

学生のセルフ・マネジメントのためのゲーミフィケーション・ プラットフォーム AOCa の設計と実装

The Design and Implementation of the AOCa Gamification Platform for Students' Self-Management

小久保 温[†], 角田 均[†], 伊藤 匠[†], 織田 将史[†], 三上 絢佳[†], 今 北斗[†], 柏谷 至[†], 工藤 雅世[†],
坂田 令[‡]

[†]青森大学, [‡] 株式会社リンクステーション

Abstract

We have developed the AOCa gamification platform for students' self-management. The platform has been constructed as a web service. Users access to the platform through the FeliCa card terminal unit or the web page for smartphones. In this paper, the design and the implementation of the platform are discussed.

Keyword: gamification, self-management, web service, FeliCa, smartphone

1 はじめに

大学生になると、親元を離れて一人暮らしをはじめめる人が増え、講義を自分で選択して受講し、空いた時間にアルバイトをはじめめるなど、自己管理の側面を持った生活上の変化が起こる。一方で、わたしたちは継続的に「運動する」など、生活を向上させると考えられる行動でも続けることは難しく、自己管理は困難である。われわれは初年次の大学生の自己管理を援助することで、大学生活を向上して欲しいと考えた。

ところで、人々を夢中にさせるゲームのしくみをゲーム以外のものに取り入れて、動機付けや社会的関係の強化に活用することが、2010年頃から「ゲーミフィケーション」(Deterding 他, 2011; McGonigal, 2011)と呼ばれ注目を集めてきた。Werbachら(2012)は、ゲーミフィケーションが有効と考えられる場面の代表例として「(企業などの)内部」「外部」「行動変容」の3つを挙げている。また、根本ら(2014)は、Twitter を利用した自発的・持続的行動支援プラットフォームを試作し実証実験している。

われわれは大学生の自己管理の援助が「行動変容」ゲーミフィケーションに相当すると考え、しくみを設計し、アプリケーションとして実装した。そして、実証実験を行い、実験参加者の実験前後

の自己効力感の変化を調査した。本論文では、アプリケーションの設計と実装について論じる。

2 取り組みの体制

アプリケーションの設計と実装を論じる前に、取り組みの体制を紹介する。

今回の取り組みは、青森大学と株式会社リンクステーションの連携プロジェクトである。連携プロジェクトは昨年度(2013年度)からはじまっている。2013年度には、電子カードの一種であるFeliCaカードを用いたスタンプラリーシステムを開発した。そして、青森大学の学生が地域(青森)について学ぶことを目的とし、青森市内の文化スポーツ施設を巡るスタンプラリーを実施した。FeliCaカードを採用したのは、株式会社リンクステーションが、これまでFeliCaカードを用いたビジネスに組み込み、開発実績などがあったためである。昨年度に引き続き、今年度(2014年度)もFeliCaカードを用いたプロジェクトを実施した。

今回のプロジェクトは、著者らがしくみを考え、アプリケーションとして実装し、実証実験を運営した(小久保ら, 2014)。実証実験は、青森大学1年生の後期 semester の必修科目「地域貢献基礎演習」で行い、担当の教員および受講した学生に「ミッション」(3節で説明)のアイデア出しや運営や参

加で協力をいただいた。事前の説明とアイデア出しを2014年10月1日に行い、実証実験自体は10月29日から12月17日までの8週間実施した。実証実験の前後で、参加者の自己効力感を測定している。

3 自己管理のためのゲーミフィケーションのしくみの設計

大学生の自己管理を支援するために、自己管理が難しい要因を検討し、「今何をすべきかがあやふやなこと」、また「すべきことを成し遂げてもよいフィードバックが得られないこと」などがあると考えた。

そこで、「コンピュータRPG(ロールプレイングゲーム)」をモチーフとして、生活を向上させるタスクのうち小さい単位で実行できるものを「ミッション」と呼び、それを達成することで「経験値」が得られ、経験値がたまる「レベル」がアップするというしくみを考えた。そして、それをわかりやすく参加者に提示してはどうかと考え、今回の取り組みを「学生生活「見える化」プロジェクト」と名付けた。

ミッションの達成の記録には、FeliCaカードを用いることにした。FeliCaカードはAOCa(AOmori university Cardの略称)と呼ぶことにした。そして、青森大学内の目立つ場所にミッション達成の記録端末を用意し、カードをかざし自己申告で記録してもらうことにした。ミッションの達成の記録にはスマートフォン・アプリを用いるなど、いろいろな方法が考えられる。われわれは、カードという物理的実体を持ったものを参加者に持ち歩いてもらうこと、目立つ場所に設置した記録端末にカードをかざすという目に付く行為などを通じて、プロジェクト自体も「見える化」しようと考えた。そして、参加者に顕在意識化してもらうことで、習慣化を強化できるのではないかと考え、この方法を採用した。

ただし、今回、予算の関係で記録端末を2台しか用意できなかった。そのため、1人の利用者が記録端末を長時間占有することはできない。そこで、スマートフォンで、自分のステータスなどの詳しい情報をじっくりと確認できるしくみも提供することにした。つまり、記録端末では記録と簡単な情報の閲覧、スマートフォンでは詳しい情報

の閲覧という枠組みにすることにした。

4 RPG的なメタファーの設計

今回のゲーミフィケーションの取り組みに、プレイヤーであるユーザーがミッションを達成してレベルアップするという形式のRPGをメタファーとして採用することにした。それを実現するためのゲーム要素の設計について論じる。

4.1 カテゴリー

学生生活を向上させる行為をミッションとして提示するときに、漠然と提示するよりも、何が学生生活を向上させるのかについて指針を示した方が効果的であると考えた。

そこで、学生生活を向上させる行為を検討して分類し、「スタディ(学習)」「ヘルス(健康)」「モラル(道徳)」「コミュニケーション(人間関係)」の4つの「カテゴリー」を設定することにした。このカテゴリーとレベルに従って、ユーザーにミッションを割り当て、ミッションを達成すると内容に応じてカテゴリーごとに経験値が獲得されることにした。レベルに応じてミッションを割り当てるには、ロジット関数型の分布を使用することにした。

具体的なイメージを、図1に示した。現在、ユーザーはステータスとしてスタディ243、ヘルス198、モラル353、コミュニケーション218の経験値と対応したレベルを持っている。そして、ユーザーにはスタディ「PCで検索する」、ヘルス「自炊する」、モラル「授業中私語をしない」、コミュニケーション「人の目を見て話す」というミッションが与えられている。このうち、ミッション「自炊する」を達成すると、スタディ5、ヘルス30、モラル5、コミュニケーション0の経験値を獲得し、レベルに応じたミッションが新たに割り振られる。

4.2 ユーザー

ミッションを達成することで、経験値を獲得し、レベルアップするしくみを実現するために、ユーザーには、名前など「個人固有の情報」、経験値などの「ステータス」、現在「割り当て」られているミッション、ミッション達成の「履歴」の大きく4種類の情報を持たせることにした。

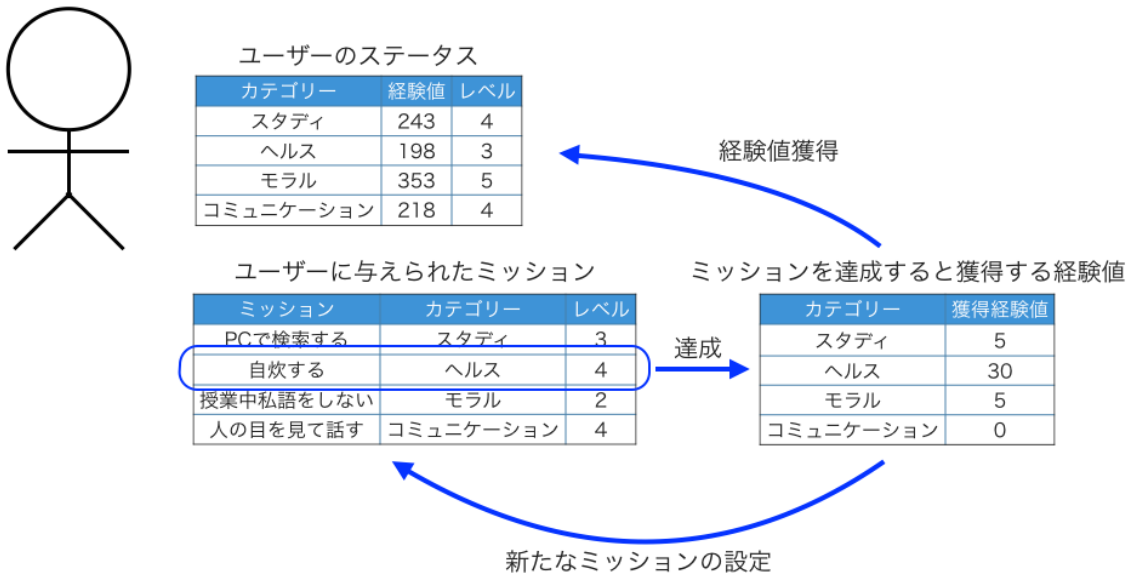


図 1: RPG 的なメタファーの設計

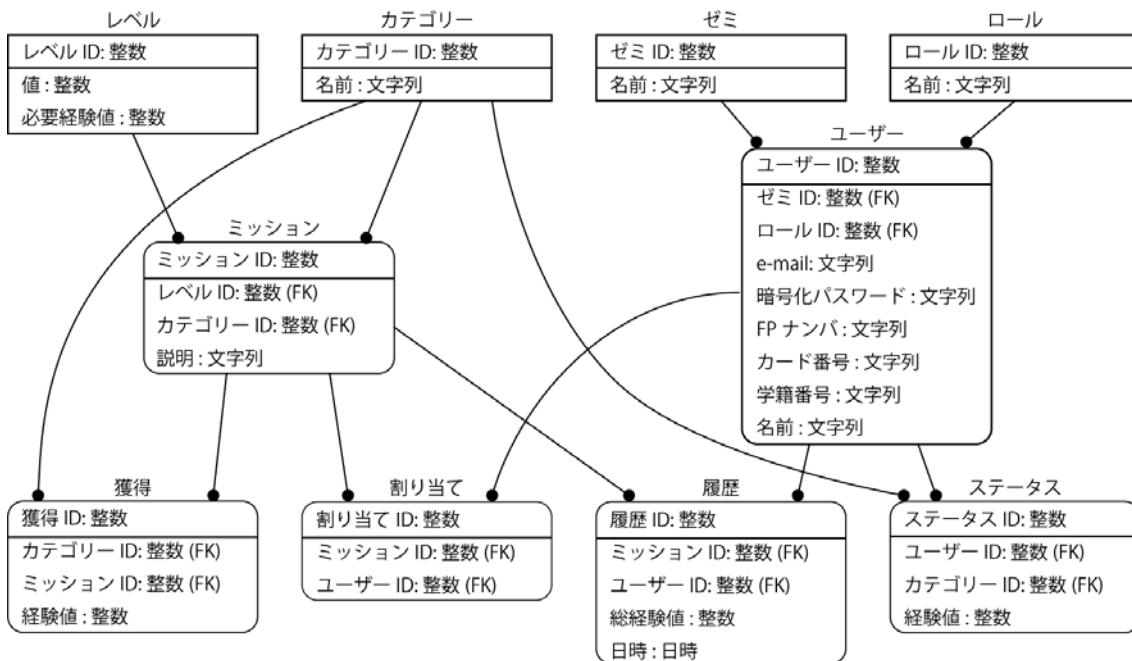


図 2: RPG 的なメタファーを実現した E-R 図

4.3 ミッション

ミッションは、レベルとカテゴリーを参照し、ミッション内容をデータとして持つものとした。各ミッションには、達成時に「獲得」できる経験値をカテゴリーごとに用意した。

以上の設計を E-R(Entity-Relationship)図にしたものが図 2 である。

5 実装

5.1 システム構成

FeliCa カードとスマートフォンから利用するゲーミフィケーションのしくみを実現するために、バックエンドに Web サービスを用意することにした。

そして、FeliCa カードで認証して、ミッションの達成を記録する端末として、Windows 8.1 タブ

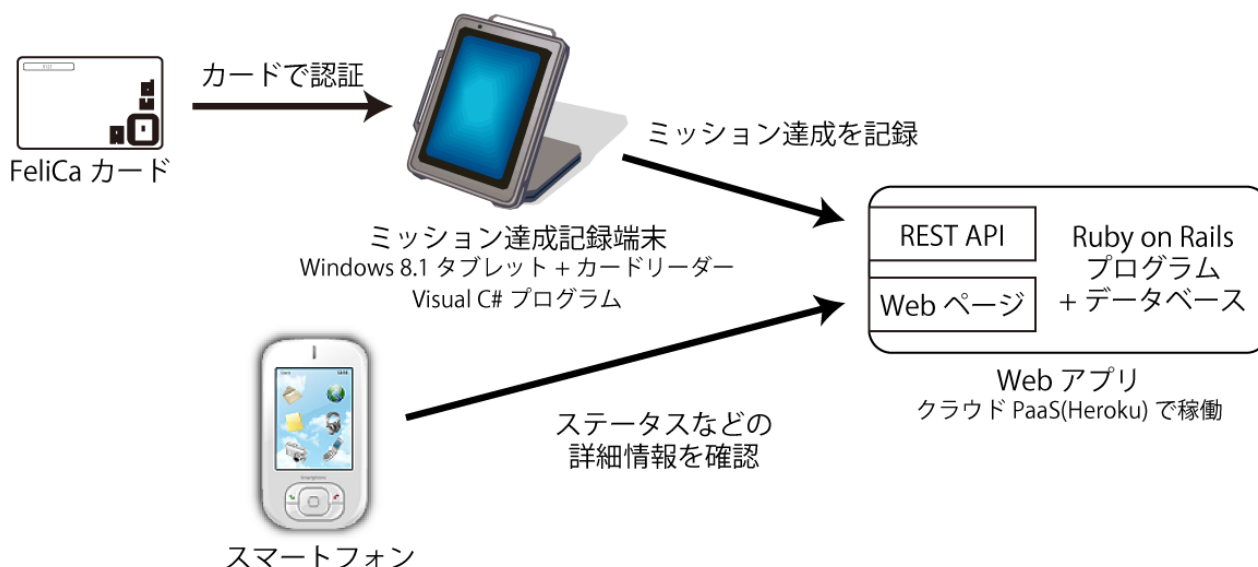


図 3: システム構成

レットを使用し、Visual C#で記述されたアプリケーションを動かすことにした。これは株式会社リンクステーションが提供する FeliCa カードの SDK が C#用であったからである。C#で記述されたアプリは、HTTP で Web サービスと通信し、ミッションの達成を記録したり、ユーザーの情報を取得して表示する。

スマートフォンからは、Web サービスと同じサーバーで提供する Web ページにブラウザでアクセスすることにした。

これらを実現するために、Web サービスでは API と Web ページを提供することにした。開発期間が 1 ヶ月しかなかったため、短期開発が可能とされる Ruby on Rails(Hansson, 2009)を用いて開発することにした。Web サービスの提供は、授業などでアクセスが集中する可能性などを考え、Ruby on Rails に対応したクラウド PaaS(Platform as a Service)の一つである Heroku(“salesforce.com, inc.”, n. d.)を利用することにした。Web サービスで使用するデータベースには、Heroku で提供されている PostgreSQL を使用することにした。

以上をまとめたシステム構成を図 3 に示す。

5.2 ミッション達成記録端末

ミッション達成記録端末では、FeliCa カードをかざしていないときには、いわゆるランキングで

ある「リーダーボード」を表示することにした。そして、カードをかざすと、現在、ユーザーに割り振られているミッションの一覧とステータスが表示され、タップすることでミッションの達成を記録することにした。ミッションの達成を記録すると、効果音とともにステータスが上昇し、新たに割り振られたミッションが表示されることになった。実際に開発した端末でのミッション達成の記録の画面を図 4 に示す。

5.3 スマートフォン用 Web ページ

スマートフォン用の Web ページでは、ミッション達成の記録端末よりも詳細な情報を見ることができるようにした。具体的には、トップページには「リーダーボード」としてランキングと最近の履歴が表示される(図 5)。ログインすると、個人の詳しい情報(図 6)やミッション達成の履歴(図 7)が見られるようにした。Web ページは、主にスマートフォンから閲覧することを想定しているが、実装には Twitter Bootstrap(Otto & Thornton, 2013)を使用し、PC などでも最適化された画面が表示されるようになっている。

5.4 Web サービス

現在の Web サービスは、Fielding(2000)の提唱する REST(Representational State Transfer)的なアーキテクチャーを持ったサービスが主流とな



図 4: ミッション達成の記録端末の記録画面

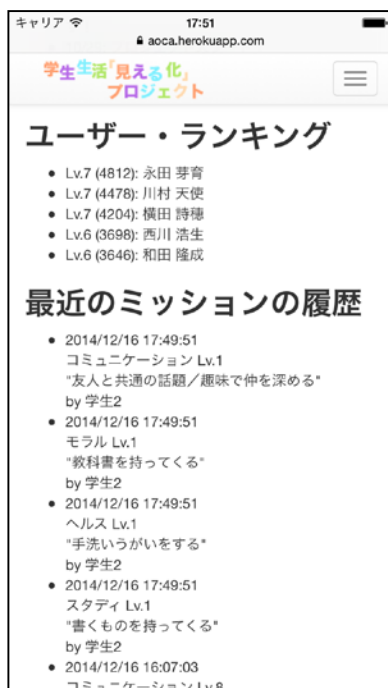


図 5: スマートフォン用 Web ページのトップページのリーダーボード



図 6: スマートフォン用 Web ページの個人の情報ページ

日時	ミッション	経験値	総合レベル
2014/12/16 17:49:51	コミュニケーション 友人と共通の話題/趣味で仲を深める	38 → 48	1 → 1
2014/12/16 17:49:51	モラル 教科書を持ってくる	28 → 38	1 → 1
2014/12/16 17:49:51	ヘルス 手洗いうがいをする	18 → 28	1 → 1
2014/12/16 17:49:51	スタディ 書くものを持ってくる	8 → 18	1 → 1
2014/12/16 17:47:22	ボーナス Webサイトアクセスボーナス	0 → 8	1 → 1

図 7: スマートフォン用 Web ページの個人のミッション達成の履歴ページ

っている(水野, 2014; Mulloy, 2012; 山本, 2010)。REST 的なアーキテクチャーでは、URI がサービスのリソースとして見え、それを HTTP メソッドで操作する。われわれの Web サービスもこれに準拠したものにするようにした。

Web サービスで提供する API は、ミッション達成の記録端末が使用する。ミッション達成の端末では、ミッション達成を記録し、カードをかざした個人の情報が表示され、「リーダーボード」としてランキングと最近達成されたミッションが表示される。よって、API もこれに対応したものにするようにした。リソースはユーザーとミッション達成の履歴の 2 種類となり、表 1 のようなア

セス方法にすることにした。

このうち、FP ナンバに対応したユーザーのミッション達成の履歴は、HTTP の GET メソッドで「/api/histories/:FP ナンバ」という URL にアクセスするという仕様としたのは設計ミスである。これでは FP ナンバに個々の履歴が対応するように見える。FP ナンバには個々のユーザーが対応し、そのユーザーが履歴を持つという意味では「/api/users/:FP ナンバ/histories」とした方が、統一が取れていたと考えられる。

API に対して HTTP の GET メソッドでアクセスすると、JSON(JavaScript Object Notation)形式の応答が返ってくるようにした。たとえば、FP ナンバに対応したユーザーの詳細情報は、コード 1 のように JavaScript のオブジェクトの形式で構造化されたデータとして取得できるようにした。API へのアクセスにエラーがあった場合にも、コード 2 のような JSON 形式の応答が返る。

また、HTTP の POST メソッドでデータを送る場合、送信は通常の POST メソッドのデータ形式で、応答は JSON 形式とした。コード 3 は、FP ナンバが F25C2D3991 のユーザーが、ミッションの ID が 20 と 12 を記録するときを送るデータの例である。

応答を JSON 形式にしたのは、クライアント・サイドの Web アプリケーションを記述するのに広く用いられている JavaScript と相性のよい JSON 形式での応答が現在主流となっているためである。JSON 形式のデータを解析するライブラリは、JavaScript 以外の環境でも広く提供されていて、開発も容易である。

表 1: Web サービスの API へのアクセス方法

機能	API へのアクセス
ユーザー一覧(ランキング上位 5 人)	GET /api/users
FP ナンバに対応したユーザーの詳細情報	GET /api/users/:FP ナンバ
ミッション達成の記録(書き込み)	POST /api/histories
ミッション達成履歴(最近 5 件)	GET /api/histories
FP ナンバに対応したユーザーのミッション達成の履歴	GET /api/histories/:FP ナンバ

コード 1: ユーザーの詳細情報の応答

```
{
  "number": "ソフトウェア情報学部教員",
  "role": "管理者",
  "seminar": "",
  "name": "小久保 温",
  "email": "kokubo@aomori-u.ac.jp",
  "average_level": 1,
  "total_experience": 67,
  "next_average_level_required_experience": 200,
  "statuses": [
    {
      "name": "スタディ",
      "level": 1,
      "experience": 16,
      "next_level_required_experience": 50
    },
    {
      "name": "ヘルス",
      "level": 1,
      "experience": 15,
      "next_level_required_experience": 50
    },
    {
      "name": "モラル",
      "level": 1,
      "experience": 26,
      "next_level_required_experience": 50
    },
    {
      "name": "コミュニケーション",
      "level": 1,
      "experience": 10,
      "next_level_required_experience": 50
    }
  ],
  "missions": [
    {"id": 1, "name": "スタディ", "level": 1, "description": "筆記用具を持ち歩く"},
    {"id": 10, "name": "ヘルス", "level": 1, "description": "朝食を食べる"},
    {"id": 20, "name": "モラル", "level": 1, "description": "ゲームは1日1時間まで"},
    {"id": 30, "name": "コミュニケーション", "level": 1, "description": "あいさつする"}
  ]
}
```

コード 2: エラーの応答(FP ナンバが存在しない場合)

```
["error": "404 Not Found", "detail": "user not found with fono=F25C2D3991"]
```

コード 3: POST で送信するデータ

```
fpono=F25C2D3991
mission_ids[]=20
mission_ids[]=12
```

Web サービスの実装には、Ruby on Rails 4 を使用した。そして、ユーザー認証には devise(Plataformatec, (2009))、API の提供には Grape(Bleigh & “Intridea, Inc.”. (2010))を用いた。

5.5 ボーナス

今回の取り組みを習慣化してもらうのを支援するために、スマートフォン向け Web ページにアクセスすると、1日1回までボーナスを付加することにした。ボーナスは、通常のミッションを達成するよりも少ないが、初期のレベルでは無視できない程度の経験値が獲得できるものとした。ボーナスの実装には Zichermann & Cunningham(2011)を参考にした。

6 まとめ

大学生を対象とした自己管理能力の向上を目指したゲーミフィケーションのしくみを設計し、システムとして実装した。また、開発したシステムを用いて、実証実験を行っている(小久保ら, 2014)。

今回の取り組みでは、学生生活を向上させると考えられるタスクのうち、小さい単位で実施できるものをミッションと呼び、ミッションを達成することでレベルアップするという、RPG 的なしくみを、E-R(Entity-Relationship)を分析して設計した。それを FeliCa カードと対応した端末、スマートフォン、Web サービスを組み合わせたシステムとして開発した。FeliCa カードと端末は、ミッションの達成を記録するのに用いる。端末には Windows 8.1 タブレットを採用し、Visual C#で開発したアプリケーションを稼働させ、FeliCa カードで認証し、Web サービスにデータを書き込み、ユーザーの簡易情報を表示する。スマートフォンからは、Web ページを通じて、詳細なユーザー情報を閲覧する。Web サービスは、設計した E-R 図

を元に Ruby on Rails で開発し、クラウド PaaS の Heroku を用いて稼働させた。完成したシステムには、ゲーミフィケーションのしくみを実現するために必要な機能をすべて実装することができた。

謝辞

本取り組みに協力してくださった青森大学 2014 年度後期「地域貢献基礎演習」を担当された先生方、受講した学生みなさんに感謝いたします。また、青森大学社会学部の鈴木康弘先生には、自己効力感の測定についてアドバイスをいただき、感謝いたします。なお、本取り組みは、平成 26 年度青森大学教育研究プロジェクトに採択され、援助を受けました。

文献

- Bleigh, M., & “Intridea, Inc.”. (2010). Grape. *REST-like API micro-framework for Ruby*. Project site: <https://github.com/intridea/grape>.
- Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: defining gamification. In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*. 9-15. ACM.
- Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures* (Doctoral dissertation, University of California).
- Hansson, D. H. (2009). Ruby on rails. Website. Project site: <http://www.rubyonrails.org>.

- 小久保 温, 角田 均, 伊藤匠, 織田 将史, 三上 絢佳, 今 北斗, 柏谷 至, 工藤 雅世, 坂田 令. (2014). 3F-05 「学生のセルフ・マネジメントのためのゲーミフィケーション・アプリケーションの開発」. 情報処理学会第 77 回全国大会講演論文集, (4), 47-48.
- McGonigal, J. (2011). *Reality is broken: Why games make us better and how they can change the world*. Penguin.
(妹尾 堅一郎 監修. 藤本 徹, & 藤井 清美 訳. (2011). 『幸せな未来はゲームが創る』. 早川書房.)
- 水野 貴明. (2014). 『Web API: The Good Parts』. オライリー・ジャパン.
- Mulloy, B. (2012). *Web API Design: Crafting Interfaces that Developers Love*. Apigee.
- 根本 啓一, 高橋 正道, 林 直樹, 水谷 美由起, 堀田 竜士, & 井上 明人. (2014). 「ゲーミフィケーションを活用した自発的・持続的行動支援プラットフォームの試作と実践」. 情報処理学会論文誌, 55(6), 1600-1613.
- Otto, M., & Thornton, J. (2013). Bootstrap. *Twitter Bootstrap*. Project site: <http://getbootstrap.com>.
- Plataformatec. (2009). Devise. *Ruby gem*. Project site: <https://github.com/plataformatec/devise>.
- “salesforce.com, inc.”. (n. d.). Heroku. *Cloud platform as a service*. Official site: <http://www.heroku.com>.
- Werbach, K., & Hunter, D. (2012). *For the win: How game thinking can revolutionize your business*. Wharton Digital Press.
(三ツ松 新 監修. 渡部 典子 訳. (2013). 『ウォートン・スクール ゲーミフィケーション集中講義』. CCC メディアハウス.)
- 山本 陽平. (2010). 『Web を支える技術: HTTP、URI、HTML、そして REST』. 技術評論社.
- Zichermann, G., & Cunningham, C. (2011). *Gamification by design: Implementing game mechanics in web and mobile apps*. “O'Reilly Media, Inc.”.

The Design and Implementation of the AOCa Gamification Platform for Students' Self-Management

Atsushi KOKUBO[†], Hitoshi TSUNODA[†], Takumi ITOU[†], Masashi ODA[†], Ayaka MIKAMI[†], Hokuto KON[†], Itaru KASHIWAYA[†], Masayo KUDOH[†], Ryo SAKATA[‡]

[†]Aomori University, [‡] LINK STATION Company, Limited

大学生のセルフ・マネジメントのためのゲーミフィケーションのプラットフォーム AOCa を開発した。プラットフォームは、Web サービスとして構築した。ユーザーは Web サービスに FeliCa カード端末やスマートフォン向けの Web ページからアクセスする。本論文では、ゲーミフィケーションのしくみを実現するための設計と実装を論じる。

キーワード: ゲーミフィケーション、セルフ・マネジメント、ウェブ・サービス、FeliCa、スマートフォン