたと考えられる。

次に、学生毎の call 受信回数(被 call 回数)を図 8 に示す。学生は毎回課題チェックを受ける必要があるため、それほど被 call 回数の分散は見られず、高々3倍ほどの差となっている。

次に、学生対応1回当たりの call 時間について、3分を

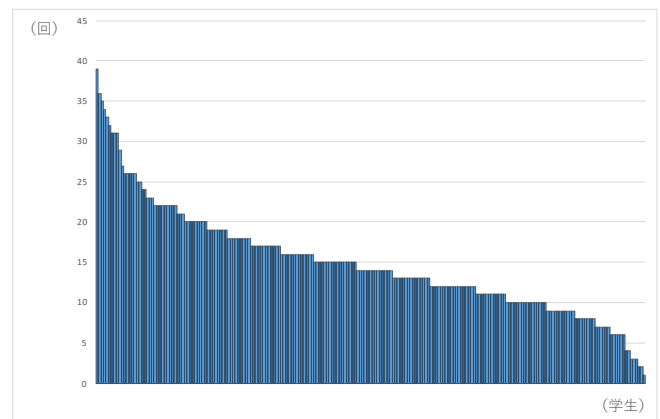


図 8 学生毎の被 call 回数

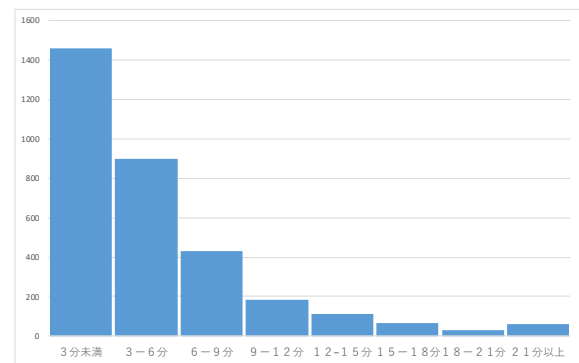


図 9 call 時間のヒストグラム

区間とするヒストグラムを図 9 に示す。本データに関しては、15秒未満のビデオ通話に関しては、何らかのトラブルが起こったものと考えてデータから除外している。3分未満が多く、徐々に短くなっていくロングテール形の結果となった。10分以内の対応が全体の9割ほどを占めている。

### 6.3 授業アンケートの年次比較

当該授業では、毎年共通の質問を利用して、プログラミングについての事前・事後意識調査を行っている。2019年度の授業で事前・事後それぞれに回答した1年生171名のデータと、2020年度の授業で事前・事後それぞれに回答した1年生191名のデータを用いた。4つの質問に対する回

表 5 授業アンケート結果の年次比較

	2019 年度 (n=171)			2020 年度 (n=191)		
	事前	事後	差	事前	事後	差
あなたはプログラミングが得意ですか？	1.89	2.03	+0.13	1.80	2.15	+0.35**
あなたはプログラミングが好きですか？	2.50	2.77	+0.26**	2.55	2.86	+0.30**
プログラミング授業では先生や TA によく質問をしましたか？	2.68	3.21	+0.53**	1.93	2.48	+0.54**
プログラミング授業ではクラスの人と話し合いましたか？	3.15	3.15	± 0.00	2.98	2.47	△ 0.52**

(\*<.05, \*\*<.01)

答項目に際して、4 段階のリッカート尺度により採取した回答を 1 から 4 の間隔尺度として解釈し、事前・事後のアンケートの各質問項目について分散分析を行った。

その結果を表 5 に示す。設問「あなたはプログラミングが得意ですか？」については、2020 年度、2019 年度ともに、事前より事後の方が高く、受講後に学生はプログラミングが得意に感じるようになっている。2020 年度、2019 年度間において、事前、事後ともに有意差は見られなかったが、学生個人の数値の変化については、2020 年度の方が大きく、2020 年度の授業の方が、学生がプログラミングを得意と感じるきっかけになっている。

設問「あなたはプログラミングが好きですか？」については、2020 年度、2019 年度ともに、事前より事後の方が高く、受講後に学生はプログラミングを好きに感じるようになっている。2020 年度、2019 年度間において、事前、事後、学生個人の数値の変化に有意差は見られなかった。これは、2020 年度はフルオンラインで授業を行ったが、学生がプログラミングを好きになるきっかけとしては対面授業である 2019 年度と変わらない成果をあげることができたと考察できる。

設問「今学期のプログラミング授業では先生や TA によく質問をしましたか？」については、2020 年度、2019 年度ともに、事前より事後の方が高く、本実習は、質問がしやすい環境を整備できたと評価できる。2020 年度、2019 年度間において、事前、事後ともに、有意差が見られたが、学生個人の数値の変化については、有意差は見られなかった。これは、2020 年度はほとんどの授業がオンライン授業であるため、2019 年度に比べ質問がしにくい環境であったが、本システムの導入により、他授業より、質問がしやすい環境を提供できたことが考えられる。

設問「今学期のプログラミング授業ではクラスの人と話し合いましたか？」については、2019 年度は、事前より事後で有意差はないが、2020 年度は、事後は事前と比較して数値が低い。2020 年度、2019 年度間において、事前では、有意差が見られないが、事後では、有意差が見られ、学生個人の数値の変化としては、2020 年度は 2019 年度に比べ、数値が下がっている。これらは、2020 年度の本授業でグループワークを導入したのが遅かったこと、グループ

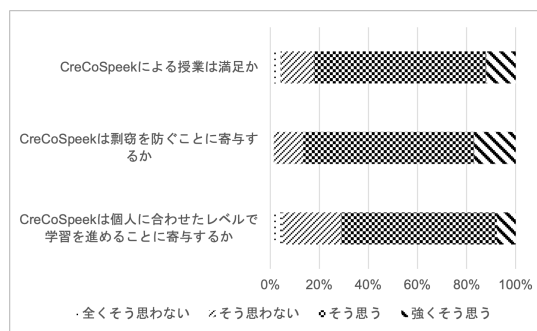


図 10 CreCoSpeak 基本項目 結果

ワーク機能の説明が不足していたことが原因であると考えられる。

## 6.4 CreCoSpeak に関するアンケート結果

2020 年度の受講生に対し、本システムについてのアンケートを行った。システム開発進行の都合で、アンケートは 2 度に分散して行っている。「基本項目」と「顔の見えるコミュニケーション」、「授業の印象」に関しては、第 1 回目のアンケート (第 10 週時点, 有効回答 142 件) によるデータである。「教え合いについては、第 2 回目のアンケート (第 15 週時点, n=195) によるデータである。自由記述に関しては、双方のアンケートから抽出している。

### 6.4.1 基本項目

基本項目は 4 段階のリッカート尺度での回答式である。基本項目の結果を、図 10 に示す。

設問「CreCoSpeak による授業は満足か」について、回答者の約 80% が「強くそう思う」、「そう思う」を選択している。これより、本システムを用いたフル・オンラインの授業の運営が、学生に受け入れられているということが考えられる。

設問「CreCoSpeak は剽窃を防ぐことに寄与するか」について、回答者の約 85% が「強くそう思う」、「そう思う」を選択している。これより、本システムには、提出課題の剽窃を防ぐ効果を見込むことができると考えられる。

設問「CreCoSpeak は個人に合わせたレベルで学習を進めることに寄与するか」について、回答者の約 70% が「強くそう思う」、「そう思う」を選択している。これより、本



表 6 call の際に、顔を隠す措置をしたことはあるか 結果

はい	11.3%
いいえ	88.7%

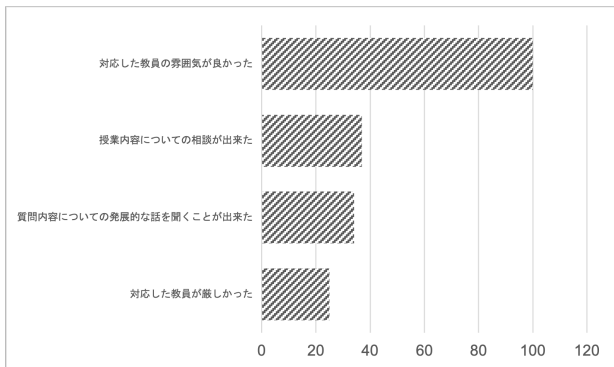


図 11 CreCoSpeak による授業の印象について 結果

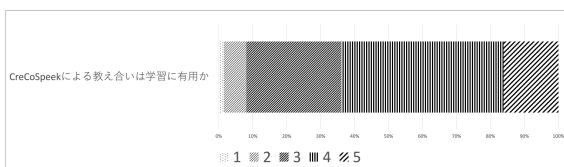


図 12 CreCoSpeak による教え合いについて 結果

システムを用いた授業は、受講している学生個人に合わせた指導を行うことができているということが考えられる。

#### 6.4.2 「顔の見えるコミュニケーション」への賛同の分析

設問「callの際に、顔を隠す措置をしたことはあるか」の結果を表 6 に示す。回答者の 88.7%が、「いいえ」を選択している。これは、「顔の見えるコミュニケーション」を構築するという本システム、本授業の目的に、大多数の学生が賛同しているものと考えられる。

#### 6.4.3 CreCoSpeak による授業の印象についての分析

設問「CreCoSpeak を利用し、先生や TA とのコミュニケーションで印象的だったことはあるか」の結果のうち最も回答数の多かった 4 つの項目を、図 11 に示す。最も回答数が多かったのは、「対応した教員の雰囲気が良かった」である。これは、回答者の約 72%が回答している。次点以降は、「授業内容についての相談が出来た」、「質問内容についての発展的な話を聞くことが出来た」、「対応した教員が厳しかった」であり、それぞれ約 26%、約 23%、約 18%が回答している。「対応した教員が厳しかった」を選択した回答者のうち、76%は「対応した教員の雰囲気が良かった」を選択している。これらより、本システムを用いた授業では、気軽に質問をできる環境を用意できたこと、TA の対応が厳しくも、明るい雰囲気を演出していたことが考えられる。

#### 6.4.4 CreCoSpeak による教え合いについての分析

設問「CreCoSpeak を使った教え合いをしたか」について、「利用した」を選択した回答者は、約 70%であった。利用した学習者に対して尋ねた設問「CreCoSpeak による教

表 7 CreCoSpeak の良かった点

P1	他の授業と比べて質問がともしやすく、オンライン授業であっても理解が十分にできました。
P2	先生や TA の方と 1 対 1 ではなせるので、気軽に質問ができ、間違っ理解していた部分を指摘してもらえる。
P3	生徒の画面が共有されることで、指示などが的確に通っていた。
P4	課題チェックの順番が見られるので、作業に集中することが出来る
P5	遠隔でもその場で教えてもらっているような感覚で教われたこと。
P6	良い緊張感を持って授業に取り組むことが出来た。
P7	オンライン授業を体面授業に限りなく近づけるためには必須のものだと思う。
P8	顔が見えるので会話しやすかった
P9	TASA さんや先生方が真摯に対応してくださった点。
P10	オンライン授業において課題チェックの効率を上げ、さらに小テストの際はカンニング防止の役割もなしていたという点で、画期的なシステムだったように感じます。
P11	誰が誰とグループワークをしているのが視覚的に分かるのが良かった。

え合いは学習に有用か」(5段階のリッカート尺度)の結果を図 12 に示す。平均は 3.71、分散 0.75、中央値 4 であった。1-2 を選択した回答者は約 8%であり、4-5 を選択した回答者は約 65%であった。

#### 6.4.5 自由記述の分析

良かった点の中で、特徴的なものを表 7 にまとめる。P1 は、質問のしやすさを指摘している。P2 は、教員と 1 対 1 で話せることで、学生は個人のペースで学習ができることを指摘している。P3 は、教員-学生間の画面共有によるコミュニケーションが、プログラミング学習に寄与していることを指摘している。P4 は、課題チェックが回ってくる順番を学生に示すことが、学生の作業への集中の手助けとなっていることを指摘している。P5, P6, P7 は、本システムを利用した学習環境が、対面に近い環境を作れており、緊張感を持った授業を展開できていることを指摘している。P8 は、顔を見て話すことが、コミュニケーションを助長していることを指摘している。P9 は、TA の誠実な対応が、質の高い授業を展開することを助長していることを指摘している。P10 は、本システムが、剽窃の防止に寄与していたことを指摘している。P11 は、グループワークにおいて、複数のセクションを使うこと、セクションに誰が入っているかを明確にすることが、重要であることを指摘している。

問題点の中で、特徴的なものを表 8 にまとめる。N1 は、質の高い課題チェックをするため、納得はしつつも学生は順番が回ってくるのが遅いことに不満であることを指摘している。N2 は、ライブ性の高い授業を展開するにあたり、画面を監視できることについて学生が不満を持っていることを指摘している、N3 は、画面を見ることと相手の表情

表 8 CreCoSpeak の問題点

N1	仕方ないけど課題チェックの待ち時間が長め
N2	自分の画面を常に見られてしまう。
N3	自分の画面を開いておかなければならないので相手 (TA) の表情が見えない。
N4	機能が十分に伝わっていなかった部分がありそうなので、もう少し説明があったらよかったです。
N5	グループワークの時間がもう少し欲しかった。
N6	グループワークは適当に割り振られた方がやりやすいと思いました。

を見ることの両立が難しい本システムの課題を指摘している。これは同時に、学生も通話相手の顔を見てコミュニケーションを取ることを重視していることも示唆していると考えられる。N4 は、本システムには、説明が不十分であり、使われていない機能があることを指摘している。N5 は、学生がグループワークの導入時期が遅かったことに不満を抱いていることを指摘している。N6 は、コロナ禍において友人をつくるのが難しいが、学生にとって、話すきっかけを作る場として、本システムが期待していたが、その目的を果たす運営ができていなかったことを指摘している。

## 7. 考察

本システム開発では、コロナ禍での新しい環境で教育を実践しながら学生や TA からの意見を収集し、必要な機能を絞り込むことで、数週目以降は安定して稼働し、約 230 名の学生各々に際し、2 コマあたり平均 2.1 回の質問に答えられるシステムを開発することができた。Zoom のブレイクアウトルームや Remo 等のシステムの利用だけではこのような品質での演習サービスを行うことは難しかったと考えられる。品質の良い TA サービスを行うことができたことで、質問への対応や、課題チェックにおける剽窃への疑念が払拭され、学生が安心して実習を行うことができた。学生の出席率が 90% を超えているという点も、対面と遜色ない動機づけを学生が保つことができる環境であったと評価できる。

学生の立場からはやはり顔を出すことに抵抗があったと思われるが、TA・講師陣が毎週明るく挨拶から顔の見えるコミュニケーションを実践することで、カメラを隠すことも許可していたにもかかわらず、約 9 割以上の学生が顔だして質問や課題チェックに取り組む姿も見られた。筆頭著者も講師として学生の対応をしたが、4 週目以降に関しては、対面と遜色なく指導に専念できたと感じており、多くの学生が顔を出して対応してくれる姿が見られ、気持ちよく指導をすることができた。学生によるデータや TA・講師陣からも同様の意見がみられ、「顔の見えるコミュニケーション」という当初の目標は概ね達成できたと評価している。

本研究では、2020 年度オンラインで実施した授業環境の変化に際し、対面で実施した 2019 同様のプログラミングに対する意識調査を行うことができたため、比較を行うことができたことも大きい。結果として、クラスの同僚との話し合いによる学習や、TA へよく質問をしたかどうかという項目に関してはポイントは低下したものの、プログラミングが得意になったか、プログラミングが好きになったか、といった項目では昨年と同様かそれ以上の成果を挙げており、ポイントの低下に関しても、自由記述欄にはコミュニケーション環境の不備に関する言及はなく、概ね好意的に評価するものであった。これらのことから、CreCoSpeak により、オンライン化によるコミュニケーション環境の悪化は軽微にとどまったのではないかと評価している。同僚との話し合いによる学習に関しては、その機能が当初から導入されていればより良い結果となったであろうと考えられる。

得られたデータと現場の実感を総合的に、提案システム「CreCoSpeak」の開発によって、対面で行うコンピューティング実習と同等以上の演習効果を得る実習を展開することができたのではないかと考えている。プログラミング教育研究としても、オンラインの活動により記録が残ることは教育研究を推進するという意味で有望性がある。今後は after コロナ時代のプログラミング教育に向けて、本研究で得られた知見を基としてハイフレックスによる授業の環境整備なども検討していく予定である。

**謝辞** 本研究開発・実践には NTT コミュニケーションズ株式会社が提供する SkyWay サービスを広範に利用させていただきました。対面当時の写真(図 1)は株式会社ギブリーの新田章太さんに提供いただきました。本研究は JSPS 科研費 19K12256 の助成を受けたものです。

## 参考文献

- [1] 新田章太, 小西俊司, 竹内郁雄: 複数言語に対応しやすいオンラインプログラミング学習・試験システム track, 情報教育シンポジウム論文集, Vol. 2019, pp. 114-121 (2019).
- [2] 安田 光, 井上亮文, 市村 哲: 学生とティーチングアシスタント間でトラブル解決過程を共有できるプログラミング演習支援システム, 情報処理学会論文誌, Vol. 53, No. 1, pp. 81-89 (2012).
- [3] Harel, I. and Papert, S.: Software Design as a Learning Environment. Interactive Learning Environments, Vol. 1, No. 1, pp. 1-32 (1990).
- [4] 野口剛史, 井手敬也, 長郷俊輔, 古賀雅伸, 矢野健太郎: PC を使う多人数講義における TA 業務支援システムの開発と評価, 研究報告コンピュータと教育 (CE-122-31) (2013).
- [5] 井垣 宏, 齋藤 俊, 井上亮文, 中村亮太, 楠本真二: プログラミング演習における進捗状況把握のためのコーディング過程可視化システム C3PV の提案, 情報処理学会論文誌, Vol. 54, No. 1, pp. 330-339 (2013).
- [6] 市村 哲, 梶並知記, 平野洋行: プログラミング演習授業における学習状況把握支援の試み, 情報処理学会論文誌, Vol. 54, No. 12, pp. 2518-2527 (2013).

- [7] 加藤利康, 石川 孝: プログラミング演習のための授業支援システムにおける学習状況把握機能の実現, 情報処理学会論文誌, Vol. 55, No. 8, pp. 1918–1930 (2014).
- [8] 横山裕紀, 江木啓訓: プログラミング演習講義における声掛けのための TA 支援システムの開発と評価, マルチメディア, 分散協調とモバイルシンポジウム 2017 論文集, pp. 851–856 (2017).
- [9] 大森 将, 垣内洋介, 松本慎平: 授業における PC 操作情報を用いた活動状況可視化手法, 教育システム情報学会誌, Vol. 36, No. 2, pp. 107–117 (2019).
- [10] 又吉康綱, 中村聡史: askTA: 消極的な受講生でも質問可能なオンライン演習講義支援システム, WISS 2020: 第 28 回インタラクティブシステムとソフトウェアに関するワークショップ (2020).