

# スマートフォンを用いた 鉋掛け技能学習支援教材の 開発と実践評価

---

宮城教育大学大学院教育学研究科(修士課程)

教科教育専攻 生活系教育専修

平成 26 年度 修士論文

13029 板垣 翔大



# 論文要旨

---

本研究では、中学校技術・家庭科(技術分野)における技能指導の問題点の解決や、技能習得を目的とした授業外学習の可能性を探ることを目指し、スマートフォンを用いた鉋掛け技能学習支援教材(以下、鉋ラーニング)、および練習データを管理するシステム(以下、PLMS: Practice Learning Management System)を開発した。

鉋ラーニングおよび PLMS の有用性を評価するため、本研究では 2 つの実験を行った。

1 つはペットボトルを鉋に見立てて、鉋ラーニングからのフィードバックを基に自宅で鉋掛けのシミュレーションにより練習することの効果を検証するものである。実験は本学学生 10 名を対象に行い、鉋ラーニングを用いる実験群 5 名と、用いない統制群 5 名の 2 群で比較を行った。結果として、鉋ラーニングを用いて練習を行うことで、引きの速さや鉋のブレの少なさ、手元と腰の動き出しのタイミングから算出される総合得点が有意に向上することが確認できた。一方で「上達は感じたが、やはりペットボトルでのシミュレーションと実際の鉋掛けでは感覚が異なる」という被験者からの意見が多くあった。本実験のような練習方法はあくまで鉋掛けの感覚を身に付けることに留まり、技能を習熟するにはためには、まだ課題が残されている。

もう 1 つの実験では、鉋ラーニングおよび PLMS を授業内で利用することの効果を検証した。仙台市内の中学校の第 1 学年の生徒 101 名を対象に行った。結果として、引きの速さ、鉋のブレの少なさ、手元と腰の動き出しのタイミングの 3 観点すべてにおいて有意な得点の向上を確認できた。授業者からも、PLMS を利用して机間指導を行うことで、技能の習得において伸び悩んでいる生徒に優先的に指導を行うことができるという点で、高評価を得ることができた。一方、実験からは、鉋ラーニングが生徒の興味・関心を高めていた反面、ゲーム的要素故に、技能の練習だけに集中しきれない者もいることが明らかになった。鉋ラーニングのみで練習を行うのではなく、練習を行うごとに得点を記入するワークシートと併用し、各自で与えられた評価を振り返らせるような工夫を行うと、より効果的であることが示唆された。また、生徒が集中しきれない原因としては、鉋ラーニングの操作の複雑性も挙げられる。操作を必要最低限に抑えることも課題である。

# 目次

---

第1章	はじめに	1
第1節	研究背景	1
第1項	技術教育の変遷	1
第2項	技術科におけるものづくりおよび技能指導	3
第3項	授業外での学習と技能学習	4
第4項	ICTを先端的に応用した技能指導における先行研究	5
第2節	本研究の目的	6
第3節	鉋掛けの技能について	8
第4節	本論文の構成	9
第2章	システム開発	12
第1節	開発方針	12
第2節	システムの概要	12
第1項	システムの概要	12
第2項	開発環境，動作環境および実行環境	13
第3節	鉋掛け技能学習支援教材について	16
第1項	機能の概要	16
第2項	専用鉋の製作	24
第3項	使用時の流れ	32
第4節	技能学習管理システムについて	38
第1項	機能の概要	38
第2項	データベース ER 図	46
第3章	開発したアプリケーションの家庭学習での利用と評価	49
第1節	調査目的と方法	49
第2節	結果と考察	52
第1項	引きの速さ	52
第2項	鉋のブレの少なさ	53
第3項	手元と腰の動き出しのタイミング	54

第4項	総合得点.....	55
第5項	考察・まとめ.....	56
第4章	開発したシステムの授業利用と評価.....	57
第1節	調査目的と方法.....	57
第2節	結果と考察.....	63
第1項	引きの速さ.....	63
第2項	鉋のブレの少なさ.....	65
第3項	手元と腰の動き出しのタイミング.....	67
第4項	総合得点.....	69
第5項	教材の動機付け効果の評価.....	71
第6項	教材の機能分析.....	75
第7項	考察・まとめ.....	77
第5章	結論.....	79
	謝辞.....	81
	本研究に関する発表論文.....	82
	図表目次.....	83

# 第1章 はじめに

---

本章では、技術科教育や技能指導の歴史的背景、技能指導や学校外での学習等における問題点を明らかにし、本研究の目的や意義について述べる。

## 第1節 研究背景

### 第1項 技術教育の変遷

現在の中学校の技術・家庭科の前身である「職業科」は1947年に成立した。当時の職業科の授業時数(必修)は3年間で420時間とされており、選択科目の選び方によっては、3年間で840時間であった。この職業科は1951年に「職業・家庭科」が発足するまでの4年間続いた。職業・家庭科となつてからは授業時数(必修)が3年間で315時間とされた。現在の「技術・家庭科」は1962年に新設された。新設当時の授業時数(必修)は職業・家庭科と同じく315時間であったが、その後5回の学習指導要領の改訂を経て、現行の学習指導要領では3年間で175時間(必修)となり、選択科目としての技術・家庭科の時間は無くなった。図1-1は職業科、職業・家庭科、技術・家庭科の授業時数(必修)の変化である。時代を追うと共に授業時数が減少していることが把握できる。

一方、授業で扱うべき内容は増加傾向にある。1962年に技術・家庭科が新設された当時は男女別学であり、男子は主に現在の技術分野(以下、技術科)で扱われている内容のみを、女子は主に家庭分野の内容のみを履修すればよかったが、1977年の学習指導要領の改訂では、男子が家庭分野の内容を、女子が技術分野の内容を履修する「相互乗り入れ」が可能になり、1989年の改定以降は、男女別学が完全に撤廃され、実質的に技術科の授業時数は半分になった[1-1]。

直近の学習指導要領改訂(2010年)では、授業時数の減少はなかったが、これまで選択の内容であった、エネルギー変換や生物育成、プログラミング等に関する内容が全て必修とされ、各領域や単元に割くことのできる時間が限られている。河合ら(2014)は、山形県の技術科教員にアンケート調査を行った[1-2]。その結果、技術科の教員が感じている指導上の問題(教科の内容以外)として授業時数不足が最も多く挙げられた。この結果からも授業時数不足が大きな課題の1つであることがわかる。

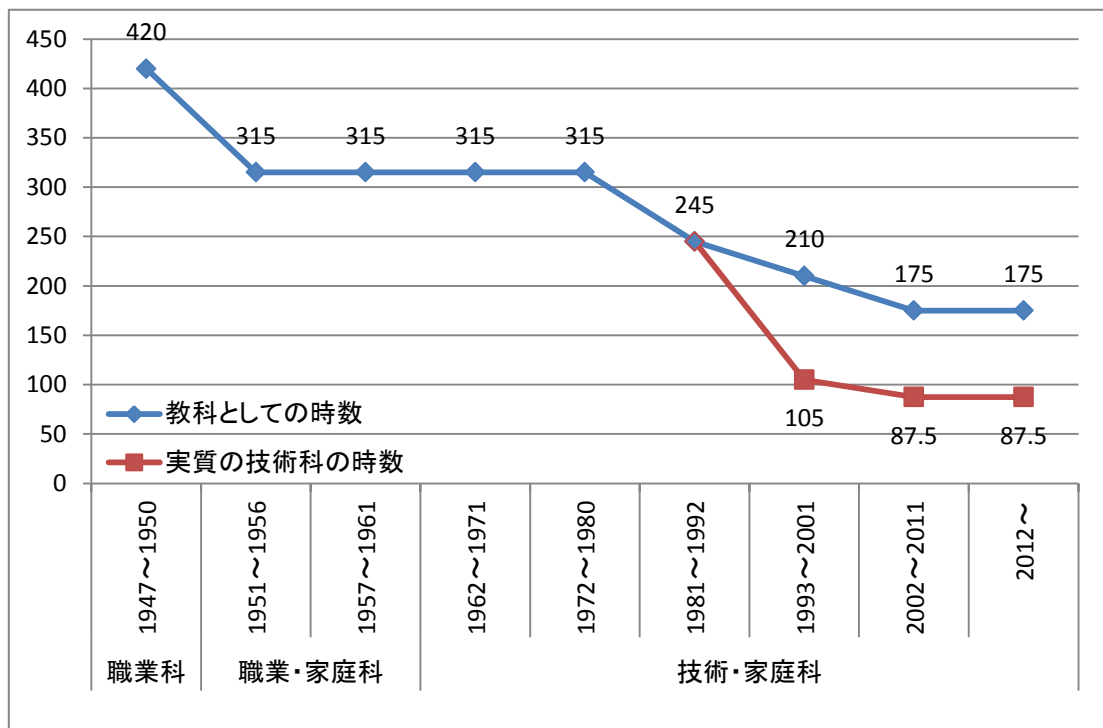


図 1-1 技術教育にあたる教科の授業時数の変化

## 第2項 技術科におけるものづくりおよび技能指導

技術科の特徴として、製品の製作等における、工具の使用などの技能学習が伴うことが挙げられる。工具の正しい使用技能を身に付けることは、製品の仕上がりの質を左右する。また、製品の仕上がりは生徒の意欲に影響する。安孫子ら(2002)の調査では、子供のものづくり経験の乏しさが指摘されており、ものづくりの機会を可能な限り確保し、生活経験が不足している分、より丁寧に指導することが必要であるとされている[1-3]。

しかし、技術科の免許を持たずに技術科の授業を担当している教員は少なくない。中菌ら(2012)によれば、鹿児島県では43.9%の教員が免許外で技術科の授業を担当している[1-4]。現在の技術・家庭科の目標は「生活に必要な基礎的・基本的な知識および技術の習得を通して、生活と技術とのかかわりについて理解を深め、進んで生活を工夫し創造する能力と実践的な態度を育てる。」と定められている[1-5]。この目標の中にもある「基礎的・基本的な知識および技術」を習得させるためには、専門的な知識および技術を有する教員が指導することが望ましい。特に、工具を使用する技能の指導にあたっては、教師が正しい技能を生徒に演示し、学ばせる形態が一般的である。免許外の教員が適切に工具の使い方を理解し、模範的な使用動作を演示できるだけ熟練しているかどうかについては疑問が残る。



### 第3項 授業外での学習と技能学習

第1項で、技術科の授業時数不足について述べた。授業での学びを充実したものにするために、学校では「宿題」として学習者に学習を求めることが多い。家庭学習では主に、教科書や問題集、学習プリント等を用いて予習や復習が行われる。昨今では、授業に先立ち知識の習得を行い、授業で事前に学んだ知識を活用するディスカッション、問題解決学習などを行う、反転授業(Flippled Learning)も注目されている[1-6]。その他にも、e-Learning やMOOC(Massive Open Online Course:大規模公開オンライン講座)など、指導者から対面で学ぶのではなく、ICT を活用し教材から学習したり、遠隔地で学習したりする学習形態もみられる[1-7]。しかし、これらの学習形態では、主に知識の習得を目的としており、技能の習得は目的とされていないのが現状である。

島(2007)は、大学の実技科目の授業に e-Learning を取り入れた[1-8]。この取り組みは、授業で行う競技のルールや基本的な動作を、学習者に事前に動画教材等で学習させる、反転授業的なe-Learningの活用である。この授業を受けた学生からは、「動作をあらかじめ知ることができる」、「あらかじめ見ておくと授業中の説明を理解しやすい」といった意見があった。一方で、実際にその動作を行うことに関しては、自分がどのような動作をしているのか、適切な動作をできているのかどうかを確認することができず、学生は動作の自学を難しく感じていた。

著者ら(2013)は、技能学習における指導者の役割を次のように捉えている[1-9]。

1. 学習者に模範的な動作を演示して、実際の動きを理解させる役割
2. 学習者の動きを見て確認する役割
3. 学習者の技能改善のための助言をする役割

そのため、特に授業外や遠隔地での技能学習、技能の反転授業等を可能にしようとした場合では、上記の 2「学習者の動きを見て(評価して)確認する役割」および 3「学習者の技能改善のために助言をする役割」を実現することが困難となる。これらの実現のためには、上記の役割を担う別の手段や方法が必要であると考えられる。島らの取り組みでは、1「学習者に模範的な動作を演示する役割」を動画教材が担っていると言える。これに加え、上記の 2「学習者の動きを見て確認する役割」や、3「学習者の技能改善のための助言をする役割」を担う道具や方法があれば、学習者が感じていた動作の自学の難しさを軽減できるのではないかと考えられる。

## 第4項 ICT を先端的に応用した技能指導における 先行研究

先端的に ICT を応用した技能指導の先行研究として、鋸挽き動作の観察教材がある[1-10]。これは、熟練者の鋸挽きの動きをモーションキャプチャし、その動作を 3 次元仮想空間内で再現したものを生徒が観察して正しい技能を捉えるためのものである。ビデオ教材と比較すると、視点の移動が可能であり、死角が存在せず自由にあらゆる角度から観察できる点が特徴である。これは前項で示した技能学習における指導者の役割の 1「学習者に模範的な動作を演示して、実際の動きを理解させる」を担うものとして位置付けられる。

また、同じく先端的に ICT を応用した先行研究として、スマートフォンを用いた鋸挽き技能習得教材(のこぎりラーニング)も開発されている[1-11]。これは、鋸に取り付けたスマートフォンに内蔵されたセンサが、学習者の鋸挽きの動きをセンシングし、評価・フィードバックを与えるものである。近年販売されているスマートフォンの多くは加速度センサやジャイロセンサ(角加速度センサ)を搭載しており、そのセンサから得られる値を、専用のアプリケーションを開発することにより利用している。これは、前項で示した技能学習における指導者の役割 2「学習者の動きを見て確認する」や 3「学習者の技能改善のための助言をする」を担うものとして位置付けられる。

## 第2節 本研究の目的

前述の中学校技術科や技能指導における課題を踏まえ、本研究では問題の所在を以下のように捉えた。

- 中学校技術科においては、必修事項が充実した一方で各内容に充てられる授業時間は減少し、十分な学習時間を確保することが困難である
- 子供のものづくり経験や生活経験が不足している中、免許外の教員が少なくなく、十分に技能の指導をできているかどうか疑問が残る
- 授業外での学習や、e-Learning, MOOC, 反転授業等は、主に知識の習得が目的とされており、技能の習得を目的とした際に現状の学習方法だけでは不十分である

第1節第4項で述べた先行研究では、上記の問題を解決すべく、鋸挽きの技能に焦点を当て、スマートフォンを用いた技能習得教材の開発・評価を行った。中学校技術科での木材加工では、手工具を使用して切る、削る、叩く、掘る、組み立てるなどの基礎的な作業が指導されている[1-12]。本研究では、その中の「削る」にあたる「鉋掛け」の技能に焦点を当て、技能学習支援教材の開発を行うこととした。鉋を用いた加工は、平削り、木端削り、木口削りがあるが、本研究では平削りの動作を対象とする。平削りは製作の仕上がりに大きく影響する技能であるが[1-13]、木材加工の中でも難しく、熟練を要する技能である。その難しさは鉋身の砥ぎや調整などにもあるが、中でも削りの動作については、従来から「勘」や「こつ」といったものに基づいた感覚的な表現による説明が目立つ[1-14]。

ところで、2014年の総務省の調査によれば、スマートフォンの国内の普及率が62.6%であり、同調査でスマートフォンに関する項目が追加された4年前(2010年)の9.7%と比較すると急速に普及が進んでいることがわかる[1-15]。各家庭に普及しているスマートフォンを用いることで、専用のセンサや装置等を整備する必要がない点や、センシングやフィードバックに必要なもの(各種センサ、ディスプレイ、スピーカ等)が1台に収まっている点に、スマートフォンを用いることの意義があると考えている。

ここで、本研究の位置付けを明確にするために、技能の授業外学習や反転授業を行うために踏むべき段階を、技術科での工具使用技能を例に、認知的領域の改訂版タクソミーを基に考える。改訂版タクソミーでは、教育目標が段階ごとに「知識次元」、「記憶する」、「理解する」、「応用する」、「分析する」、「評価する」、「創造する」と分類されている[1-16]。これに則った対面での技能の学習としては、以下のようなものが考えられる。

1. 工具に対する適切な言語情報が理解できる(知識次元, 記憶する, 理解する)
2. 指導者の模範的な工具使用動作を観察し識別できる(記憶する, 理解する, 分析する)
3. 実際に工具を使用して指導者の動作を模倣できる(理解する, 応用する)
4. 指導者から技能を改善するための助言を受けて, 自己修正の目途を立てる(分析する, 評価する)
5. 繰り返し練習を行い, 自己の身体動作を自己修正する(分析する, 評価する, 創造する)
6. 4や5を繰り返し, 身体動作を自動化, 最適化しはじめる(分析する, 評価する, 創造する)

以上を踏まえ, 技能の授業外学習や反転授業を行うために必要なステップを図 1-2 のように考えた。

第 1 段階は知識ベースの学習であるため, 従来通りのアプローチで実現できると考えられる。

第 2 段階では, 活用できる教材の 1 つとしてビデオ教材が挙げられる。また, 前節で触れた 3 次元仮想空間内での技能観察教材[1-10]もここに位置付けられる。

同じく前節で触れたスマートフォンを用いた鋸挽き技能学習教材[1-11]は第 3 段階に相当する。本研究もこの第 3 段階に位置付けている。

正しい動作習得の後に, 第 4 段階として, その動作での加工により, 切削面等が実際に好ましい状態に仕上がっているのかどうかを評価する必要がある。また, 最終的には下位の 4 段階を踏まえて, 授業実践を交えながら, これらを効果的に活用するための方法を検討する必要があると考えている。

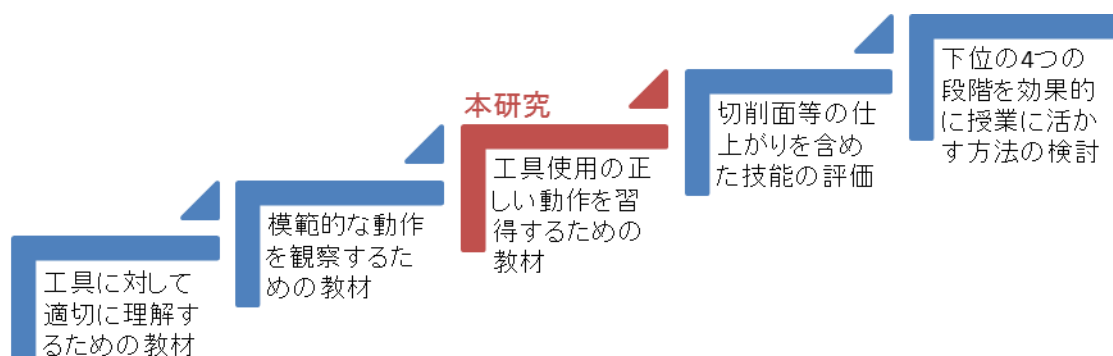


図 1-2 技能の授業外学習や反転授業を行うためのステップと本研究の位置付け

## 第3節 鉋掛けの技能について

鉋掛け(平削り)の技能学習支援教材を開発するにあたり、先行研究から理想的な鉋掛けの動作や評価すべき観点等をまとめる。

田中ら(1990)は、鉋掛けの動作における身体重心の移動を、熟練者、経験者、初心者の動作の比較を通して分析した[1-17]。結果として、熟練者ほど、下肢や重心の移動を使って、全身で削っていることが明らかとなった。また、鉋を引く速度に関して、熟練者ほど速度(特に初速度)が速く、また、削り終わりに鉋を一気に静止させている様子が確認された。

安孫子ら(2005)は、デジタルカメラを用いた3次元動作分析で、熟練者と未熟練者の動作を比較した[1-18]。比較の結果、熟練者の特徴として、初速度が速く、引き終わりに意識的に鉋を静止させていた、腰の引きを使った動作がみられる、などが挙げられた。

橘田ら(2003)は、中学生の鉋掛けの動作を筋電位と画像解析から分析した[1-19]。結果として、1. 鉋切削の初速を大きくして一気に引く、2. 鉋を押さえつけることより引く力を有効に発揮させる、3. 上肢に依存するより、下肢を踏ん張って体重移動を有効に利用する、4. 削り終わり時は鉋を押さえ、落とさずに水平に移動するよう支配する、の4点を身に付けた生徒が効果的な鉋掛けを行うことがわかった。

また、橘田ら(2003)は、鉋台の握り位置が切削動作に及ぼす影響について分析し、台の中央部を握るよりも、台尻部分を握るよう指導すべきであるとの判断をした[1-20]。その際、分析の観点として、1. 平均切削速度、2. 切削面の精度(傷出現数、削り屑厚さ、削りの腰面積)、3. 筋電位分布の3つを用いた。

田中ら(1991)は「かんな削り訓練装置」を開発した[1-14]。この装置では 1. 被削材の先端から末端までを削っているかの判定、2. 鉋削りの速度の結果を即座に学習者にフィードバック、3. 学習者と熟練者の鉋削りの速度の比較の3つを行うことができる装置である。訓練の効果を検証する実験からは、「従来よりも短い練習時間で熟練者と同じような一定速度による鉋削りができるようになった」という結果や、装置を使用したことで「ただ漫然と削るのではなく、評価されることで学習目標を常に維持できる」、「熟練者との速度の違いを比較できるので、自分の欠点がどこか理解できる」といった使用者からの感想が得られた。

以上の先行研究より、本研究で鉋掛け技能学習支援教材の開発をするにあたり、熟練者と学習者の動きを比較しながら、1. 鉋の引きの速さ、2. 鉋のブレの少なさ、3. 全身を使って引いているか、の3観点を評価するものが有効ではないかと考えた。引きの速さと全身の使い方は、多くの先行研究で扱われている観点であり、熟練者と未熟練者の差が明らかなものであり、評価・改善することで鉋掛けが上達すると考えられる。鉋のブレの少なさは、ただ力任せに引く

だけでは、切削面が美しく仕上がらず、傷や削り残しが残る。速く引きつつも丁寧に引く必要があることから、ブレの少なさも重要な要素と捉え、評価の対象とした。

## 第4節 本論文の構成

第1章(本章)では、技術科の歴史的背景や、技能の授業外学習をしようとした際の課題、および、研究の目的等について述べた。

第2章では、開発した鉋掛け技能学習支援アプリケーションと、それをを用いて練習した際のデータを管理するシステムの機能の概要等について述べる。また、本研究で用いる、スマートフォンを装着することのできる専用鉋の製作についても触れる。

第3章では、開発したアプリケーションの家庭学習での利用可能性を検証する調査の調査方法、結果および考察について述べる。本調査では、大学生を対象にし、自宅ではペットボトルを鉋に見立てて、鉋掛けのシミュレーションを通して練習を行った。

第4章では、開発したアプリケーションやシステムの授業利用における有用性の検証をする調査の方法、結果および考察について述べる。本調査では、仙台市内の中学校の1年生を対象に、専用鉋を用いて実際に鉋掛けをすることを通して練習を行った。

第5章では、本研究の結論として、研究全体のまとめや、得られた知見、今後の課題等についてまとめる。

なお、付録として、本研究に関する発表論文、および開発したアプリケーションやシステムのソースコードを加える。

## 参考文献

- [1-1] 日本産業技術教育学会 技術教育分科会:新 技術科教育総論, pp.1-22 (2009).
- [1-2] 河合康則, 瀬尾和哉, 日高貴志夫, 三澤正太:山形県における技術科教育の実態と課題に関する調査研究, 日本産業技術教育学会第 32 回東北支部大会講演論文集, pp.35-36 (2014).
- [1-3] 安孫子啓, 安藤明伸, 河合康則:小学校におけるものづくり教育の実態に関する調査研究, 宮城教育大学紀要, 第 37 巻, pp.223-232 (2002).
- [1-4] 中藪政彦:技術・家庭科教育の過去から未来を見る。(調査研究), 第一工業大学研究報告, 第 24 号, pp.63-74 (2012).
- [1-5] 文部科学省:中学校学習指導要領, pp.98-104 (2008).
- [1-6] 重田勝介:反転授業 ICT による教育改革の進展, 情報管理, 第 56 巻, 第 10 号, pp.677-694 (2013).
- [1-7] 船守美穂:21 世紀の新たな高等教育形態 MOOCs(1)世界で広がる無料のオンライン講義とは, カレッジマネジメント, 第 31 巻, 第 4 号, pp.36-41 (2013).
- [1-8] 島健:e ラーニングを使用した体育実技の授業方法に関する研究～実技科目への LMS の導入と問題点～, 上智大学体育, 第 40 巻, pp.1-13 (2007).
- [1-9] 板垣翔大, 安藤明伸, 高久敏宏, 安孫子啓:鉋の平削り動作の学習履歴管理システムに必要な要件とは何か, 日本産業技術教育学会第 19 回技術教育分科会講演要旨集, pp.11-12 (2013).
- [1-10] 安藤明伸, 住川泰希:モーションキャプチャと仮想空間を利用した鋸挽き動作観察教材の開発と機能評価, 日本教育工学会論文誌, 第 36 巻, 第 2 号, pp.103-110 (2012).
- [1-11] 安藤明伸, 高久敏宏, 板垣翔大, 鳥居隆司, 竹野英俊:スマートフォンによるセンサーアシストトレーニング法の開発, モバイル学会モバイル'13 研究論文集, pp.71-72 (2012).
- [1-12] 藤冲零士, 田中通義, 向井辰徳:平鉋による平面削り指導のための右手の力と左手の力の計測システムの開発と評価, 日本産業技術教育学会誌, 第 55 巻, 第 3 号, pp.207-212 (2013).
- [1-13] 安孫子啓:平かんな削りの指導に関する基礎的研究, 宮城教育大学紀要, 第 32 巻, pp.227-231 (1997).
- [1-14] 田中通義, 篠田功, 山下晃功, 安孫子啓:かんな削り訓練装置の開発と訓練効果, 日本産業技術教育学会誌, 第 33 巻, 第 4 号, pp.251-256 (1991).
- [1-15] 総務省:平成 26 年度情報通信白書, p.169 (2014).

- [1-16] 石井英真:「改訂版タキノミー」によるブルーム・タキノミーの再構築—知識と認知過程の二次元構成の検討を中心に—, 日本教育方法学会紀要「教育方法学研究」第28巻, pp.47-58 (2002).
- [1-17] 田中通義, 安孫子啓:かんな削り動作における身体重心の移動分析について, 日本産業技術教育学会誌, 第32巻, 第4号, pp.249-255 (1990).
- [1-18] 安孫子啓, 安藤明伸, 池田晃一, 寺岡武:平かんな削りの指導に関する基礎的研究 未熟練者の切削動作における三次元分析, 宮城教育大学紀要, 第40巻, pp193-199 (2005).
- [1-19] 橘田紘洋, 近藤文彦, 白井博成:中学生における効果的なかんな切削動作に関する研究, 愛知教育大学教育実践総合センター紀要, 第7巻, pp297-303 (2004).
- [1-20] 橘田紘洋, 近藤文彦, 白井博成:中学生における効果的なかんな切削動作に関する研究 かんな台の握り位置が切削動作に及ぼす影響について, 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, 第52巻, pp35-39 (2003).



## 第2章 システム開発

本章では、開発したスマートフォンアプリケーションと Web アプリケーションの機能の概要やシステム全体の構成等について述べる。

### 第1節 開発方針

本研究での開発は、第1章で述べた、スマートフォンを用いた鋸挽き技能習得システムを拡張する形で行った。そのため、UI(User Interface)を極力統一し、操作感の違いによる使い勝手の悪さがないようにした。また第1章で述べたように、本研究で開発するシステムでは、手元の動きの評価だけでなく、腰を使って全身で鉋を引くことができているかどうかを評価する。そこで、スマートフォン2台を1セットとして使用し、1台は鉋に取り付けて手元の動きを、もう1台は腰付近に取り付けて腰の動きをセンシングすることとした。

### 第2節 システムの概要

#### 第1項 システムの概要

本システムは、主に学習者が使用するスマートフォンアプリケーションと、主に指導者が使用する Web アプリケーションで構成されている(図 2-1)。学習者がスマートフォンアプリケーションを用いて鉋掛けの練習を行うと、そのデータがサーバへ送信・蓄積され、指導者はそのデータを一元管理できる。詳細は、アプリケーションについて第3節で、システムについて第4節で述べる。なお、本研究では、開発したスマートフォンアプリケーションを「鉋ラーニング」と呼称し、開発したシステムを PLMS (Practice Learning Management System) と呼称する。

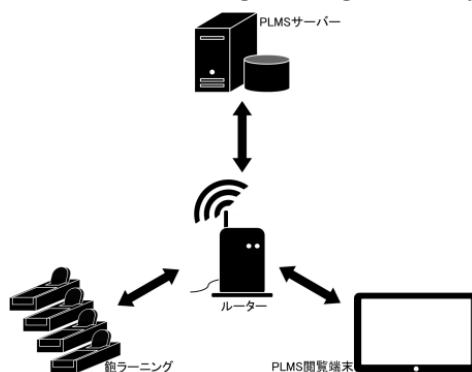


図 2-1 開発したシステムの構成

## 第2項 開発環境，動作環境および実行環境

本研究で鉋ラーニングおよび PLMS の開発に用いた PC の環境を表 2-1 に示す。また，鉋ラーニングの動作環境を表 2-2 に示す。PLMS は Web サーバソフトウェアがインストールされており，無線もしくは有線 LAN での接続に対応しているコンピュータで動作する。システムの構成の詳細については本章第 4 節で述べる。

表 2-1 開発環境

OS	Windows 8.1Pro with Media Center
CPU	Intel Core i3 540 3.07GHz
メモリ	16GB
HDD	1.5TB
その他	Eclipse 4.2.1 JUNO

表 2-2 鉋ラーニングの動作環境

OS	Android2.3.3 以上
通信	Wi-Fi, Bluetooth
センサ	加速度センサ, ジャイロセンサ

本研究での飽ラーニングの実行環境について、使用したスマートフォンのスペックを表 2-3 および表 2-4 に示す。使用した端末は、LG エレクトロニクス社の Nexus 5[2-1]およびサムスン電子社の Galaxy Nexus[2-2]である。PLMS の実行には Microsoft 社の Surface Pro 2[2-3]を用いた。詳細なスペックや使用したソフトウェアは表 2-5 の通りである。

表 2-3 Nexus 5 のスペック

OS	Android4.4 → Android5.0(アップデート)
CPU	Qualcomm Snapdragon™ 800, 2.26 GHz
RAM	2GB
ストレージ	16GB
サイズ	69.17 × 137.84 × 8.59mm
重量	130g
	4.95 インチ フル HD(1920×1080)
ディスプレイ	静電容量式 マルチタッチ
通信	Wi-Fi(2.4G/5G) 802.11 a/b/g/n/ac Bluetooth 4.0 LE
センサ	加速度計, GPS, コンパス, 近接/周囲光, ジャイロスコープ, 気圧計
バッテリー	2,300 mAh

表 2-4 Galaxy Nexus のスペック

OS	Android4.0 → Android4.2.2(アップデート)
CPU	Texas Instruments OMAP 4460 1.2GHz
RAM	1GB
ストレージ	16GB
サイズ	68 × 136 × 8.9mm
重量	138g
	4.7 インチ HD(1280×720)
ディスプレイ	静電容量式 マルチタッチ
通信	Wi-Fi(2.4G/5G) 802.11 a/b/g/n/ac Bluetooth v3.0
センサ	加速度計, GPS, コンパス, 近接センサ, ジャイロスコープ, 気圧計
バッテリー	1750 mAh

表 2-5 Surface Pro 2 のスペック

OS	Windows 8.1 Pro, Office Home and Business 2013
CPU	Intel Core i5 4300U 2.50GHz
RAM	8GB
ストレージ	256GB
サイズ	257 × 173 × 13.5mm
重量	907 g
	10.6 インチ フル HD(1920×1080)
ディスプレイ	静電容量式 マルチタッチ
通信	Wi-Fi (2.4G/5G) 802.11 a/b/g/n/ Bluetooth 4.0
センサ	光センサ, 加速度センサ, ジャイロスコープ, 電子コンパス
バッテリー	1750 mAh

## 第3節 鉋掛け技能学習支援教材について

### 第1項 機能の概要

本アプリケーション(以下、鉋ラーニング)は、2 台のスマートフォンを用いた鉋掛け技能学習支援教材である。2 台のうち、1 台を鉋に取り付け(以下、鉋端末)、もう 1 台を腰付近に装着して(以下、腰端末)鉋掛けの練習を行う。図 2-2 は鉋ラーニングを用いて鉋掛けの練習を行っている様子である。学習者のズボンのポケットには腰端末が入っている。

鉋ラーニングでは、事前に熟練者の鉋掛けの動きをモデルとして登録しておく。学習者が本アプリケーションを用いて練習すると、モデルの動きと比較され、評価される。モデルの登録の詳細に関しては後述する。



図 2-2 鉋ラーニングを用いた鉋掛けの練習の様子

学習者が鉋ラーニングを用いて練習を行うと、1. 引きの速さ、2. 鉋のブレの少なさ、3. 手元と腰の動き出しのタイミングの 3 つの観点において動きが評価され、結果が表示される。なお、引きの速さおよび手元と腰の動き出しのセンシングには加速度センサを用い、鉋のブレのセンシングにはジャイロセンサを用いている(表 2-6)。

表 2-6 使用しているセンサ

センシングの対象	使用しているセンサ
1. 引きの速さ	鉋端末の加速度センサ
2. 鉋のブレの少なさ	鉋端末のジャイロセンサ
3. 手元と腰の動き出しのタイミング	両端末の加速度センサ

2 台の端末は Bluetooth により接続されている。腰端末でセンシングされた全てのデータは鉋端末へ送信される。鉋端末では、鉋端末のセンサから取得したデータと、腰端末から受信したデータの両方を集計し、学習者へ結果と助言を表示する。

結果の表示には以下の 3 種類がある。それぞれについて、鉋ラーニングの画面と共に詳細を述べる。

1. 言葉による結果表示とアドバイス(図 2-3)
2. 速度および加速度のグラフ(図 2-4)
3. 速度を再現するリプレイアニメーション(図 2-5)

1. 言葉による結果表示と KR 情報としてのアドバイスではあらかじめ登録されているモデルの速度やブレの少なさを 100 点とした場合の点数や、それに応じたランクが表示される。ランクは星の数で表され、最大で星が 3 つ、最少で星が 1 つである。また、その結果を踏まえ、次回以降の練習で意識すべき点がアドバイスとして表示される。

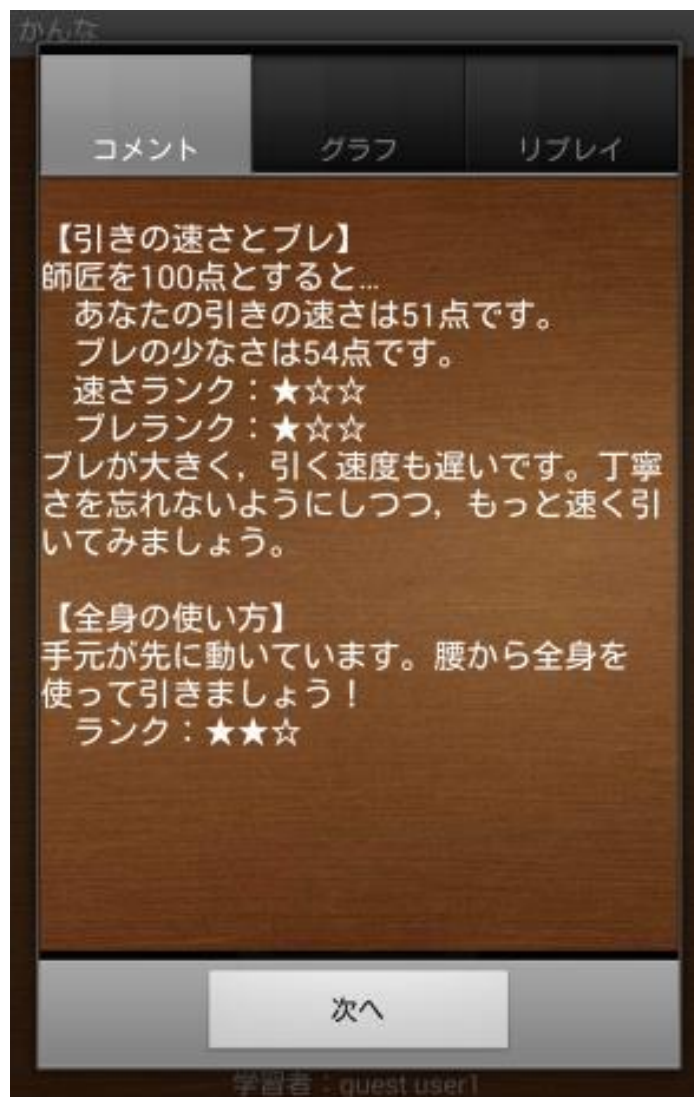


図 2-3 言葉による結果表示とアドバイス

2. 速度および加速度のグラフでは、練習した際の速度や加速度の変化のグラフが、あらかじめ登録されているモデルのグラフと重ね合わせて表示される。また、腰が動き出したタイミングは、横軸上に「○」で表示される。学習者は熟練者との速度の違いを視覚的に確認することができる。例えば図 2-4 のような結果が表示された場合、黄色で示された熟練者の動きは、幅が狭く速度が大きいことに対し、青で示された学習者の動きは、幅が広く速度が小さい。このことから、学習者は熟練者に比較し、素早く一気に引くことができていないことがわかる。また、熟練者は速度が上昇する前に「○」表示されているが、学習者は速度が上昇してから「○」が表示されている。このことから、熟練者は手元よりも腰が先に動いているが、学習者は手元が先に動いていることがわかる。

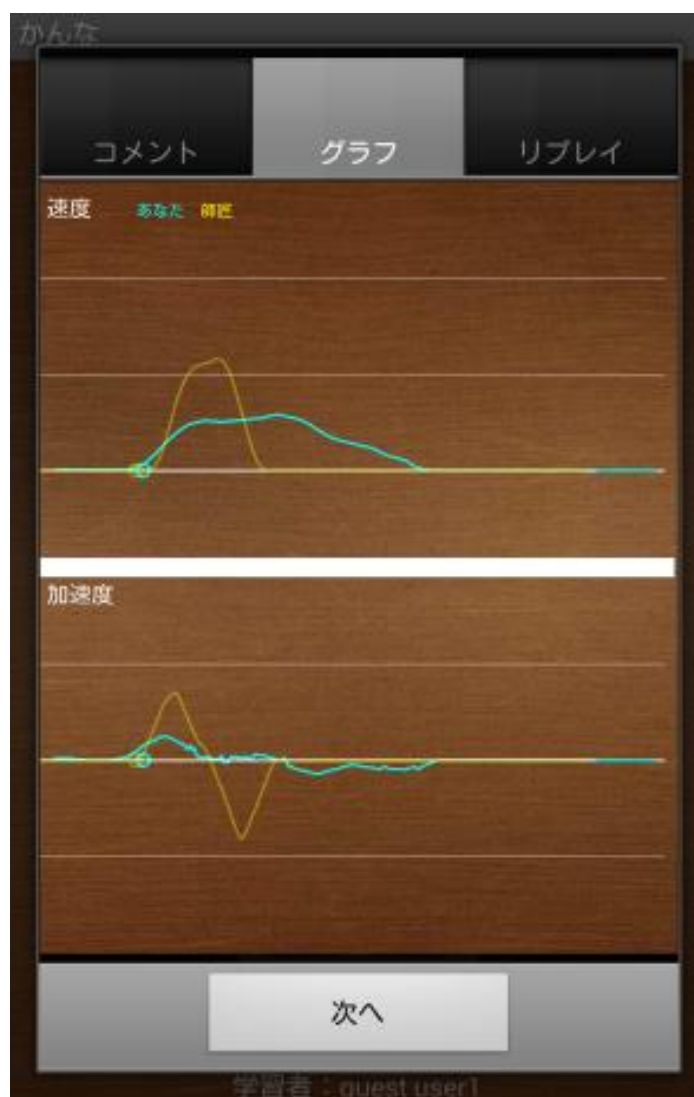


図 2-4 速度および加速度のグラフ



3. 速度を再現するリプレイアニメーションでは、練習時の匏の速度と、熟練者の匏の速度がアニメーションで再現される。学習者は速度の違いをより直感的に比較することが可能である。



図 2-5 速度を再現するリプレイアニメーション

結果が表示された後、図 2-6 のような結果の履歴が表示される。過去の点数のデータはサーバに保存されているものを取得して描画している。これにより、学習者は自身がどのように上達しているのか、どの点で伸び悩んでいるのかなどを把握することができる。サーバとの関係等の詳細については第 4 節で述べる。

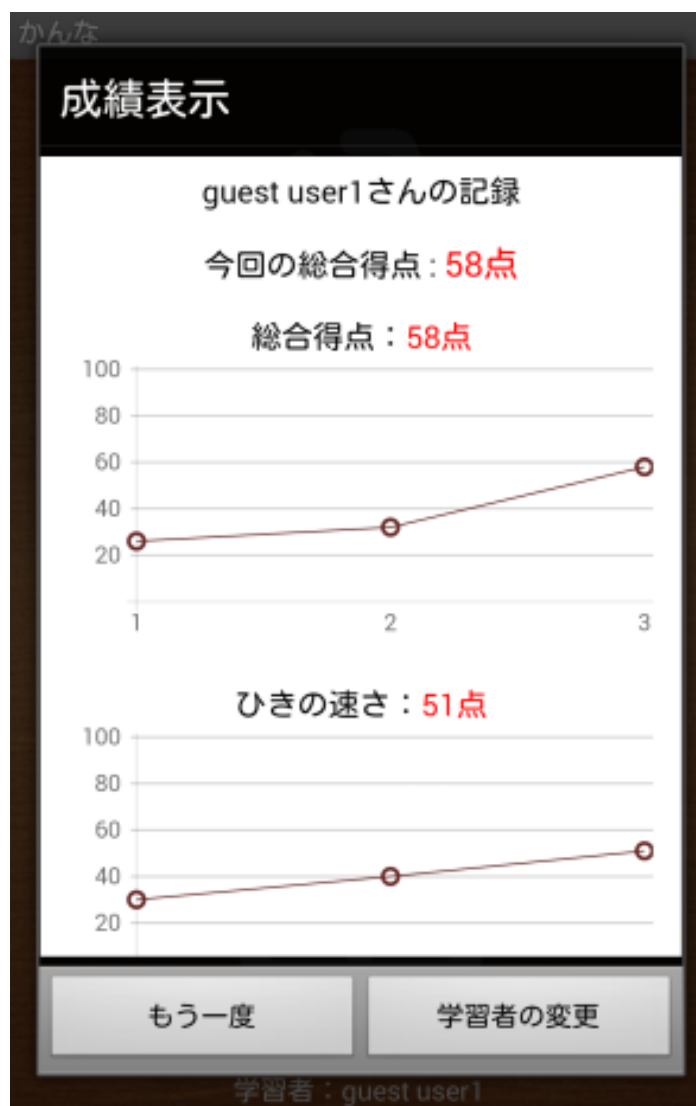


図 2-6 上達履歴表示

## 各得点の算出方法について

引きの速さは、登録されたモデルの速度の最大値を 100 とした場合の割合を得点としている。学習者の速度がモデルの速度を超えた場合は全て 100 点となるようにした。次式は、学習者の速度の最大値を  $V_L$ 、モデルの速度の最大値を  $V_M$  とした際の、引きの速さの得点  $P_V$  を求める式である。例えば、モデルの引きの速さの最大値が 2000[mm/s] で、学習者が 1500[mm/s] だった場合、速さの得点は 75 点となる。

$$(V_L < V_M \text{ のとき}) \quad P_V = \frac{V_L}{V_M} \times 100$$

$$(V_L \geq V_M \text{ のとき}) \quad P_V = 100$$

ブレの少なさは、学習者のジャイロセンサの値の最大値を 100 とした場合の、登録されたモデルの値の割合を得点としている。学習者の値がモデルの値以下であった場合は全て 100 点となるようにした。次式は学習者のブレの値を  $B_L$ 、モデルのブレの値を  $B_M$  とした際の、ブレの少なさの得点  $P_B$  を求める式である。例えば、モデルのジャイロセンサの値の最大値が 1.5[rad/s<sup>2</sup>] で、学習者が 2.0[rad/s<sup>2</sup>] だった場合、ブレの少なさの得点  $P_B$  は 75 点となる。

$$(B_L > B_M \text{ のとき}) \quad P_B = \frac{B_M}{B_L} \times 100$$

$$(B_L \leq B_M \text{ のとき}) \quad P_B = 100$$

手元と腰の動き出しのタイミングは3段階評価である。最も理想的なものが「腰から全身で引いている」動作である。その次に「腰も動いているが手元から引いている」動作であり、最も理想的でないものが「手元だけで引いている」動作である。それぞれの動作を表 2-7 にまとめた。本来は3段階の評価だが、便宜上、理想的なものから順にそれぞれ 100 点、70 点、30 点、とした。

表 2-7 「手元と腰の動き出し」の観点における動作

動作	動き出しの順序	得点
腰から全身で引いている	腰 → 手元	100
腰も動いているが手元から引いている	手元 → 腰	70
手元だけで引いている	手元のみ	30

総合得点は、引きの速さ、鉋のブレの少なさ、手元と腰の動き出しのタイミングの3観点のそれぞれの得点から算出している。総合得点  $P$  を求める式を以下に示す。鉋を引く際に、鉋身が被削材に食い込んだときの切削抵抗に負けてしまうとそもそも削ることができないことや、田中ら(1990, 1991)[2-4][2-5], 橘田ら(2003, 2004)[2-6][2-7], 安孫子ら(2005)[2-8]のように、鉋を引く速度について言及している先行研究が多数みられることから、引きの速さの得点に重みを付けた。引きの速さの得点  $P_V$  を4割、ブレの少なさの得点  $P_B$  と手元と腰の動き出しのタイミングの得点  $P_W$  をそれぞれ3割、を重み付けし、総合得点  $P$  とした。

$$P = (0.4 \times P_V) + (0.3 \times P_B) + (0.3 \times P_W)$$

## 第2項 専用鉋の製作

鉋ラーニングを用いて鉋掛けを行う際、スマートフォンを装着することができる専用の鉋を用いる。本項では専用鉋の詳細について述べる。

鉋ラーニング専用の鉋を図 2-7 に示す。専用鉋の鉋台の寸法は  $275 \times 80 \times 36\text{mm}$  であり、加工前の寸法と同じである。重量は  $930\text{g}$  で、スマートフォンを装着した状態では、 $1060\text{g}$  になる(表 2-8)。加工前の重量  $1104\text{g}$  と比較すると  $44\text{g}$  軽いですが、わずかな差であるため鉋の使用に際しての影響は少ないと判断した。



図 2-7 鉋ラーニング専用鉋

表 2-8 鉋の重量

鉋	重量[g]
加工前	1104
加工後(スマートフォン未装着時)	930
加工後(スマートフォン装着時)	1060

砲台の加工後の図面を図 2-8, 図 2-9 に示す。図 2-9 には寸法を記し, 加工に関係のない砲身を差し込む溝については, 細い線で記して隠れ線は省略した。加工の大部分はフライス盤を用いた。その後, 鑿を用いて細部を仕上げた。

図 2-10 および図 2-11 はスマホを固定するアタッチメント(A, B)である。これは 3D プリントを用いて製作した, ばね(線径 0.65mm×外径 6.3mm×長さ 52mm)の両端をアタッチメント A, B それぞれの半円部分に引っ掛けて組み立てる。砲台とアタッチメントは  $\phi 3 \times 10\text{mm}$  のタッピングネジ(鍋)で接合した。図 2-12 はカバーの図面である。カバーは厚さ 2mm のアクリル板をレーザーカッターで加工した。図 2-13 はカバーを砲台に取り付ける蝶番の図面である。この蝶番もアタッチメントと同様に 3D プリントを用いて製作した。蝶番とカバーはアクリル用接着剤で接合し, 砲台との接合には  $\phi 3 \times 5\text{mm}$  のタッピングネジ(皿)を用いた。表 2-9 に専用砲の製作に必要な部品をまとめた。

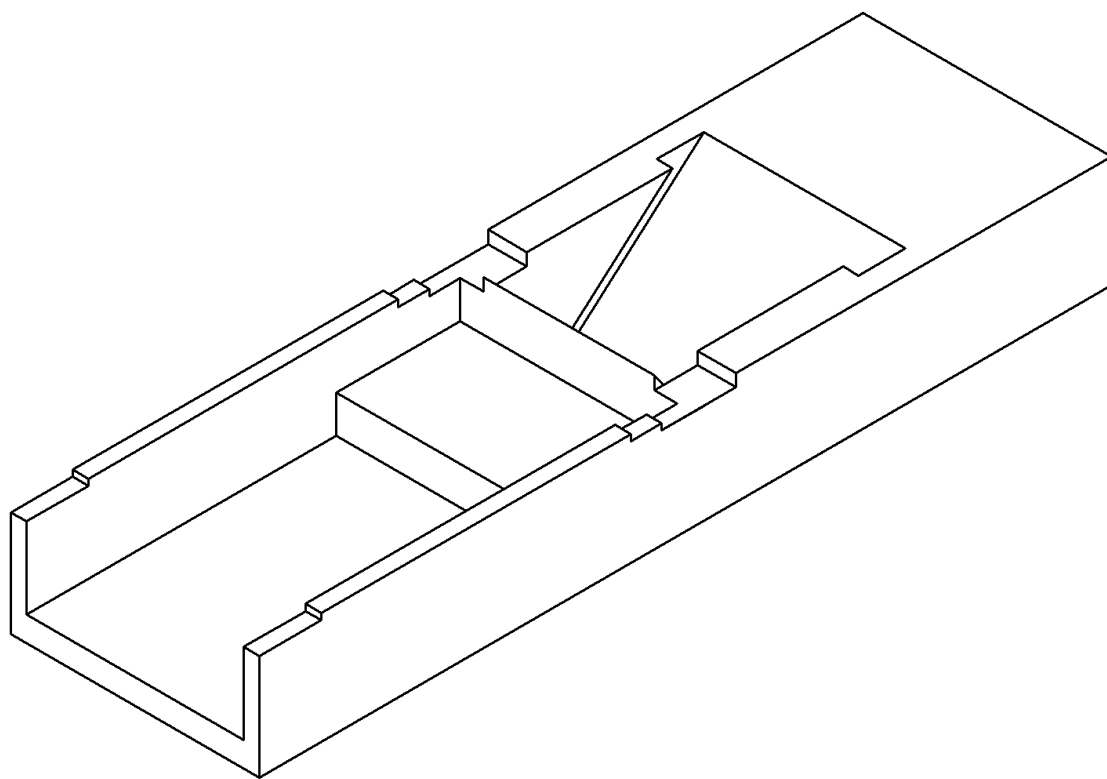


図 2-8 砲台の加工後の図面(等角図)

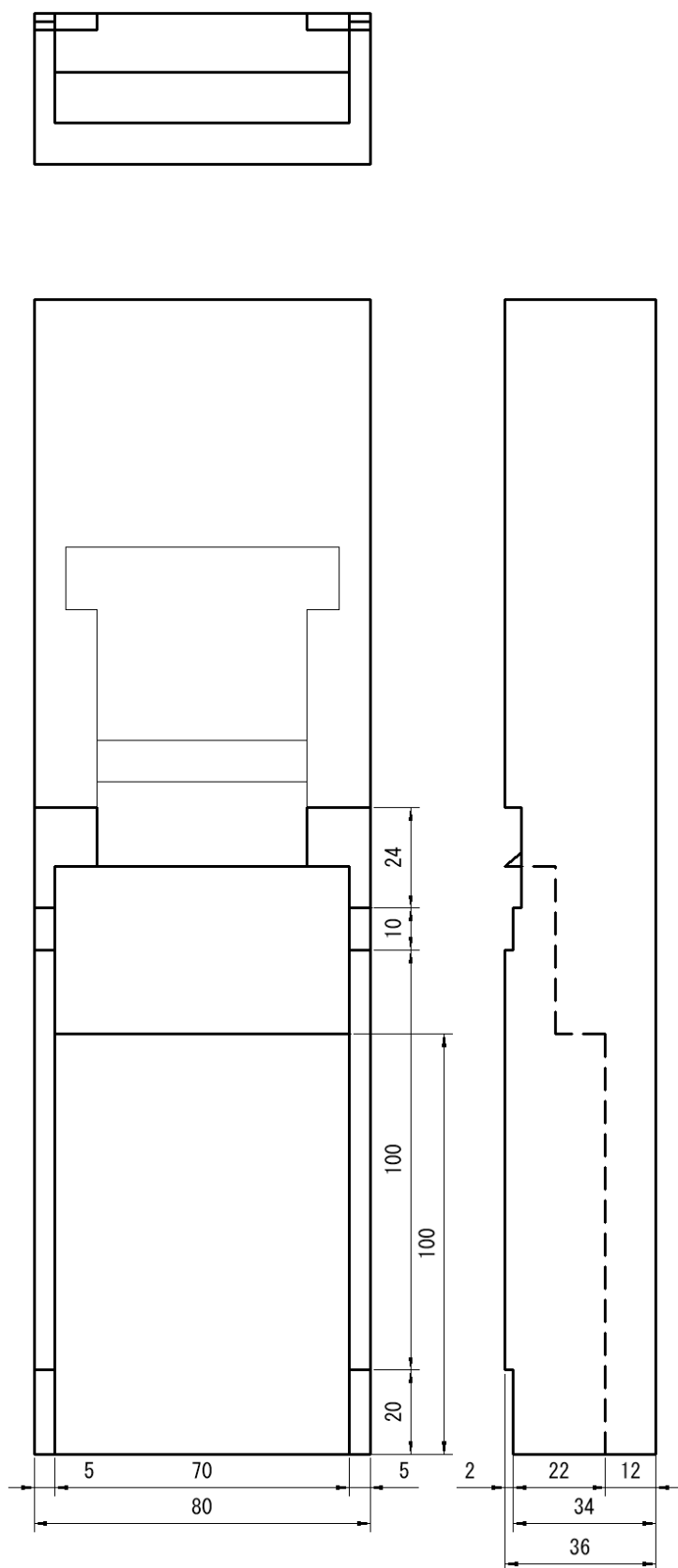


図 2-9 砲台の加工後の図面(三面図)

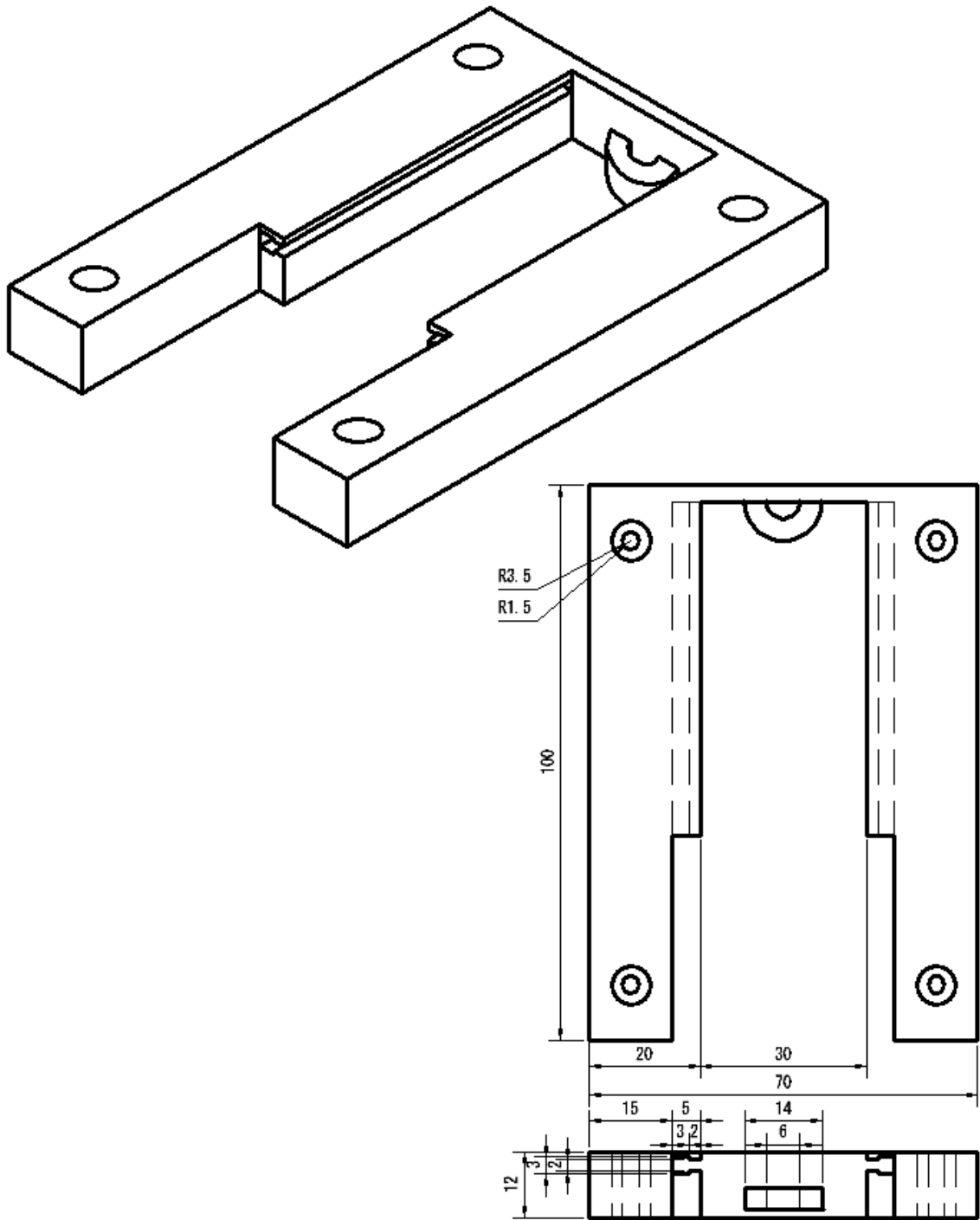


図 2-10 専用鉋のアタッチメント(A)



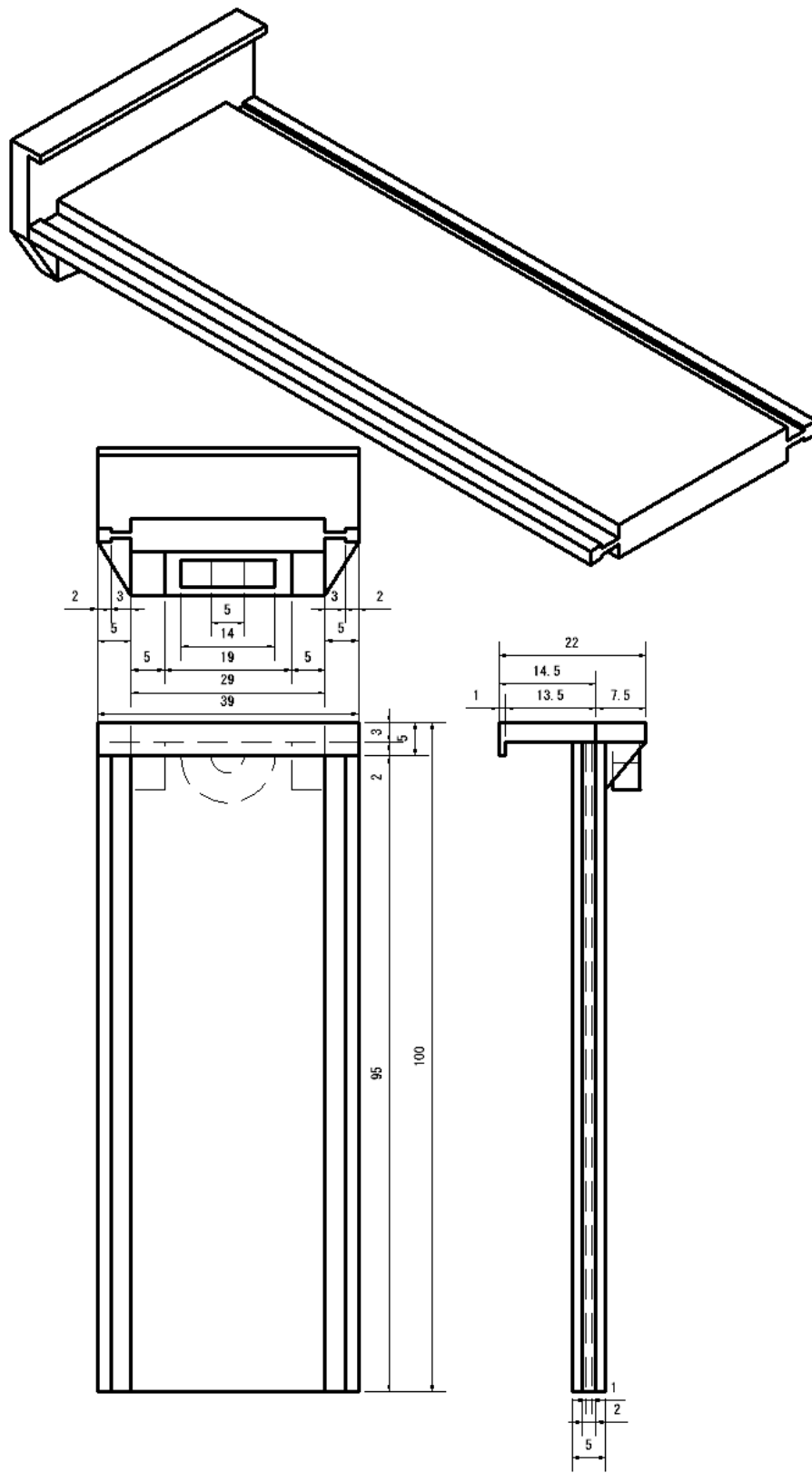


図 2-11 専用鉋のアタッチメント(B)

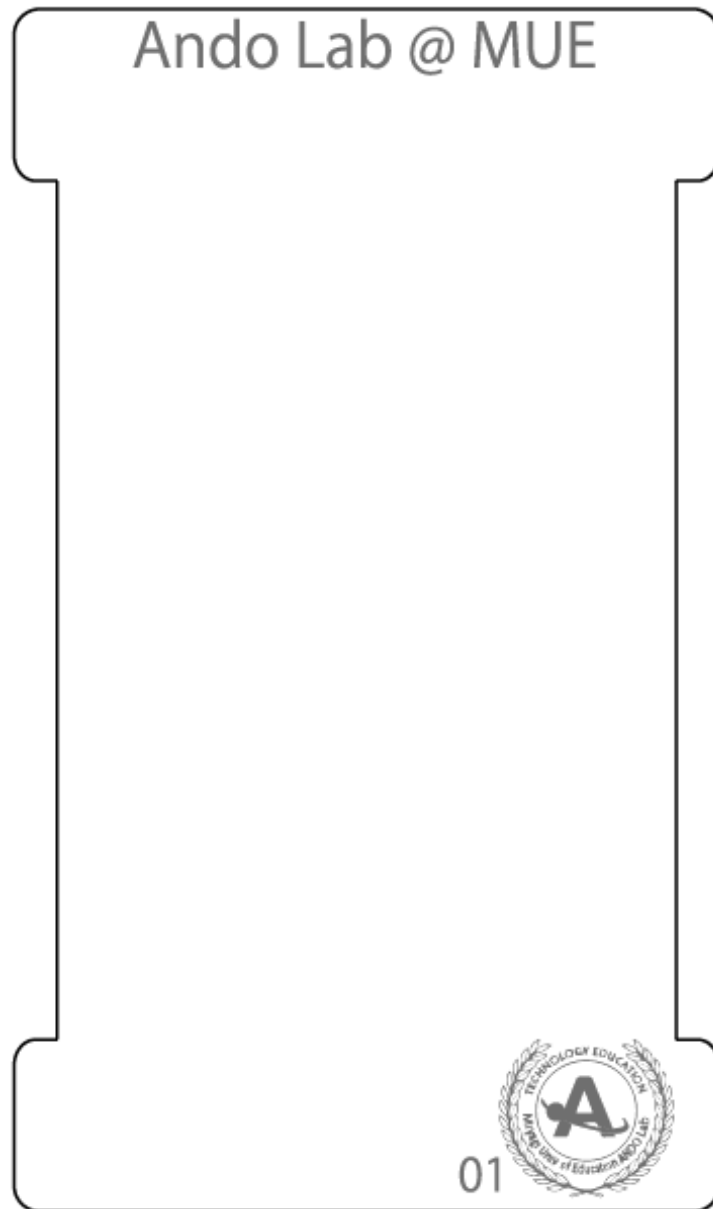


図 2-12 専用鉋のカバー

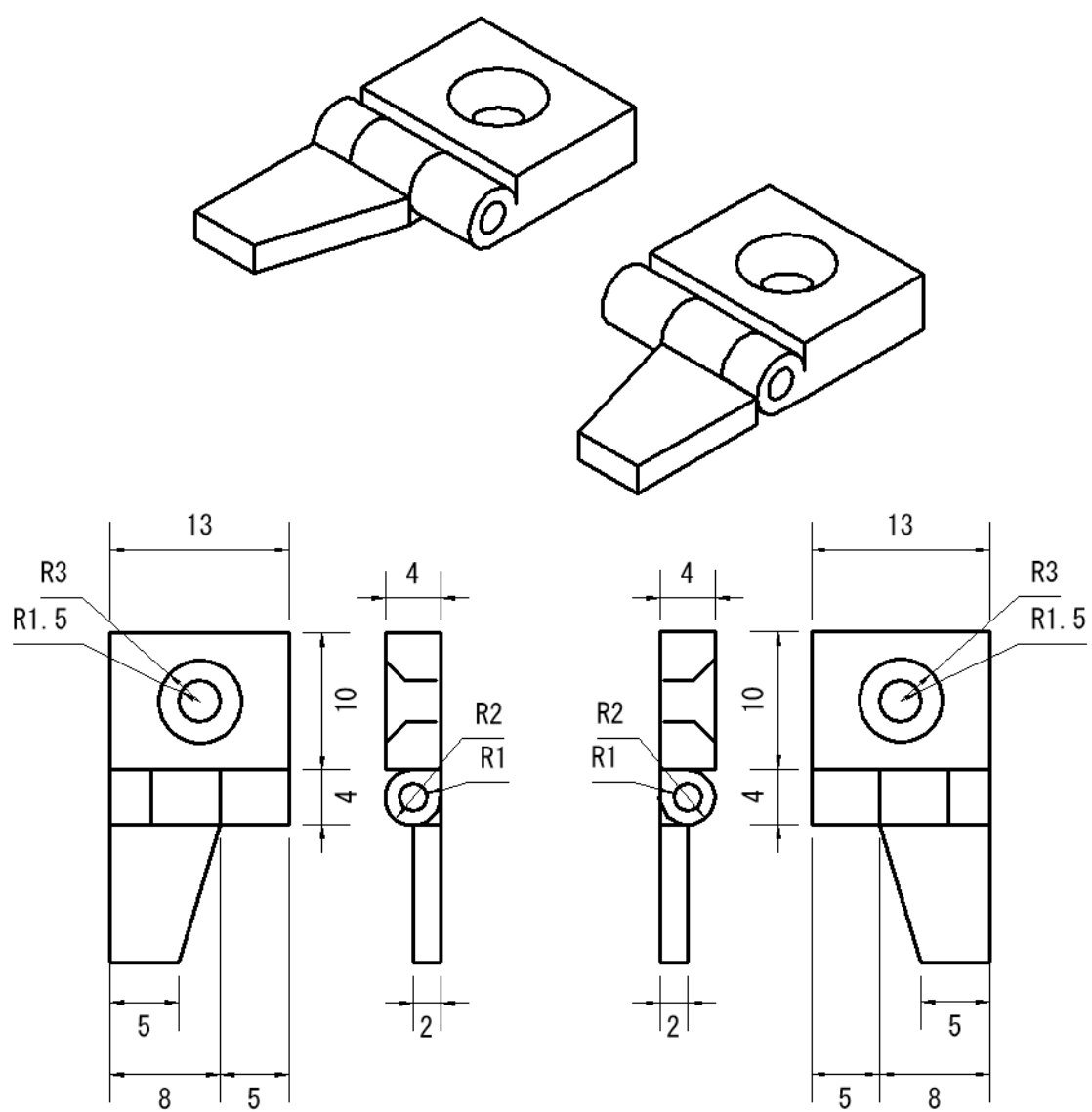


図 2-13 カバーを鉋台に取り付ける蝶番

表 2-9 専用鉋部品表

	数量	寸法	
寸六鉋	1	275×80×36[mm]	-
アタッチメント	A, B各1	図 2-10, 図 2-11 参照	3D プリンタで製作
蝶番	左右各1	図 2-13 参照	3D プリンタで製作
カバー	1	図 2-12 参照	レーザーカッターで製作
ばね	1	0.65×6.3×52[mm]	寸法は線径×外径×長さ
タッピングネジ(鍋)	4	φ3×10[mm]	-
タッピングネジ(皿)	2	φ3×5[mm]	-
接着剤	-	-	-

## 第3項 使用時の流れ

本項では、開発したアプリケーションを使用する際の流れを 1. 学習時の流れと、2. モデル登録時の流れに分けて述べる。

### 1. 学習時の流れ

本アプリケーションを用いて学習を行う場合、用意できるスマートフォンの台数に応じて、使用する台数を「1 台で練習」もしくは「2 台で練習」から選択することができる。ただし、1 台で練習を行う場合は、第 1 項で述べた 3 観点のうち、「手元と腰の動き出しのタイミング」の評価をすることはできない。ここでは、2 台の端末を用いて練習を行う際の流れを述べる。

図 2-14 は学習時の流れを示すフローチャートで、図 2-15 は画面遷移図である。アプリケーション使用前の準備として、システムのサーバと通信するための無線 LAN ルーターへの接続や、端末間の Bluetooth ペアリングを行う必要がある。また、2 台のうち 1 台を専用鉋にあらかじめ装着しておく。

アプリケーションを起動すると、使用するスマートフォンの台数を選択する画面が表示される。ここでは「2 台で練習」を選択する。すると、「学習」、「鍛錬」および「登録」の 2 つのボタンが表示される。「登録」を選択すると、モデルの登録を行うことができる。詳細については本項で後述する。練習を行う際は「鍛錬」を選択する。選択すると、2 台のうち、どちらを鉋端末とし、どちらを腰端末とするのかを選択するダイアログが表示される。鉋と腰のそれぞれを選択すると、両端末が Bluetooth で接続される。接続が完了すると、腰端末には「腰付近のポケットに入れてください」のメッセージが表示される。この後、腰端末はセンシングのみの端末となり、操作の必要はないため、ポケットへ入れておく。一方、鉋端末では Bluetooth 接続が完了すると、学習者を指定する操作へ入る。はじめに、クラス選択ダイアログが表示される。クラスを選択すると、次に出席番号を選択するダイアログが表示される。学習者が自身の出席番号を選択すると、練習前の準備は完了となる。

準備が完了すると、画面に「開始」のボタンが現れる。鉋掛けをできる状態で静止し、「開始」ボタンを押すと、「3・2・1・始め」と音声で鉋を引くタイミングを指示されるので、合図に合わせて平削りを 1 度行う。すると、結果が表示される。結果を確認した後、「次へ」のボタンをタップすると、これまでの点数の履歴が表示される。

ここまで一連の練習のサイクルが終了する。この後、引き続き練習を行う場合は「もう一度」のボタンをタップすると、「開始」のボタンが表示された画面へ戻る。練習を終え、次の学習者と交代する場合は「学習者の変更」ボタンをタップする。すると、出席番号を選択するダイアログが再び表示される。その後の手順は同様である。

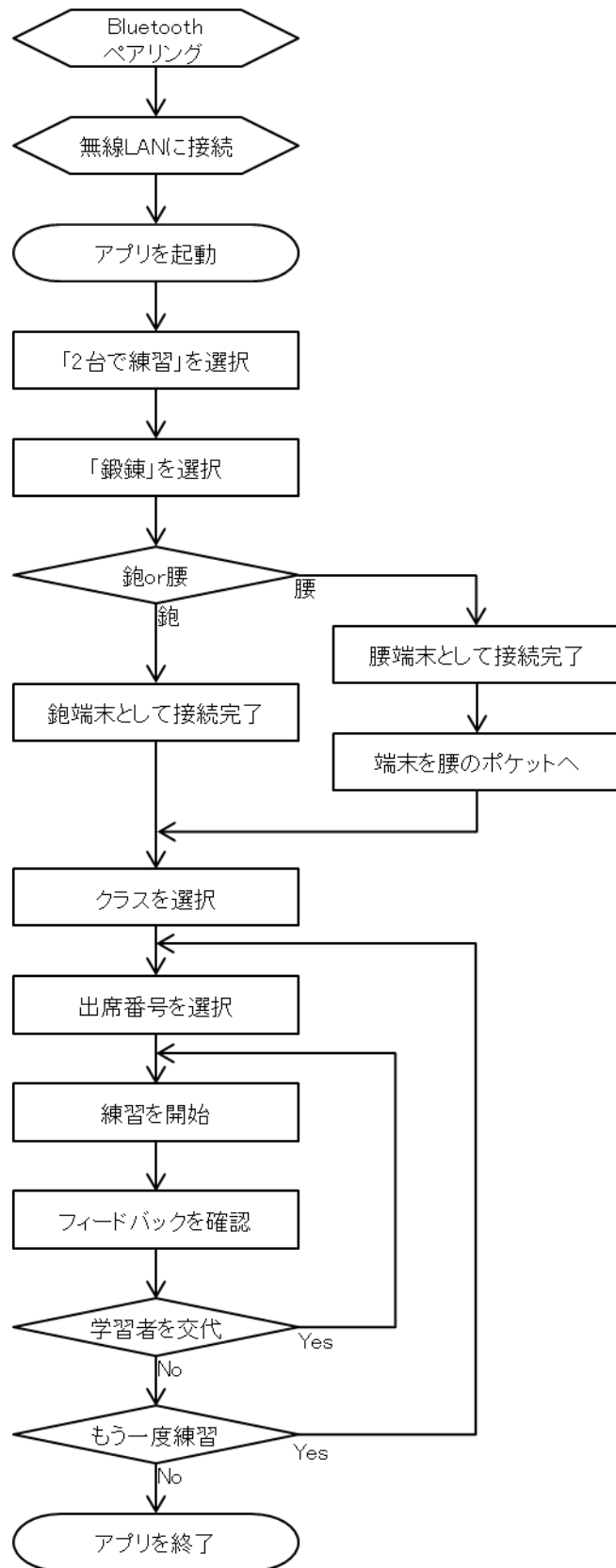


図 2-14 練習時のフローチャート

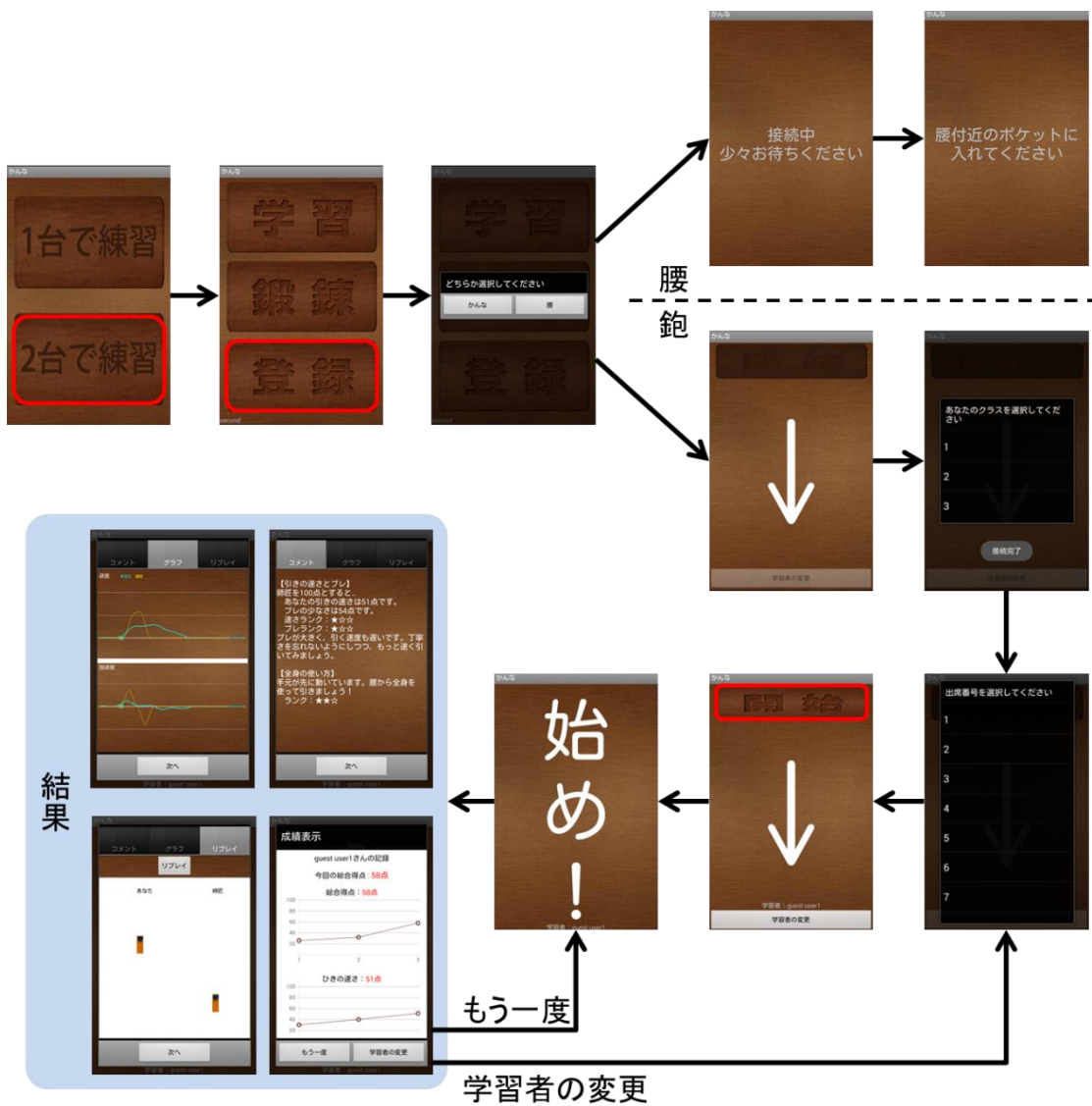


図 2-15 練習時の画面遷移図

## 2. モデル登録時の流れ

モデル登録時のフローチャートを図 2-16 に、操作の流れを図 2-17 に示す。事前の準備や、両端末をそれぞれ鉋端末、腰端末と指定し、Bluetooth 接続を完了させ「開始」ボタンを表示させるところまでの操作は同様である。「開始」ボタンをタップし、鉋掛けを行うと、速度やブレの大きさ、手元と腰のタイミングが表示される。また、練習時と同様にグラフやリプレイアニメーションも確認することができる。データを登録し直したい際は「もう一度」ボタンをタップする。取れたデータで問題がない場合は、「これを登録する」のボタンをタップし次へ進む。すると、モデルの名前を入力するダイアログが表示される。ここでは、登録者の氏名などをモデルの名前として入力する。入力を行い、最後に「OK」ボタンをタップすると、端末や PLMS サーバにモデルのデータが登録される。登録したデータを練習対象として使用する場合の操作については次節で述べる。



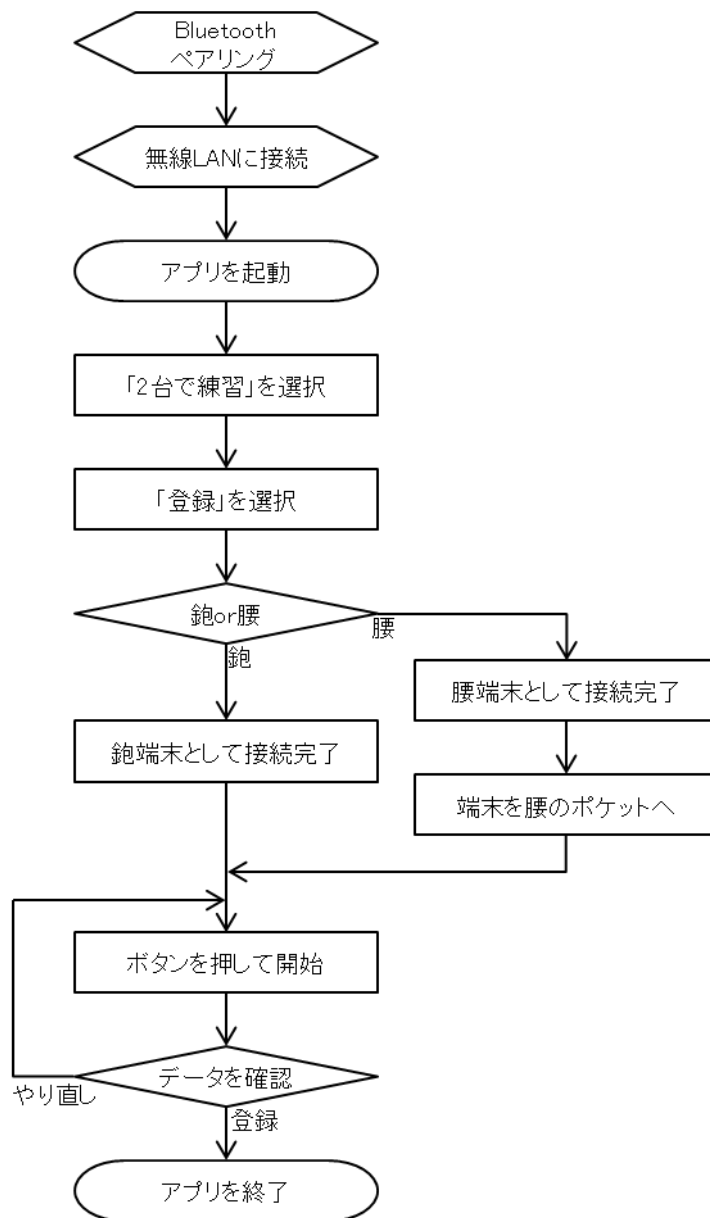


図 2-16 モデル登録時のフローチャート

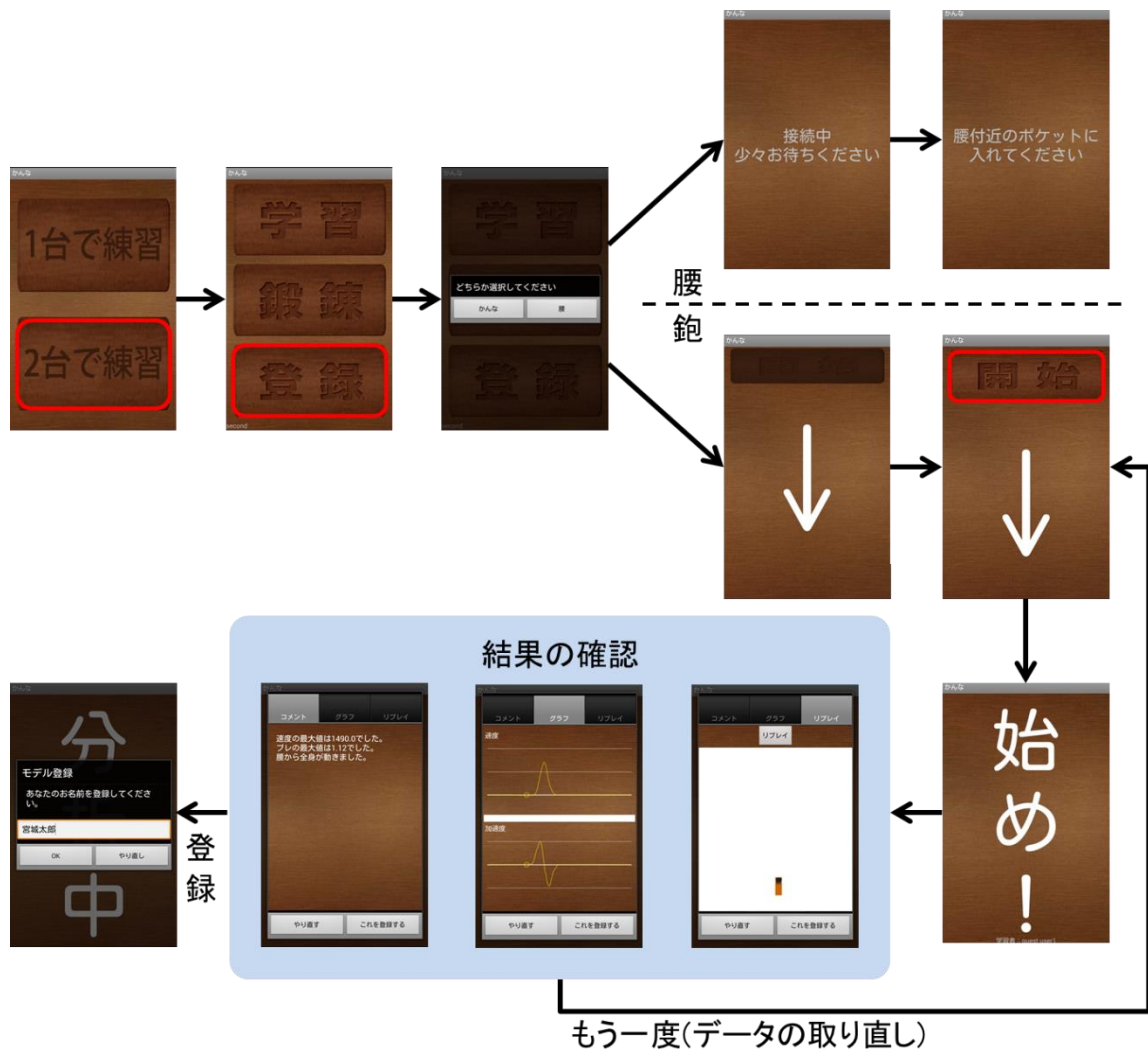


図 2-17 モデル登録時の操作の流れ

## 第4節 技能学習管理システムについて

### 第1項 機能の概要

本節で述べる技能学習管理システム(以降, システムまたは PLMS と表記)は, 技能学習のデータを管理する Web アプリケーションである。図 2-18 は本システムのページの構成である。本システムは主に 1. 練習データの管理, および 2. 生徒情報等の管理や設定の 2 つの機能を持つ。本項ではこれらの 2 つに分け詳細を述べる。

本システムのトップページを図 2-19 に示す。鋸の技能学習管理システムは, 前章で述べたもの[2-9]であり, 玄翁の技能学習管理システムは, 現在本研究室で開発を行っている。

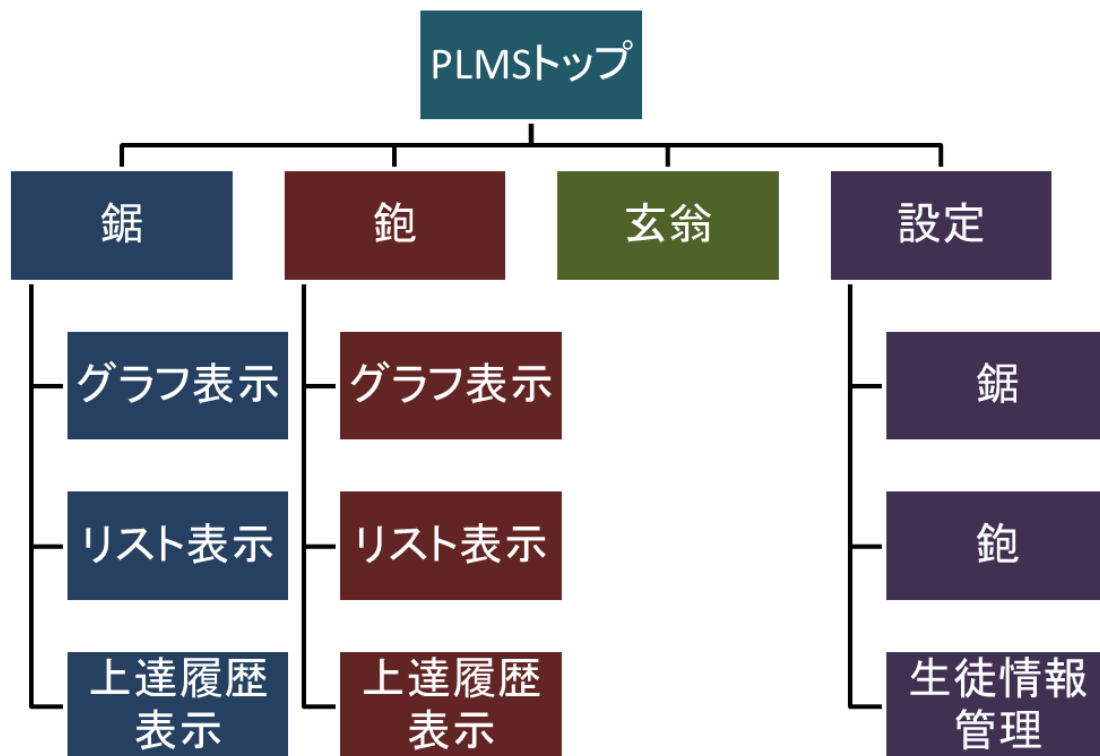


図 2-18 PLMS のページ構成

# PLMS (Practice Learning Management System)



図 2-19 PLMS トップページ

## 1. 練習データの管理

ここでは、本研究で開発した鉋掛けにおける技能学習管理機能について述べる。学習者が鉋ラーニングを用いて練習を行ったデータは PLMS サーバに蓄積されており、指導者はそのデータを 1. グラフ表示, 2. リスト表示, 3. 上達履歴表示の 3 つの表示で確認することができる。

グラフ表示の画面を図 2-20 に示す。グラフ表示では、学習者が練習したデータが速度、加速度、およびレーダーチャートの 3 つのグラフで表示される。速度および加速度のグラフでは、鉋ラーニングでの結果表示と同様に、登録されたモデルの速度、加速度のグラフと重ね合わせて表示される。PLMS では、学習者のグラフが赤の線で、モデルの動きが緑の線で表示される。また、学習者のブレの大きさが黄色で表示される。レーダーチャートは、引きの速さ、ブレの小ささ、および腰の動き出しのタイミングのそれぞれの点数を示す。

なお、データが多数存在する場合に、必要なデータのみを確認できるよう、クラスや期間、学習者名での絞り込みや、日時での並べ替えを行うことができる。また、引きの速さやブレの大きさに応じた並べ替えも行うことができる。

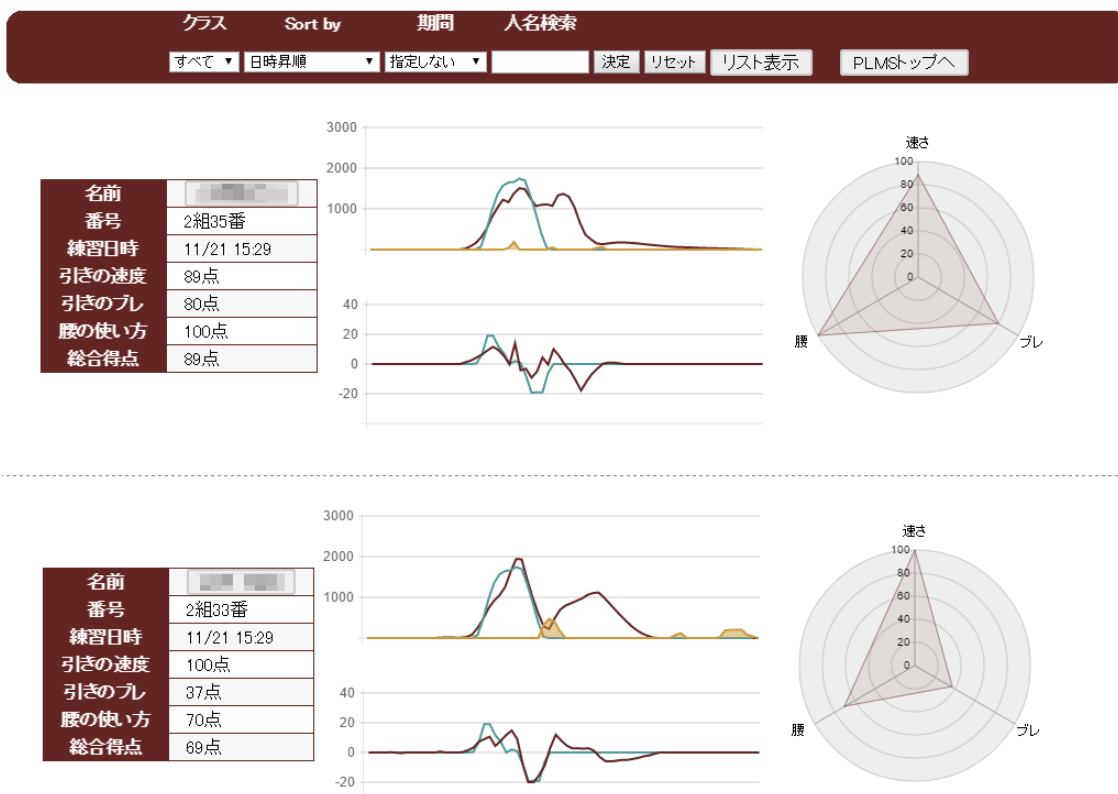


図 2-20 グラフ表示

リスト表示の画面を図 2-21 に示す。リスト表示では、学習者の練習のデータを表形式で一覧表示する。グラフ表示と比較すると詳細なデータを確認することはできないが、多くの学習者の総合得点や引きの速さ、ブレの少なさ、腰の動き出しのタイミング等を一目で確認することができる。また、一定以上の点数の学習者を青で、一定以下の点数の学習者を赤でハイライトすることで、優先的に指導すべき学習者を逸早く確認することができる。出席番号、得点、引きの速さ等の各項目をクリックすることで、それぞれの項目での並べ替えを行うことができる。

クラス	番号	名前	得点	引き速さ	ブレの少なさ	腰の使い方	日時
2	4		61点	★★★	★★	★	11/21 15:30
2	8		95点	★★★	★★★	★★★	11/21 15:30
2	10		80点	★★★	★★★	★★	11/21 15:30
2	1		69点	★★★	★	★★★	11/21 15:30
2	2		100点	★★★	★★★	★★★	11/21 15:30
2	35		85点	★★★	★	★★★	11/21 15:30
2	4		62点	★★★	★★	★	11/21 15:29
2	35		89点	★★★	★★	★★★	11/21 15:29
2	33		69点	★★★	★	★★	11/21 15:29
2	26		72点	★★	★	★★★	11/21 15:29
2	22		31点	★	★★	★	11/21 15:29
2	5		61点	★★★	★	★	11/21 15:29
2	17		83点	★	★★★	★★★	11/21 15:29
2	21		100点	★★★	★★★	★★★	11/21 15:29
2	8		100点	★★★	★★★	★★★	11/21 15:29
2	3		95点	★★★	★★★	★★★	11/21 15:29
2	13		97点	★★★	★★★	★★★	11/21 15:29
2	1		90点	★★★	★★	★★★	11/21 15:29
2	4		69点	★★★	★★★	★	11/21 15:29
2	23		73点	★★	★	★★★	11/21 15:29

図 2-21 リスト表示

上達履歴表示の画面を図 2-22 に示す。グラフ表示もしくはリスト表示の学習者の氏名をクリックすることで、その学習者の上達履歴を確認することができる。上達履歴表示では、その学習者のそれまでの点数の遷移を確認することができる。その学習者がどのような点で技能が向上しており、どのような点で伸び悩んでいるのかを確認することができる。なお、この上達履歴表示は、鉋ラーニングでの練習後の結果表示の際に、図 2-23 のように表示されるため、学習者自身もそれぞれの観点でどのように点数が推移しているのかを確認することができる。

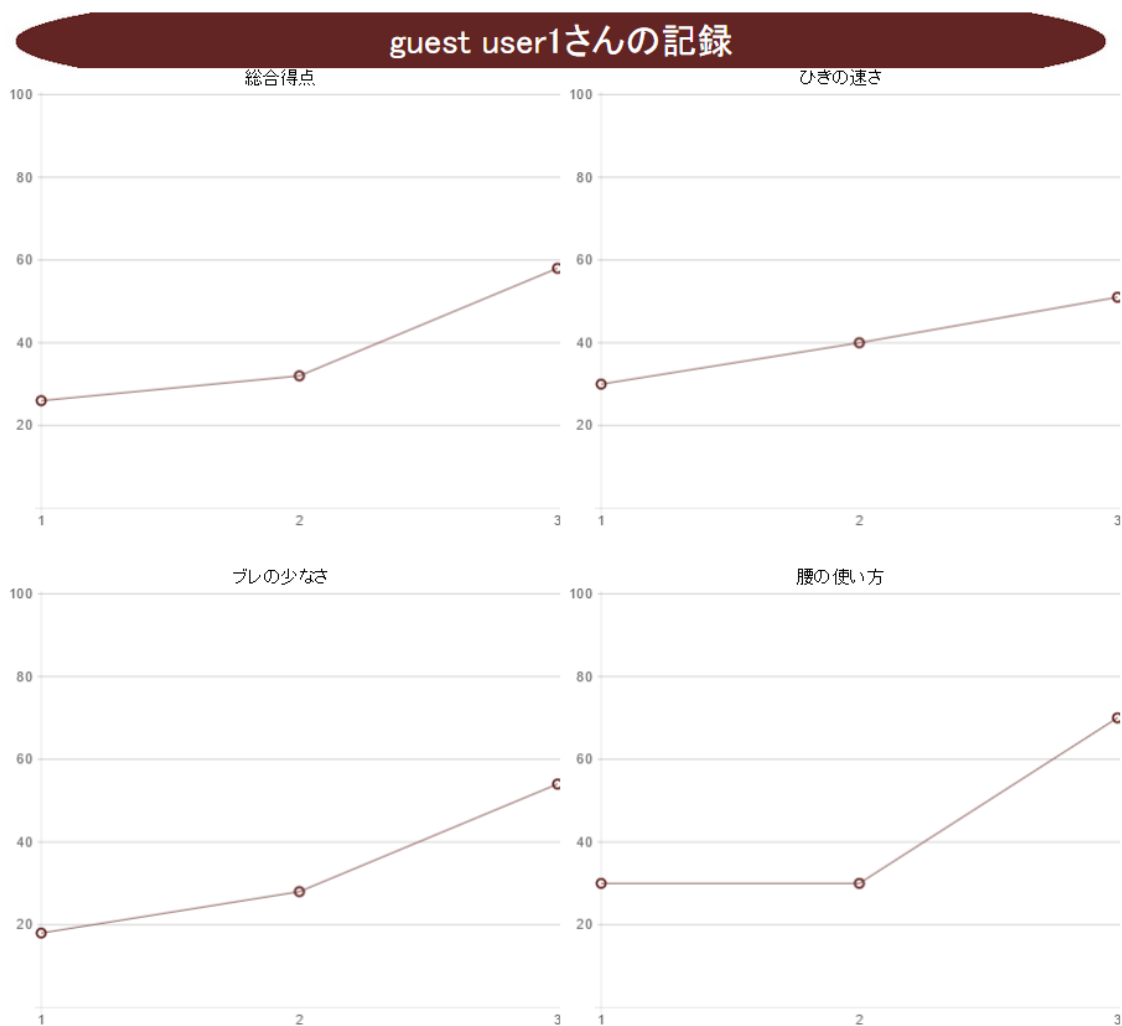


図 2-22 上達履歴表示

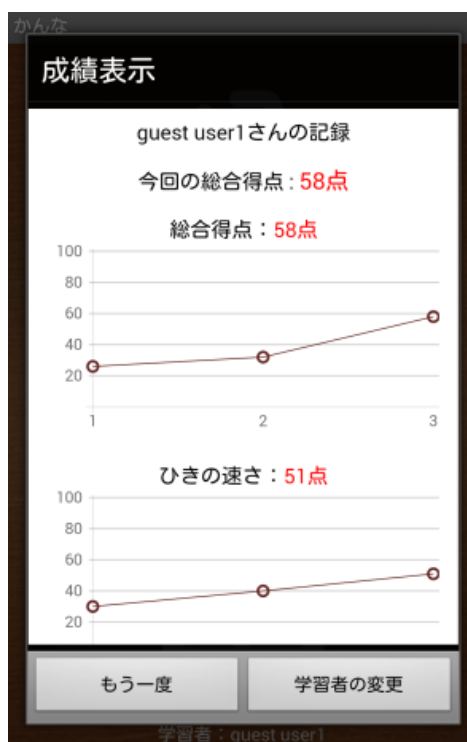


図 2-23 学習者端末に表示される上達履歴表示

## 2. 生徒情報等の管理や設定

PLMS では、システムの設定や、学習者情報の管理を行うことができる。

図 2-24 は鉋掛けに関する設定ページである。ここでは、学習者が練習を行う際のモデルの選択や、星の数でランクを表す際の基準となる点数を設定することができる。ここで設定したデータは、学習者が鉋ラーニングのアプリケーションを起動した際に読み込まれ、アプリケーション側に反映される。図 2-24 にある「モデルを選択する」のボタンをクリックすると、図 2-25 に示すモデル選択ページへ移る。モデル選択ページでは、過去に登録されたモデルの一覧が表示され、学習時に使用したいモデルを選択することができる。



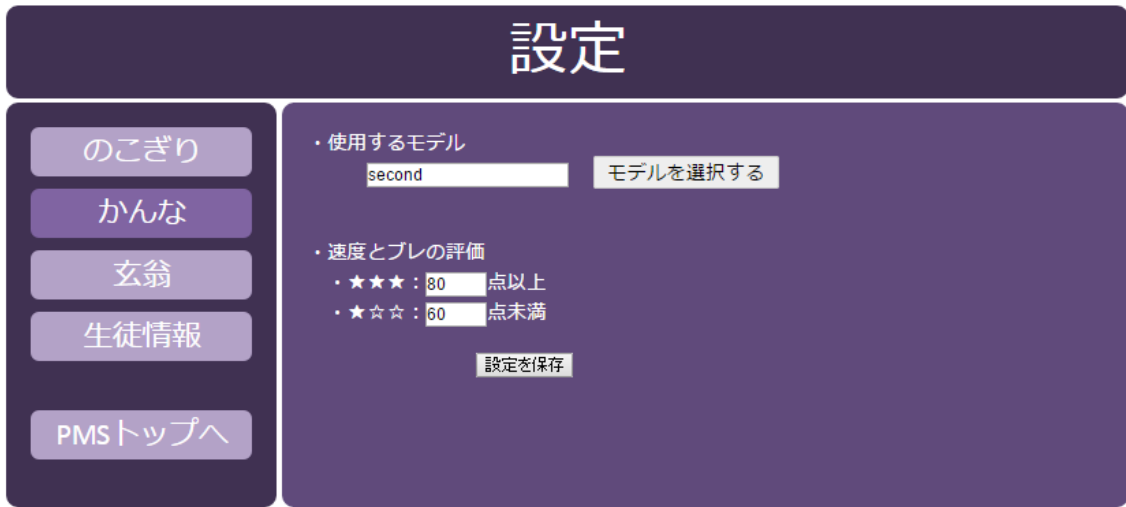


図 2-24 砲掛けに関する設定のページ



図 2-25 モデル選択ページ

図 2-26 は生徒情報の管理ページである。「生徒情報確認」では、現在登録されている学習者の一覧を確認することができる。「生徒情報追加」では、学習者の情報を 1 人ずつ登録することができる(図 2-27)。「生徒情報一括追加」では、CSV ファイルをアップロードすることにより、複数の学習者の情報を一括して登録することができる。「生徒情報更新」からは、同じく CSV ファイルをアップロードすることにより、一括して学習者の情報を更新することができる。

なお、ここで登録している学習者情報は、学習者の学年、クラス、出席番号、氏名、性別である。図 2-26 に示す学習者情報管理ページの下部にある「CSV フォーマット」のボタンから専用の様式をダウンロードすることができ、その様式に沿ったデータを作成、アップロードすることで追加または更新を行うことができる。「追加」を行うと、既存の学習者情報に新しい学習者情報が追加され、「更新」を行うと、既存のデータが削除され、新しいデータが上書きされる。

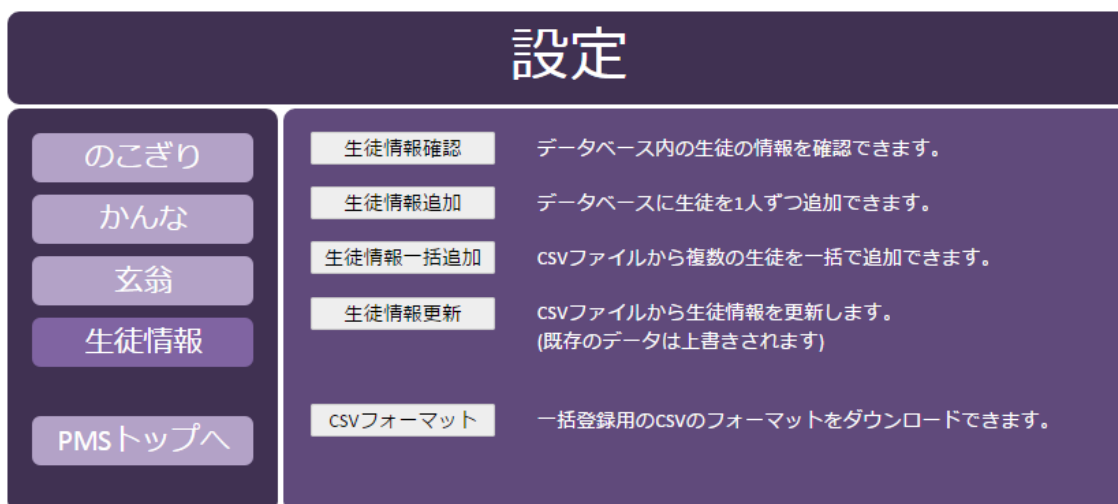


図 2-26 生徒情報管理ページ



図 2-27 生徒情報追加ページ

## 第2項 データベース ER 図

PLMS のデータベース ER 図を図 2-28 に示す。

学習者名テーブルには学習者の氏名が格納されている。学習者情報テーブルには、学習者それぞれの学年、クラス等のデータが格納されており、練習データテーブルには、飽ラーニングを用いて練習を行った際の速度等のデータや各観点の得点が格納されている。これら 3 つのテーブルは学習者 ID で紐づけられている。学習者 ID は、前項で述べた生徒情報の追加や更新の操作が行われた際に、学年、クラスおよび出席番号から自動生成される一意なものである。

設定データテーブルには、使用するモデルの ID や星の数で示されるランク付けのボーダーとなる得点が格納されている。このテーブルのデータは飽ラーニングの起動時にスマートフォンへ読み込まれる。その際、モデル ID で紐づけられたモデルの詳細データがモデルテーブルから読み込まれる。

なお、各テーブル作成の SQL 文は下記の通りである。

### 【学習者名テーブル】

```
CREATE TABLE nameTb (ID INTEGER, name TEXT);
```

### 【学習者情報テーブル】

```
CREATE TABLE userTb (ID INTEGER, year INTEGER, class INTEGER, number INTEGER, sex TEXT);
```

### 【練習データテーブル】

```
CREATE TABLE planeTb (ID INTEGER, uTime INTEGER, speed TEXT, acc TEXT, gyro TEXT, koshi TEXT, gap INTEGER, accPoint INTEGER, gyroPoint INTEGER, koshiPoint INTEGER, totalPoint INTEGER);
```

### 【モデルテーブル】

```
CREATE TABLE planeModelTb (modelID INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT, speed TEXT, acc TEXT, gyro TEXT, koshi TEXT, gap INTEGER, koshiTime INTEGER, time);
```

### 【設定データテーブル】

```
CREATE TABLE planeSetTb (modelID INTEGER, border1 INTEGER, border2 INTEGER);
```

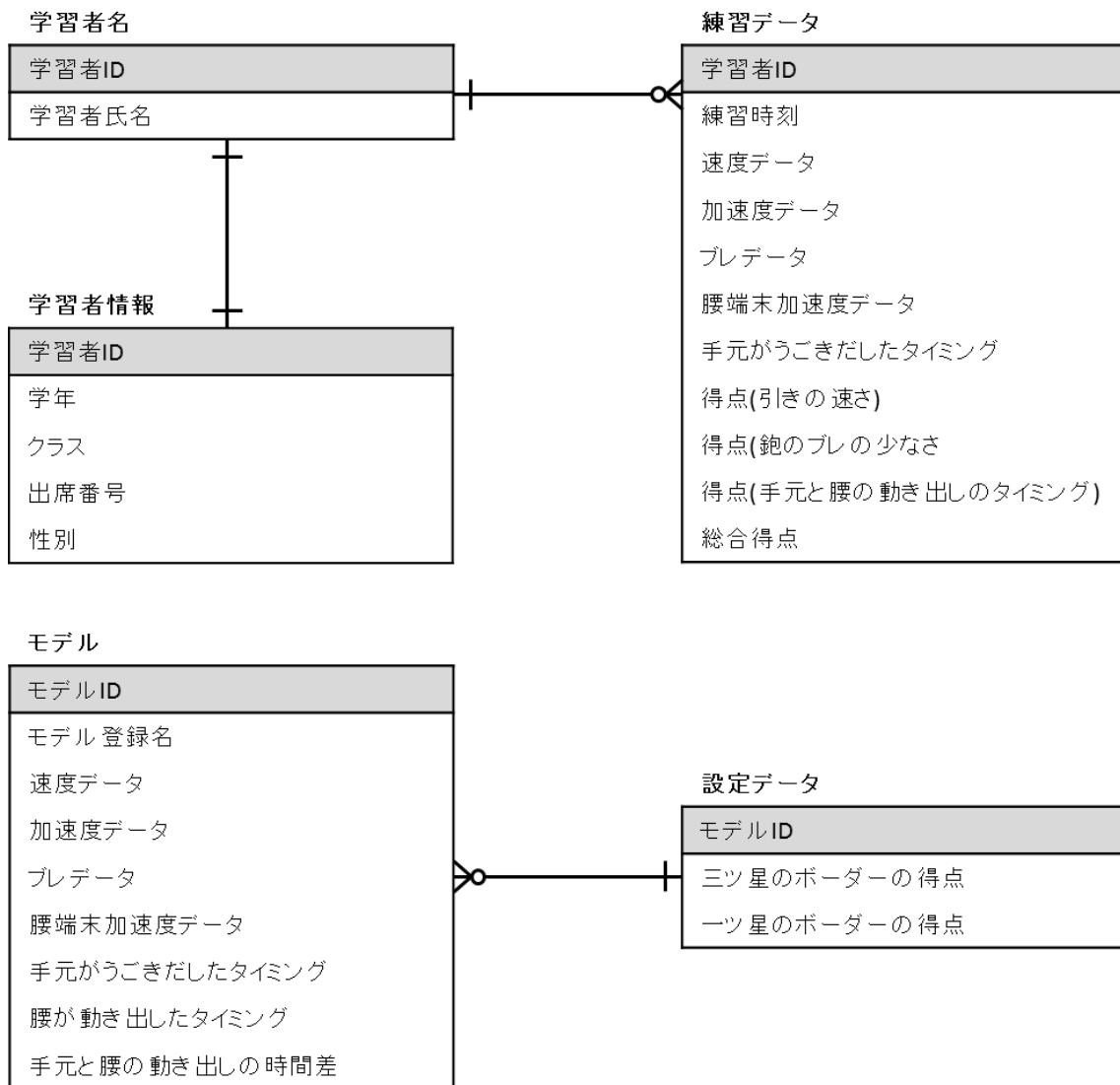


図 2-28 PLMS のデータベース ER 図

## 参考文献

- [2-1] Nexus 5 – Google Play の端末: [https://play.google.com/store/devices/details/Nexus\\_5\\_16\\_GB\\_%E3%83%96%E3%83%A9%E3%83%83%E3%82%AF?id=nexus\\_5\\_black\\_16gb&hl=ja](https://play.google.com/store/devices/details/Nexus_5_16_GB_%E3%83%96%E3%83%A9%E3%83%83%E3%82%AF?id=nexus_5_black_16gb&hl=ja) (2014.12.31 取得)
- [2-2] Galaxy Nexus SAMSUNG 日本: <http://www.samsung.com/jp/consumer/mobilephone/smartphone/docomo/SGH-N044TSNDCM> (2014.12.31 取得)
- [2-3] Surface Pro 2: <http://www.microsoft.com/surface/ja-jp/products/surface-pro-2> (2014.12.31 取得)
- [2-4] 田中通義, 安孫子啓: かんな削り動作における身体重心の移動分析について, 日本産業技術教育学会誌, 第 32 巻, 第 4 号, pp.249-255 (1990).
- [2-5] 田中通義, 篠田功, 山下晃功, 安孫子啓: かんな削り訓練装置の開発と訓練効果, 日本産業技術教育学会誌, 第 33 巻, 第 4 号, pp.251-256 (1991).
- [2-6] 橘田紘洋, 近藤文彦, 白井博成: 中学生における効果的なかんな切削動作に関する研究 かんな台の握り位置が切削動作に及ぼす影響について, 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, 第 52 巻, pp.35-39 (2003).
- [2-7] 橘田紘洋, 近藤文彦, 白井博成: 中学生における効果的なかんな切削動作に関する研究, 愛知教育大学教育実践総合センター紀要, 第 7 巻, pp.297-303 (2004).
- [2-8] 安孫子啓, 安藤明伸, 池田晃一, 寺岡武: 平かんな削りの指導に関する基礎的研究 未熟練者の切削動作における三次元分析, 宮城教育大学紀要, 第 40 巻, pp.193-199 (2005).
- [2-9] 安藤明伸, 高久敏宏, 板垣翔大, 鳥居隆司, 竹野英俊: スマートフォンによるセンサーアシストトレーニング法の開発, モバイル学会モバイル'13 研究論文集, pp.71-72 (2012).

# 第3章 開発したアプリケーション の家庭学習での利用と評価

---

本章および次章では、鉋ラーニングの技能向上における有用性を検証する実験やその結果について述べる。本章では、鉋ラーニングを用いて、学習者の自宅で練習を行った際の効果について検証した。

## 第1節 調査目的と方法

本研究で開発したアプリケーション 鉋ラーニングを自宅で簡易的な疑似シミュレーション練習に用いることの効果を調査するための実験を行った。技能に対する指導者が不在である自宅で、簡易的であっても鉋ラーニングを用いて練習することの意義や、中学校等で実際に製作を行う前の段階として、工具使用動作の練習を事前に自宅で行うことの最低限の可能性を検証することも目的としている。

本調査では、高精度の技能習得に対する教育的な成果を主眼とするのではなく、最低限の学習者負担での環境下でどの程度効果が出るのかを把握する。実際に自宅に鉋や削り台、被削材等の環境を整えることにおける被験者の負担を考慮し、1.5 リットルのペットボトルにスマートフォンを取り付けて鉋掛けのシミュレーションで体の動かし方の感覚を身に付けるよう練習を行った。容易に入手できる点、握ったときの幅が鉋に近い点、比較的固く、強い力で握っても潰れない点などを踏まえ、本調査では 1.5 リットルのペットボトルを用いることとした。ペットボトルには図 3-1 のように、両面テープとゴムマットを用いてスマートフォンを取り付けた(図 3-2)。表 3-1 は鉋とペットボトルの重量の比較である。通常の高鉋や鉋ラーニング専用鉋を比較すると 400g ほど重い。また、ペットボトルとの寸法の比較を表 3-2 に示す。高さは 76mm と実際の鉋よりも高いが、握った感覚に最も影響すると予想される幅は 88mm であり、実際の鉋との差は 8mm である。



図 3-1 ペットボトルに両面テープで貼り付けたゴムマット



図 3-2 ペットボトルへのスマートフォンの装着

表 3-1 鉋とペットボトルの重量の比較

	重量[g]
通常の寸六鉋	1104
専用鉋(スマートフォン装着時)	1060
ペットボトル	1556

表 3-2 鉋とペットボトルの寸法の比較

鉋[mm]		ペットボトル[mm]
275	縦	305
80	横	88
36	高さ	76

調査の詳細を表 3-3 に示す。鉋掛け経験がほぼ皆無の本学学生 10 名(男 5 名, 女 5 名)を対象とした。被験者を, 鉋ラーニングからのアドバイスを基に練習を行う実験群と, アドバイスを見ずに練習を行う統制群の 2 群に分け, 結果を比較した。

表 3-3 本調査の詳細

	10 名 (男 5 名, 女 5 名)
被験者	実験群:5 名 (男 2 名, 女 3 名), フィードバックあり 統制群:5 名 (男 2 名, 女 3 名), フィードバックなし
	事前調査:5 回
試技回数	自宅練習:1 日 5 回×4 日間 事後調査:5 回

本調査は, 1. 事前調査, 2. 自宅での練習, 3. 事後調査の 3 つの段階に分けられる。

1. 事前調査では, 被験者の鉋掛け練習前の状態を調査した。専用鉋を用いて, 動きをセンシングしながら 5 回の鉋掛けを行った。このとき, 画面には結果やアドバイスが表示されないアプリケーションを使い, 被験者へフィードバックは与えないようにした。

2. 自宅での練習では, 被験者が自宅で練習を行った。練習回数は 1 日 5 回, 練習期間は 4 日間とした。このとき, 実験群は結果やアドバイスが表示されるアプリケーションを使用し, 統制群は表示されないアプリケーションを使用した。実験群は表示された結果やアドバイスを参考に技能を向上させるように練習をした。統制群は教科書に記載されている情報を基に, 技能が向上するよう各自で考え, 練習をした。

3. 事後調査では, 実験群, 統制群ともに自宅での練習後の状態を調査した。このとき使用したアプリケーションは, 事前調査時と同様に, 結果やアドバイスが表示されないものである。

事前調査と事後調査では, 引きの速さ, 鉋のブレの少なさ, 手元と腰の動き出しのタイミングの 3 観点における得点および総合得点の 4 つを記録した。本調査では, 事前調査および事後評価で得られたそれぞれの得点の比較と, 事後に行ったヒアリング調査から得られた回答を基に結果を考察した。

なお, 統制群が使用する教科書は, 現在市販されている 3 社の技術科の教科書のうち, 平削りの動作について最も詳細に説明が記されていた開隆堂のもの[3-1]とした。



## 第2節 結果と考察

本調査から得られたデータを基に、練習前後の得点の比較、および群間の得点の比較を行った。データの処理にあたっては、各群の被験者数が5人と少ない点を考慮し、ノンパラメトリック検定を用いることとした。練習前後の得点の比較には、Wilcoxon の符号付順位検定を、群間の得点の比較には、Mann-Whitney の U 検定を用いた。

### 第1項 引きの速さ

練習の前後における引きの速さの得点の差を調べた(表 3-4, 図 3-3)。結果として、実験群は練習後に得点が向上し、統制群は低下したが、どちらも有意なものではなかった。また、群間の得点の差についても、練習前後のいずれの時点においても有意な差は確認できなかった。

表 3-4 引きの速さにおける得点の変化と比較

		Before	After	<i>p</i>	
実験群	平均	46.3	51.6	0.686	n.s.
	SD	10.6	11.7		
統制群	平均	45.9	37.3	0.144	n.s.
	SD	21.3	21.6		

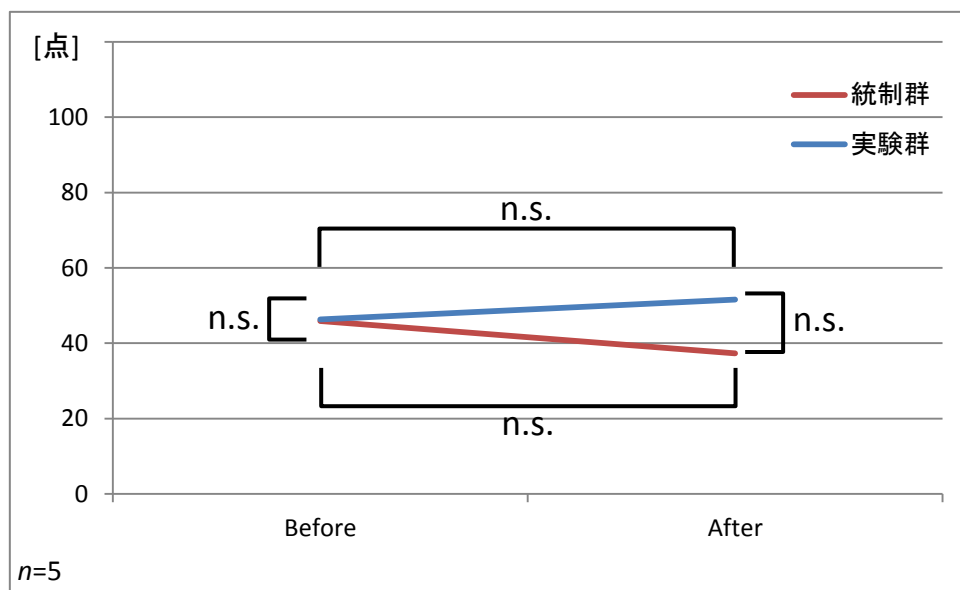


図 3-3 引きの速さにおける得点の変化と比較

## 第2項 鉋のブレの小ささ

練習前後における鉋のブレの小ささの得点の差を調べた(表 3-5, 図 3-4)。実験群においては練習前から練習後への得点の向上が見られるが, どちらの群においても, 練習前後の得点の差は有意なものではなかった。また, 両群の間についても, 練習前, 練習後どちらの時点においても有意な差は見られなかった。

表 3-5 鉋のブレの小ささにおける得点の変化と比較

		Before	After	<i>p</i>	
実験群	平均	65.5	71.1	0.588	n.s.
	SD	20.7	21.7		
統制群	平均	80.7	79.9	1.000	n.s.
	SD	17.1	28.1		

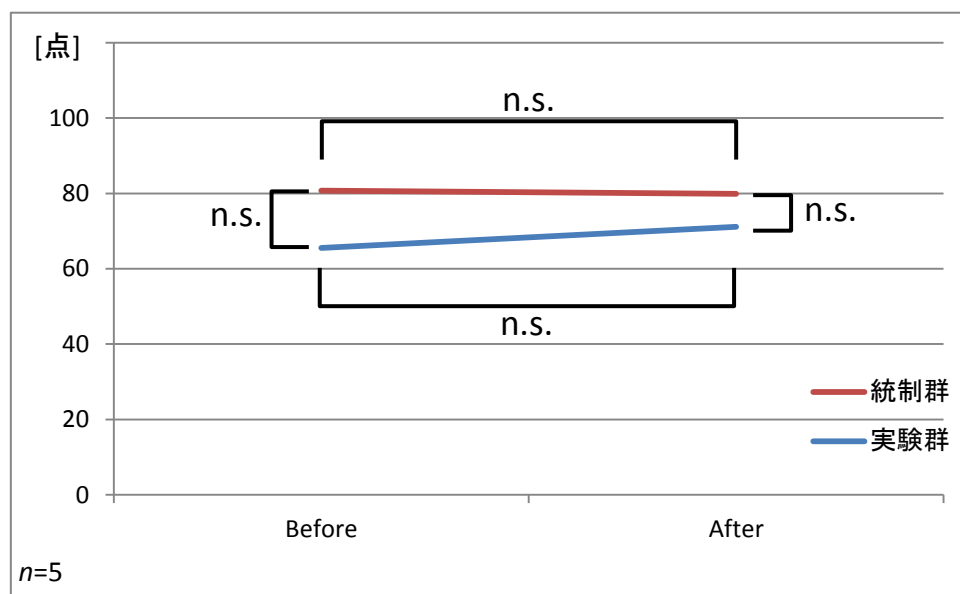


図 3-4 鉋のブレの小ささにおける得点の変化と比較

### 第3項 手元と腰の動き出しのタイミング

練習前後の手元と腰の動き出しのタイミングの得点の差を比較した(表 3-6, 図 3-5)。結果として, 実験群においては練習後に得点が向上したが, 有意なものではなかった。また, 群間の得点の差についても, 練習前後のいずれの時点においても有意な差は確認できなかった。

表 3-6 手元と腰の動き出しのタイミングにおける得点の変化と比較

		Before	After	<i>p</i>	
実験群	平均	85.9	89.2	0.257	n.s.
	SD	9.7	7.8		
統制群	平均	76.5	74.3	1.000	n.s.
	SD	13.9	30.5		

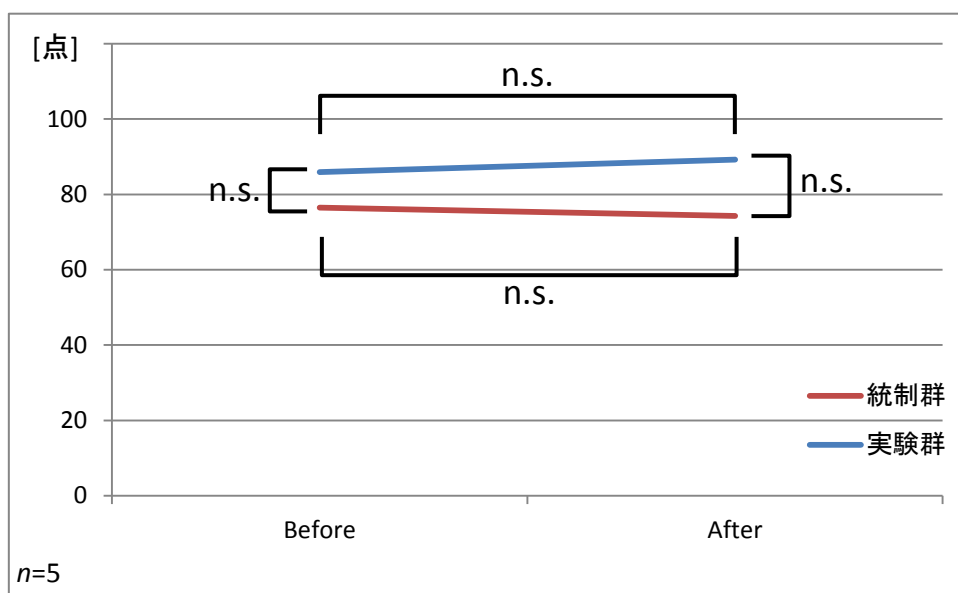


図 3-5 手元と腰の動き出しのタイミングにおける得点の変化と比較

## 第4項 総合得点

練習前後の総合得点の差を比較した(表 3-7, 図 3-6)。実験群では, 練習後に得点が 10% 水準で有意に向上したことが確認できた( $Z=-1.753$ ,  $p=0.080$ )。一方, 統制群では練習後に得点が低下していた( $Z=-1.826$ ,  $p=0.068$ )。両群の間の得点の差については, 練習前後のいずれの時点においても有意な差は確認できなかった。

表 3-7 総合得点の変化と比較

総合		Before	After	$p$	
実験群	平均	63.9	68.9	0.080	*
	SD	3.6	8.5		
統制群	平均	65.5	61.2	0.068	*
	SD	10.9	13.0		

\*:10%水準で有意

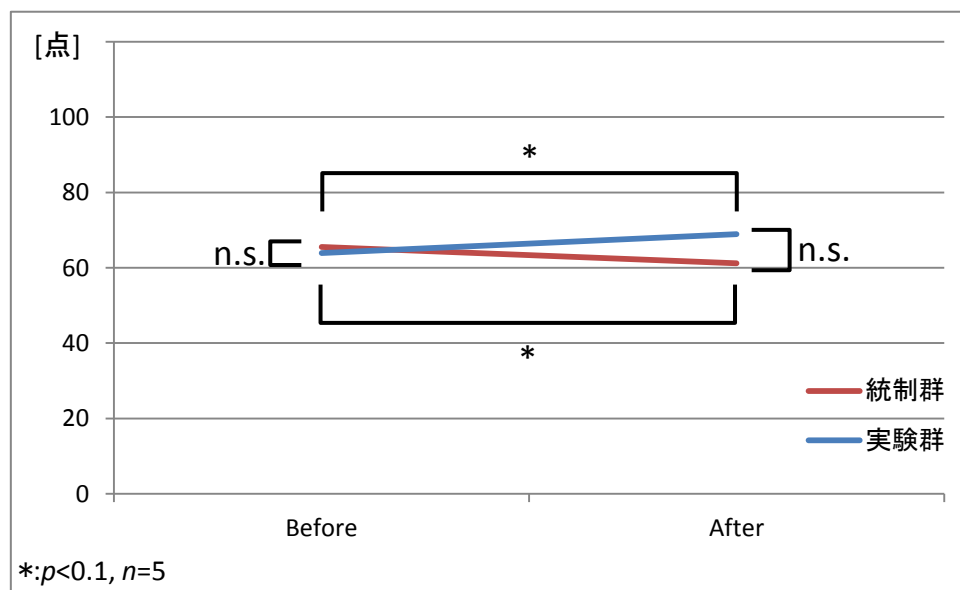


図 3-6 総合得点の変化と比較

## 第5項 考察・まとめ

本調査から、鉋ラーニングとペットボトルを用いて鉋掛けの家庭学習を行うことで、総合得点が有意に向上したことが確認できた。各観点(引きの速さ、鉋のブレの少なさ、手元と腰の動き出しのタイミング)においては、有意な差を確認することができなかった。しかし、いずれの観点においても被験者全体の平均得点は向上していた。その内訳としては、実験群の5名の被験者のうち、引きの速さと手元と腰の動き出しのタイミングにおいては4名の、鉋のブレの少なさにおいては3名の得点が向上していた。

事後に行った被験者へのヒアリング調査からは、被験者自身の実感として、上達を感じたとの意見が実験群から多く挙がった。一方、統制群からは、教科書に書いてある情報を意識しながら練習はしたものの、自分の動きが実際にどのようなになっているのかが分からず、その状態で良いのか、教科書通りの動きになっているのかがわからなかったという意見が多かった。第1章で触れた島(2007)の知見[3-2]と同様に、指導者が不在で、かつ、鉋ラーニングのような動作をセンシングする道具を用いない場合、学習者が自身の動きがどういった状態であるのかをメタ認知することが困難であることが、本調査からも示唆された。

以上より、本調査からは指導者が不在である自宅で、鉋ラーニングとペットボトルを用いる鉋掛けのシミュレーションを行った場合に、鉋掛けの基礎的な技能向上を十分に確認することができなかった。しかし、本調査では被験者が各群5人と少数であった。総合得点が有意に向上していた点や、各観点においても多くの被験者の得点が向上していた点も踏まえ、追加実験を行い、被験者数を増やした上で、再度検証を行いたい。

また、事後のヒアリング調査からは両群に共通して、「ペットボトルでのシミュレーションと実際の鉋掛けではやはり感覚が異なる」といった意見も得られた。シミュレーションによる技能の学習においては、「鉋掛けの感覚を掴む」という点がメインになり、「技能をマスターする」という段階まで達するには、まだ多くの課題が残っている点を認識する必要がある。

### 参考文献

- [3-1] 開隆堂出版株式会社:技術・家庭科[技術分野], p.31 (2012).
- [3-2] 島健:e ラーニングを使用した体育実技の授業方法に関する研究～実技科目へのLMSの導入と問題点～, 上智大学体育, 第40巻, pp.1-13 (2007).

# 第4章 開発したシステムの 授業利用と評価

---

本章でも前章に引き続き、鉋ラーニングの有用性を検証するための実験および結果について述べる。本章では、鉋ラーニングおよび PLMS を中学校技術科での授業で用いた際の効果等についてまとめた。

## 第1節 調査目的と方法

本研究で開発したシステム PLMS を授業で利用した際の技能向上における有用性を評価するための実践利用を行った。実践利用は、仙台市立第二中学校の技術・家庭科(技術分野)の授業で行い、第1学年の3クラス105名のうち、当日の授業に出席し鉋掛けの練習を行った101名を対象とした。実施の詳細は表4-1の通りである。

授業での生徒の練習風景を図4-1および図4-2に示す。練習は2人1組のペアで行った。鉋掛けの練習および結果の確認を5回行うごとに交代し、授業時間内で可能な限り練習を繰り返すよう指示した。授業者はタブレット端末で PLMS を使用し、生徒の練習のデータを確認しながら机間指導を行った(図4-3)。なお、本調査では、本学技官の鉋掛けのデータを熟練者のモデルとして使用した。

生徒は、1組では鉋ラーニングのみを用いて練習を行った。3組では鉋ラーニングに加え、図4-4に示すワークシートを用いて、1回練習を行うごとに点数をワークシートに記入しながら練習を行った。2組では、同じワークシートを用いて点数を記入させ、かつ練習を5回行った時点で、それまでの5回の練習を振り返り、6回目以降の練習で意識したい点を各自で考え記入する時間を設けた。授業の流れを表4-2に示す。

表 4-1 授業実施の詳細

実施場所	仙台市立第二中学校		
実施日	2014年11月		
実施学年	1学年 101名(男61名,女40名)		
実施学級	1組	35名(男21名,女14名)	ワークシート不使用
	2組	34名(男21名,女13名)	ワークシート使用*
	3組	32名(男19名,女13名)	ワークシート使用
活動グループ	2人1組(5回練習するごとに交代)		
使用したモデル	本学技官の鉋掛けのデータ		

※5回練習終了時にそれ以降の練習で意識したい点を各自考えて記入



図 4-1 鉋ラーニングを使用し練習を行っている様子



図 4-2 2人1組で互いの結果を確認し合う生徒



図 4-3 授業者による机間指導時の PLMS 使用の様子



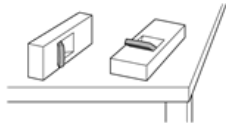
1年技術 「生活の中で役に立つものを作ろう」～材料の切削～②

今日の目標

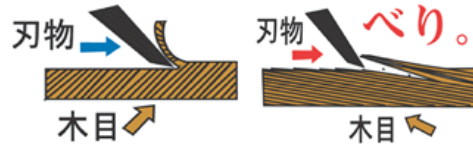
1年 組 番 氏名 \_\_\_\_\_

1. 前回の確認

① かなの置き方は？



② 削る方向は？



2. 自分の「かなけずり」振り返ろう

	1回目	2回目	3回目	4回目	5回目
引きの速度					
引きのプレ					
腰の使い方					
総合得点					

・アドバイスをまとめてみよう

\_\_\_\_\_

\_\_\_\_\_

	6回目	7回目	8回目	9回目	10回目	
引きの速度						
引きのプレ						
腰の使い方						
総合得点						

3. 今日の授業を振り返って ～かなけずり をやってみての感想～

\_\_\_\_\_

\_\_\_\_\_

図 4-4 授業で使ったワークシート

表 4-2 授業の流れ(2組)

学習段階	概要
事前準備	<ul style="list-style-type: none"> <li>● 削り台, 被削材を生徒の机の上に置いておく</li> </ul>
導入(5分)	<ul style="list-style-type: none"> <li>● 前時までの学習内容の確認させる</li> <li>● ビデオ教材を見せる(宮大工による鉋仕上げ)</li> <li>● 授業の目標を明確にする</li> </ul>
展開(40分)	<ul style="list-style-type: none"> <li>● デジタル教科書および付属ビデオ教材で鉋掛けの動作を見せる</li> <li>● 鉋ラーニングの操作説明動画を見せる</li> <li>● 教師演示により操作方法と鉋掛け動作を確認させる</li> <li>● 鉋, スマートフォン等を配る</li> <li>● 鉋ラーニングを用いて5回ずつ練習させる(PLMSを使用して机間指導)</li> <li>● 5回分の得点を振り返らせ, 意識すべき点をプリントにまとめさせる</li> <li>● 鉋ラーニングを用いて再び5回練習させる(PLMSを使用して机間指導)</li> <li>● 鉋, スマートフォン等を回収する</li> </ul>
まとめ(5分)	<ul style="list-style-type: none"> <li>● 授業の感想をワークシートに記入・発表させる</li> <li>● 次時の予告をする</li> </ul>

また, 本調査では, 授業実施前後にそれぞれ生徒にアンケート調査を行った。アンケート調査の目的は, 鉋ラーニングの動機付け効果の検証と, 教材の機能の分析である。

動機付け効果の検証に関するアンケート項目を表 4-3 に示す。授業の前に3項目, 授業後にその3項目を含む4項目について回答を得た。調査は「1(いいえ)~5(はい)」の5件法で行った。項目の選定にあたっては, 山本ら(2013)による調査[4-1]を参考にした。

教材の機能の分析に関するアンケートの項目を表 4-4 に示す。この項目は星野ら(2003)による教材の機能分析の評定尺度[4-2]や, 安藤ら(2013)による教材の機能分析調査[4-3]を参考に作成した。このアンケートに関しては, 授業前には実施せず授業後のみ行った。このアンケートは教材の機能を6つの観点から分析するものである。「ストレス」と「集中度」の項目は全て逆転項目である。前述のアンケート同様に「1(いいえ)~5(はい)」の5件法にて回答を得た。

また, これらの項目に加え, 授業後に授業の感想を自由記述で回答を得た。

表 4-3 動機付けの効果の検証に関するアンケート項目

設問文	実施
1. 木材の表面を削る加工をしたいと思いますか?	授業前後
2. かんなを使って木材の表面をきれいにする自信はありますか?	授業前後
3. かんながけの練習をして身に付けた知識や技能が今後の生活や製作に役立つと思いますか?	授業前後
4. スマートフォンやアプリを使ったことで, 練習へのやる気が高まりましたか?	授業後のみ

表 4-4 教材の機能の分析に関するアンケート項目

観点	観点の概要	設問文
興味・ 関心	授業や学習内容への興味・関心の程度を評価する	1. かんなを使って木材を削ることは楽しいと思いましたが？ 2. もっと長く練習したいと思いましたが？ 3. 意欲的に取り組むことができましたか？
知識・ 理解	学習内容の理解度を評価する	1. 正しいかんなの使い方を理解できましたか？ 2. 正しいからだ全身の使い方を理解できましたか？ 3. よく考えて作業できるようになりましたか？
態度	授業への取り組みの態度を評価する	1. 落ち着いた態度で作業をすることができましたか？ 2. 先生の説明をしっかりと聞けたと思いますか？ 3. 友達と余分な話をしないで授業を受けましたか？
深化	学習内容を深めることができたかを評価する	1. かんながけについて、もっと色々なことを知りたいと思いましたが？ 2. 今日の授業で技能が身に付いたと思いますか？ 3. 今日のかんながけの練習を作品の製作に活かしていけると思いますか？
ストレス	授業や教材から受けるストレスの程度を評価する	1. 操作や使い方がわからなくなることがありましたか？ 2. 練習するときに嫌になることがありましたか？ 3. 練習するときに困ることがありましたか？
集中度	授業に熱心に取り組めたかを評価する	1. 授業の中で何をするかわからなくなることはありませんでしたか？ 2. 授業中に「ボーっ」とすることがありましたか？ 3. 授業にもっと集中できるとよかったですか？

## 第2節 結果と考察

本実験では、引きの速さ、鉋のブレの少なさ、手元と腰の動き出しのタイミング、総合得点の4項目において、練習前と練習後における得点の比較、およびクラス間の得点の比較を行った。練習前後の得点の比較には t 検定を用いた。また、クラス間の得点の比較には一元配置分散分析を用い、その後の検定として Tukey-Kramer の検定を行った。その際、スマートフォンの操作ミスや、腰端末の渡し忘れなどによる明らかな異常値はデータ全体の精度を下げることから、一部のデータを分析対象から除外した。

### 第1項 引きの速さ

全生徒の練習前後における引きの速さの得点について平均の比較を行った(表 4-5, 図 4-5)。結果として練習前の 54.3 点から、練習後の 71.3 点に上昇しており、0.1%水準で有意な差( $t_{101}=-6.575$ ,  $p<0.001$ )が見られた。

表 4-5 引きの速さにおける練習前後の得点の比較

	Before	After	$p$	
平均	54.3	71.3	<0.001	***
SD	28.2	26.4		

\*\*\*:0.1%水準で有意

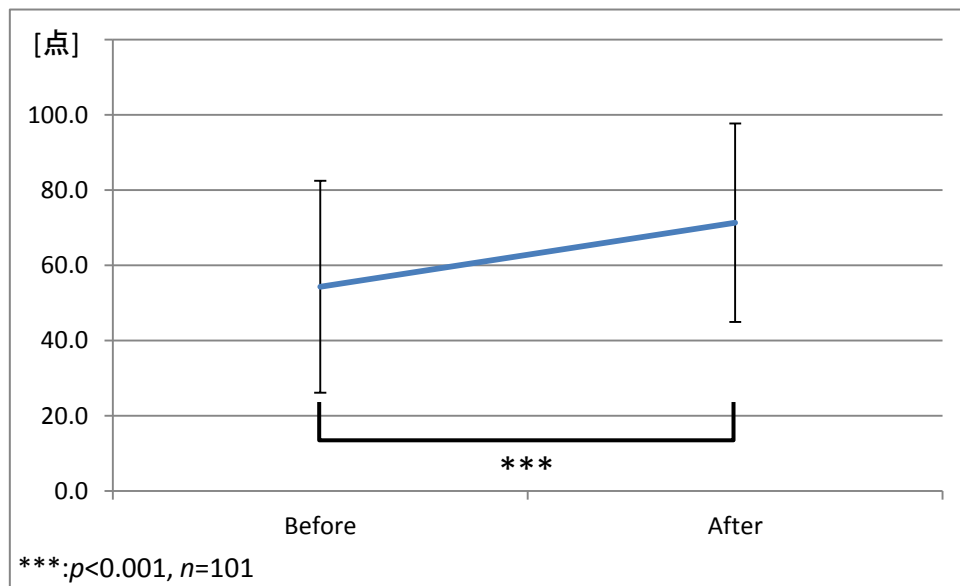


図 4-5 引きの速さにおける練習前後の得点の変化

次に、引きの速さにおいて、クラス間の得点の比較を行った(図 4-6)。練習前はどのクラスの間にも有意差は見られなかったが、練習後は 10%水準で有意差( $F_{(2,98)}=3.30$ ,  $p=0.041$ )があった。Tukey-Kramer を用いた多重比較では、1 組と 2 組の間に有意差があることが示された。

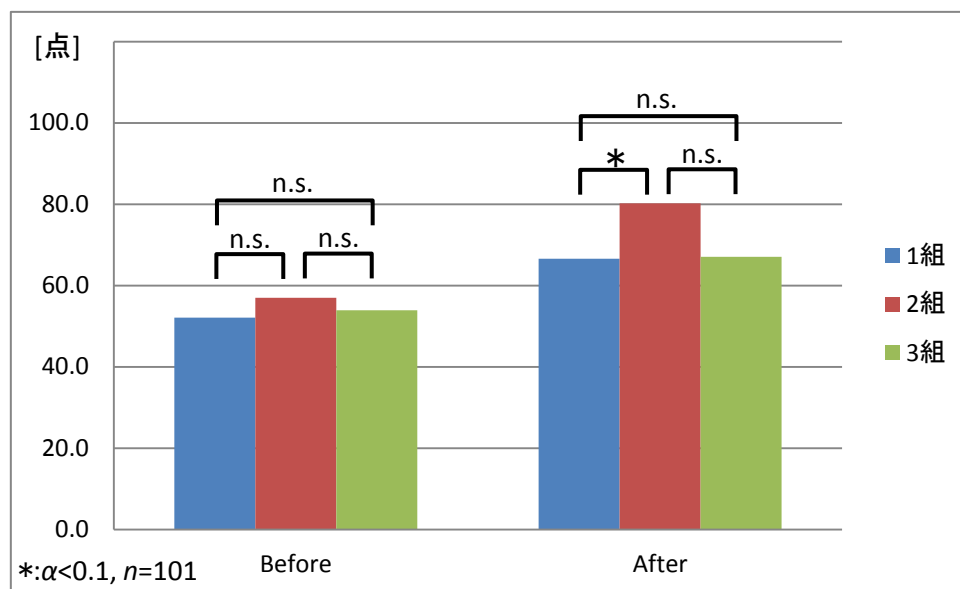


図 4-6 引きの速さにおけるクラス間の比較

## 第2項 砲のブレの少なさ

全生徒の練習前後における砲のブレの少なさの得点について平均の比較を行った(表 4-6, 図 4-7)。結果として練習前の 68.2 点から, 練習後 75.7 点に上昇しており, 5%水準で有意な差( $t_{101}=-2.185$ ,  $p=0.031$ )が見られた。

表 4-6 砲のブレの少なさにおける練習前後の得点の比較

	Before	After	$p$	
平均	68.2	75.7	0.031	*
SD	28.1	26.0		

\*:5%水準で有意

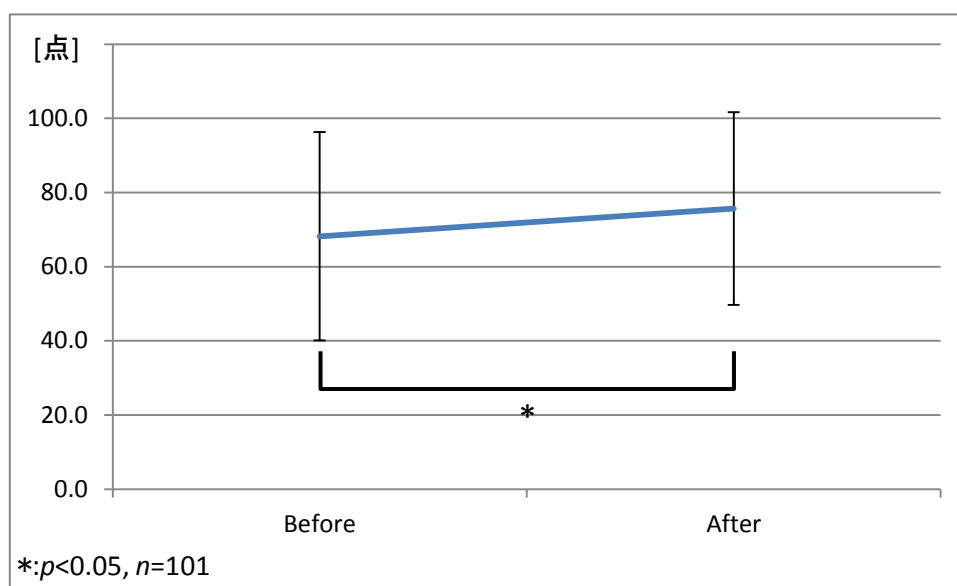


図 4-7 砲のブレの少なさにおける練習前後の得点の変化

次に、鉋のブレの少なさにおいて、クラス間の得点の比較を行った(図 4-8)。点数の向上が最も顕著であったのは 2 組であったが、授業前後ともに、どのクラスの間にも有意な差は見られなかった。

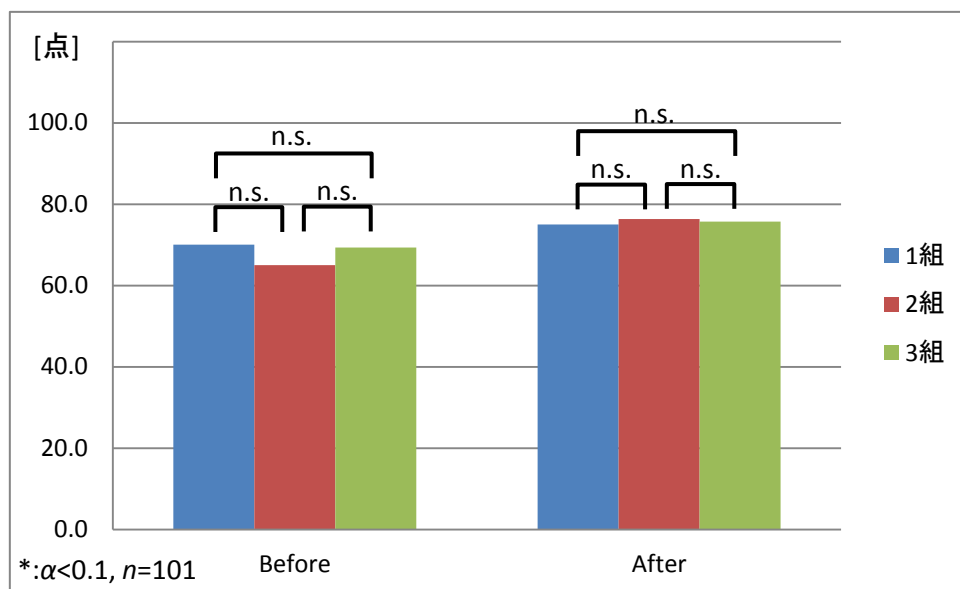


図 4-8 鉋のブレの少なさにおけるクラス間の比較

### 第3項 手元と腰の動き出しのタイミング

全生徒の練習前後における手元と腰の動き出しのタイミングの得点について平均の比較を行った(表 4-7, 図 4-9)。結果として練習前の 80.4 点から, 練習後 88.5 点に上昇しており, 1% 水準で有意な差( $t_{101}=-3.140$ ,  $p=0.002$ )が見られた。

表 4-7 手元と腰の動き出しのタイミングにおける練習前後の得点の比較

	Before	After	$p$	
平均	80.4	88.5	0.002	**
SD	26.6	22.9		

\*\* : 1%水準で有意

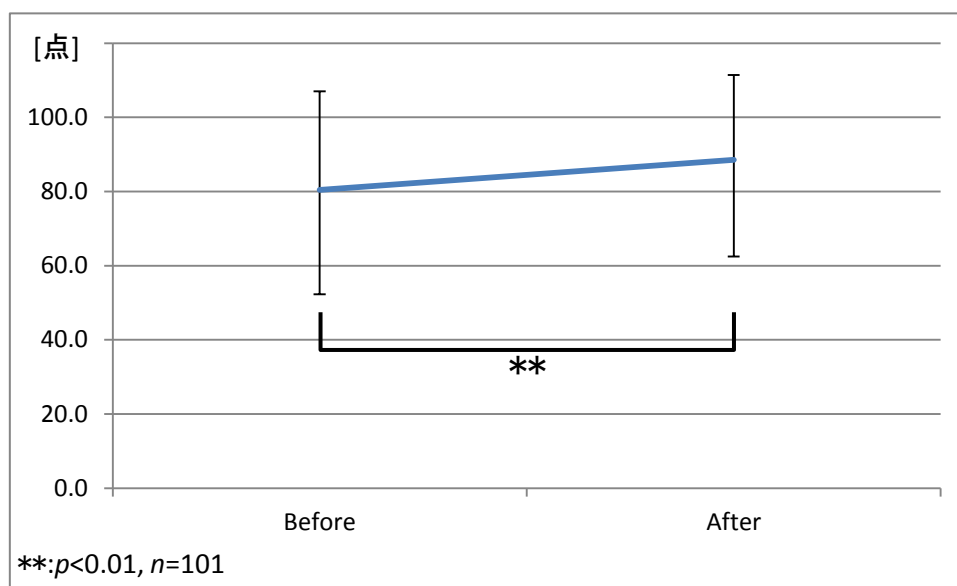


図 4-9 手元と腰の動き出しのタイミングにおける練習前後の得点の変化



次に、手元と腰の動き出しのタイミングにおいて、クラス間の得点の比較を行った(図 4-10)。練習前はどのクラス間の有意差は見られなかったが、練習後は 10%水準で有意差 ( $F_{(2,98)}=2.77, p=0.067$ )があった。Tukey-Kramer を用いた多重比較では、引きの速さと同様に 1 組と 2 組の間に有意差があることが示された。

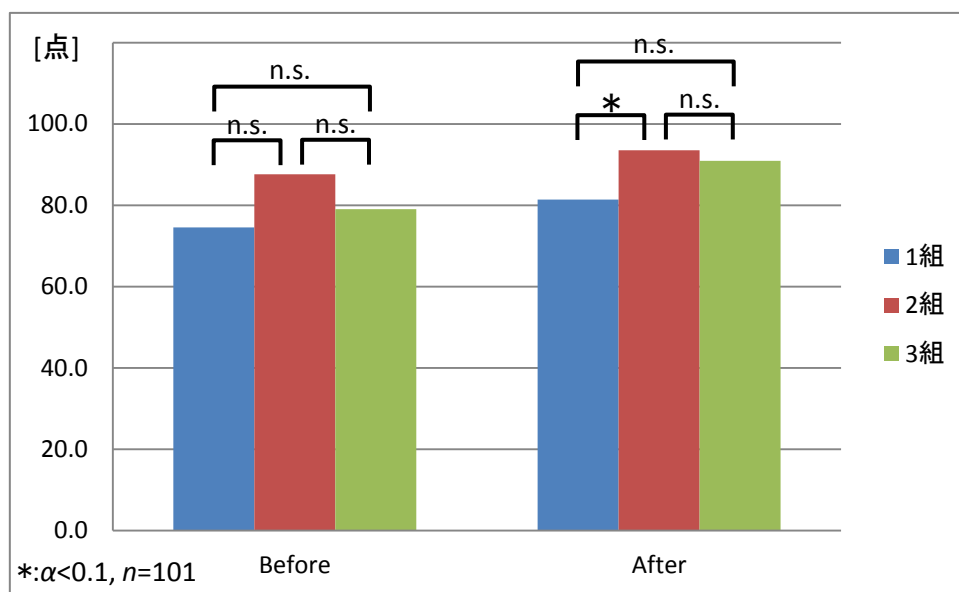


図 4-10 手元と腰の動き出しのタイミングにおけるクラス間の比較

## 第4項 総合得点

全生徒の練習前後における総合得点について平均の比較を行った(表 4-8, 図 4-11)。結果として練習前の 67.3 点から, 練習後 78.2 点に上昇しており, 0.1%水準で有意な差 ( $t_{101}=-3.140$ ,  $p=0.002$ )が見られた。

表 4-8 練習前後の総合得点の比較

	Before	After	$p$	
平均	67.3	78.2	<0.001	***
SD	15.8	14.4		

\*\*\*:0.1%水準で有意

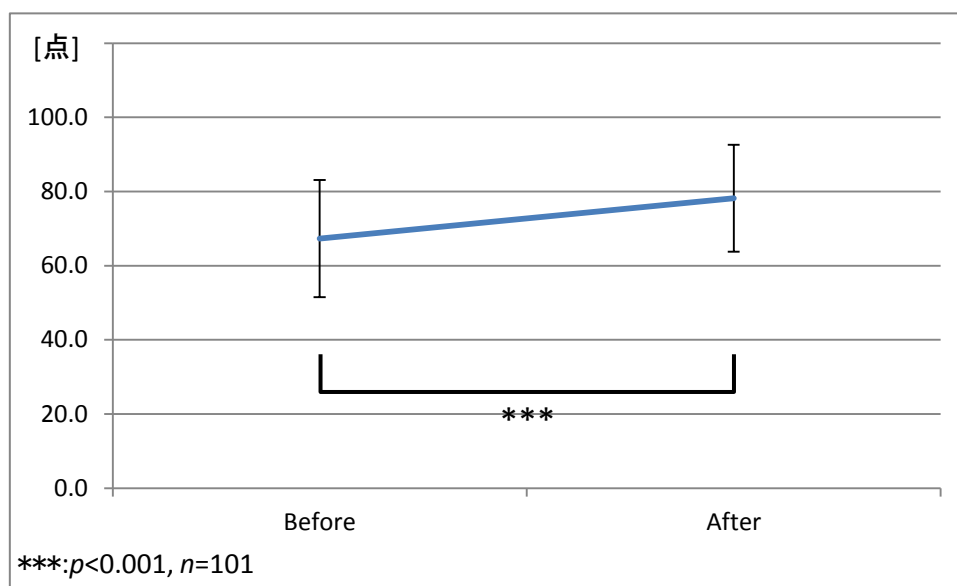


図 4-11 練習前後の総合得点の変化

次に、クラス間の総合得点の比較を行った(図 4-12)。練習前はどのクラス間の有意差は見られなかったが、練習後は 10% 水準で有意差 ( $F_{(2,98)}=3.64$ ,  $p=0.030$ ) があった。Tukey-Kramer を用いた多重比較では、引きの速さや手元と腰の動き出しのタイミングと同様に 1 組と 2 組の間に有意差があることが示された。

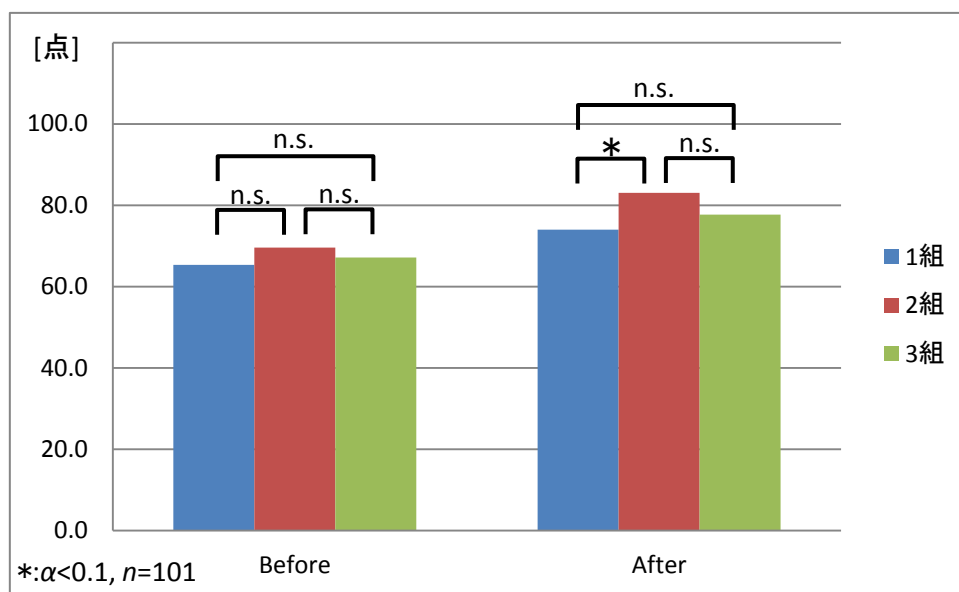


図 4-12 総合得点のクラス間の比較

## 第5項 教材の動機付け効果の評価

授業実施の前後に調査したアンケートの結果(表 4-9, 図 4-13)から, 鉋ラーニングの動機付け効果の評価を行った。評価にあたっては, 本章第 1 節の表 4-3 に示す項目の授業実施前後の結果をt検定により比較した。なお, 授業前後の両方に回答があった数が97であり, そのうち有効回答数は96であった。

「1. 木材の表面を削る加工をしたいと思いますか?」では, 授業前の回答の平均が 4.40, 授業後の回答の平均が 4.31 であった(図 4-14)。授業前後の回答に有意な差は見られなかった( $t_{96}=0.072$ ,  $p=0.472$ )。

「2. かんなを使って木材の表面をきれいにする自信はありますか?」では, 授業前の回答の平均が 3.04, 授業後の回答の平均が 3.00 であった(図 4-15)。授業前後の回答に有意な差は見られなかった( $t_{96}=0.261$ ,  $p=0.794$ )。

「3. かんながけの練習をして身に付けた知識や技能が今後の生活や製作に役立つと思いますか?」では, 授業前の回答の平均が 3.36, 授業後の回答の平均が 3.61 であり(図 4-16), 10%水準で有意な差( $t_{96}=-1.887$ ,  $p=0.062$ )を確認することができた。

「4. スマートフォンやアプリを使ったことで, 練習へのやる気が高まりましたか?」では, 回答の平均が 4.20 であった(授業後のみ実施)。

項目 4 の回答で 4.20 と比較的高い結果となったにも関わらず, 項目 1 や項目 2 において授業前後で有意な差が見られなかったことについて考察する。授業前に授業者により実施された調査から, 生徒の 81%が鉋の使用経験がないことがわかっている。また, 自由記述による授業の感想では「思っていた以上に鉋掛けが難しかった」という旨の記述が多数見られた。これらのことから, 生徒は鉋掛けの実際についてよく知らず, もっと簡単に削ることができるであろうと予想していたが, 実際に鉋掛けを行い, その難しさを感じたと考えられる。一方で, 鉋ラーニングを用いて練習したことで, 技能の向上や意欲の高まりがあり, そのマイナス面とプラス面が相殺され, 項目 1 や項目 2 においては, 授業前後の回答に差が見られなかったのではないかと考えられる。

表 4-9 教材の動機付け効果に関するアンケートの結果

		Before	After	<i>p</i>	
1. 木材の表面を削る加工をしたいと思いますか？	平均	4.40	4.30	0.472	n.s.
	SD	0.98	1.05		
2. かんなを使って木材の表面をきれいにする自信はありますか？	平均	3.04	3.00	0.794	n.s.
	SD	1.19	1.41		
3. かんながけの練習をして身に付けた知識や技能が今後の生活や製作に役立つと思いますか？	平均	3.36	3.61	0.062	*
	SD	1.20	1.22		
4. スマートフォンやアプリを使ったことで、練習へのやる気が高まりましたか？	平均	-	4.20	-	-
	SD	-	1.22		

\*:10%水準で有意

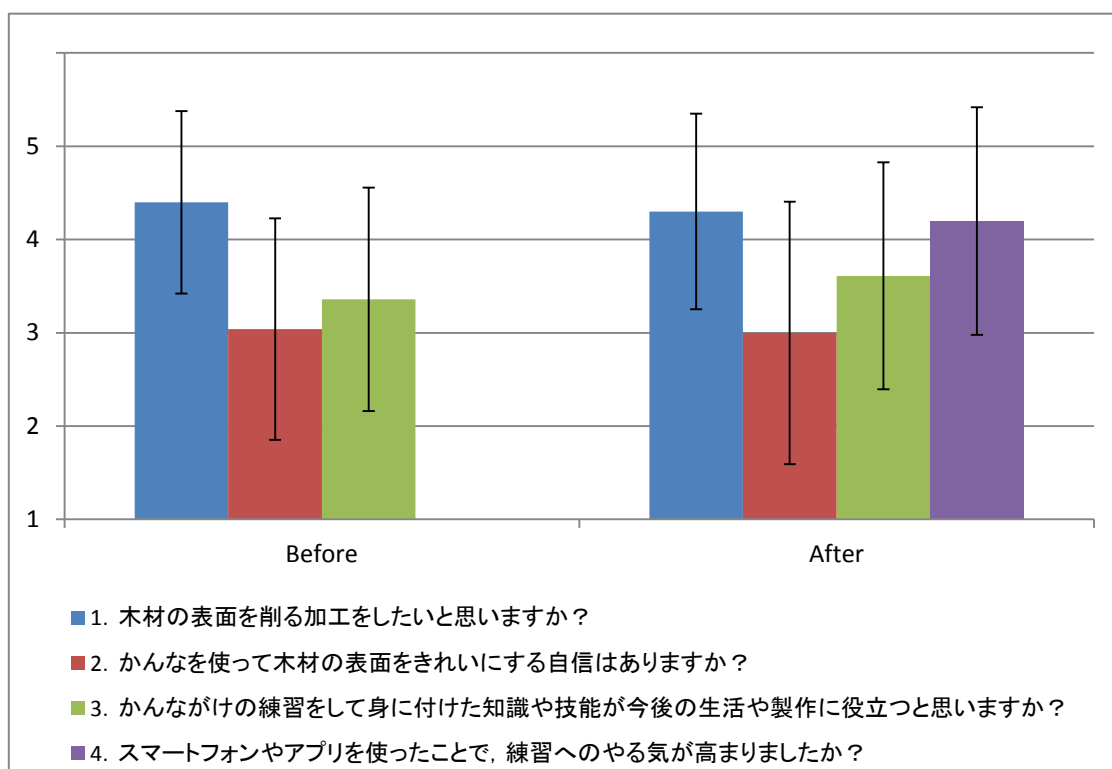


図 4-13 教材の動機付け効果に関するアンケートの結果

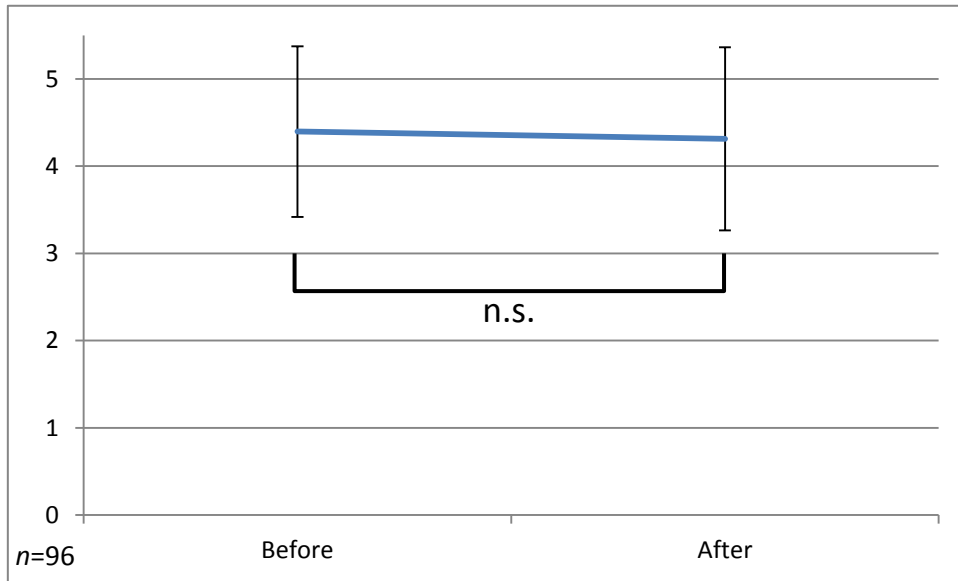


図 4-14 授業前後の比較 : 1. 木材の表面を削る加工をしたいか

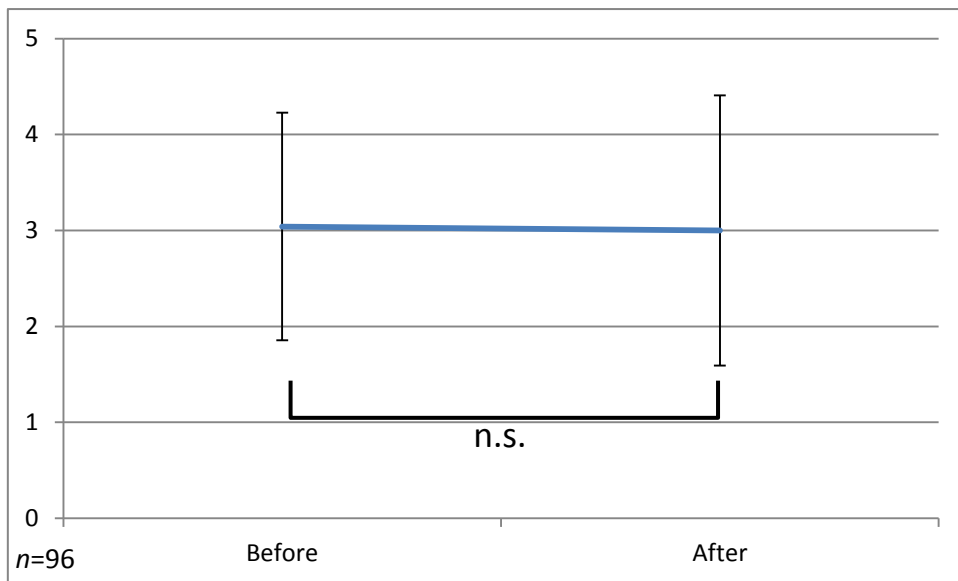


図 4-15 授業前後の比較 : 2. 表面をきれいにする自信はあるか

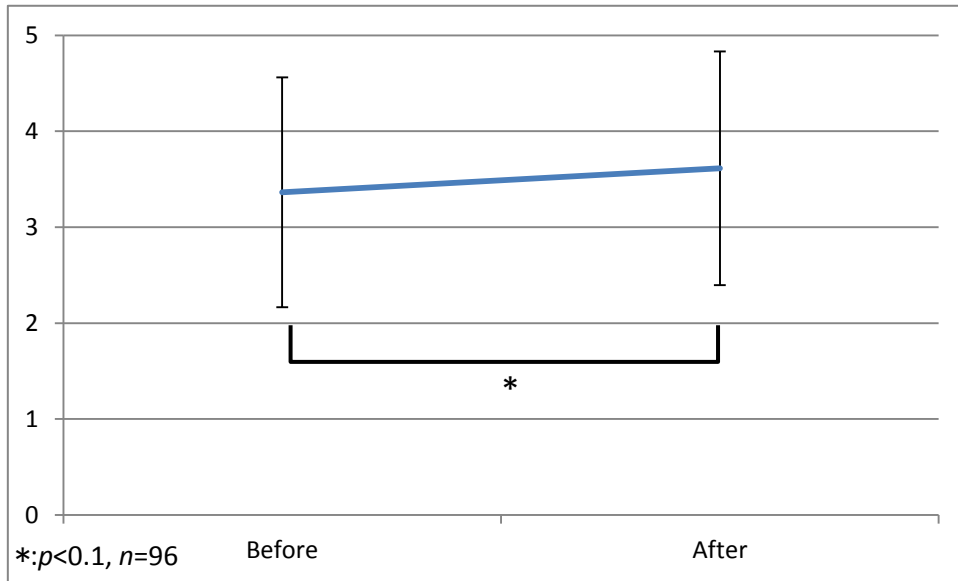


図 4-16 授業前後の比較 : 3. 今後の生活や製作に役立つと思うか

## 第6項 教材の機能分析

教材の機能分析に関するアンケートの結果を図 4-17 および図 4-18 に示す。本来、ストレスと集中度に関する項目は逆転項目であるが、集計の際、その他の項目と同様に評価できるよう、1 を 5 に、2 を 4 に置き換えた。

興味・関心に関する項目では、3 項目すべてにおいて平均が 4.5 を上回る結果となり、飽ラーニングの使用が生徒の興味・関心を高めることに繋がっていることがうかがえる。

ストレス 1「操作や使い方がわからなくなることがありましたか？」では平均値が 3.5 を下回る結果となった。練習前に 2 台の接続を行う点など操作の複雑さが原因であると考えられる。

態度 3「友達と余分な話をしないで授業を受けましたか？」や、集中度 3「授業にもっと集中できるとよかったなと思いますか？」も比較的低い結果となった。普段の授業では用いていないスマートフォンが目の前にあったという点や、教材のゲーム的要素が起因していると考えられる。

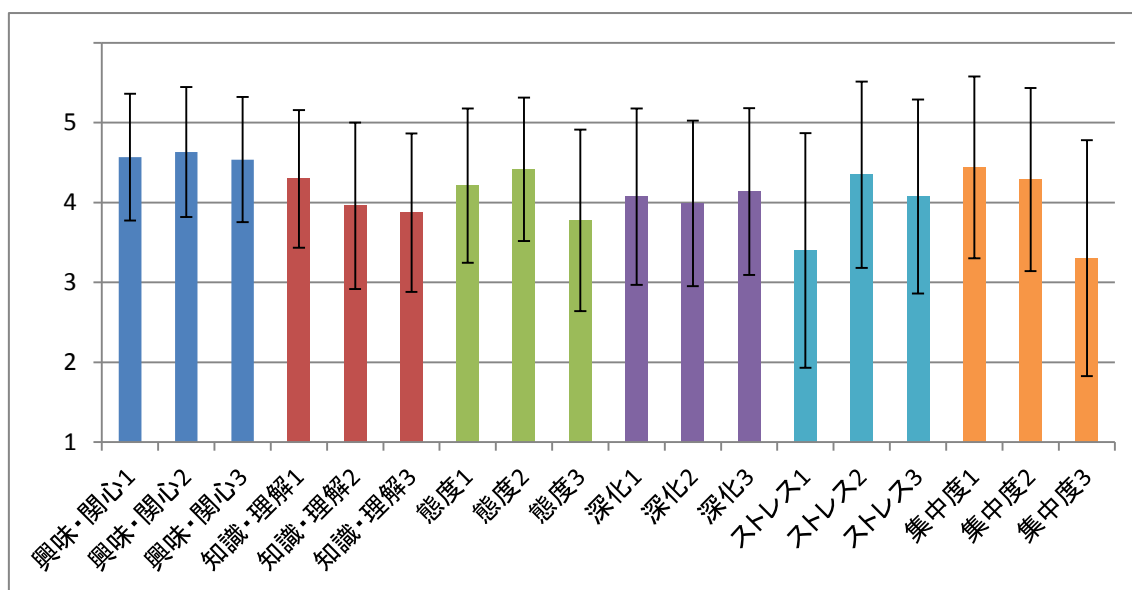


図 4-17 教材の機能分析に関するアンケートの結果(項目ごと)



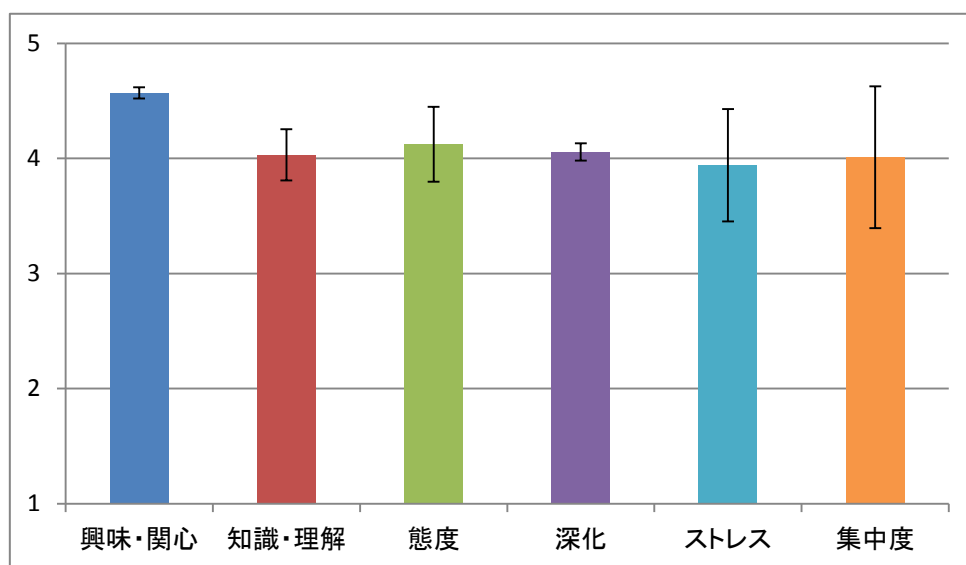


図 4-18 教材の機能分析に関するアンケートの結果(観点ごと)

## 第7項 考察・まとめ

授業前後の得点の比較からは、引きの速さ、鉋のブレの少なさ、手元と腰の動き出しのタイミングの3観点すべてにおいて、得点の有意な向上が見られた。また、これらの3観点の得点から算出される総合得点においても有意に得点の向上が認められた。

3つの観点の中でも、引きの速さの得点が最も顕著に向上していた。初歩段階として、鉋で材料を削るためには、切削抵抗に負けないよう力強く鉋を引く必要がある。そのため、生徒が“削ること”を意識し、速さや腰使いの動きが上達したと考えられる。一方、鉋のブレの少なさは、“きれいに削る”ための観点であり、段階としては速く引くことの次にあると考えられる。また、引きの速さなどと比べると緻密な制御が求められるため、鉋のブレの少なさに比べ、引きの速さや手元と腰の動き出しのタイミングの方が顕著に上達したと考察できる。

クラス間の比較においては、引きの速さ、手元と腰のタイミング、総合得点の3観点において、1組と2組の間に有意な差が確認された。いずれの観点においても得点は1組より2組の方が高かった。これは鉋ラーニングを用いて練習を行う際にワークシートを用い、生徒に点数を振り返らせ、その後の練習にどう活かすのかを考えさせる機会を設けたことが起因していると考えられる。1組の生徒の中には、鉋ラーニングを用いての練習がゲーム感覚になりすぎてしまい、引きの速さで満点を出すことだけに熱中し、その他の観点の得点や表示された結果・アドバイスを確認しない生徒も見受けられた。鉋ラーニングの活用方法として、生徒に結果やアドバイスを確実に振り返らせる手立てが必要であることが確認できた。

アンケートの結果からは、鉋ラーニングによる動機付けの効果に関して、生徒が授業の前後で「鉋掛けが今後の生活や製作に役立つ」と強く感じるようになることが分かった。

教材の機能分析に関するアンケートからは、生徒の興味・関心を強く引く教材であることが明らかになった。一方で、アプリケーションの操作の複雑さなどから生徒を困惑させてしまう面があることも明らかになった。また、鉋ラーニングのゲーム的要素から集中力を欠いてしまう生徒も一部で見受けられた。技能の上達に集中させる工夫が必要である。

授業後に行った授業者への聞き取り調査では、授業者は、鉋ラーニングが生徒を意欲的に取り組むきっかけになることを第一に述べていた。普段の授業と比較して格段に意欲的に取り組んでいたことがわかった。また、タブレット端末を使ってPLMSでデータを確認しながらの机間指導では、技能が未熟な生徒の下に直接向かい指導をすることができたと述べた。本来であれば、直接見ることでしか指導が必要な生徒を見つけることはできない。また、複数人でグループを組んで練習していると、練習している生徒以外の状態は確認することができない。そういったタイミングの問題をPLMSが解決した点を高評価していた。一方で、本研究で製作した専用鉋は、スマートフォンを装着できる最も小さなサイズのものであったが、中学生が使用するにあたってはもう一回り小さい鉋の方が適している点や、操作の複雑さを指摘していた。また、

今後実装を期待する機能として、木口削りや木端削りの動作のセンシングなどを挙げていた。

授業後に行われた検討会では、参観者から以下のような意見や質問が挙げられた。

1. 専用鉋に取り付けられたスマートフォンを操作しようとする、刃が机に触れる向きで置かざるを得ず、鉋の正しい置き方をすることができない
2. 操作が複雑で鉋掛けの練習に集中することができない
3. 鉋ラーニングによる評価と実際の材料の仕上りの質は一致しているのか

1 点目については、箸置きのように、鉋を置いたときに刃先がどこにも触れないような台を設けることや、腰端末にフィードバック等を表示し、鉋端末では操作を必要としないような仕様にするなどが解決策として考えられる。

2 点目についての解決策としては、鉋端末と腰端末との接続を、アプリケーションを起動したときに自動で行われるようにすること、端末 ID と生徒 ID をあらかじめ紐づけておき学習者選択の操作を省くこと、鉋が水平になり静止したことを検知し自動で計測が開始されるようにし「開始」ボタン押下の処理を省くことなどが挙げられる。

3 点目については、本研究では、鉋ラーニングの評価と実際の仕上りの関係を調査していないが、今後調査すべき重要な課題である。

## 参考文献

- [4-1] 山本利一, 佐藤正直: 中学校技術・家庭科栽培学習におけるタブレット端末の活用と授業実践, 日本教育情報学会誌, 第 29 巻, 第 1 号, pp.45-53 (2013).
- [4-2] 星野敏彦, 近藤明博, 加藤幸一: 技術科教育における教具の機能分析及び評価—電気・情報基礎領域の場合—, 日本産業技術教育学会誌, 第 45 巻, 第 2 号, pp.91-98 (2003).
- [4-3] 安藤明伸, 住川泰希: モーションキャプチャと仮想空間を利用した鋸挽き動作観察教材の開発と機能評価, 日本教育工学会論文誌, 第 38 巻, 第 2 号, pp.103-110 (2012).

## 第5章 結論

---

本研究では、スマートフォンを用いた鉋掛け技能学習支援アプリケーション 鉋ラーニングと、その練習データを管理するシステム PLMS (Practice Learning Management System)の開発を行った。鉋ラーニングは2台のスマートフォンを使用するアプリケーションであり、1台を鉋に、もう1台を腰に装着して鉋掛けの練習を行うと、引きの速さ、鉋のブレの少なさ、手元と腰の動き出しのタイミングの3つの観点においてセンシングを行い、学習者に結果の表示と、それを踏まえたアドバイスが与えられる。鉋ラーニングを用いて練習すると、練習データがPLMSのサーバへ送信・蓄積される。指導者はそのデータを一元管理することができる。

鉋ラーニングおよび PLMS の有用性を評価するために、2つの実験を行った。

1つ目の実験では、鉋ラーニングの家庭学習での利用可能性を評価することを目的とした。この実験では、ペットボトルを鉋に見立てて、鉋ラーニングを用いて鉋掛けのシミュレーションにより練習を行うことの効果について検証した。大学生10名を対象に、鉋ラーニングからのフィードバックを基に練習する実験群5名、教科書に記載された情報を基に練習する統制群5名に分けた。データの分析は、各群における練習前後の得点の比較と、練習前後各時点における群間の得点の比較を行った。

結果として、実験群では、総合得点において得点の有意な向上が見られた。評価の3観点(引きの速さ、鉋のブレ、手元と腰の動き出しのタイミング)においては、練習後に得点が向上してはいたものの、有意なものではなかった。事後のヒアリング調査からは、実験群の方が統制群よりも楽しんで練習をし、上達を感じている傾向にあることがわかった。一方で、両群に共通して、「ペットボトルでのシミュレーションと実際の鉋掛けはやはり異なる」という意見が挙がった。この実験で行ったような練習方法は、「鉋掛けの感覚を掴む」という目的のもとで行うべきもので、技能を習熟させるという段階までは至らないものと捉えている。

本調査では被験者が各群5人と少数であった。有意な向上が確認できたのは総合得点のみだったが、各観点の得点においても多くの被験者が向上していた。このことを踏まえ、追加実験を行い、被験者数を増やした上で、再度検証を行いたい。

2つ目の実験では、鉋ラーニングおよび PLMS の授業利用における有用性を評価することを目的とした。この実験では、仙台市立第二中学校1学年3クラスの授業で鉋ラーニングを使用した。授業後には引きの速さ、鉋のブレの少なさ、手元と腰の動き出しのタイミングの全ての観点において得点の向上が有意に認められた。また、授業を受けた生徒から得たアンケートの結果より、鉋ラーニングが生徒の関心・意欲を効果的に高める教材であることがわかった。一方で、アンケートの結果や事後の授業検討会などから、鉋ラーニングが持つゲーム的要素により生徒は集中力を欠いてしまったり、高い得点を得ることだけに熱中してしまい技能の上

達に対しての取り組みが疎かになってしまったりすることがわかった。授業実践の中からは、鉋ラーニングとワークシートを併用し、練習するたびに得点を振り返らせることが有効な手立ての 1 つであることが明らかになった。また、鉋ラーニングの操作の複雑さが生徒を困惑させたり、集中力を欠いたりしていることもわかった。技能の上達において不要な操作は最低限に抑え、より簡単に集中して練習に取り組めるようなアプリケーションに改良する必要がある。

机間指導時に授業者が PLMS を使用することについては、指導が必要な学習者を優先的に見つけ出し指導を行うことに効果があった。授業者が 1 人で生徒全員の動作を同時に確認することは難しい。また、グループになっての活動では、練習している生徒と順番を待っている生徒が生まれてしまう点も、生徒全員に指導を行うことを難しくする。そういった状況の中で、PLMS で生徒の練習の状態を一覧することができたことが、指導を支援した点を特に強く述べていた。

# 謝辞

---

本論文は、宮城教育大学安孫子啓教授および安藤明伸准教授のご指導の下、多くの方々からのご協力をいただき進めてきた研究の成果をまとめたものです。特に、安藤准教授からは、本研究の遂行にあたり、全面的にご指導いただきました。また、私が学部学生として研究室に配属されたときから、研究に関することはもちろんのこと、日常生活においても長い時間を共にしていただきました。快適に過ごせる研究室の環境の提供や、修了後の進路に関することなどについてもご助言をいただけたおかげで、研究の遂行に集中し専念することができました。心より感謝の意を表します。

開発したシステムの有用性検証の調査にあたりましては、鈴木丈治教諭をはじめとする仙台市立第二中学校の皆様、ならびに宮城教育大学バドミントン部の皆様に多大なるご協力をいただきました。鈴木教諭におかれましては、システムを授業の中で有効に活用する方法を深く検討していただき、システム改善のためのご助言等もいただきました。心より感謝申し上げます。

研究室の後輩である、福谷遼太氏、太田あゆみ氏、芹川郁子氏、潟岡冴子氏、川田拓氏、伊藤亮太氏、氏家裕樹氏、渡邊将大氏、志賀勇人氏、鳥村理人氏、泉健介氏、そして高橋耀氏とは、本研究を遂行するにあたり、多くの時間を同じ研究室で過ごしました。いつも明るい研究室で、毎日楽しく研究を進めることができました。特に福谷氏とは学部時代から本研究に関する助言や指摘をたくさんいただきました。心より感謝の意を表します。

最後に、本研究や大学院での生活を陰ながら支えてくださった家族や友人に、改めて感謝いたします。

平成 27 年 1 月 16 日 板垣翔大

# 本研究に関する発表論文

---

板垣翔大, 安藤明伸, 安孫子啓

複数のスマートフォンを用いたかんながけ動作習得用教材の開発  
日本産業技術教育学会第 18 回技術教育分科会 (2012)

板垣翔大, 安藤明伸, 高久敏宏, 鳥居隆司, 竹野英俊

平鉋削りの身体コーディネイトのずれをスマートフォンで取得する試み  
モバイル学会モバイル'13 研究論文集, pp.73-76 (2013)

安藤明伸, 竹野英敏, 鳥居隆司, 板垣翔大, 高久敏宏

技能の家庭学習を実現するための ICT 活用の枠組み  
日本産業技術教育学会第 56 回全国大会講演要旨集, p.50 (2013)

板垣翔大, 安藤明伸, 安孫子啓, 高久敏宏, 浅水智也

2 台のスマートフォンに対応させた鉋がけ動作習得用アプリケーションの開発  
日本産業技術教育学会第 56 回全国大会講演要旨集, p.51 (2013)

板垣翔大, 安藤明伸, 高久敏宏, 福谷遼太

技能の可視化は MOOTs(Massive Open Online Trainings)を可能にするか  
日本教育工学会第 29 回全国大会講演論文集 pp.255-256 (2013)

板垣翔大, 安藤明伸, 高久敏宏, 安孫子啓

鉋の平削り動作の学習履歴管理システムに必要な要件とは何か  
日本産業技術教育学会第 19 回技術教育分科会講演要旨集, pp.11-12 (2013)

Shota Itagaki, Akinobu Ando, Toshihiro Takaku, Hidetoshi Takeno, Takashi Torii  
Development of a Skill Practice Management System (PMS) for Learning Japanese  
Traditional Craft Tools by Using Smartphones  
Proceedings of World Conference on Educational Multimedia, Hypermedia and  
Telecommunications, Vol.2014, pp.1001-1009 (2014)

Akinobu Ando, Shota Itagaki, Hidetoshi Takeno, Takashi Torii, Darold Davis  
Development a Multiple Skill Practice Management and Result Feedback System  
for “Smart Vocational Learning” by Using Smartphones  
Proceedings of 2015 IEEE 4th Global Conference on Consumer Electronics ,  
pp.323-326 (2014)

# 図表目次

---

図 1-1	技術教育にあたる教科の授業時数の変化	2
図 1-2	技能の授業外学習や反転授業を行うためのステップと本研究の位置付け	7
図 2-1	開発したシステムの構成	12
図 2-2	鉋ラーニングを用いた鉋掛けの練習の様子	16
図 2-3	言葉による結果表示とアドバイス	18
図 2-4	速度および加速度のグラフ	19
図 2-5	速度を再現するリプレイアニメーション	20
図 2-6	上達履歴表示	21
図 2-7	鉋ラーニング専用鉋	24
図 2-8	鉋台の加工後の図面(等角図)	25
図 2-9	鉋台の加工後の図面(三面図)	26
図 2-10	専用鉋のアタッチメント(A)	27
図 2-11	専用鉋のアタッチメント(B)	28
図 2-12	専用鉋のカバー	29
図 2-13	カバーを鉋台に取り付ける蝶番	30
図 2-14	練習時のフローチャート	33
図 2-15	練習時の画面遷移図	34
図 2-16	モデル登録時のフローチャート	36
図 2-17	モデル登録時の操作の流れ	37
図 2-18	PLMS のページ構成	38
図 2-19	PLMS トップページ	39
図 2-20	グラフ表示	40
図 2-21	リスト表示	41
図 2-22	上達履歴表示	42
図 2-23	学習者端末に表示される上達履歴表示	43
図 2-24	鉋掛けに関する設定のページ	44
図 2-25	モデル選択ページ	44
図 2-26	生徒情報管理ページ	45
図 2-27	生徒情報追加ページ	45
図 2-28	PLMS のデータベース ER 図	47
図 3-1	ペットボトルに両面テープで貼り付けたゴムマット	50
図 3-2	ペットボトルへのスマートフォンの装着	50
図 3-3	引きの速さにおける得点の変化と比較	52



図 3-4 鉋のブレの少なさにおける得点の変化と比較.....	53
図 3-5 手元と腰の動き出しのタイミングにおける得点の変化と比較 .....	54
図 3-6 総合得点の変化と比較 .....	55
図 4-1 鉋ラーニングを使用し練習を行っている様子.....	58
図 4-2 2人1組で互いの結果を確認し合う生徒.....	59
図 4-3 授業者による机間指導時の PLMS 使用の様子.....	59
図 4-4 授業で使用したワークシート .....	60
図 4-5 引きの速さにおける練習前後の得点の変化.....	63
図 4-6 引きの速さにおけるクラス間の比較.....	64
図 4-7 鉋のブレの少なさにおける練習前後の得点の変化.....	65
図 4-8 鉋のブレの少なさにおけるクラス間の比較.....	66
図 4-9 手元と腰の動き出しのタイミングにおける練習前後の得点の変化.....	67
図 4-10 手元と腰の動き出しのタイミングにおけるクラス間の比較 .....	68
図 4-11 練習前後の総合得点の変化.....	69
図 4-12 総合得点のクラス間の比較.....	70
図 4-13 教材の動機付け効果に関するアンケートの結果.....	72
図 4-14 授業前後の比較:1. 木材の表面を削る加工をしたいか .....	73
図 4-15 授業前後の比較:2. 表面をきれいにする自信はあるか.....	73
図 4-16 授業前後の比較:3. 今後の生活や製作に役立つと思うか.....	74
図 4-17 教材の機能分析に関するアンケートの結果(項目ごと).....	75
図 4-18 教材の機能分析に関するアンケートの結果(観点ごと).....	76
表 2-1 開発環境 .....	13
表 2-2 鉋ラーニングの動作環境.....	13
表 2-3 Nexus 5 のスペック .....	14
表 2-4 Galaxy Nexus のスペック.....	14
表 2-5 Surface Pro 2 のスペック.....	15
表 2-6 使用しているセンサ .....	17
表 2-7 「手元と腰の動き出し」の観点における動作.....	22
表 2-8 鉋の重量.....	24
表 2-9 専用鉋部品表 .....	31
表 3-1 鉋とペットボトルの重量の比較.....	50
表 3-2 鉋とペットボトルの寸法の比較.....	50
表 3-3 本調査の詳細.....	51
表 3-4 引きの速さにおける得点の変化と比較.....	52
表 3-5 鉋のブレの少なさにおける得点の変化と比較.....	53

表 3-6	手元と腰の動き出しのタイミングにおける得点の変化と比較 .....	54
表 3-7	総合得点の変化と比較 .....	55
表 4-1	授業実施の詳細 .....	58
表 4-2	授業の流れ(2組) .....	61
表 4-3	動機付けの効果の検証に関するアンケート項目 .....	61
表 4-4	教材の機能の分析に関するアンケート項目 .....	62
表 4-5	引きの速さにおける練習前後の得点の比較.....	63
表 4-6	鉋のブレの少なさにおける練習前後の得点の比較.....	65
表 4-7	手元と腰の動き出しのタイミングにおける練習前後の得点の比較.....	67
表 4-8	練習前後の総合得点の比較.....	69
表 4-9	教材の動機付け効果に関するアンケートの結果.....	72

## 付録

# ソースコード

鉋ラーニングおよび PLMS の主要部分のソースコードを以下に記す。

## 【鉋ラーニング】

Kanna2Activity.java (鉋端末として指定した際に呼び出されるプログラム)

```
0001. package jp.anlab.kanna2PMS;
0002. import java.io.BufferedReader;
0003. import java.io.BufferedWriter;
0004. import java.io.File;
0005. import java.io.FileNotFoundException;
0006. import java.io.FileOutputStream;
0007. import java.io.FileReader;
0008. import java.io.IOException;
0009. import java.io.InputStream;
0010. import java.io.OutputStream;
0011. import java.io.OutputStreamWriter;
0012. import java.io.UnsupportedEncodingException;
0013. import java.math.BigDecimal;
0014. import java.net.HttpURLConnection;
0015. import java.net.URL;
0016. import java.util.ArrayList;
0017. import java.util.Calendar;
0018. import java.util.Set;
0019. import java.util.UUID;
0020. import java.util.Vector;
0021. import org.apache.http.HttpResponse;
0022. import org.apache.http.client.methods.HttpGet;
0023. import
0024.     org.apache.http.impl.client.DefaultHttpClient;
0025. import org.apache.http.util.EntityUtils;
0026. import android.app.Activity;
0027. import android.app.AlertDialog;
0028. import android.app.ProgressDialog;
0029. import android.bluetooth.BluetoothAdapter;
0030. import android.bluetooth.BluetoothDevice;
0031. import android.bluetooth.BluetoothServerSocket;
0032. import android.bluetooth.BluetoothSocket;
0033. import android.content.Context;
0034. import android.content.DialogInterface;
0035. import android.content.SharedPreferences;
0036. import android.content.SharedPreferences.Editor;
0037. import android.graphics.Bitmap;
0038. import android.graphics.Bitmap.CompressFormat;
0039. import android.graphics.BitmapFactory;
0040. import android.graphics.Canvas;
0041. import android.graphics.Color;
0042. import android.graphics.Paint;
0043. import android.graphics.Path;
0044. import android.graphics.PorterDuff;
0045. import android.hardware.Sensor;
0046. import android.hardware.SensorEvent;
0047. import android.hardware.SensorEventListener;
0048. import android.hardware.SensorManager;
0049. import android.media.MediaPlayer;
0050. import android.os.Bundle;
0051. import android.os.Environment;
0052. import android.os.Handler;
0053. import android.os.Message;
0054. import android.text.format.Time;
0055. import android.util.Log;
0056. import android.view.Gravity;
0057. import android.view.KeyEvent;
0058. import android.view.LayoutInflater;
0059. import android.view.View;
0060. import android.view.View.OnClickListener;
0061. import android.view.ViewGroup;
0062. import android.webkit.WebView;
0063. import android.widget.AdapterView;
0064. import android.widget.AdapterView.OnItemClickListener;
0065. import android.widget.ArrayAdapter;
0066. import android.widget.Button;
0067. import android.widget.LinearLayout;
0068. import android.widget.LinearLayout.LayoutParams;
0069. import android.widget.ListView;
0070. import android.widget.TabHost;
0071. import android.widget.TabHost.TabSpec;
0072. import android.widget.TextView;
0073. import android.widget.Toast;
0074. public class Kanna2Activity extends Activity
0075.     implements SensorEventListener {
0076.     AgraphView AgraphView;
0077.     AlertDialog dialog2;
0078.     AlertDialog.Builder alertDialogBuilder;
0079.     AlertDialog.Builder replayDlg;
0080.     android.app.AlertDialog.Builder dialog = null;
0081.     ArrayList<Float> blurList = new ArrayList<Float>();
0082.     ArrayList<Float> bureList = new ArrayList<Float>();
0083.     ArrayList<Float> koshiList = new ArrayList<Float>();
0084.     ArrayList<Float> model_Ylist = new
0085.         ArrayList<Float>();
0086.     ArrayList<Float> speedList = new ArrayList<Float>();
0087.     ArrayList<Float> Ylist = new ArrayList<Float>();
0088.     ArrayList<Float> Ylist2 = new ArrayList<Float>();
0089.     ArrayList<Float> YlistA = new ArrayList<Float>();
0090.     ArrayList<Float> YlistA2 = new ArrayList<Float>();
0091.     ArrayList<Integer> docchi = new
0092.         ArrayList<Integer>();
0093.     ArrayList<String> arrayGakuban;
0094.     ArrayList<String> arrayKeyId;
0095.     ArrayList<String> arrayName;
0096.     ArrayList<String> saveList = new
0097.         ArrayList<String>();
0098.     Bitmap kanna;
0099.     BluetoothAdapter mAdapter;
0100.     BluetoothSocket[] socket = new BluetoothSocket[1];
```

```

0097. boolean finishflg = false;
0098. boolean graphKannaMove = false;
0099. boolean isKannaMove = false;
0100. boolean isKoshiMove = false;
0101. boolean koshiPlot = false;
0102. Button endBtn;
0103. Button okBtn;
0104. Button replayBtn;
0105. Button replayBtn2;
0106. Calendar cld;
0107. ConnectedThread[] connection = new
    ConnectedThread[1];
0108. Editor e;
0109. final int A = 0;
0110. final int B = 1;
0111. final int C = 2;
0112. final int KANNA = 1;
0113. final int KOSHI = 0;
0114. final String NAME = "BLUETOOTH_SAMPLE";
0115. float Acc;
0116. float AccKako;
0117. float bureBorder1 = 70;
0118. float bureBorder2 = 90;
0119. float docchif = 0;
0120. float gyro_value[];
0121. float gyroValue;
0122. float hosei = (float) Math.cos(Math.PI / 4f);
0123. float kakoVel;
0124. float model_replayPosition = 0;
0125. float modelBure;
0126. float moveLine = 3;
0127. float postBlur;
0128. float postSpeed;
0129. float replayCenter = 0;
0130. float replayLeft = 0;
0131. float replayLength = 0;
0132. float replayPosition = 0;
0133. float replayRatio = 0;
0134. float replayRight = 0;
0135. float Spa;
0136. float span = 0;
0137. float speedBorder1 = 70;
0138. float speedBorder2 = 90;
0139. float Vel;
0140. float y_hosei;
0141. float z_hosei;
0142. float[] acc = new float[3];
0143. float[] gyr = new float[3];
0144. Handler dHandler = new Handler();
0145. Handler mHandler = new Handler();
0146. int blurRank;
0147. int burePoint = 0;
0148. int graphDownInt = graphDownReset;
0149. int graphDownReset = 5;
0150. int intCount = 4;
0151. int jikan;
0152. int kessonchi = 0;
0153. int koshiPoint = 0;
0154. int modelZureInt;
0155. int practiceNo = 1;
0156. int replayHeight = 0;
0157. int replayWidth = 0;
0158. int speedPoint = 0;
0159. int speedRank;
0160. int totalPoint = 0;

```

```

0161. int userClassNum = 2;
0162. int userId;
0163. int XYnum = 0;
0164. int zureInt;
0165. LinearLayout btnLayout;
0166. long kanna_time = 0;
0167. long koshi_time = 0;
0168. long koshi_time_int = 0;
0169. long t;
0170. long unixTime;
0171. Paint paintBorder = new Paint();
0172. Paint paintLine = new Paint();
0173. Paint paintLine2 = new Paint();
0174. Paint paintSensor = new Paint();
0175. Paint paintText = new Paint();
0176. Path path = new Path();
0177. Path pathA = new Path();
0178. private Bitmap bmp = null;
0179. private Bitmap bmp_model = null;
0180. private Bitmap bmp_modelA = null;
0181. private Bitmap bmpA = null;
0182. private boolean divflag_kanna = false;
0183. private boolean divflag_koshi = false;
0184. private boolean isRepeat;
0185. private boolean startFlag;
0186. private Button changeUser;
0187. private Button countBtn;
0188. private Canvas bmpCanvas;
0189. private Canvas bmpCanvasA;
0190. private File file;
0191. private File fileHosei;
0192. private float avarage_koshi;
0193. private float div_koshi;
0194. private float goukei;
0195. private float goukei_x;
0196. private float mYOffset;
0197. private float mYOffsetA;
0198. private float pastY = 200;
0199. private float pastYA = 200;
0200. private float pastYs = 200;
0201. private float x_button;
0202. private float x_kosei;
0203. private float x_kosei2;
0204. private float x_sensor;
0205. private float xGraph = 20;
0206. private float y_sensor;
0207. private float y_SensorValue;
0208. private float[] currently_koshi;
0209. private int avarageInt_kanna;
0210. private int divInt_kanna;
0211. private int hight;
0212. private int maxInt = 0;
0213. private int minInt = 0;
0214. private int point;
0215. private int REPEAT_INTERVAL;
0216. private int timesInt;
0217. private int toggle;
0218. private int wide;
0219. private int[] currentlyInt_kanna;
0220. private LinearLayout countLayout;
0221. private long alltime;
0222. private long start;
0223. private long stop;
0224. private MediaPlayer mp = null;
0225. private Runnable looper;

```

```

0226. private SensorManager sensorManager;
0227. private SharedPreferences modelpref;
0228. private String date;
0229. private String date2;
0230. private String str_rank;
0231. private TextView countTextView;
0232. private TextView textView;
0233. private TextView userTextView;
0234. private Thread thread;
0235. private Thread threads;
0236. private Time time;
0237. ProgressDialog progressDialog;
0238. ReplayView replayView;
0239. Runnable countRunnable;
0240. Runnable replayer;
0241. SensorCalibration SC = new SensorCalibration();
0242. ServerThread serverThread;
0243. static ArrayAdapter<String> adapter;
0244. String Data = null;
0245. String filename;
0246. String filenameHosei;
0247. String host = "(専用サーバのホスト)";
0248. String ID = "";
0249. String logstr;
0250. String modeldata_str;
0251. String modelName;
0252. String result;
0253. String saveBure;
0254. String saveHosei;
0255. String saveKoshi;
0256. String saveText;
0257. String saveTime;
0258. String speedRankStr;
0259. String strint;
0260. String timing;
0261. String username;
0262. String ViewStr;
0263. Thread replay;
0264. UUID MY_UUID =
    UUID.fromString("11111111-0000-1000-1111-00AAEECCAA
    FF");
0265. Vector<BluetoothDevice> mDevices;
0266. VgraphView VgraphView;
0267. View resultView;
0268.
0269. @Override
0270. public void onCreate(Bundle savedInstanceState) {
0271.     super.onCreate(savedInstanceState);
0272.
0273.     // SensorManager インスタンスを取得
0274.     sensorManager = (SensorManager)
        getSystemService(Context.SENSOR_SERVICE);
0275.     gyro_value = new float[4];
0276.
0277.     // モデル未選択時の警告ダイアログ
0278.     AlertDialog.Builder a = new
        AlertDialog.Builder(this);
0279.     a.setMessage("モデルが選択されていません");
0280.     a.setPositiveButton("OK", new
        DialogInterface.OnClickListener() {
0281.         public void onClick(DialogInterface dialog, int id)
        {
0282.             finish();
0283.         }
0284.     });
0285.
0286.     // 結果表示用の View
0287.     textView = new TextView(this);
0288.     textView.setText("ここに結果が表示されます");
0289.
0290.     // グラフ描画の用 View
0291.     VgraphView = new VgraphView(this);
0292.     AgraphView = new AgraphView(this);
0293.
0294.     // 各ボタン等レイアウト
0295.     btnLayout = new LinearLayout(this);
0296.     btnLayout.setGravity(Gravity.BOTTOM |
        Gravity.CENTER_HORIZONTAL);
0297.     okBtn = new Button(this);
0298.     endBtn = new Button(this);
0299.     replayBtn = new Button(this);
0300.     countLayout = new LinearLayout(this);
0301.     countLayout.setOrientation(LinearLayout.VERTICAL);
0302.     countLayout.setBackgroundResource(R.drawable.start
        );
0303.     setContentView(countLayout);
0304.     countTextView = new TextView(this);
0305.     countTextView.setText("↓");
0306.     countTextView.setTextColor(Color.WHITE);
0307.     countTextView.setTextSize(250);
0308.     countTextView.setGravity(Gravity.CENTER |
        Gravity.CENTER);
0309.     countBtn = new Button(this);
0310.     countBtn.setEnabled(false);
0311.     countBtn.setBackgroundResource(R.drawable.go_xml);
0312.     userTextView = new TextView(this);
0313.     changeUser = new Button(this);
0314.     changeUser.setText("学習者の変更");
0315.     changeUser.setEnabled(false);
0316.     countLayout.addView(countBtn, new
        LinearLayout.LayoutParams(
0317.         ViewGroup.LayoutParams.FILL_PARENT, 250));
0318.     LinearLayout.LayoutParams lp = new
        LinearLayout.LayoutParams
        (ViewGroup.LayoutParams.FILL_PARENT,
0320.         ViewGroup.LayoutParams.FILL_PARENT);
0321.     lp.weight = 1f;
0322.     countLayout.addView(countTextView, lp);
0323.     lp = new LinearLayout.LayoutParams
        (ViewGroup.LayoutParams.FILL_PARENT,
0324.         ViewGroup.LayoutParams.FILL_PARENT);
0325.     countLayout.addView(userTextView);
0326.     userTextView.setGravity(Gravity.CENTER_HORIZONTAL);
0327.     countLayout.addView(changeUser);
0328.
0329.     // グラフ描画の色等
0330.     paintSensor.setColor(Color.CYAN);
0331.     paintSensor.setStyle(Paint.Style.STROKE);
0332.     paintSensor.setStrokeWidth(4);
0333.     paintSensor.setAntiAlias(true);
0334.     paintLine.setColor(Color.WHITE);
0335.     paintLine.setStyle(Paint.Style.FILL);
0336.     paintLine.setStrokeWidth(3);
0337.     paintLine2.setColor(Color.WHITE);
0338.     paintLine2.setStyle(Paint.Style.FILL);
0339.     paintLine2.setStrokeWidth(1);
0340.     paintBorder.setColor(Color.WHITE);
0341.     paintBorder.setStyle(Paint.Style.FILL);
0342.     paintBorder.setStrokeWidth(30);
0343.     paintText.setColor(Color.CYAN);

```

```

0344.   paintText.setStyle(Paint.Style.FILL);
0345.   paintText.setStrokeWidth(2);
0346.   paintText.setTextSize(30);
0347.
0348.   // リプレイアニメーション用 View
0349.   replayView = new ReplayView(this);
0350.   kanna = BitmapFactory.decodeResource(getResources(),
R.drawable.kanna);

0351.
0352.   // 結果のタブ
0353.   LayoutInflater inflater = LayoutInflater.from(this);
0354.   resultView = inflater.inflate(R.layout.result_view,
null);
0355.   TabHost host = (TabHost)
resultView.findViewById(R.id.tabhost);
0356.   host.setup();
0357.   TabSpec tabComment = host.newTabSpec("Comment");
0358.   tabComment.setIndicator("コメント");
0359.   tabComment.setContent(R.id.resultComment);
0360.   host.addTab(tabComment);
0361.   TabSpec tabGraph = host.newTabSpec("Graph");
0362.   tabGraph.setIndicator("グラフ");
0363.   tabGraph.setContent(R.id.resultGraph);
0364.   host.addTab(tabGraph);
0365.   TabSpec tabReplay = host.newTabSpec("Replay");
0366.   tabReplay.setIndicator("リプレイ");
0367.   tabReplay.setContent(R.id.resultReplay);
0368.   host.addTab(tabReplay);
0369.
0370.   // 開始ボタン押下時の処理
0371.   countBtn.setOnClickListener(new
View.OnClickListener() {
0372.   public void onClick(View v) {
0373.   graphKannaMove = true;
0374.   mp = MediaPlayer.create(Kanna2Activity.this,
R.raw.button);
0375.   try {
0376.   mp.prepare();
0377.   } catch (IllegalStateException e) {
0378.   e.printStackTrace();
0379.   } catch (IOException e) {
0380.   e.printStackTrace();
0381.   }
0382.   mp.start();
0383.   mp = MediaPlayer.create(Kanna2Activity.this,
R.raw.count);
0384.   try {
0385.   mp.prepare();
0386.   } catch (IllegalStateException e) {
0387.   e.printStackTrace();
0388.   } catch (IOException e) {
0389.   e.printStackTrace();
0390.   }
0391.   mp.start();
0392.   Countdown();
0393.   countBtn.setVisibility(View.GONE);
0394.   changeUser.setVisibility(View.GONE);
0395.   countTextV.setTextSize(400);
0396.   }
0397.   });
0398.
0399.   // 学習者変更ボタン押下時の処理
0400.   changeUser.setOnClickListener(new
View.OnClickListener() {
0401.   public void onClick(View v) {
0402.   getuser();
0403.   }
0404.   });
0405.
0406.   // モデルデータの読み込み
0407.   modelpref = getSharedPreferences("2modelpref",
MODE_WORLD_READABLE
| MODE_WORLD_WRITEABLE);
0408.   modeldata_str = modelpref.getString("2modelint",
"");
0409.   strint = modelpref.getString("2modelint", "");
0410.   modelName = modelpref.getString("2name", "");
0411.   if (modelName != null) {
0412.   if (!modelName.equals("")) {
0413.   modelZureInt =
Integer.parseInt(modelpref.getString(modelName
+ "2zure", ""));
0414.   logsttr = modelpref.getString(modelName + "2maxInt",
"");
0415.   modelBure =
Marumeru(Float.parseFloat(modelpref.getString
(modelName + "2maxBure", "")), 2);
0416.   username = modelpref.getString("username", "");
0417.   BufferedReader br = null;
0418.   StringBuilder sb = new StringBuilder();
0419.   try {
0420.   try {
0421.   br = new BufferedReader(new FileReader(Environment
.getExternalStorageDirectory().getPath()
+ "/"
+ getPackageName()
+ "/two/"
+ modelName
+ ".txt"));
0422.   String str;
0423.   while ((str = br.readLine()) != null) {
0424.   sb.append(str);
0425.   } finally {
0426.   if (br != null) {
0427.   br.close();
0428.   }
0429.   } catch (IOException e) {
0430.   }
0431.   String allData = sb.toString();
0432.   String rawData[] = allData.split(",");
0433.   for (int i = 0; i < rawData.length; i++) {
0434.   try {
0435.   model_Ylist.add(Float.parseFloat(rawData[i]));
0436.   replayLength += Float.parseFloat(rawData[i]);
0437.   } catch (NumberFormatException e1) {
0438.   model_Ylist.add(0f);
0439.   }
0440.   }
0441.   replayRatio = (replayHeight - 50) / replayLength;
0442.
0443.   // 腰端末との Bluetooth 接続
0444.   mDevices = new Vector<BluetoothDevice>();
0445.   mAdapter = BluetoothAdapter.getDefaultAdapter();
0446.   Set<BluetoothDevice> devices =
mAdapter.getBondedDevices();
0447.   for (BluetoothDevice device : devices) {
0448.   mDevices.add(device);
0449.   }

```

```

0461. serverThread = new ServerThread();
0462. serverThread.start();
0463. } else {
0464. a.show();
0465. }
0466. } else {
0467. a.show();
0468. }
0469. }
0470.
0471. // 組選択ダイアログ
0472. public void classView() {
0473. final String[] classStr = { "1", "2", "3" };
0474. ListView LV = new ListView(this);
0475. adapter = new ArrayAdapter<String>(this,
0476. android.R.layout.simple_list_item_1, classStr);
0477. dialog = new AlertDialog.Builder(this);
0478. dialog2 = dialog.create();
0479. dialog2.setCanceledOnTouchOutside(false);
0480. dialog2.setTitle("あなたのクラスを選択してください");
0481. dialog2.setView(LV);
0482. dialog2.show();
0483. LV.setAdapter(adapter);
0484. LV.setOnItemClickListener(new
0485. OnItemClickListener(){
0486. public void onItemClick(AdapterView<?> items, View
0487. view,
0488. int position, long id) {
0489. userClassNum =
0490. Integer.parseInt(classStr[position]);
0491. dialog2.dismiss();
0492. getuser();
0493. }
0494. });
0495. }
0496. // カウントダウン
0497. public void CountDown() {
0498. final int REPEAT_INTERVAL = 1000;
0499. final Handler handler = new Handler();
0500. countRunnable = new Runnable() {
0501. public void run() {
0502. intCount--;
0503. countTextView.setText(String.valueOf(intCount));
0504. if (intCount == 0) {
0505. handler.removeCallbacks(countRunnable);
0506. countTextView.setTextSize(150);
0507. countTextView.setText("始\nめ\nめ\n!");
0508. onBtn();
0509. } else {
0510. handler.postDelayed(this, REPEAT_INTERVAL);
0511. }
0512. handler.postDelayed(countRunnable,
0513. REPEAT_INTERVAL);
0514. }
0515. }
0516. // 計測中の処理
0517. public void onBtn() {
0518. currentlyInt_kanna = new int[10];
0519. currently_koshi = new float[10];
0520. mHandler = new Handler();
0521. startFlag = true;
0522. isRepeat = false;
0523. REPEAT_INTERVAL = 5;
0524. time = new Time("Asia/Tokyo");
0525. looper = new Runnable() {
0526. public void run() {
0527. while (isRepeat) {
0528. try {
0529. Thread.sleep(REPEAT_INTERVAL);
0530. } catch (InterruptedException e) {
0531. }
0532. mHandler.post(new Runnable() {
0533. public void run() {
0534. doSomething();
0535. }
0536. });
0537. }
0538. };
0539. isRepeat = true;
0540. goukei = 0;
0541. goukei_x = 0;
0542. start = 0;
0543. stop = 0;
0544. maxInt = 0;
0545. minInt = 0;
0546. timesInt = 0;
0547. start = System.currentTimeMillis();
0548. time.setToNow();
0549. thread = new Thread(looper);
0550. thread.start();
0551. textV.setTextSize(75);
0552. textV.setTextColor(Color.WHITE);
0553. textV.setText("計測中...");
0554. textV.setPadding(0, 100, 0, 0);
0555. textV.setGravity(Gravity.CENTER);
0556. float y_btn = y_sensor * 10;
0557. y_btn = Math.round(y_btn);
0558. y_btn = y_btn / 10;
0559. SC.setBtnSensorValue(y_btn);
0560. x_kosei = x_sensor * 10;
0561. x_kosei = Math.round(x_kosei);
0562. x_kosei = x_kosei / 10;
0563. x_button = x_kosei;
0564. }
0565. }
0566. // 繰り返しの動作
0567. private void doSomething() {
0568. jikan++;
0569. gyro_value[3] = gyro_value[3] +
0570. Math.abs(gyro_value[0])
0571. + Math.abs(gyro_value[1]) + Math.abs(gyro_value[2]);
0572. int finishjikan = jikan / 10;
0573. y_SensorValue = SC.setNowSensorValue(y_sensor);
0574. x_kosei = x_sensor * 10;
0575. x_kosei = Math.round(x_kosei);
0576. x_kosei = x_kosei / 10;
0577. if (startFlag == true) {
0578. timesInt++;
0579. for (int i = 9; i >= 1; i--) {
0580. currentlyInt_kanna[i] = currentlyInt_kanna[i - 1];
0581. currently_koshi[i] = currently_koshi[i - 1];
0582. }
0583. currentlyInt_kanna[0] = (int) y_SensorValue;
0584. currently_koshi[0] = Float.parseFloat(Data);
0585. avarageInt_kanna = (currentlyInt_kanna[0] +
0586. currentlyInt_kanna[1]

```



```

0585. + currentlyInt_kanna[2] + currentlyInt_kanna[3]
0586. + currentlyInt_kanna[4] + currentlyInt_kanna[5]
0587. + currentlyInt_kanna[6] + currentlyInt_kanna[7]
0588. + currentlyInt_kanna[8] + currentlyInt_kanna[9]) /
10;
0589. avarage_koshi = (currently_koshi[0] +
currently_koshi[1]
0590. + currently_koshi[2] + currently_koshi[3]
0591. + currently_koshi[4] + currently_koshi[5]
0592. + currently_koshi[6] + currently_koshi[7]
0593. + currently_koshi[8] + currently_koshi[9]) / 10;
0594. int[] deviationInt_kanna = new int[10];
0595. float[] deviation_koshi = new float[10];
0596. for (int i = 0; i <= 9; i++) {
0597. deviationInt_kanna[i] =
Math.abs(currentlyInt_kanna[i]
0598. - avarageInt_kanna);
0599. deviation_koshi[i] = Math.abs(currently_koshi[i]
0600. - avarage_koshi);
0601. }
0602. divInt_kanna = (deviationInt_kanna[0] +
deviationInt_kanna[1]
0603. + deviationInt_kanna[2] + deviationInt_kanna[3]
0604. + deviationInt_kanna[4] + deviationInt_kanna[5]
0605. + deviationInt_kanna[6] + deviationInt_kanna[7]
0606. + deviationInt_kanna[8] + deviationInt_kanna[9]) /
10;
0607. div_koshi = (deviation_koshi[0] + deviation_koshi[1]
0608. + deviation_koshi[2] + deviation_koshi[3]
0609. + deviation_koshi[4] + deviation_koshi[5]
0610. + deviation_koshi[6] + deviation_koshi[7]
0611. + deviation_koshi[8] + deviation_koshi[9]) / 10;
0612. if (divflag_koshi == false && div_koshi > 0.7f) {
0613. koshi_time = System.currentTimeMillis()-75;
0614. divflag_koshi = true;
0615. koshiPlot = true;
0616. docchi.add(KOSHI);
0617. koshi_time_int = jikan;
0618. }
0619. if (divflag_kanna == false && divInt_kanna > 0) {
0620. divflag_kanna = true;
0621. isKannaMove = true;
0622. kanna_time = System.currentTimeMillis();
0623. docchi.add(KANNA);
0624. }
0625. if (finishjikan == 36) {
0626. divflag_kanna = false;
0627. finishflg = true;
0628. countTextV.setText("分¥n 析¥n 中");
0629. dostop();
0630. }
0631. bureList.add(gyroValue);
0632. if (maxInt < -y_SensorValue) {
0633. maxInt = (int) -y_SensorValue;
0634. }
0635. if (minInt > -y_SensorValue) {
0636. minInt = (int) -y_SensorValue;
0637. }
0638. goukei = goukei + Math.abs(y_SensorValue);
0639. x_kosei2 = ((x_kosei) * 10 - (x_button) * 10) / 10;
0640. goukei_x = goukei_x + Math.abs(x_kosei2);
0641. if (jikan == 1) {
0642. unixTime = System.currentTimeMillis();
0643. AccKako = Acc;
0644. } else {
0645. t = System.currentTimeMillis() - unixTime;
0646. unixTime = System.currentTimeMillis();
0647. Acc = -acc[1];
0648. if (t > 10) {
0649. kessonchi = 5;
0650. } else {
0651. if (kessonchi > 0) {
0652. kessonchi--;
0653. if (Acc <= 0.5f && Acc >= -0.5f) {
0654. Spa = AccKako * t;
0655. YlistA.add(Marumeru(AccKako,1));
0656. } else {
0657. Spa = Acc * t;
0658. AccKako = Acc;
0659. YlistA.add(Marumeru(Acc,1));
0660. }
0661. } else {
0662. Spa = Acc * t;
0663. YlistA.add(Marumeru(Acc,1));
0664. }
0665. kakoVel = Vel;
0666. Vel = Vel + Spa;
0667. Ylist.add(Vel);
0668. }
0669. }
0670. saveList.add(username + "," + practiceNo + ","
0671. + System.currentTimeMillis() + "," + acc[0] + "," +
acc[1]
0672. + "," + acc[2] + "," + gyr[0] + "," + gyr[1] + "," +
gyr[2]
0673. + "," + Data + ",");
0674. blurList.add(gyroValue);
0675. speedList.add(postSpeed);
0676. koshiList.add(Float.valueOf(Data));
0677. VgraphView.invalidate();
0678. AgraphView.invalidate();
0679. }
0680. }
0681.
0682. private void dostop() {
0683. isRepeat = false;
0684. pastY = pastYs;
0685. xGraph = 20;
0686. boolean isGraphStart = false;
0687. boolean isGraphZero = false;
0688. float max = 0;
0689. int maxLim = 3;
0690. boolean isMax = false;
0691. float y2 = 0;
0692. float yA2 = 0;
0693. float henkyoku = 0;
0694. int henkyokuten = 0;
0695. float kyokusho = 1000000;
0696. float kyokushoA = 0;
0697. int kyokushoten = 0;
0698. float down = 0;
0699. int zeroten = 0;
0700. for (int i = 0; i < Ylist.size(); i++) {
0701. y2 = Ylist.get(i);
0702. if (max <= y2) {
0703. max = y2;
0704. maxLim = 3;
0705. }
0706. if (!isGraphStart) {
0707. zureInt++;

```

```

0708. }
0709. if (!isGraphStart && y2 >= 250) {
0710. isGraphStart = true;
0711. }
0712. if (isGraphStart && max > y2) {
0713. maxLim--;
0714. }
0715. if (maxLim == 0) {
0716. isMax = true;
0717. }
0718. if (isMax && i > 0) {
0719. if (henkyoku <= y2 - Ylist.get(i - 1)) {
0720. henkyoku = y2 - Ylist.get(i - 1);
0721. henkyokuten = i - 7;
0722. }
0723. }
0724. if (isMax && kyokusho >= y2) {
0725. kyokusho = y2;
0726. kyokushoten = i;
0727. kyokushoA = yA2;
0728. }
0729. if (!isGraphZero && isMax && y2 <= 0) {
0730. isGraphZero = true;
0731. zeroten = i;
0732. }
0733. }
0734. down = kyokushoten - henkyokuten;
0735. if (down < 0) {
0736. down = 7;
0737. }
0738. for (int i = 0; i < Ylist.size(); i++) {
0739. y2 = Ylist.get(i);
0740. yA2 = YlistA.get(i);
0741. if (isGraphZero) {
0742. if (i >= zeroten) {
0743. y2 = 0;
0744. yA2 = 0;
0745. }
0746. } else {
0747. if (i >= kyokushoten) {
0748. y2 = 0;
0749. yA2 = 0;
0750. } else if (i >= henkyokuten) {
0751. y2 = y2 - (((i - henkyokuten) / down) * kyokusho);
0752. yA2 = yA2 + (((i - henkyokuten) / down) * kyokushoA);
0753. }
0754. }
0755. Ylist2.add(y2);
0756. YlistA2.add(yA2);
0757. replayPosition = 0;
0758. model_replayPosition = 0;
0759. }
0760. threads = null;
0761. stop = System.currentTimeMillis();
0762. alltime = (stop - start) / 1000;
0763. if (alltime == 0) {
0764. alltime = 1;
0765. }
0766. // 速度ランクを判定
0767. float speedMax =
Float.parseFloat(Maximum(Ylist2)[0]);
0768. int modelSpeed = (int) Float.parseFloat(logsttr);
0769. speedPoint = (int) ((100 * speedMax) / modelSpeed);
0770. if (speedPoint > 100) {
0771. speedPoint = 100;
0772. }
0773. if (speedPoint < speedBorder1) {
0774. speedRank = C;
0775. speedRankStr = "★☆☆";
0776. } else if (speedPoint >= speedBorder2) {
0777. speedRank = A;
0778. speedRankStr = "★★★";
0779. } else {
0780. speedRank = B;
0781. speedRankStr = "★★☆";
0782. }
0783. // プレ度ランクを判定
0784. String[] bureResult = Maximum(bureList);
0785. float bure = Float.parseFloat(bureResult[0]);
0786. burePoint = 0;
0787. if (bure == 0f) {
0788. burePoint = 100;
0789. } else {
0790. burePoint = (int) (Marumeru(((100 * modelBure) / bure),
0));
0791. }
0792. if (burePoint > 100) {
0793. burePoint = 100;
0794. }
0795. if (burePoint < bureBorder1) {
0796. blurRank = C;
0797. str_rank = "★☆☆";
0798. } else if (burePoint >= bureBorder2) {
0799. blurRank = A;
0800. str_rank = "★★★";
0801. } else {
0802. blurRank = B;
0803. str_rank = "★★☆";
0804. }
0805. saveBure = String.valueOf(bure);
0806. // 速度とプレからアドバイス
0807. if (speedRank == A) {
0808. if (blurRank == A) {
0809. result = "プレもなく、良いスピードで引くことができています。
その調子!";
0810. } else if (blurRank == B) {
0811. result = "良いスピードで引けていますが、少しプレがあります。
もう少し丁寧に引いてみましょう。";
0812. } else if (blurRank == C) {
0813. result = "良いスピードで引けていますが、プレが大きいです。
力任せに引くのではなく、丁寧に心掛けましょう。";
0814. }
0815. } else if (speedRank == B) {
0816. } else if (speedRank == C) {
0817. if (blurRank == A) {
0818. result = "プレことなく丁寧に引けています。
次はそのままもう少し速く引いてみましょう。";
0819. } else if (blurRank == B) {
0820. result = "少しかなががプレしているようです。
丁寧に、かつもう少し速く引いてみましょう。";
0821. } else if (blurRank == C) {
0822. result = "かんなのプレが大きいです。もっと丁寧に、
かつもう少し速く引いてみましょう。";
0823. }
0824. } else if (speedRank == C) {
0825. if (blurRank == A) {
0826. result = "丁寧に引くことができていますが、少し遅いです。
もっと速く引いてみましょう。";
0827. } else if (blurRank == B) {
0828. result = "引く速度が遅いです。少しプレがあるので、
丁寧に忘れないようにしつつ、もっと速く引いてみましょう。";
0829. } else if (blurRank == C) {
0830. result = "もっと速く引いてみましょう。";
0831. }
0832. } else if (blurRank == B) {
0833. result = "引く速度が遅いです。少しプレがあるので、
丁寧に忘れないようにしつつ、もっと速く引いてみましょう。";
0834. }

```

```

0835. } else if (blurRank == C) {
0836.     result = "ブレが大きく、引く速度も遅いです。";
0837.     丁寧さを忘れないようにしつつ、もっと速く引いてみましょう。";
0838. }
0839. }
0840. if (docchi.size() == 2) {
0841.     docchif = (kanna_time - koshi_time) / 1000f;
0842.     if (docchif > 0) {
0843.         docchif = (kanna_time - koshi_time) / 1000f;
0844.         timing = "腰から全身でしっかり引くことができます！\n
ランク：★★★";
0845.         saveKoshi = "KOSHI->TE";
0846.         koshiPoint = 100;
0847.     } else {
0848.         timing = "手元が先に動いています。腰から全身を使って引きま
しょう！\n
ランク：★★☆";
0849.         saveKoshi = "TE->KOSHI";
0850.         koshiPoint = 70;
0851.     }
0852. }
0853. saveTime = String.valueOf(docchif);
0854. } else {
0855.     timing = "上半身だけで引いています。腰から全身で引きましよ
う！\n
ランク：★☆☆";
0856.     saveKoshi = "TE only";
0857.     saveTime = "-";
0858.     koshiPoint = 30;
0859. }
0860. ViewStr = "\n【引きの速さとブレ】" + "\n 師匠を 100 点とす
ると..."
0861. + "\n あなたの引きの速さは" + speedPoint
0862. + "点です。" + "\n ブレの少なさは" + burePoint + "点で
す。"
0863. + "\n 速さランク："
0864. + speedRankStr + "\n ブレランク：" + str_rank + "\n"
+ result
0865. + "\n\n【全身の使い方】" + "\n" + timing;
0866. textV.setTextColor(Color.WHITE);
0867. textV.setText("計測完了");
0868. textV.setText(ViewStr);
0869. textV.setTextSize(17);
0870. textV.setPadding(0, 10, 0, 0);
0871. textV.setGravity(Gravity.LEFT);
0872. startFlag = false;
0873. mp = MediaPlayer.create(this, R.raw.result);
0874. try {
0875.     mp.prepare();
0876. } catch (IllegalStateException e) {
0877.     e.printStackTrace();
0878. } catch (IOException e) {
0879.     e.printStackTrace();
0880. }
0881. mp.start();
0882. mHandler.removeCallbacks(looper);
0883. postToDB();
0884.
0885. // リプレイアニメーションの再生
0886. replayView = new ReplayView(Kanna2Activity.this);
0887. replayBtn2 = new Button(Kanna2Activity.this);
0888. replayBtn2.setText("リプレイ");
0889. replayBtn2.setLayoutParams(new LayoutParams(
0890.     LayoutParams.WRAP_CONTENT,
0891.     LayoutParams.WRAP_CONTENT));
0891. replayBtn2.setOnClickListener(new OnClickListener()
{
0892.     public void onClick(View v) {
0893.         replayPosition = 0;
0894.         model_replayPosition = 0;
0895.         replayer = new Runnable() {
0896.             public void run() {
0897.                 mHandler.post(new Runnable() {
0898.                     public void run() {
0899.                         replayView.invalidate();
0900.                     }
0901.                 });
0902.                 try {
0903.                     Thread.sleep(500);
0904.                 } catch (InterruptedException e) {
0905.                 }
0906.                 for (int i = zureInt; i < Ylist2.size()
0907.                     || modelZureInt + i - zureInt < model_Ylist
0908.                         .size(); i++) {
0909.                     if (i < Ylist2.size()) {
0910.                         replayPosition += Ylist2.get(i);
0911.                     } else {
0912.                         replayPosition += 0;
0913.                     }
0914.                     if (modelZureInt + i - zureInt < model_Ylist
0915.                         .size()) {
0916.                         model_replayPosition += model_Ylist
0917.                             .get(modelZureInt + i - zureInt);
0918.                     } else {
0919.                         model_replayPosition += 0;
0920.                     }
0921.                     try {
0922.                         Thread.sleep(REPEAT_INTERVAL * 5);
0923.                     } catch (InterruptedException e) {
0924.                     }
0925.                     mHandler.post(new Runnable() {
0926.                         public void run() {
0927.                             replayView.invalidate();
0928.                         }
0929.                     });
0930.                 }
0931.             }
0932.         };
0933.         replay = new Thread(replay);
0934.         replay.start();
0935.     }
0936. });
0937. replayDlg = new
AlertDialog.Builder(Kanna2Activity.this);
0938. replayDlg.setView(resultView);
0939. VgraphView = new VgraphView(Kanna2Activity.this);
0940. AgraphView = new AgraphView(Kanna2Activity.this);
0941. TextView rt = (TextView) resultView
.findViewById(R.id.resultText);
0942. LinearLayout g = (LinearLayout) resultView
.findViewById(R.id.resultGraph);
0943. LinearLayout r = (LinearLayout) resultView
.findViewById(R.id.resultReplay);
0944. rt.setText(ViewStr);
0945. rt.setTextSize(17);
0946. rt.setTextColor(Color.WHITE);
0947. LinearLayout.LayoutParams lp = new
LinearLayout.LayoutParams(
0948.     LinearLayout.LayoutParams(
0949.         LinearLayout.LayoutParams.FILL_PARENT,
0950.         LinearLayout.LayoutParams.WRAP_CONTENT);
0951.     lp.weight = 1;
0952.     g.addView(VgraphView, lp);

```

```

0954. g.addView(AgraphView, lp);
0955. r.setGravity(Gravity.CENTER_HORIZONTAL);
0956. r.addView(replayBtn2);
0957. r.addView(replayView);
0958. replayDlg.setCancelable(false);
0959. replayDlg.setNegativeButton("次へ",
0960. new DialogInterface.OnClickListener() {
0961. public void onClick(DialogInterface dialog, int id)
0962. {
0963. webView();
0964. }
0965. });
0966. replayDlg.show();
0967. }
0968. // 速度グラフの描画
0969. private class VgraphView extends View {
0970. public VgraphView(Context context) {
0971. super(context);
0972. setFocusable(true);
0973. }
0974. protected void onSizeChanged(int w, int h, int oldw,
0975. int oldh) {
0976. super.onSizeChanged(w, h, oldw, oldh);
0977. wide = w;
0978. hight = h;
0979. pastYs = h / 2;
0980. pastY = (hight / 4) * 3;
0981. pastYA = (hight / 4) * 2;
0982. if (bmp == null) {
0983. bmp = Bitmap.createBitmap(w, h,
0984. Bitmap.Config.ARGB_4444);
0985. bmp_model = Bitmap.createBitmap(w, h,
0986. Bitmap.Config.ARGB_4444);
0987. bmp_model = BitmapFactory.decodeFile(Environment
0988. .getExternalStorageDirectory().getPath()
0989. + "/" + getPackageName() + "/two/" + modelName +
0990. ".png");
0991. }
0992. bmpCanvas = new Canvas(bmp);
0993. bmpCanvas.drawColor(Color.argb(0, 200, 255, 255));
0994. path.moveTo(20, (hight / 4) * 3);
0995. pathA.moveTo(20, (hight / 4) * 2);
0996. span = (wide-40)/360f;
0997. for (int i = 0; i < Ylist2.size(); i++) {
0998. mYOffset = MoveAverage(i)[0];
0999. mYOffsetA = MoveAverage(i)[1];
1000. int y = (int) (mYOffset);
1001. int yA = (int) (mYOffsetA);
1002. path.quadTo(xGraph, pastY, (xGraph + xGraph + span)
1003. / 2,
1004. (pastY + y) / 2);
1005. pathA.quadTo(xGraph, pastYA, (xGraph + xGraph + span)
1006. / 2,
1007. (pastYA + yA) / 2);
1008. xGraph = xGraph + span;
1009. pastY = y;
1010. pastYA = yA;
1011. }
1012. bmpCanvas.drawLine(20, hight / 2, wide - 20, hight /
1013. 2, paintLine2);
1014. bmpCanvas.drawLine(20, (hight / 4) * 1, wide - 20,
1015. (hight / 4) * 1,
1016. paintLine2);
1017. bmpCanvas.drawLine(20, (hight / 4) * 3, wide - 20,
1018. (hight / 4) * 3,
1019. paintLine2);
1020. bmpCanvas.drawCircle(20 + (koshi_time_int-30)*span,
1021. (hight / 4) * 3, 10,
1022. paintSensor);
1023. bmpCanvas.drawLine(5, hight, wide - 5, hight,
1024. paintBorder);
1025. VgraphView.invalidate();
1026. }
1027. @Override
1028. protected void onDraw(Canvas canvas) {
1029. canvas.drawBitmap(bmp, 0, 0, null);
1030. if (isRepeat) {
1031. bmpCanvas.drawLine(20, hight / 2, wide - 20, hight /
1032. 2,
1033. paintLine2);
1034. bmpCanvas.drawLine(20, (hight / 4) * 1, wide - 20,
1035. (hight / 4) * 1, paintLine2);
1036. bmpCanvas.drawLine(20, (hight / 4) * 3, wide - 20,
1037. (hight / 4) * 3, paintLine2);
1038. if (startFlag == true) {
1039. if (koshiPlot) {
1040. koshiPlot = false;
1041. }
1042. }
1043. bmpCanvas.drawLine(5, hight, wide - 5, hight,
1044. paintBorder);
1045. if (isKannaMove) {
1046. isKannaMove = false;
1047. }
1048. } else {
1049. canvas.drawBitmap(bmp_model,
1050. (zureInt-modelZureInt)*span, 0, null);
1051. paintText.setColor(Color.WHITE);
1052. paintText.setTextSize(35);
1053. canvas.drawText("速度", 10, 55, paintText);
1054. paintText.setColor(Color.CYAN);
1055. paintText.setTextSize(25);
1056. canvas.drawText("あなた", 150, 55, paintText);
1057. paintText.setColor(Color.YELLOW);
1058. paintText.setTextSize(25);
1059. canvas.drawText("師匠", 250, 55, paintText);
1060. }
1061. }
1062. }
1063. // 加速度グラフの描画
1064. private class AgraphView extends View {
1065. public AgraphView(Context context) {
1066. super(context);
1067. setFocusable(true);
1068. }
1069. protected void onSizeChanged(int w, int h, int oldw,
1070. int oldh) {
1071. super.onSizeChanged(w, h, oldw, oldh);
1072. wide = w;
1073. hight = h;
1074. pastYs = h / 2;
1075. pastY = h / 2;
1076. if (bmpA == null) {
1077. bmpA = Bitmap.createBitmap(w, h,
1078. Bitmap.Config.ARGB_4444);
1079. bmp_modelA = Bitmap.createBitmap(w, h,
1080. Bitmap.Config.ARGB_4444);

```

```

1066. bmp_modelA = BitmapFactory.decodeFile(
1067. Environment.getExternalStorageDirectory().getPath(
1068. + "/" + getPackageName() + "/two/" + modelName +
    "A.png");
1069. }
1070. bmpCanvasA = new Canvas(bmpA);
1071. bmpCanvasA.drawColor(Color.argb(0, 200, 255, 255));
1072. bmpCanvasA.drawLine(20, height / 2, width - 20, height
    / 2, paintLine);
1073. bmpCanvasA.drawLine(20, (height / 4) * 1, width - 20,
1074. (height / 4) * 1, paintLine2);
1075. bmpCanvasA.drawLine(20, (height / 4) * 3, width - 20,
1076. (height / 4) * 3, paintLine2);
1077. bmpCanvasA.drawPath(pathA, paintSensor);
1078. bmpCanvasA.drawCircle(20 + (koshi_time_int-30)*span,
    height / 2, 10,
1079. paintSensor);
1080. bmpCanvasA.drawLine(5, 0, width - 5, 0, paintBorder);
1081. AgraphView.invalidate();
1082. }
1083. @Override
1084. protected void onDraw(Canvas canvas) {
1085. canvas.drawBitmap(bmpA, 0, 0, null);
1086. if (isRepeat) {
1087. bmpCanvasA.drawLine(5, 0, width - 5, 0, paintBorder);
1088. bmpCanvasA.drawLine(20, height / 2, width - 20, height
    / 2,
1089. paintLine);
1090. bmpCanvasA.drawLine(20, (height / 4) * 1, width - 20,
1091. (height / 4) * 1, paintLine2);
1092. bmpCanvasA.drawLine(20, (height / 4) * 3, width - 20,
1093. (height / 4) * 3, paintLine2);
1094. } else {
1095. canvas.drawBitmap(bmp_modelA,
    (zureInt-modelZureInt)*span, 0,
1096. null);
1097. paintText.setTextSize(35);
1098. paintText.setColor(Color.WHITE);
1099. canvas.drawText("加速度", 10, 55, paintText);
1100. }
1101. }
1102. }
1103.
1104. @Override
1105. public void onResume() {
1106. super.onResume();
1107. Sensor sensor = sensorManager
1108. .getDefaultSensor(Sensor.TYPE_LINEAR_ACCELERATION)
    ;
1109. Sensor sensor2 =
    sensorManager.getDefaultSensor(Sensor.TYPE_GYROSCO
    PE);
1110. sensorManager.registerListener(this, sensor,
1111. SensorManager.SENSOR_DELAY_FASTEST);
1112. sensorManager.registerListener(this, sensor2,
1113. SensorManager.SENSOR_DELAY_FASTEST);
1114. }
1115.
1116. @Override
1117. public void onPause() {
1118. super.onPause();
1119. sensorManager.unregisterListener(this);
1120. }
1121.
1122. // センサの値を取得
1123. public void onSensorChanged(SensorEvent event) {
1124. switch (event.sensor.getType()) {
1125. case Sensor.TYPE_LINEAR_ACCELERATION:
1126. acc = event.values;
1127. y_sensor = event.values[1];
1128. x_sensor = event.values[0];
1129. postSpeed = SC.setNowSensorValue(-y_sensor);
1130. break;
1131. case Sensor.TYPE_GYROSCOPE:
1132. gyr = event.values;
1133. gyro_value[0] = Math.abs(Marumeru(event.values[0],
    2));
1134. gyro_value[1] = Math.abs(Marumeru(event.values[1],
    2));
1135. gyro_value[2] = Math.abs(Marumeru(event.values[2],
    2));
1136. gyroValue = Marumeru(event.values[1], 2)
    + Marumeru(event.values[2], 2);
1137. break;
1138. }
1139. }
1140. }
1141.
1142. // 学習者データをサーバから取得
1143. public void getuser() {
1144. InputStream in = null;
1145. HttpURLConnection http = null;
1146. try {
1147. URL url = new URL(host + "studentsData.txt");
1148. http = (HttpURLConnection) url.openConnection();
1149. http.setRequestMethod("GET");
1150. http.connect();
1151. in = http.getInputStream();
1152. byte[] data = new byte[4096];
1153. in.read(data);
1154. String src = new String(data);
1155. String[] strAry = src.split("\n");
1156. arrayName = new ArrayList<String>();
1157. arrayGakuban = new ArrayList<String>();
1158. arrayKeyId = new ArrayList<String>();
1159. for (int i = 0; i < strAry.length; i++) {
1160. String[] strAryPerson = strAry[i].split(",");
1161. if (Integer.parseInt(strAryPerson[1]) ==
    userClassNum) {
1162. arrayGakuban.add(strAryPerson[2]);
1163. arrayName.add(strAryPerson[3]);
1164. arrayKeyId.add(strAryPerson[5]);
1165. }
1166. }
1167. } catch (Exception e) {
1168. } finally {
1169. try {
1170. if (http != null)
1171. http.disconnect();
1172. if (in != null)
1173. in.close();
1174. } catch (Exception e) {
1175. }
1176. }
1177. listView();
1178. }
1179.
1180. // 学習者選択処理
1181. public void listView() {
1182. ListView LV = new ListView(this);

```

```

1183. adapter = new ArrayAdapter<String>(this,
1184. android.R.layout.simple_list_item_1, arrayGakuban);
1185. LV.setAdapter(adapter);
1186. LV.setOnItemClickListener(new OnItemClickListener()
1187. {
1188.     public void onItemClick(AdapterView<?> items, View
1189.         view,
1190.         int position, long id) {
1191.             e.putString("username",
1192.                 arrayGakuban.get(position).toString());
1193.             e.putString("userid",
1194.                 arrayKeyId.get(position).toString().trim());
1195.             userId =
1196.                 Integer.valueOf(arrayKeyId.get(position).toString(
1197.                     )
1198.                     .trim());
1199.             e.commit();
1200.             ID = arrayKeyId.get(position).toString().trim();
1201.             username = arrayGakuban.get(position).toString();
1202.             countBtn.setEnabled(true);
1203.             changeUser.setEnabled(true);
1204.             userTextV.setText("学習者:" +
1205.                 arrayName.get(position).toString());
1206.             dialog2.dismiss();
1207.         }
1208.     });
1209.     dialog = new AlertDialog.Builder(this);
1210.     dialog2 = dialog.create();
1211.     dialog2.setCanceledOnTouchOutside(false);
1212.     dialog2.setTitle("出席番号を選択してください");
1213.     dialog2.setView(LV);
1214.     dialog2.show();
1215.     }
1216.     // 練習データをサーバへポスト
1217.     public String postToDB() {
1218.         String speed = "";
1219.         String acc = "";
1220.         String blur = "";
1221.         String koshi = "";
1222.         for (int i = 0; i < Ylist2.size(); i++) {
1223.             speed += Math.round(Ylist2.get(i));
1224.             if (i != Ylist2.size() - 1) {
1225.                 speed += ",";
1226.             }
1227.         }
1228.         for (int i = 0; i < YlistA2.size(); i++) {
1229.             acc += Math.round(YlistA2.get(i));
1230.             if (i != YlistA2.size() - 1) {
1231.                 acc += ",";
1232.             }
1233.         }
1234.         for (int i = 0; i < blurList.size(); i++) {
1235.             blur += Math.round(blurList.get(i));
1236.             if (i != blurList.size() - 1) {
1237.                 blur += ",";
1238.             }
1239.         }
1240.         String s = host + "plane/senddata.php?ID=" + ID
1241.             + "&speed=" + speed
1242.             + "&acc=" + acc
1243.             + "&gyro=" + blur
1244.             + "&koshi=" + koshi
1245.             + "&gap=" + zureInt
1246.             + "&accPoint=" + speedPoint
1247.             + "&gyroPoint=" + burePoint
1248.             + "&koshiPoint=" + koshiPoint
1249.             + "&totalPoint=" + (speedPoint + burePoint +
1250.                 koshiPoint)/3;
1251.         HttpGet method = new HttpGet(s.toString());
1252.         try {
1253.             DefaultHttpClient client = new DefaultHttpClient();
1254.             HttpResponse response = client.execute(method);
1255.             EntityUtils.toString(response.getEntity(),
1256.                 "UTF-8");
1257.             return EntityUtils.toString(response.getEntity(),
1258.                 "UTF-8");
1259.         } catch (Exception e) {
1260.             return null;
1261.         }
1262.         // Bluetooth 接続処理(サーバ側)
1263.         private class ServerThread extends Thread {
1264.             private final BluetoothServerSocket mmServerSocket;
1265.             public ServerThread() {
1266.                 BluetoothServerSocket tmp = null;
1267.                 try {
1268.                     tmp = mAdapter
1269.                         .listenUsingRfcommWithServiceRecord(NAME, MY_UUID);
1270.                 } catch (IOException e) {
1271.                 }
1272.                 mmServerSocket = tmp;
1273.             }
1274.             @Override
1275.             public void run() {
1276.                 while (true) {
1277.                     try {
1278.                         socket[0] = mmServerSocket.accept();
1279.                     } catch (IOException e) {
1280.                     }
1281.                     if (socket[0] != null) {
1282.                         manageConnectedSocket(socket[0]);
1283.                         Message msg = new Message();
1284.                         msg.obj = "";
1285.                         toastHandler.sendMessage(msg);
1286.                         connection[0].start();
1287.                     }
1288.                     break;
1289.                 }
1290.             }
1291.             @SuppressWarnings("unused")
1292.             public void cancel() {
1293.                 try {
1294.                     mmServerSocket.close();
1295.                 } catch (IOException e) {
1296.                 }
1297.             }
1298.         }
1299.         // Bluetooth 接続確立時の処理
1300.         private class ConnectedThread extends Thread {

```

```

1303. private final BluetoothSocket mmSocket;
1304. private final InputStream mmInStream;
1305. private final OutputStream mmOutStream;
1306. String[] strAry = new String[5];
1307. public ConnectedThread(BluetoothSocket socket) {
1308.     mmSocket = socket;
1309.     InputStream tmpIn = null;
1310.     OutputStream tmpOut = null;
1311.     try {
1312.         tmpIn = socket.getInputStream();
1313.         tmpOut = socket.getOutputStream();
1314.     } catch (IOException e) {
1315.     }
1316.     mmInStream = tmpIn;
1317.     mmOutStream = tmpOut;
1318.     }
1319.     @Override
1320.     public void run() {
1321.         byte[] buffer = new byte[1024];
1322.         @SuppressWarnings("unused")
1323.         int bytes;
1324.         while (true) {
1325.             try {
1326.                 bytes = mmInStream.read(buffer);
1327.                 String readMsg = new String(buffer, "UTF-8");
1328.                 Message msg = new Message();
1329.                 msg.obj = readMsg;
1330.                 strAry = readMsg.split("xxx");
1331.                 Data = strAry[0];
1332.             } catch (IOException e) {
1333.                 break;
1334.             }
1335.         }
1336.     }
1337.     @SuppressWarnings("unused")
1338.     public void write(byte[] bytes) {
1339.         try {
1340.             mmOutStream.write(bytes);
1341.         } catch (IOException e) {
1342.         }
1343.     }
1344.     @SuppressWarnings("unused")
1345.     public void cancel() {
1346.         try {
1347.             mmSocket.close();
1348.         } catch (IOException e) {
1349.         }
1350.     }
1351.     }
1352.
1353.     public void manageConnectedSocket(BluetoothSocket
socket) {
1354.         connection[0] = new ConnectedThread(socket);
1355.     }
1356.     final Handler toastHandler = new Handler() {
1357.         public void handleMessage(Message msg) {
1358.             Toast.makeText(Kanna2Activity.this, msg.obj + "接続
完了",
1359.                 Toast.LENGTH_SHORT).show();
1360.             classView();
1361.         }
1362.     };
1363.
1364.     // Back ボタン押下時の処理
1365.     public boolean onKeyDown(int keyCode, KeyEvent event)
{
1366.         if (keyCode == KeyEvent.KEYCODE_BACK) {
1367.             System.exit(0);
1368.         } else if (keyCode == KeyEvent.KEYCODE_MENU) {
1369.             openOptionsMenu();
1370.         }
1371.         return true;
1372.     }
1373.
1374.     // 四捨五入
1375.     public float Marumeru(float f, int i) {
1376.         BigDecimal xbi = new BigDecimal(String.valueOf(f));
1377.         float result = xbi.setScale(i,
BigDecimal.ROUND_HALF_UP).floatValue();
1378.         return result;
1379.     }
1380.
1381.     // 最大値取得
1382.     public String[] Maximum(ArrayList<Float> list) {
1383.         String[] result = new String[3];
1384.         float f = 0f;
1385.         int index = 0;
1386.         String when = "";
1387.         for (int i = 0; i < list.size(); i++) {
1388.             if (f < list.get(i)) {
1389.                 f = list.get(i);
1390.                 index = i;
1391.             }
1392.         }
1393.         result[0] = String.valueOf(f);
1394.         result[1] = String.valueOf(index);
1395.         return result;
1396.     }
1397.
1398.     // 移動平均算出
1399.     public float[] MoveAverage(int i) {
1400.         float[] result = new float[2];
1401.         float Vsum = 0;
1402.         float Asum = 0;
1403.         if (i > 10 && i < Ylist2.size() - 10) {
1404.             for (int j = -5; j < 5; j++) {
1405.                 Vsum += Ylist2.get(i + j);
1406.             }
1407.             for (int j = -10; j < 10; j++) {
1408.                 Asum += YlistA2.get(i + j);
1409.             }
1410.             result[0] = (hight / 4) * 3 + (-Vsum / 10) / 10;
1411.             result[1] = hight / 2 + (-Asum / 20) * 7;
1412.         } else {
1413.             result[0] = (hight / 4) * 3 + (-Ylist2.get(i)) / 10;
1414.             result[1] = hight / 2 + (-YlistA2.get(i)) * 7;
1415.         }
1416.         return result;
1417.     }
1418.
1419.     // リブレイアアニメーション描画
1420.     public class ReplayView extends View {
1421.         public ReplayView(Context context) {
1422.             super(context);
1423.         }
1424.         protected void onSizeChanged(int w, int h, int oldw,
int oldh) {
1425.             super.onSizeChanged(w, h, oldw, oldh);
1426.             replayHeight = h;
1427.             replayWidth = w;

```

```

1428. replayRight = 1 * (replayWidth / 4);
1429. replayCenter = 2 * (replayWidth / 4);
1430. replayLeft = 3 * (replayWidth / 4);
1431. replayRatio = (replayHeight - 50) / replayLength;
1432. }
1433. @Override
1434. protected void onDraw(Canvas canvas) {
1435. canvas.drawColor(Color.WHITE);
1436. canvas.drawBitmap(kanna, replayRight,
replayPosition * replayRatio,
1437. null);
1438. canvas.drawBitmap(kanna, replayLeft,
model_replayPosition
1439. * replayRatio, null);
1440. paintText.setTextSize(35);
1441. paintText.setColor(Color.BLACK);
1442. canvas.drawText("あなた", replayRight, 100,
paintText);
1443. canvas.drawText("師匠", replayLeft, 100, paintText);
1444. }
1445. }
1446.
1447. // 上達履歴表示 View
1448. public void webView() {
1449. final WebView webView = new WebView(this);
1450. webView.getSettings().setJavaScriptEnabled(true);
1451. webView.loadUrl(host +
"plane/growthRecordMobile.php?ID=" + userId);
1452. AlertDialog.Builder dlg = new
AlertDialog.Builder(this);
1453. AlertDialog.Builder dialog2 = new
AlertDialog.Builder(this);
1454. dialog2.setTitle("成績表示");
1455. dialog2.setView(webView);
1456. dialog2.setPositiveButton(" もう一度 ",
1457. new DialogInterface.OnClickListener() {
1458. public void onClick(DialogInterface dialog, int id)
{
1459. intCount = 4;
1460. jikan = 0;
1461. koshi_time_int = 0;
1462. zureInt = 1;
1463. pastY = pastYs;
1464. pastYA = pastYs;
1465. xGraph = 20;
1466. countBtn.setVisibility(View.VISIBLE);
1467. changeUser.setVisibility(View.VISIBLE);
1468. Paint p = new Paint();
1469. p.setColor(Color.WHITE);
1470. if (bmpCanvas != null && bmpCanvasA != null) {
1471. bmpCanvas.drawColor(Color.TRANSPARENT,
1472. PorterDuff.Mode.CLEAR);
1473. bmpCanvasA.drawColor(Color.TRANSPARENT,
1474. PorterDuff.Mode.CLEAR);
1475. }
1476. countTextV.setText("↓");
1477. countTextV.setTextSize(250);
1478. divflag_koshi = false;
1479. divflag_kanna = false;
1480. currentlyInt_kanna = new int[10];
1481. currently_koshi = new float[10];
1482. docchi.clear();
1483. bureList.clear();
1484. speedList.clear();
1485. blurList.clear();
1486. koshiList.clear();
1487. saveList.clear();
1488. unixTime = 0;
1489. t = 0;
1490. y_hosei = 0;
1491. z_hosei = 0;
1492. Acc = 0;
1493. AccKako = 0;
1494. Spa = 0;
1495. Vel = 0;
1496. kakoVel = 0;
1497. kessonchi = 0;
1498. graphDownInt = graphDownReset;
1499. path = new Path();
1500. pathA = new Path();
1501. XYnum = 0;
1502. Ylist.clear();
1503. Ylist2.clear();
1504. YlistA.clear();
1505. YlistA2.clear();
1506. }
1507. });
1508. dialog2.setNegativeButton("学習者の変更",
1509. new DialogInterface.OnClickListener() {
1510. public void onClick(DialogInterface dialog, int id){
1511. intCount = 4;
1512. jikan = 0;
1513. koshi_time_int = 0;
1514. zureInt = 1;
1515. pastY = pastYs;
1516. pastYA = pastYs;
1517. xGraph = 20;
1518. countBtn.setVisibility(View.VISIBLE);
1519. changeUser.setVisibility(View.VISIBLE);
1520. Paint p = new Paint();
1521. p.setColor(Color.WHITE);
1522. if (bmpCanvas != null && bmpCanvasA != null) {
1523. bmpCanvas.drawColor(Color.TRANSPARENT,
1524. PorterDuff.Mode.CLEAR);
1525. bmpCanvasA.drawColor(Color.TRANSPARENT,
1526. PorterDuff.Mode.CLEAR);
1527. }
1528. countTextV.setText("↓");
1529. countTextV.setTextSize(250);
1530. divflag_koshi = false;
1531. divflag_kanna = false;
1532. currentlyInt_kanna = new int[10];
1533. currently_koshi = new float[10];
1534. docchi.clear();
1535. bureList.clear();
1536. speedList.clear();
1537. blurList.clear();
1538. koshiList.clear();
1539. saveList.clear();
1540. unixTime = 0;
1541. t = 0;
1542. y_hosei = 0;
1543. z_hosei = 0;
1544. Acc = 0;
1545. AccKako = 0;
1546. Spa = 0;
1547. Vel = 0;
1548. kakoVel = 0;
1549. kessonchi = 0;
1550. graphDownInt = graphDownReset;

```



```

1551. path = new Path();
1552. pathA = new Path();
1553. XYnum = 0;
1554. Ylist.clear();
1555. Ylist2.clear();
1556. YlistA.clear();
1557. YlistA2.clear();

```

```

1558. getuser();
1559. }
1560. });
1561. dialog2.show();
1562. }
1563. }

```

## KoshiActivity.java (腰端末として指定した際に呼び出されるプログラム)

```

0001. package jp.anlab.kanna2PMS;
0002. import java.io.IOException;
0003. import java.io.InputStream;
0004. import java.io.OutputStream;
0005. import java.math.BigDecimal;
0006. import java.util.List;
0007. import java.util.Set;
0008. import java.util.UUID;
0009. import java.util.Vector;
0010. import android.app.Activity;
0011. import android.bluetooth.BluetoothAdapter;
0012. import android.bluetooth.BluetoothDevice;
0013. import android.bluetooth.BluetoothSocket;
0014. import android.content.Context;
0015. import android.hardware.Sensor;
0016. import android.hardware.SensorEvent;
0017. import android.hardware.SensorEventListener;
0018. import android.hardware.SensorManager;
0019. import android.os.Bundle;
0020. import android.os.Handler;
0021. import android.os.Message;
0022. import android.os.Vibrator;
0023. import android.util.Log;
0024. import android.view.KeyEvent;
0025. import android.widget.TextView;
0026. import android.widget.Toast;
0027.
0028. public class KoshiActivity extends Activity {
0029.     BluetoothAdapter mAdapter;
0030.     boolean send = false;
0031.     boolean ss = false;
0032.     boolean start = false;
0033.     ClientThread clientThread;
0034.     ConnectedThread connection;
0035.     Context mContext;
0036.     final String NAME = "BLUETOOTH_SAMPLE";
0037.     private float[] accelerometerValues = new float[3];
0038.     private float[] gyroscopeValues = new float[3];
0039.     private float[] linearaccelerationValues = new
float[3];
0040.     private float[] orientationValues = new float[3];
0041.     private float[] rotationvectorValues = new float[3];
0042.     private Runnable looper;
0043.     private Sensor accelerometerSensor;
0044.     private Sensor gyroscopeSensor;
0045.     private Sensor linearaccelerationSensor;
0046.     private Sensor orientationSensor;
0047.     private Sensor rotationvectorSensor;
0048.     private static final String TAG = "BLUETOOTH_SAMPLE";
0049.     private String accelerometerText;
0050.     private String gyroscopeText;
0051.     private String linearaccelerationText;
0052.     private String orientationText;
0053.     private String rotationvectorText;
0054.     private String sendText;
0055.     SensorManager sm;
0056.     String values;
0057.     TextView clientTextView;
0058.     TextView koshiText;
0059.     Thread thread;
0060.     UUID MY_UUID =
UUID.fromString("11111111-0000-1000-1111-00AAEECCAA
FF");
0061.     Vector<BluetoothDevice> mDevices;
0062.
0063.     private SensorEventListener listener = new
SensorEventListener() {
0064.     public void onAccuracyChanged(Sensor sensor, int
accuracy) {
0065.     }
0066.
0067.     // センサの値取得
0068.     public void onSensorChanged(SensorEvent event) {
0069.     if (event.sensor.getType() ==
Sensor.TYPE_LINEAR_ACCELERATION) {
0070.     linearaccelerationValues = event.values;
0071.     linearaccelerationText =
String.valueOf(Math.abs(Math.abs(linearacceleratio
nValues[0])
0072.     +
Math.round(Math.abs(linearaccelerationValues[2])),
1));
0073.
0074.     }
0075.     sendText = linearaccelerationText + "xxx";
0076.     values = sendText;
0077.     }
0078.     };
0079.
0080.     @Override
0081.     public void onCreate(Bundle savedInstanceState) {
0082.     super.onCreate(savedInstanceState);
0083.     setContentView(R.layout.koshi);
0084.     koshiText = (TextView) findViewById(R.id.koshiText);
0085.     sm = (SensorManager)
getSystemService(SENSOR_SERVICE);
0086.     List<Sensor> sensorList;
0087.     sensorList =
sm.getSensorList(Sensor.TYPE_LINEAR_ACCELERATION);
0088.     linearaccelerationSensor = sensorList.get(0);
0089.     sm.registerListener(listener,
linearaccelerationSensor,
SensorManager.SENSOR_DELAY_GAME);
0090.     mDevices = new Vector<BluetoothDevice>();
0091.     mAdapter = BluetoothAdapter.getDefaultAdapter();
0092.     Set<BluetoothDevice> devices =
mAdapter.getBondedDevices();

```

```

0094. for (BluetoothDevice device : devices) {
0095.     mDevices.add(device);
0096.     Toast.makeText(this, "Name:" + device.getName(),
0097.         Toast.LENGTH_SHORT).show();
0098.     }
0099.     }
0100.     for (int i = 0; i < mDevices.size(); i++) {
0101.         clientThread = new
0102.             ClientThread(mDevices.elementAt(i));
0103.         }
0104.         looper = new Runnable() {
0105.             public void run() {
0106.                 while (true) {
0107.                     try {
0108.                         Thread.sleep(50);
0109.                     } catch (InterruptedException e) {
0110.                         }
0111.                     if (start) {
0112.                         if (values != null && !values.equals("")) {
0113.                             connection.write(values.getBytes());
0114.                         }
0115.                     }
0116.                 }
0117.             }
0118.         };
0119.     }
0120.
0121.     // Bluetooth 接続処理(クライアント側)
0122.     private class ClientThread extends Thread {
0123.         private final BluetoothSocket mmSocket;
0124.         @SuppressWarnings("unused")
0125.         private final BluetoothDevice mmDevice;
0126.         public ClientThread(BluetoothDevice device) {
0127.             BluetoothSocket tmp = null;
0128.             mmDevice = device;
0129.             try {
0130.                 tmp =
0131.                     device.createRfcommSocketToServiceRecord(MY_UUID);
0132.             }
0133.             mmSocket = tmp;
0134.         }
0135.         @Override
0136.         public void run() {
0137.             mAdapter.cancelDiscovery();
0138.             try {
0139.                 mmSocket.connect();
0140.             } catch (IOException connectException) {
0141.                 try {
0142.                     mmSocket.close();
0143.                 } catch (IOException closeException) {
0144.                     }
0145.                 return;
0146.             }
0147.             manageConnectedSocket(mmSocket);
0148.             Message msg = new Message();
0149.             msg.obj = "koshi";
0150.             mHandler.sendMessage(msg);
0151.             start = true;
0152.             thread = new Thread(looper);
0153.             thread.start();
0154.         }
0155.         @SuppressWarnings("unused")
0156.         public void cancel() {
0157.             try {
0158.                 mmSocket.close();
0159.             } catch (IOException e) {
0160.                 }
0161.             }
0162.         }
0163.         public void manageConnectedSocket(BluetoothSocket
0164.             socket) {
0165.             connection = new ConnectedThread(socket);
0166.             connection.start();
0167.         }
0168.         // Bluetooth 接続確立時の処理
0169.         private class ConnectedThread extends Thread {
0170.             private final BluetoothSocket mmSocket;
0171.             private final InputStream mmInputStream;
0172.             private final OutputStream mmOutputStream;
0173.             String[] strAry = new String[5];
0174.             long[] longAry = new long[5];
0175.             public ConnectedThread(BluetoothSocket socket) {
0176.                 mmSocket = socket;
0177.                 InputStream tmpIn = null;
0178.                 OutputStream tmpOut = null;
0179.                 try {
0180.                     tmpIn = socket.getInputStream();
0181.                     tmpOut = socket.getOutputStream();
0182.                 } catch (IOException e) {
0183.                     }
0184.                 mmInputStream = tmpIn;
0185.                 mmOutputStream = tmpOut;
0186.             }
0187.             @Override
0188.             public void run() {
0189.                 byte[] buffer = new byte[1024];
0190.                 @SuppressWarnings("unused")
0191.                 int bytes;
0192.                 while (true) {
0193.                     try {
0194.                         bytes = mmInputStream.read(buffer);
0195.                         String readMsg = new String(buffer, "UTF-8");
0196.                         Message msg = new Message();
0197.                         msg.obj = readMsg;
0198.                         strAry = readMsg.split("xxx");
0199.                         for(int i=0; i<longAry.length; i++){
0200.                             longAry[i] = Long.parseLong(strAry[i]);
0201.                         }
0202.                         vibration(longAry);
0203.                         mHandler.sendMessage(msg);
0204.                     } catch (IOException e) {
0205.                         break;
0206.                     }
0207.                 }
0208.             }
0209.             public void write(byte[] bytes) {
0210.                 try {
0211.                     mmOutputStream.write(bytes);
0212.                 } catch (IOException e) {
0213.                     }
0214.             }
0215.             @SuppressWarnings("unused")
0216.             public void cancel() {
0217.                 try {
0218.                     mmSocket.close();
0219.                 } catch (IOException e) {
0220.                     }

```

```

0221. }
0222. }
0223. public void vibration(long[] longAry){
0224. Vibrator vibrator =
    Vibrator.getSystemService(Context.VIBRATOR_SERVICE
    );
0225. vibrator.vibrate(longAry, 1);
0226. }
0227. }
0228. final Handler mHandler = new Handler() {
0229. public void handleMessage(Message msg) {
0230. }
0231. };
0232. final Handler mHandler1 = new Handler() {
0233. public void handleMessage(Message msg) {
0234. koshiText.setText("腰付近のポケットに¥n入れてください
    ");
0235. };
0236.

0237. // 四捨五入
0238. public float Marumeru(float f, int i) {
0239. BigDecimal xbi = new BigDecimal(String.valueOf(f));
0240. float result = xbi.setScale(i,
    BigDecimal.ROUND_HALF_UP).floatValue();
0241. return result;
0242. }
0243.
0244. // Back ボタン押下時の処理
0245. public boolean onKeyDown(int keyCode, KeyEvent event)
    {
0246. if (keyCode == KeyEvent.KEYCODE_BACK) {
0247. System.exit(0);
0248. } else if (keyCode == KeyEvent.KEYCODE_MENU) {
0249. openOptionsMenu();
0250. }
0251. return true;
0252. }
0253. }

```

## 【PLMS】

### graphview.php (グラフ表示)

```

0001. <?php
0002. date_default_timezone_set ( 'Asia/Tokyo' );
0003. session_start ();
0004.
0005. // ログインのチェック
0006. if (! isset ( $_SESSION ["user_name"] )) {
0007. $no_login_url = "../login.php";
0008. header ( "Location: {$no_login_url}" );
0009. exit ();
0010. }
0011.
0012. function checkSelected($var, $selected_value) {
0013. if ($var == $selected_value) {
0014. return ' selected="selected"';
0015. }
0016. return '';
0017. }
0018.
0019. // 並び替え用の配列
0020. $sort_ary = array (
0021. "5" => "日時降順",
0022. "1" => "速度が速い順",
0023. "2" => "速度が遅い順",
0024. "3" => "ブレ値が大きい順",
0025. "4" => "ブレ値が小さい順"
0026. );
0027.
0028. // 移動平均を算出
0029. function moveAverage($data) {
0030. $sum = 0;
0031. $n = 0;
0032. for($i = 0; $i < count ( $data ); $i ++ ) {
0033. $sum = ($sum) + ($data [$i]);
0034. $n ++;
0035. }
0036. $result = ($sum) / ($n);
0037. return $result;

0038. }
0039.
0040. // グラフ得点表示用のデータ整形
0041. function mathGraphData($userData, $gap) {
0042. $var = array ();
0043. $var = explode ( ",", $userData );
0044. for($i = $gap; $i < count ( $var ) + $gap; $i = $i +
    5) {
0045. $returnData = $returnData . "," . (($var [($i - 2)]
    + $var [($i - 1)] + $var
    0046. [$i] + $var [($i + 1)]
    + $var [($i + 2)]) / 5);
0047. }
0048. return $returnData;
0049. }
0050.
0051. // ブレ値グラフの表示整形
0052. function mathGraphGyroData($userData, $gap) {
0053. $var = array ();
0054. $var = explode ( ",", $userData );
0055. for($i = $gap; $i < count ( $var ) + $gap; $i = $i +
    5) {
0056. $returnData = $returnData . "," . ($var [$i] * 2);
0057. }
0058. return $returnData;
0059. }
0060.
0061. $class_list = $_GET ['class_list'];
0062. $kikan = $_GET ['kikan'];
0063. $sort_num = $_GET ['sort_num'];
0064. $search_name = $_GET ['search_name'];
0065. $view = $_GET ['view'];
0066. $p = $_GET ['p'];
0067. $time = time ();
0068. $ID_ar = array ();
0069. $uTime_ar = array ();
0070. $speed_ar = array ();

```

```

0071. $acc_ar = array ();
0072. $gyro_ar = array ();
0073. $koshi_ar = array ();
0074. $gap_ar = array ();
0075. $accPoint_ar = array ();
0076. $gyroPoint_ar = array ();
0077. $koshiPoint_ar = array ();
0078. $totalPoint_ar = array ();
0079. $year_ar = array ();
0080. $class_ar = array ();
0081. $number_ar = array ();
0082. $sex_ar = array ();
0083. $name_ar = array ();
0084. $modelID = "";
0085. $modelSpeed = "";
0086. $modelAcc = "";
0087. $modelGyro = "";
0088. $i = 0;
0089. $n = 0;
0090.
0091. // データベースとのやり取りの準備
0092. try {
0093. $pdo = new PDO ( "sqlite://elearning.db" );
0094. $pdo->setAttribute ( PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION );
0095. chmod ( "elearning.db", 0777 );
0096. if ( ! $pdo ) {
0097. die ( '接続失敗です。' . $sqliteerror );
0098. } else {
0099. $isUsertb = false;
0100. $isDatatb = false;
0101. $isTypetb = false;
0102. print ( '接続に成功しました。<br>' );
0103. }
0104.
0105. // SQL 文の生成
0106. $sql_query1 = "SELECT plainTb.*, userTb.year,
userTb.class, userTb.number,
0107. userTb.sex, nameTb.name FROM plainTb, userTb, nameTb
ON
0108. plainTb.ID=userTb.ID and plainTb.ID=nameTb.ID";
0109. $sql_query2 = "SELECT count (*) FROM plainTb , userTb
ON userTb.ID=plainTb.ID";
0110. if ( $kikan != null && $kikan != "null" ) {
0111. if ( $kikan == 1 ) {
0112. $viewtime = $time - 86400;
0113. $sql_query = $sql_query . " where uTime > $viewtime";
0114. } else if ( $kikan == 2 ) {
0115. $viewtime = $time - 604800;
0116. $sql_query = $sql_query . " where uTime > $viewtime";
0117. } else if ( $kikan == 3 ) {
0118. $viewtime = $time - 2419200;
0119. $sql_query = $sql_query . " where uTime > $viewtime";
0120. }
0121. }
0122. if ( $search_name != "" ) {
0123. foreach ( $pdo->query (
0124. "SELECT * FROM nameTb WHERE name LIKE
'%$search_name%' ) as $row){
0125. $viewuserid = $row ["ID"];
0126. }
0127. $sql_query = $sql_query . " and userTb.ID =
¥$viewuserid¥";
0128. }
0129. if ( 0 < $class_list && $class_list < 5 ) {
0130. $sql_query = $sql_query . " and class =
¥$class_list¥";
0131. }
0132. if ( 0 < $sort_num && $sort_num < 6 ) {
0133. if ( $sort_num == 1 ) {
0134. $sql_query = $sql_query . " ORDER BY accPoint DESC";
0135. } else if ( $sort_num == 2 ) {
0136. $sql_query = $sql_query . " ORDER BY accPoint ASC";
0137. } else if ( $sort_num == 3 ) {
0138. $sql_query = $sql_query . " ORDER BY gyroPoint ASC";
0139. } else if ( $sort_num == 4 ) {
0140. $sql_query = $sql_query . " ORDER BY gyroPoint DESC";
0141. } else if ( $sort_num == 5 ) {
0142. $sql_query = $sql_query . " ORDER BY uTime ASC";
0143. }
0144. } else {
0145. $sql_query = $sql_query . " ORDER BY uTime DESC";
0146. }
0147. $sql_query1 = $sql_query1 . $sql_query;
0148. $sql_query2 = $sql_query2 . $sql_query;
0149. if ( $p > 0 ) {
0150. $offset = $p * 10;
0151. } else {
0152. $offset = 0;
0153. }
0154. $sql_query1 = $sql_query1 . " LIMIT 10 OFFSET $offset";
0155. $countSQL = $pdo->query ( $sql_query2 )->fetchAll ();
0156. $maxpage = floor ( $countSQL [0] [0] / 10 );
0157. ?>
0158.
0159. <html lang="ja">
0160. <head>
0161. <title>グラフ表示</title>
0162. <link rel="stylesheet" href="default.css"
type="text/css" />
0163. <script
0164. src="http://ajax.googleapis.com/ajax/libs/jquery/1
.8.2/jquery.min.js"></script>
0165. <script type="text/javascript"
src="js/jquery.zoomooz.min.js">
</script>
0166. <script type="text/javascript">
0167. $(document).ready(function() {
0168. $(".zoomTarget").zoomTarget();
0169. });
0170. </script>
0171. <script src="./js/confirm.js"></script>
0172. <meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
0173. <meta http-equiv="Content-Language" content="ja" />
0174. <meta http-equiv="Content-Style-Type"
content="text/css" />
0175. <meta http-equiv="Content-Script-Type"
content="text/javascript" />
0176. <script src="Chart.js"></script>
0177. <meta name="viewport" content="initial-scale = 1,
user-scalable = no">
0178. <style>
0179. canvas {
0180.
0181. }
0182. }
0183. </style>
0184. </head>
0185. <body>
0186. <div id="main">

```

```

0187. <div id="top">
0188. <FORM method="get" action="graphview.php">
0189. <table align='center' valign='middle' class='top'>
0190. <TR>
0191. <TH>クラス</TH>
0192. <TH>Sort by</TH>
0193. <TH>期間</TH>
0194. <TH>人名検索</TH>
0195. </TR>
0196. <TR>
0197. <TH><select name=class_list>
0198. <option value="null">すべて</option>
0199. <?php
0200. for($classid = 1; $classid < 5; $classid++){
0201. echo "<option value=¥\"{$classid}¥\" .
0202. checkSelected ( $classid, $class_list )
0203. . ">{$classid}</option>¥n";
0204. }
0205. ?>
0206. </select></TH>
0207. <TH><select name=sort_num>
0208. <option value="null">日時昇順</option>
0209. <?php
0210. foreach ( $sort_ary as $key => $value ) {
0211. echo "<option value=¥\"{$key}¥\" .
0212. checkSelected ( $sort_num, $key ) .
0213. ">{$value}</option>¥n";
0214. }
0215. ?>
0216. </select></TH>
0217. <TH><select name=kikan>
0218. <option value="null">指定しない</option>
0219. <option value="1"
0220. <?php
0221. if ($kikan == 1) {
0222. echo "selected";
0223. }
0224. ?>
0225. >24 時間以内</option>
0226. <option value="2"
0227. <?php
0228. if ($kikan == 2) {
0229. echo "selected";
0230. }
0231. ?>
0232. >一週間以内</option>
0233. <option value="3"
0234. <?php
0235. if ($kikan == 3) {
0236. echo "selected";
0237. }
0238. ?>
0239. >一ヶ月以内</option>
0240. </select></TH>
0241. <TH><INPUT TYPE="text" NAME="search_name" size="8"
0242. <?php
0243. if ($search_name != "") {
0244. echo "value=$search_name";
0245. }
0246. ?>
0247. ></TH>
0248. <TH><INPUT type="submit" value="決定"></TH>
0249. </form>
0250. <form method="get" action="graphview.php">
0251. <TH><INPUT type="submit" value="リセット"></TH>
0252. </form>
0253. <td>
0254. <p><div class='buttonDiv'>
0255. <BR> <a HREF="listview.php">リスト表示</a>
0256. </div></p>
0257. </td>
0258. <td>
0259. <p><div class='buttonDiv'>
0260. <BR> <a HREF="..">PLMS トップへ</a>
0261. </div></p>
0262. </td>
0263. </TR>
0264. </Table>
0265. </div>
0266. <?php
0267. // y 軸時間の設定
0268. $yAxis = "'', '', '', '', '', '', '', '', '', '',
0269. '', '', '', '', '', '', '', '', '', '', '',
0270. '', '', '', '', '', '', '', '', '', '', '',
0271. '', '', '', '', '', '', '', '', '', '', '',
0272. '', '', '', '', '', '', '', '', '', '', '',
0273. '', '', '', '', '', '', '', '', '', '', '',
0274. '', '', '', '', '', '', '', '', '', '',
0275. '', '', '', '', '', '', '',
0276. // データベースから値取得
0277. foreach ( $pdo->query ( "SELECT modelID FROM
0278. plainSetTb" ) as $row ) {
0279. $modelID = $row ["modelID"];
0280. }
0281. foreach ( $pdo->query ( "SELECT * FROM plainModelTb
0282. WHERE modelID = '$modelID'" )
0283. as $row ) {
0284. $modelSpeed = $row ["speed"];
0285. $modelAcc = $row ["acc"];
0286. $modelGyro = $row ["gyro"];
0287. $modelGap = $row ["gap"];
0288. }
0289. foreach ( $pdo->query ( $sql_query1 ) as $row ) {
0290. array_push ( $ID_ar, $row ["ID"] );
0291. array_push ( $uTime_ar, $row ["uTime"] );
0292. array_push ( $speed_ar, $row ["speed"] );
0293. array_push ( $acc_ar, $row ["acc"] );
0294. array_push ( $gyro_ar, $row ["gyro"] );
0295. array_push ( $koshi_ar, $row ["koshi"] );
0296. array_push ( $gap_ar, $row ["gap"] );
0297. array_push ( $accPoint_ar, $row ["accPoint"] );
0298. array_push ( $gyroPoint_ar, $row ["gyroPoint"] );
0299. array_push ( $koshiPoint_ar, $row ["koshiPoint"] );
0300. array_push ( $totalPoint_ar, $row ["totalPoint"] );
0301. array_push ( $year_ar, $row ["year"] );
0302. array_push ( $class_ar, $row ["class"] );
0303. array_push ( $number_ar, $row ["number"] );
0304. array_push ( $sex_ar, $row ["sex"] );
0305. array_push ( $name_ar, $row ["name"] );
0306. $n ++;
0307. }
0308. // グラフの描画
0309. $rader = "";
0310. $line = array ();
0311. $line2 = array ();
0312. $gyr = array ();
0313. for($j = 0; $j < $n; $j ++) {
0314. $datastr = "";
0315. $databurestr = "";
0316. $data_gyro = explode ( ",", $gyro_ar [$j] );
0317. for($count = 0; $count < count ( $data_gyro ) - 1;

```

```

0310.     $count ++) {
0311.     $databurestr = $databurestr . (($data_gyro [$count -
0312.     1] + $data_gyro [$count] + $data_gyro [$count + 1])
0313.     * 100 / 3) . ",";
0314.     }
0315.     $databurestr = $databurestr . abs ( $data_gyro [count
0316.     ( $data_gyro ) - 1] *
0317.     100 );
0318.     $gap = $gap_ar [$j] - $modelGap;
0319.     $graphData = mathGraphData ( $speed_ar [$j], $gap );
0320.     $graphDataM = mathGraphData ( $modelSpeed, 0 );
0321.     $graphDataA = mathGraphData ( $acc_ar [$j], $gap );
0322.     $graphDataAM = mathGraphData ( $modelAcc, 0 );
0323.     $databurestr = mathGraphGyroData ( $databurestr,
0324.     $gap );
0325.     $line [] = "
0326.     var lineChartData$j = {
0327.     labels : [$yAxis],
0328.     datasets : [{
0329.     fillColor : 'rgba(99, 37, 35, 0)',
0330.     strokeColor : 'rgba(99, 37, 35, 1)',
0331.     pointDot : false,
0332.     data : [$graphData]
0333.     }, {
0334.     fillColor : 'rgba(99, 37, 35, 0)',
0335.     strokeColor : 'rgba(82,154,157, 1)',
0336.     pointDot : false,
0337.     data : [$databurestr]
0338.     } ]
0339.     }
0340.     var option$j = {
0341.     scaleSteps : 3,
0342.     scaleStepWidth : 1000,
0343.     scaleStartValue : 0
0344.     }
0345.     var myLine$j = new
0346.     Chart(document.getElementById('canvas$j'))
0347.     .getContext('2d').Line(lineChartData$j,
0348.     option$j);";
0349.     $line2 [] = "
0350.     var lineChartData2$j = {
0351.     labels : [$yAxis],
0352.     datasets : [{
0353.     fillColor : 'rgba(99, 37, 35, 0)',
0354.     strokeColor : 'rgba(82,154,157, 1)',
0355.     pointDot : false,
0356.     data : [$graphDataAM]
0357.     }, {
0358.     fillColor : 'rgba(99, 37, 35, 0)',
0359.     strokeColor : 'rgba(99, 37, 35, 1)',
0360.     pointDot : false,
0361.     data : [$graphDataA]
0362.     } ]
0363.     }
0364.     var option2$j = {
0365.     scaleSteps : 4,
0366.     scaleStepWidth : 20,
0367.     scaleStartValue : -40,
0368.     scaleLabel : "%" <%=value%>%"
0369.     }
0370.     var myLine2$j = new
0371.     Chart(document.getElementById('canvas2$j'))
0372.     .getContext('2d').Line(lineChartData2$j,
0373.     option2$j);";
0374.     $rader_data = array (
0375.     $accPoint_ar [$j],
0376.     $gyroPoint_ar [$j],
0377.     $koshiPoint_ar [$j]
0378.     );
0379.     $rader .= "
0380.     var rc$j = new html5jp.graph.radar('rader$j');
0381.     if( ! rc$j ) { return; }
0382.     var items = [
0383.     ['xxx',$rader_data[0],$rader_data[1],$rader_data[2
0384.     ]]
0385.     ];
0386.     var params = {
0387.     aCap: ['速さ', 'ブレ', '腰'],
0388.     aMax: 100,
0389.     aMin: 0,
0390.     chartShape: 'circle',
0391.     aLinePositions: [0,20,40,60,80,100]
0392.     };
0393.     rc$j.draw(items, params);";
0394.     echo "<script type='text/javascript'
0395.     src='rendering-mode.js'></script>
0396.     <!--[if IE]><script type='text/javascript'
0397.     src='html5jp/
0398.     excanvas/excanvas.js'></script><![endif]-->
0399.     <script type='text/javascript'
0400.     src='html5jp/graph/radar.js'></script>
0401.     <script type='text/javascript'>
0402.     window.onload = function() {
0403.     $rader
0404.     };
0405.     </script>";
0406.     $date = date ( 'm/d H:i', $uTime_ar [$i] );
0407.     echo "
0408.     <div id='userData'>
0409.     <table align='center' valign='middle' border='0'>
0410.     <TR>
0411.     <TD>
0412.     <table class='sample' >
0413.     <TR>
0414.     <TH>名前</TH>
0415.     <TD><div class='buttonDiv'><A HREF =
0416.     growthRecord.php?
0417.     ID={$ID_ar[$i]}>$name_ar[$i]</A></div></TD>
0418.     </TR>
0419.     <TR>
0420.     <TH>番号</TH>
0421.     <TD>$class_ar[$i]組$number_ar[$i]番</TD>
0422.     </TR>
0423.     <TR>
0424.     <TH>練習日時</TH>
0425.     <TD>$date</TD>
0426.     </TR>
0427.     <TR>
0428.     <TH>引きの速度</TH>
0429.     <TD>$accPoint_ar[$i]点</TD>
0430.     </TR>
0431.     <TR>
0432.     <TH>引きのブレ</TH>
0433.     <TD>$gyroPoint_ar[$i]点</TD>
0434.     </TR>

```

```

0426. <TR>
0427. <TH>腰の使い方</TH>
0428. <TD>$koshiPoint_ar[$i]点</TD>
0429. </TR>
0430. <TR>
0431. <TH>総合得点</TH>
0432. <TD>$totalPoint_ar[$i]点</TD>
0433. </TR>
0434. </TABLE>
0435. </TD>
0436. <TD>
0437. <div class=zoomTarget>
0438. <BR>
0439. <canvas id='canvas$i' height='140'
width='400'></canvas>
0440. <script> $line[$i] </script>
0441. </div>
0442. <div class=zoomTarget>
0443. <BR>
0444. <canvas id='canvas2$i' height='140'
width='400'></canvas>
0445. <script> $line2[$i] </script>
0446. </div>
0447. </TD>
0448. <TD>
0449. <div class=zoomTarget>
0450. <div><canvas width='300' height='300'
id='rader$i'></div>
0451. </div>
0452. </TD>
0453. </TR>
0454. </table>
0455. </div>";
0456. $i ++;
0457. $datastr = "";
0458. $databurestr = "";
0459. }
0460. } catch ( PDOException $e ) {

```

```

0461. print $e->getMessage ();
0462. die ();
0463. }
0464. $pdo = null;
0465. echo "<table align='center' valign='middle'><tr>";
0466. $urlname =
"&class_list=$class_list&sort_num=$sort_num
0467. &kikan=$kikan&search_name=$search_name&view=$view"
;
0468. if ($p > 0) {
0469. $Prev = $p - 1;
0470. $PrevPage = "<td><div class='buttonDiv'><a
href='graphview.php
0471. ?p={$Prev}{$urlname}'>前の10件</a></div></td>";
0472. echo $TopPage = "<td><div class='buttonDiv'><a
href='graphview.php
0473. ?{$urlname}'><<</a></div></td> ";
0474. echo $PrevPage . " ";
0475. }
0476. if ($p < $maxpage) {
0477. $Next = $p + 1;
0478. $NextPage = "<td><div class='buttonDiv'><a
href='graphview.php
0479. ?p={$Next}{$urlname}'>次の10件</a></div></td> ";
0480. echo $NextPage;
0481. echo $TopPage = "<td><div class='buttonDiv'><a
href='graphview.php
0482. ?{$urlname}&p={$maxpage}'>>></a></div></td>";
0483. }
0484. echo "</tr></table><BR>";
0485. ?>
0486. <div class='buttonDiv' align="right">
0487. <a href="#maindiv">▲ページの先頭へ戻る</a>
0488. </div>
0489. </div>
0490. <BR>
0491. </body>
0492. </html>

```

## listview.php (リスト表示)

```

0001. <?php
0002. date_default_timezone_set('Asia/Tokyo');
0003. session_start();
0004.
0005. // ログインのチェック
0006. if (!isset($_SESSION["user_name"])) {
0007. $no_login_url = "../login.php";
0008. header("Location: {$no_login_url}");
0009. exit;
0010. }
0011. function viewData($miss, $good) {
0012. if ($miss > 1) {
0013. return "bgcolor='feldsper'";
0014. }else if($good==3){
0015. return "bgcolor='LightCyan'";
0016. }
0017. return '';
0018. }
0019. }
0020. //ソート id の計算
0021. function getSortId($var){

```

```

0022. if($var!=null){
0023. if($var==1||$var==3){
0024. $var=1;
0025. }else{
0026. $var=2;
0027. }
0028. return $var;
0029. }
0030. }
0031.
0032. //エラー表示は1
0033. ini_set('display_errors', 0);
0034. $ID_ar = array();
0035. $class_ar = array();
0036. $number_ar = array();
0037. $name_ar = array();
0038. $totalPoint_ar = array();
0039. $speed_ar = array();
0040. $gyro_ar = array();
0041. $koshi_ar = array();
0042. $uTime_ar = array();
0043. $i=0;

```

```

0044.
0045. $sortNumber = $_GET['sortNumber'];
0046. $sortNumber = getSortId($sortNumber);
0047. $sortPoint = $_GET['sortPoint'];
0048. $sortPoint = getSortId($sortPoint);
0049. $sortSpeed = $_GET['sortSpeed'];
0050. $sortSpeed = getSortId($sortSpeed);
0051. $sortBure = $_GET['sortBure'];
0052. $sortBure = getSortId($sortBure);
0053. $sortKoshi = $_GET['sortKoshi'];
0054. $sortKoshi = getSortId($sortKoshi);
0055. $sortDate = $_GET['sortDate'];
0056. $sortDate = getSortId($sortDate);
0057.
0058. if($_POST["selectClass"]!=null){
0059. $_SESSION["selectClass"] = $_POST["selectClass"];
0060. }
0061. ?>
0062.
0063. <html>
0064. <head>
0065. <title>一覧画面</title>
0066. <link rel="stylesheet" href="list.css"
0067. type="text/css" />
0068. </head>
0069. <body>
0070. <div class="mainviv">
0071. <div class='buttonDiv'><a href="#">PLMS トップへ
0072. </a></div>
0073. <br />
0074. <div class='buttonDiv'><a href="graphview.php">グラフ
0075. 表示へ切り替え</a></div><br>
0076. クラス選択 :
0077. <FORM style="display:inline;"
0078. action="listview.php" method="post">
0079. <SELECT name='selectClass'
0080. onChange='this.form.submit()'>
0081. <OPTION value='0'>すべて</OPTION>
0082. <OPTION value='1' <?php if($_SESSION["selectClass"]
0083. == 1){echo "selected";}?>>1組</OPTION>
0084. <OPTION value='2' <?php if($_SESSION["selectClass"]
0085. == 2){echo "selected";}?>>2組</OPTION>
0086. <OPTION value='3' <?php if($_SESSION["selectClass"]
0087. == 3){echo "selected";}?>>3組</OPTION>
0088. <OPTION value='4' <?php if($_SESSION["selectClass"]
0089. == 4){echo "selected";}?>>4組</OPTION>
0090. </SELECT>
0091. </FORM>
0092. <hr width="1000" size="2" align="left">
0093. <?php
0094. try{
0095. $pdo = new PDO("sqlite:../elearning.db");
0096. $pdo->setAttribute(PDO::ATTR_ERRMODE,
0097. PDO::ERRMODE_EXCEPTION);
0098. $sql_query = "SELECT plainTb.*, userTb.year,
0099. userTb.class, userTb.number, userTb.sex, nameTb.name
0100. FROM plainTb, userTb, nameTb ON plainTb.ID=userTb.ID
0101. and plainTb.ID=nameTb.ID ";
0102. if($_SESSION["selectClass"]>0){
0103. $sql_query = $sql_query." AND class =
0104. ".$_SESSION['selectClass'];
0105. }
0106. // サーバからデータ取得
0107. if($sortNumber==1){
0108. $sql_query = $sql_query." ORDER BY number ASC";
0109. }else if($sortNumber==2){
0110. $sql_query = $sql_query." ORDER BY number DESC";
0111. }else if($sortPoint==1){
0112. $sql_query = $sql_query." ORDER BY totalPoint DESC";
0113. }else if($sortPoint==2){
0114. $sql_query = $sql_query." ORDER BY totalPoint ASC";
0115. }else if($sortSpeed==1){
0116. $sql_query = $sql_query." ORDER BY accPoint ASC";
0117. }else if($sortSpeed==2){
0118. $sql_query = $sql_query." ORDER BY accPoint DESC";
0119. }else if($sortBure==1){
0120. $sql_query = $sql_query." ORDER BY gyroPoint DESC";
0121. }else if($sortBure==2){
0122. $sql_query = $sql_query." ORDER BY gyroPoint ASC";
0123. }else if($sortKoshi==1){
0124. $sql_query = $sql_query." ORDER BY koshiPoint DESC";
0125. }else if($sortKoshi==2){
0126. $sql_query = $sql_query." ORDER BY koshiPoint ASC";
0127. }else if($sortDate == 1){
0128. $sql_query = $sql_query." ORDER BY uTime ASC";
0129. }else if($sortDate == 2){
0130. $sql_query = $sql_query." ORDER BY uTime DESC";
0131. }else{
0132. $sql_query = $sql_query." ORDER BY uTime DESC";
0133. }
0134. }
0135. $sortNumber++;
0136. $sortPoint++;
0137. $sortSpeed++;
0138. $sortBure++;
0139. $sortKoshi++;
0140. $sortDate++;
0141. ?>
0142. <table align='center' valign='middle'
0143. class='sample'>
0144. <tr align = 'center'><th>クラス</th>
0145. <?php echo "
0146. <th><div class='buttonDiv'><a href =
0147. 'listview.php?sortNumber=$sortNumber'>番号
0148. </a></div></th>
0149. <th>名前</th>
0150. <th><div class='buttonDiv'><a href =
0151. 'listview.php?sortPoint=$sortPoint'>得点
0152. </a></div></th>
0153. <th><div class='buttonDiv'><a href =
0154. 'listview.php?sortSpeed=$sortSpeed'>引く速さ
0155. </a></div></th>
0156. <th><div class='buttonDiv'><a href =
0157. 'listview.php?sortBure=$sortBure'>プレの少なさ
0158. </a></div></th>
0159. <th><div class='buttonDiv'><a href =
0160. 'listview.php?sortKoshi=$sortKoshi'>腰の使い方
0161. </a></div></th>
0162. <th><div class='buttonDiv'><a href =
0163. 'listview.php?sortDate=$sortDate'>日時
0164. </a></div></th>";
0165. foreach($pdo->query($sql_query) as $row){
0166. array_push($ID_ar,$row['ID']);
0167. array_push($class_ar,$row['class']);
0168. array_push($number_ar,$row['number']);
0169. array_push($name_ar,$row['name']);
0170. array_push($totalPoint_ar,(int)$row['totalPoint']);
0171. ;
0172. array_push($speed_ar,$row['accPoint']);

```



```

0146. array_push($gyro_ar,$row['gyroPoint']);
0147. array_push($koshi_ar,$row['koshiPoint']);
0148. array_push($uTime_ar,$row['uTime']);
0149. $missInt      = 0;
0150. $goodInt       = 0;
0151.
0152. // 基準点以下を赤で表示
0153. if ($totalPoint_ar[$i] < 50) {
0154. $totalPoint_ar[$i] = "<font
color='red'>$totalPoint_ar[$i]点</font>";
0155. } else {
0156. $totalPoint_ar[$i] = "$totalPoint_ar[$i]点";
0157. }
0158.
0159. //スピードランク
0160. if($speed_ar[$i]<60){
0161. $speed_ar[$i] = "<font color='red'>★</font>";
0162. $missInt ++;
0163. }else if($speed_ar[$i]<=80){
0164. $speed_ar[$i] = "★★";
0165. }else{
0166. $speed_ar[$i] = "★★★";
0167. $goodInt ++;
0168. }
0169.
0170. // ブレランク
0171. if($gyro_ar[$i]<60){
0172. $gyro_ar[$i] = "<font color='red'>★</font>";
0173. $missInt ++;
0174. }else if($gyro_ar[$i]<=80){
0175. $gyro_ar[$i] = "★★";
0176. }else{
0177. $gyro_ar[$i] = "★★★";
0178. $goodInt ++;
0179. }
0180. // 腰ランク
0181. if($koshi_ar[$i]<60){
0182. $koshi_ar[$i] = "<font color='red'>★</font>";

```

```

0183. $missInt ++;
0184. }else if($koshi_ar[$i]<=80){
0185. $koshi_ar[$i] = "★★";
0186. }else{
0187. $koshi_ar[$i] = "★★★";
0188. $goodInt ++;
0189. }
0190.
0191. $timeview = date("m/d H:i",$uTime_ar[$i]);
0192. $viewColor = ViewData($missInt,$goodInt);
0193. echo "
0194. <TR ALIGN = 'center'><TD>$class_ar[$i]</TD>
0195. <TD>$number_ar[$i]</TD>
0196. <TD $viewColor><A HREF =
'growthRecord.php?ID=$ID_ar[$i]'>$name_ar[$i]</A><
/TD>
0197. <TD>$totalPoint_ar[$i]</TD>
0198. <TD Align='left'>$speed_ar[$i]</TD>
0199. <TD Align='left'>$gyro_ar[$i]</TD>
0200. <TD Align='left'>$koshi_ar[$i]</TD>
0201. <TD>$timeview</TD>
0202. </TR>";
0203. $i++;
0204. }
0205. }catch(PDOException $e){
0206. print $e->getMessage();
0207. die();
0208. }
0209. echo "</table>";
0210. $pdo = null;
0211. ?>
0212. </div>
0213. <BR>
0214. <div class='buttonDiv' align="right">
0215. <a href="#maindiv">▲ページの先頭へ戻る</a>
0216. </div>
0217. </body>
0218. </html>

```

## growthrecord.php (上達履歴表示)

```

0001. <?php
0002. session_start();
0003. // ログインのチェック
0004. if (!isset($_SESSION["user_name"])) {
0005. $no_login_url = "../login.php";
0006. header("Location: {$no_login_url}");
0007. exit;
0008. } else {
0009. }
0010.
0011. $ID = $_GET['ID'];
0012. $writeCsvDecision = $_GET['write'];
0013.
0014. function connectionStr($var,$useData,$totalData){
0015. if($totalData==0){
0016. $var = round($useData);
0017. }else{
0018. $var = $var.",".round($useData);
0019. }
0020. return $var;
0021. }
0022.

```

```

0023. //グラフ共通部分
0024. function graphWrite($xData,$yData,$num){
0025. $var="
0026. <script>
0027. var lineChartData$num = {
0028. labels : [$yData],
0029. datasets : [
0030. {
0031. fillColor : 'rgba(99, 37, 35, 0)',
0032. strokeColor : 'rgba(99, 37, 35, 1)',
0033. pointDot : false,
0034. data : [$xData]
0035. }
0036. ]
0037. }
0038. var myLine$num = new
Chart(document.getElementById('canvas$num')).getCon
text('2d')).Line(lineChartData$num);
0039. </script>
0040. ";
0041. return $var;
0042. }

```

```

0043.
0044. // データベースから値を取得
0045. try {
0046. $pdo = new PDO("sqlite:../elearning.db");
0047. $pdo -> setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
0048. if (!$pdo) {
0049. die('接続失敗です。' . $sqliteerror);
0050. } else {
0051. }
0052. }catch(PDOException $e) {
0053. print $e -> getMessage();
0054. die();
0055. }
0056.
0057. $sql_query = "SELECT plainTb.*, nameTb.name FROM
plainTb, nameTb ON plainTb.ID=nameTb.ID WHERE
plainTb.ID = $ID ORDER BY uTime ASC";
0058. $totalPoint = "";
0059. $accPoint = "";
0060. $gyroPoint = "";
0061. $koshiPoint = "";
0062. $totalData=0;
0063. foreach ($pdo->query($sql_query) as $row) {
0064. $userName = $row["name"];
0065. $totalPoint .= $row["totalPoint"].",";
0066. $accPoint .= $row["accPoint"].",";
0067. $gyroPoint .= $row["gyroPoint"].",";
0068. $koshiPoint .= $row["koshiPoint"].",";
0069.
0070. $totalData++;
0071. }
0072. $totalDataHalf = round($totalData / 2);
0073. ?>
0074. <html>
0075. <head>
0076. <title><?php echo $userName?>さんの記録</title>
0077. <link rel="stylesheet" href="growth.css"
type="text/css" />
0078. <script
0079. src="http://ajax.googleapis.com/ajax/libs/jquery/1
.8.2/jquery.min.js"></script>
0080. <script type="text/javascript"
src="./js/jquery.zoomooz.min.js">
0081. </script>
0082. <script type="text/javascript">
0083. $(document).ready(function() {
0084. $("zoomTarget").zoomTarget();
0085. });
0086. </script>
0087. <script type="text/javascript"
src="./js/parts.js"></script>
0088. <script src="ChartGr.js"></script>
0089. <meta name = "viewport" content = "initial-scale = 1,
user-scalable = no">
0090. <style>
0091. canvas{
0092. }
0093. </style>
0094. </head>
0095. <body align="center">
0096. <div class='buttonDiv'>
0097. <a href="..">PLMS トップ</a>
0098. <a href="graphview.php">グラフ表示</a>
0099. <a href="listview.php">リスト表示</a>
0100. </div><br /><br />
0101. <div class='buttonDiv'><a href =
'graphview.php?search_name=<?php echo
('$userName'); ?>'><?php echo ('$userName'); ?>さん
の1回ごとの結果</a></div><br>
0102. <div id='growTop'>
0103. <p>
0104. <font size="6" color="#ffffff"><?php echo
$userName?>さんの記録</font>
0105. </p>
0106. </div>
0107. <div id="main" align="center">
0108. <?php
0109. $x = "";
0110. for($j=1; $j<$totalData; $j++) {
0111. $x .= $j.',';
0112. if ($j == $totalData-1) {
0113. $x .= $totalData;
0114. }
0115. }
0116. echo ("<table align = 'center'><tr align =
'center'><td>");
0117. echo "総合得点<br /><canvas id='canvas0' height='400'
width='500'></canvas><br /><br />";
0118. echo graphWrite($totalPoint,$x,0);
0119. echo ("</td>");
0120. echo ("<td>");
0121. echo "ひきの速さ<br /><canvas id='canvas1'
height='400' width='500'></canvas><br /><br />";
0122. echo graphWrite($accPoint,$x,1);
0123. echo ("</td></tr>");
0124. echo ("<tr align = 'center'><td>");
0125. echo "ブレの少なさ<br /><canvas id='canvas2'
height='400' width='500'></canvas><br /><br />";
0126. echo graphWrite($gyroPoint,$x,2);
0127. echo ("</td>");
0128. echo ("<td>");
0129. echo "腰の使い方<br /><canvas id='canvas3'
height='400' width='500'></canvas><br /><br />";
0130. echo graphWrite($koshiPoint,$x,3);
0131. echo ("</td></tr></table>");
0132. ?>
0133. </div>
0134. <br><br>
0135. <div class='buttonDiv'><a href =
'graphview.php?search_name=<?php echo
('$userName'); ?>'><?php echo ('$userName'); ?>さん
の1回ごとの結果</a></div>
0136. <div class='buttonDiv'>
0137. <br /><br />
0138. <div class='buttonDiv'>
0139. <a href="..">PLMS トップ</a>
0140. <a href="graphview.php">グラフ表示</a>
0141. <a href="listview.php">リスト表示</a>
0142. </div>
0143. </body>
0144. </html>

```