修　士　論　文

題　目　　オープンソースソフトウェアにおける
ゲーム開発とGUIアプリケーション開発間の差異

主任指導教員　　水野　修　教授

指導教員　　崔　恩瀞　助教

京都工芸繊維大学大学院 工芸科学研究科

情報工学専攻

学生番号　　18622052

氏　　名　　脇上　幸洋

令和2年2月10日提出

様式 2 号

学位論文内容の要旨（和文）

令和 2 年 2 月 10 日

京都工芸繊維大学大学院
工芸科学研究科長　殿

工芸科学研究科　　情報工学専攻
平成 30 年入学
学生番号　　18622052
氏　　名　　脇上　幸洋　㊞

（主任指導教員　　水野　修　㊞　）

本学学位規則第 4 条に基づき、下記のとおり学位論文内容の要旨を提出いたします。

1.　論文題目

オープンソースソフトウェアにおけるゲーム開発と GUI アプリケーション開発間の差異

2.　論文内容の要旨（400 字程度）

ゲーム開発プロジェクトと従来のソフトウェア開発プロジェクトとの間にはいくつかの差異が存在している事が報告されている．しかし，従来のソフトウェアという枠組みの中には性質が異なる様々なドメインが存在するにも関わらず，各ドメインに対して詳細に比較した研究は存在しない．本研究の目的はゲームと似た特徴を持つ GUI アプリケーションに着目し比較する事で，ゲーム開発プロジェクトと非ゲーム開発プロジェクトとの間に差異だけではなく何らかの類似点が存在する可能性を示す事である．そこで，オープンソースで提供されているそれぞれのソフトウェア開発プロジェクトを収集し，調査した．その結果，含まれている素材ファイルの割合がゲーム開発プロジェクトの方が多いという差異を確認すると共に，貢献している開発者の属性が似通っている事や，自動テストが書かれているプロジェクトが少ないという二つの類似点を発見した．

Difference between Game Development and GUI Application Development in Open Source Software

2020 18622052 *WAKIGAMI Yukihiro*

Abstract

Recent research has provided evidence that development of video games differs from development of software systems in other domains such as client applications and databases. However, other domains include various domains having different characteristics. Nevertheless, no study has yet considered differences between domains.

In this paper, I compare the structure of the repository, the diversity of developers, and the test code included in the repositories of game development project and the GUI application development project. To verify whether similarities exist between development of games and development of GUI apps. Fot this purpose, I collected and compared 64 game development projects and 44 GUI development projects from open source repositries.

The comparison confirms the existence of differences between game and GUI application development. The difference is that the game development projects contains more material files than the GUI application development projects. On the other hand, the comparison also finds various similarities. The similarities include the diversity of developers' specializations and the existence of test code.

目　次

1. 緒言

ゲーム開発は以前よりクライアントアプリケーションやデータベースといった他ドメインに属するソフトウェアの開発との差異が指摘されてきた．Murphy-Hill らは初めてゲーム開発と非ゲームソフトウェアの開発の差異について研究を行った [1]．彼らは企業に属しているゲーム開発者に対して調査をし，様々な差異を指摘した．その差異の中には，非ゲームソフトウェア開発と異なりゲーム開発では多種多様なアセットが必要な事や，ゲームの持つ複雑性から自動テストがあまり書かれない事などが含まれる．また，Pascarella らは上記の結果を基にオープンソースソフトウェア（OSS）の文脈でゲーム開発と非ゲームソフトウェア開発の間の差異を確認する研究を行った [2]．OpenHub を用いて OSS におけるゲーム開発リポジトリと非ゲームソフトウェア開発リポジトリをそれぞれ 30 件収集し，上記で示した差異の内いくつかが再度認められた事を示した．

しかし，Pascarella らは自身の研究において非ゲーム開発プロジェクトを収集したが，対象となったソフトウェアには異なる特徴を持つ様々なドメインのソフトウェアが含まれている．例えば，サーバー上で動作するデータベースのようなプロジェクトは，ゲーム開発において一般的に含まれるとされる画像や音楽といった素材ファイルが通常存在し得ない．こうしたドメインが多数含まれている為に，特定のドメインとの間には差異とは別に類似点が存在しているにも関わらず，その類似点が見え辛くなっている可能性が考えられる．しかし，特定のドメインに絞りゲーム開発と比較した研究は未だ存在しない．

本論文の目的は複数存在する非ゲームソフトウェアドメインの中から比較対象を一つに絞り，ゲーム開発との間に存在する差異と類似点を確認する事である．

本研究ではその中でもゲームと類似した特徴を持つと考えられるグラフィカルユーザーインターフェース（GUI）アプリケーションに着目した．GUI アプリケーションとは，マウスやタッチパネルのような入力デバイスによって操作される事を前提としたアプリケーションの総称である．そして，多くの場合ユーザーに視覚的，聴覚的効果を与える為に画像や音といった素材ファイルが含まれる．これは，ゲームと近い特徴であると考えられる．今回はこのような特徴を持つソフトウェアドメインの一つ GUI アプリケーションを比較の対象とする事で，ゲーム開発と非ゲーム

ソフトウェア開発との間に類似点が存在する可能性を検証する．

具体的な研究の手順は，本研究を行うきっかけとなったPascarellaらの研究に従う．Pascarellaらは非ゲームソフトウェア開発プロジェクトのリポジトリを収集したが，本研究ではそこに，GUIアプリケーションドメインという制限を加える．その上で64件のゲーム開発プロジェクトと44件のGUIアプリケーション開発プロジェクトのリポジトリを収集した．その後，各リポジトリに対しカテゴリ毎のファイル分布や，それぞれのリポジトリに関わる各開発者のコミット先ファイルの分布，そしてテストコードが含まれるプロジェクトの数を調査し，比較した．

本紙は次のように構成される．第2章では本研究と関連する過去の研究について述べる．第3章では今回の研究設問を示す．第4章ではリポジトリ収集の手法と，それぞれの研究設問に対する調査方法を述べる．第5章で調査結果を述べ，第6章では得られた結果を基に考察する．第7章では本研究における妥当性への脅威を示す．第8章では本研究の要約および今後の研究について述べる．

2. 関連研究

本章では，本研究と関連する以前の研究について述べる．ゲーム開発とソフトウェア工学を結び付けて研究を行う試みはこれまでにも行われており，ゲーム開発の分野に対し様々な示唆を与えている．Politowski らは，ゲーム開発プロジェクトにおいて開発プロセスと関連する問題に関してブラジルのゲーム開発者を対象に調査している [3]．その結果，どのようなプロセスであっても体系的なプロセスに沿ったプロジェクトではより高品質な製品が開発されることを明らかにした．その他にもゲーム開発者が最も頻繁に遭遇する問題として開発の遅延や非現実的な規模の要求，開発における文書の不足という問題が存在している事を示した．また，ゲーム開発プロジェクトにおいてそれぞれのプロジェクトの状況に合わせた最適な開発プロセスを推薦するシステムの開発などを行っている [4]．Aleem らは体系的にこれまでのゲーム開発に関わる文献をレビューし，ポストプロダクションに関する研究が不足している事を指摘した [5]．Musil らはオーストラリアに存在するゲーム会社 13 件を調査し，アジャイル手法が広く使われている事を明らかにした [6]．Al-azawi らはゲーム開発にて使用される既存アジャイル手法を分析し，エージェント指向プログラミングとアジャイル開発を組み合わせた新たなゲーム開発手法を提案した [7]．Callel らはゲーム開発プロジェクトに関する 50 件の事後分析から，ゲーム開発における要件定義に関する失敗の多くはリプロダクションとプロダクションの間で起こる事を発見した [8]．

こうしたゲーム開発の開発プロセスに着目した研究の他に，ゲーム開発プロジェクトの持つ固有の特性を前提とした研究が存在する．Politowski らは，GamasutraPortal を用いてゲーム開発プロジェクトの事後分析を 20 件収集し，それぞれの開発プロセスについて分析した [9]．結果として，ゲーム開発においてウォーターフォールモデルのような古典的な開発プロセスを用いているプロジェクトは減少傾向だが，未だに使用しているプロジェクトも存在していることを明らかにした．Godoy らはゲーム開発の持つ固有の特性に着目し，それらの特性を考慮した開発プロセスの研究を行った [10]．ゲーム開発にはアジャイル的な手法が用いられているが，ゲーム開発における様々な固有の問題を解決しているような手法はほとんど存在していないとし，新たにゲームスクラムと呼ばれる手法を提案した．また，ゲームスクラムを実際に

小規模のチームで実践し，その効果を確認した．Kanode らはゲーム開発プロジェクトで発生する課題を分析し，ソフトウェア工学における従来の手法を用いて解決する方法を提供した [11]．Schetinger らはアジャイル開発にてよく用いられているユーザーストーリーというプラクティスに着目し，ゲーム開発プロジェクトに合ったアプローチを提案した [12]．Tschang らはゲームに関する 65 件の事後分析の結果から，ゲーム開発と非ゲームソフトウェアの開発との間にゲームデザインやコンテンツの複雑な組み合わせに起因する差異を発見した [13]．

先で挙げた研究にはゲーム開発と非ゲームソフトウェア開発の差異を前提として置いている研究が含まれている．しかし，対照的にそれぞれの類似点を前提とする研究も存在する．Ampatzoglou らは，従来のソフトウェア開発とゲームにおける共通部分の概観を提供し，実証的な研究の不足を指摘している [14]. Petrillo らは 20 件のゲームの事後分析からゲーム開発と非ゲームソフトウェア開発において発生した問題や使用している開発プロセスがほぼ同様である事を発見している [15, 16, 17]. こうした研究はゲーム開発と非ゲームソフトウェア開発との間における類似点の存在を示唆しているが，その内容までは不明である．

3. 研究設問

本研究の目的は，ゲーム開発とGUIアプリケーション開発との間の差異や類似点を明らかにすることである．先行研究であるMurphy-Hillら[1]とPascarellaら[2]の研究により，いくつかの興味深い差異が示された．本研究では，それらの差異の中からGUIアプリケーションとの比較に関連する差異を選択し，改めて検証することを研究設問とする．

3.1 研究設問1

一つ目の研究設問は，**ゲーム開発プロジェクトはGUIアプリケーション開発プロジェクトと比較してもプロジェクトに含まれる素材ファイルの割合が多くなる傾向にあるか**である．ゲーム開発は非ゲームソフトウェア開発と比べ，画像や音楽といった多種多様な素材ファイルが必要になるという差異が報告されている．この差異は，ゲームではユーザ体験をより良いものとするために，視覚的，聴覚的な効果が重要であることから発生していると考えられる．しかし，GUIアプリケーションもまたこうした効果によってユーザ体験をより良い物にすることが求められるドメインの一つである．その為，含まれている素材ファイルの数について，ゲームとの類似点が存在し得ると考えられる．

3.2 研究設問2

二つ目の研究設問は，**GUIアプリケーション開発プロジェクトの開発者にも，素材ファイルにしかコミットしない開発者が存在しているか**である．一つ目の差異と関連して，ゲーム開発ではほかのソフトウェア開発と比べて多分野の専門家が必要になるという差異が報告されている．これは，ゲーム開発プロジェクトにのみ見られた，コードファイルに対しては開発せず，画像や音楽のような素材ファイルのみを開発している開発者の存在を根拠としている．この差異に関しても，GUIアプリケーションは上記で述べた通り素材ファイルを必要とする為，類似点として同様な開発者の存在を確認できる可能性が考えられる．

3.3 研究設問 3

最後の研究設問は，**ゲーム開発プロジェクトは，GUI アプリケーション開発プロジェクトと比較しても，テストコードを含むプロジェクトが少ない傾向にあるか**である．ゲーム開発プロジェクトは，ほかのソフトウェア開発プロジェクトと比較して，テストコードの量が不足しているという差異が挙げられている．この差異が発生する原因の一つとして，先行研究では，入力デバイスが特殊であり，ユーザーインターフェースをソフトウェアから分離することが難しいことを挙げている．しかし，GUI アプリケーションもまたマウスやタッチパッドといった入力デバイスを前提としており，ユーザーインターフェースの分離が難しい．よって，この点についても GUI アプリケーションはゲームと同様な問題を抱えていると考えられ，類似点となる可能性がある．

4. 方法

4.1 プロジェクトの選定と収集

本研究ではゲーム開発プロジェクトとGUIアプリケーション開発プロジェクトを選定し，収集する必要がある．今回はPascarellaらの研究に則り，それらのプロジェクトの選定の為にOpenHub[18]を利用した．OpenHubはOSSプロジェクトの索引を持つプラットフォームサイトであり，各アプリケーションが持つリモートリポジトリのURLに加え総コミット数のようなメトリクスを含む様々な情報を提供している．

OpenHubではタグ機能によって各ドメインに属するアプリケーションを検索できる．例えば，GUIアプリケーションに関するプロジェクトであればguiというタグが，ゲームに関するプロジェクトあればgameというタグが付いている事が多い．そのため，本研究ではまずOpenHub上でgameあるいはguiのタグが設定されたプロジェクトを検索した．しかし，これだけでは開発に使用するライブラリやフレームワークといった，指定したドメインには関わるもののアプリケーションではないプロジェクトが含まれてしまう．そこで，加えてアプリケーションではないプロジェクトである可能性を示すタグを含むプロジェクトに対して除外する処理を行った．今回除外処理に使用したタグを以下に示す．

- engine
- gamedev
- emulator
- framework
- library
- server
- cheat
- mod
- tool
- plugin
- addon
- assets

なお，収集の過程で game と gui 双方のタグを設定しているプロジェクトがいくつか確認されたが，その場合はアプリケーションを分類する上でより狭い範囲となるゲームプロジェクトとして分類した．

次に，各開発プロジェクトのリポジトリを GitHub[19] を用いて収集した．GitHub とは，プロジェクトのリポジトリをホスティングするプラットフォームサイトである．GitHub 上にもタグ機能が存在しており，OpenHub と同様にそのリポジトリが属するドメインの情報が提供されている事がある．その為，この機能を用いて GitHub からリポジトリを収集する際にも上記と同様のフィルタリングを実施した．これにより更にいくつかのリポジトリを除外できたが，中身が空のリポジトリや，タグ機能によるフィルタリングだけでは除外しきれなかったリポジトリが存在した．こうしたリポジトリに関しては，リポジトリ内をそれぞれ目視で確認し，除外した．

最後に，使用しているプログラミング言語によるフィルタリングを行った．特にゲームにおいては，既存のプログラミング言語以外の特別な言語を用いるツールによって開発されることがある．こうしたツールで独自形式のファイルが使用されている場合，後述する分類手法を使えなくなる可能性があり，調査に不都合が生じる．その為，今回はそうしたプロジェクトを除外する事を目的としてフィルタリングを行った．今回はゲーム開発と GUI アプリケーション開発両方に使用される言語から，C#，Java，Python の三言語を選定した．それぞれ双方のプロジェクトにおいて広く使用される言語であり，調査に十分な数のプロジェクトが収集できると考えた為である．

4.2 研究設問に対する調査方法

4.2.1 研究設問 1

まず，各プロジェクトに含まれているファイルを全て，一定の基準に従って分類する．本研究において各ファイルはコードカテゴリ，素材カテゴリ，その他カテゴリのいずれかに分類される．なお，これらのカテゴリは Pascarella らの研究で用いられたものを参考に設定した．先行研究では各カテゴリをさらに細分化しているが，本研究では不要であると考え，これらのカテゴリに対してのみ分類する．コードカテゴリに属するファイルは，ソフトウェア本体のコードや，開発に使用されたライブラ

リ，フレームワークといったファイルである．また，README.md のような文書ファイルもコードカテゴリに含む．素材カテゴリに属するファイルは，画像や音楽といった素材ファイルである．他にも，フォントや 3D モデルといった素材が含まれる．上記二つのカテゴリに含まれないファイルは，その他カテゴリに分類される．これらの基準に従い各ファイルを分類した後，それぞれのリポジトリにおける素材カテゴリに分類されたファイルの割合を調査する事によって一つ目の研究設問に回答する．

Pascarella らの研究は，ファイルの分類にそのファイルが含まれるディレクトリのパスと，そのファイルの持つ拡張子を基準として用いている．本研究でも，同様の基準を用いてファイルを分類する．

本研究ではまず，ファイルの持つ拡張子を用いて分類した．この段階では，.cs や.java といった拡張子を持つファイルはソースコードとしてコードカテゴリに，.png や.wav のような拡張子を持つファイルは素材に関するファイルとして素材カテゴリに分類される．しかし，.xml という拡張子を持つファイルはソフトウェア本体のソースコードの可能性もあれば，他の素材ファイルに関する様々な定義が含まれたファイルの可能性もあり，拡張子だけでは分類が難しい．こうした特徴の拡張子を持つファイルの場合には，もう一つの基準であるディレクトリのパスを使用する．ここでは，src というディレクトリに含まれるファイルはソースコードとしてコードカテゴリに，images というディレクトリに含まれるファイルは画像素材に関するファイルとして素材カテゴリへと分類した．なお，拡張子が存在していないファイルや，上記の方法では分類不可能だったファイルはその他カテゴリに分類される．

最後に，上記の分類に用いた識別子を表 4.1 に示す．本研究では Pascarella らが公開していた識別子を基本として設定しているが，それだけでは分類しきれないファイルが存在した為，独自の拡張を加えることとした．その拡張では，分類不可能だったファイルを検出しながら試行を繰り返し，各リポジトリにおいて 90 % を超えるファイルがなるべくその他カテゴリ以外に分類されるように識別子を追加した．

4.2.2 研究設問 2

それぞれのプロジェクトの各開発者がどのカテゴリに含まれるファイルを主に編集しているかを調査する．この調査の為に，Pascarella らが論文内で用いていたファイルの所有者という概念 [20] を採用した．ここで所有者とは，ある特定のファイル

表 4.1　分類に使用した識別子

カテゴリ	拡張子	ディレクトリパス
コード	'c', 'cc', 'cpp', 'h', 'hpp', 'in', 'py', 'pl', 'js', 'lua','mk', 'cmake', 'm4', 'a', 'so', 'lib', 'dll', 'so', 'zip', 'rar', '7z', 'gz', 'bz2', 'po', 'pot', 'i18n', 'tex', 'html', 'htm', 'css', 'pdf', 'cs', 'java','md', 'txt', 'meta', 'rb', 'conf','sh', 'sgml', 'eps', 'php', 'sql', 'less', 'inc', 'ts', 'qss', 'bat', 'pyc', 'resx', 'csv', 'asset', 'jar', 'scss', 'tmx', 'pyx', 'pkg','cmd', 'rst', 'hs', 'pas', 'glade', 'nupkg', 'xaml', 'tsx'	'source', 'src', 'tool', 'include', 'util', 'test','include', 'build', 'comp', 'lib', 'data', 'os', 'arch', 'language', 'languages','lng', 'i18n', 'translation', 'doc', 'man', 'license', 'guide', 'package','documentation', 'sources', 'plugins','library', 'spec', 'tests', 'ext', 'localization', 'docs', 'scripts', '3rdparty', 'docker', 'locale', 'manual', 'views', 'modules','packages', 'libraries', 'plugin', 'translations', 'libs', 'classes', 'utils', 'thirdparty', 'components', 'win32', 'engine'
素材	'wav', 'ogg', 'mp3', 'dsp', 'png', 'rgb', 'ttf', 'cfg', 'map', 'gif', 'ico', 'svg', 'dds', 'xcf', '3ds', 'eff', 'properties', 'canvas', 'effects', 'commands', 'electrical', 'extensions', 'desktop','tiff', 'tif', 'bmp', 'jpg', 'stl', 'gcode', 'blend', 'stamp','m4a','spr', 'mat', 'vert', 'frag', 'shader', 'wv'	'image', 'asset', 'assets', 'icons', 'images', 'icon', 'model','scenery', 'texture', 'graphic', 'planet', 'font', 'model', 'data', 'resources', 'fonts', 'models', 'resource', 'audio', 'mapeffects', 'objects', 'sounds', 'weapons', 'sprites', 'levels','maps', 'animations', 'items', 'enemies', 'powers', 'cutscenes','rsrc', 'graphics', 'tiledmaps', 'shader'

に対し，そのファイルに対して行われたコミット全体の 75%を超えるコミットを行った開発者を指す．

本研究ではまず各ファイルに対し所有者となる開発者を判別し，そこから開発者ごとに所有者となっているファイルのカテゴリの分布を調査した．所有者の判別には Pascarella らが使用した Git の機能を用いる手法を使用した．まず，各ファイルに対するコミットの数をカウントした．その後，各ファイルにおける各開発者のコミットの数を集計し，そのファイルの持つ全コミットの 75%を超えるコミットを行った開発者をそのファイルの所有者として記録した．最後にそれぞれの開発者がどのようなファイルに対して所有権を持っているかを調査し，素材カテゴリに属しているファイルに多くコミットを行っている開発者が存在しているかを調査した．なお，全コミットに対する個人のコミットの割合が 75 %を超える開発者が存在せず，所有者と呼べる開発者が定まらないファイルに関しては，先行研究と同様に所有者が存在しないファイルとして扱う．

4.2.3 研究設問 3

各プロジェクトに対して単体テストが書かれているかを調査する．上記 2 つの研究設問に対する調査では今回収集したリポジトリ全てに対し調査したが，この研究設問においては手法の都合上，Python で書かれたコードが含まれるプロジェクトのみを調査対象とする．

Python というプログラミング言語には単体テストを実行できるツールとして，unittest というパッケージが標準で提供されている．本研究ではリポジトリ内の unittest が使用されているファイルを検出する事で，そのリポジトリに単体テストが書かれている事を確認する．

まず，test という文字列が含まれたファイル名を持つファイルを全て抽出した．これは，前提として，テストコードが書かれたファイルでは通常 test という文字列を含むファイル名が使用されるためである．しかし，test という文字列がファイル名に含まれていても，単体テストとは無関係のコードが書かれている可能性が存在する．その為，次に unittest という文字列がコードファイル内に含まれているかどうかを探索した．最後に，各ドメインにおいて Python のコードファイルが含まれているリポジトリの数と単体テストが含まれているリポジトリの数を比較する．

5. 結果

最終的に収集されたプロジェクトの数は，ゲーム開発プロジェクトが 64 件，GUI アプリケーション開発プロジェクトが 44 件という結果となった．以下に各プロジェクトに対し第 4 章で述べた方法を用いて調査した結果を述べる．

5.1 研究設問 1

調査対象の各リポジトリにおいて，ファイルの総数と各カテゴリ毎のファイル数を表 5.1 と表 5.2 に，割合を表 5.3 と表 5.4 に示す．また，調査対象の各リポジトリにおいて，素材カテゴリに分類されたファイルの割合に関する分布を表 5.5 に示す．

表 5.1　ゲーム開発プロジェクトにおけるファイル総数と各カテゴリ毎のファイル数

	合計	平均	中央値
ファイル総数	109,762	1,829.4	768
コード	41,783	696.4	328.5
素材	65,500	1,091.7	213.5
その他	2,479	41.3	17

表 5.2　GUI アプリケーション開発プロジェクトにおけるファイル総数と各カテゴリ毎のファイル数

	合計	平均	中央値
ファイル総数	51,140	1,217.6	315.5
コード	39,764	946.8	184.5
素材	9,016	214.7	36.5
その他	2,360	56.2	16.5

まずは，リポジトリ内における素材ファイルの割合について結果を述べる．各プロジェクトにおいてリポジトリ全体に含まれる素材ファイルの割合の平均は，ゲーム開発プロジェクトでは 40.86%，GUI アプリケーション開発プロジェクトでは 17.55%という結果となった．なお，中央値は，ゲーム開発プロジェクトが 44.41%，GUI アプリケーション開発プロジェクトが 9.27%である．

次に，素材ファイルが含まれる割合の分布について結果を述べる．GUI アプリケーション開発プロジェクトでは，素材カテゴリに属するファイルの割合がリポジトリ全体の 10 %に満たないリポジトリの割合が全体の約 50 %を占めている．一方，ゲーム開発プロジェクトでは 80 %を超えるプロジェクトが何らかの素材ファイルを 10 %以上含んでいる．

他にも，ゲーム開発プロジェクトでは，リポジトリ全体の約 40 %において，素材カテゴリに属するファイルの割合が 50 %以上を占めているという結果となった．中には yorg のように，リポジトリの 90 %以上をコードファイルではなく素材ファイルが占めるプロジェクトも存在している．一方 GUI アプリケーション開発プロジェクトでは，50 %以上を素材カテゴリに属するファイルが占めるリポジトリは全体の約 10 %未満に留まる. 今回調査した GUI アプリケーション開発プロジェクトの中で最も素材ファイルの割合が高かったのは，et-otp の 58.8%である．

最後に，素材ファイルが一つも含まれていないプロジェクトの存在についても調査した．その結果，GUI アプリケーション開発プロジェクトでは JSpaceAlertMission-Generator，ark，ceres の 3 件が，ゲーム開発プロジェクトでは tbrpg の 1 件のみが該当した．

5.2 研究設問 2

今回の調査によって，所有者となるファイルを一つでも持つ開発者の総数を表 5.6 に示す．各ドメインにおいて該当する開発者は，ゲーム開発プロジェクトでは 512 人，GUI アプリケーション開発プロジェクトでは 243 人であった．加えて，双方のドメインについてコードカテゴリもしくは素材カテゴリに属するファイルのカテゴリ別所有率に関する開発者の分布を表 5.7 と表 5.8 に示す．このカテゴリ別所有率とは，所有者となっているファイルの総数に対する各カテゴリに属するファイルの割合を

表 **5.3**　ゲーム開発プロジェクトにおける各カテゴリに含まれるファイルの割合

	平均 (%)	中央値 (%)
コード	52.93	49.79
素材	40.86	44.41
その他	6.21	3.30

表 **5.4**　**GUI** アプリケーション開発プロジェクトにおける各カテゴリに含まれるファイルの割合

	平均 (%)	中央値 (%)
コード	73.37	77.58
素材	17.55	9.27
その他	9.09	5.76

表 **5.5**　開発プロジェクトに含まれる素材ファイルの数と総数に対する割合の分布

割合 (%)	ゲーム開発プロジェクト数	総数に対する割合 (%)	GUI アプリケーション開発プロジェクト数	総数に対する割合 (%)
0-10	9	15.00	22	52.38
11-20	6	10.00	6	14.29
21-30	6	10.00	3	7.14
31-40	6	10.00	5	11.90
41-50	10	16.67	2	4.76
51-60	11	18.33	4	9.52
61-70	3	5.00	0	0.00
71-80	5	8.33	0	0.00
81-90	2	3.33	0	0.00
91-100	2	3.33	0	0.00

指す．

どちらのドメインにおいても，素材カテゴリの所有率が 90%を超える開発者の存在を確認できる．例えばゲーム開発プロジェクトである frogatto というプロジェクトにおいて，こうした特徴を持つ開発者を識別できる．同様に，GUI アプリケーション開発プロジェクトでは ninja や roundcubeemail というプロジェクトにおいて，同じ特徴を持つ開発者を確認できる．こうした開発者の割合は，ゲーム開発プロジェクトでは 30%以上を占めているが，GUI アプリケーション開発プロジェクトでは 7%程度に留まっている．

また，同様にコードカテゴリの所有率が 90%を超える開発者を確認可能である．こうした開発者の割合は，ゲーム開発プロジェクトでは 30%以上を，GUI アプリケーション開発プロジェクトでは 50%以上を示している．

最後に，コードカテゴリと素材カテゴリの双方において所有率が 10%を超え，90%未満となっている開発者の存在も確認された．こうした特徴を持つ開発者の割合は，どちらのドメインにおいても 25%程度を示しており，中にはそのリポジトリ全体において 90%を超えるファイルの所有者となっている開発者も存在した．

なお，上記で示した特徴を持たない開発者がゲーム開発プロジェクトでは 7%，GUI アプリケーション開発プロジェクトでは 14%程度確認できる．こうした開発者が所有者となっているファイルの多くは，その他カテゴリに属している．

5.3 研究設問 3

研究設問 3 に対する今回の調査結果を表 5.9 に示す．

調査対象としたリポジトリの内，Python を用いて書かれたソースコードが含まれていたリポジトリはゲーム開発プロジェクトの物が 41 件，GUI アプリケーション開発プロジェクトの物が 30 件であった．その内，テストコードが含まれていたリポジトリはゲーム開発プロジェクトでは 8 件，GUI アプリケーション開発プロジェクトでは 6 件であった．各ドメインに属する全リポジトリに対する割合としては，どちらも全体の約 20%以下である．

表 **5.6**　各ドメインにおいて所有者となっているファイルを一つ以上持つ開発者の総数

	ゲーム開発プロジェクト	GUI アプリケーション開発プロジェクト
人数	512	243

表 **5.7**　ゲーム開発プロジェクトにおけるコード，素材カテゴリファイルの各所有率に属する開発者の分布

所有率 (%)	コード (人)	開発者全体に対する割合 (%)	素材 (人)	開発者全体に対する割合 (%)
0-10	184	35.94	203	39.65
11-20	21	4.10	17	3.32
21-30	16	3.13	14	2.73
31-40	23	4.49	18	3.52
41-50	32	6.25	23	4.49
51-60	17	3.32	14	2.73
61-70	17	3.32	20	3.91
71-80	16	3.13	18	3.52
81-90	17	3.32	18	3.52
91-100	169	33.01	167	32.62

表 5.8　GUI アプリケーション開発プロジェクトにおけるコード，素材カテゴリファイルの各所有率に属する開発者の分布

所有率 (%)	コード (人)	開発者全体に対する割合 (%)	素材 (人)	開発者全体に対する割合 (%)
0-10	33	13.58	156	64.20
11-20	8	3.29	18	7.41
21-30	6	2.47	7	2.88
31-40	10	4.12	8	3.29
41-50	14	5.76	13	5.35
51-60	12	4.94	7	2.88
61-70	8	3.29	9	3.70
71-80	12	4.94	2	0.82
81-90	17	7.00	6	2.47
91-100	123	50.62	17	7.00

表 5.9　テストコードが含まれたリポジトリ数

	Game	GUI
全体のリポジトリ数	41	30
テストコードを含むリポジトリ数	8	6

6. 考察

6.1 研究設問 1

研究設問 1 の結果から，GUI アプリケーション開発プロジェクトよりもゲーム開発プロジェクトの方が多く素材ファイルを含んでいる事が分かった．

この差異が生じる理由としては，それぞれが目的とするアプリケーションの特徴の違いや，それぞれのプログラミング言語が持つフレームワークの存在が考えられる．今回調査対象とした C#，Java，Python というプログラミング言語は，いずれも GUI アプリケーション開発を対象としたフレームワークを持つ．そのフレームワークには既にボタンやテキストボックスに使用されるデフォルトの素材が含まれているため，開発者はこの素材を使用する場合自身で素材を用意する必要がない．このことから，GUI アプリケーション開発プロジェクトでは素材ファイルの必要性が低い傾向にあると考えられる．

一方，ゲーム開発プロジェクトでは，一般的に既存の素材ではなく独自の素材を用意する強い動機が存在する．これは，ゲームではユーザーに対して独自の体験を与える事を重視する傾向が強く，この独自性がゲームにおいてユーザーの体験を高める事に繋がっているためである．また，GUI アプリケーションで必要となる素材は主にインターフェースに関連するものだが，ゲームではインターフェースに限らず，様々な場面で個別の素材を要求する．例えば，ゲームの世界観を示す背景や建物といったオブジェクトや，ユーザーが動かす対象となるキャラクター，強烈な印象をユーザーに与えるエフェクトなどが代表的である．このことから，ゲーム開発プロジェクトでは素材ファイルの必要性や重要性が高い傾向にあると考えられる．

以上の理由から，GUI アプリケーション開発プロジェクトとゲーム開発プロジェクトの間には，画像や音楽といった素材ファイルの量において差異が生じていると考えられる．

6.2 研究設問 2

研究設問 2 の結果から，ゲーム開発者と GUI アプリケーション開発者が次の 4 種類に分類される事が判明した．

- コードカテゴリに属するファイルに多くコミットをしている開発者
- 素材カテゴリに属するファイルに多くコミットをしている開発者
- 両方のカテゴリに属するファイルに対し一定の割合でコミットをしている開発者
- 上記に分類されない開発者

上記で挙げた開発者の中で，Pascarella らの研究では，非ゲーム開発プロジェクトにおいて素材カテゴリに属するファイルへ多くコミットをしている開発者の存在は確認されなかったとしている．しかし，本研究では，非ゲーム開発プロジェクトである GUI アプリケーション開発プロジェクトにおいてもそのような特徴を持つ開発者の存在を発見した．これは GUI アプリケーション開発プロジェクトとゲーム開発プロジェクトの間の類似点といえる．

なお，この特徴を持つ開発者が存在しているゲーム開発プロジェクトには，含まれる素材ファイルの割合に偏りは見られなかった．この傾向は GUI アプリケーション開発プロジェクトでも同様に見られる．例えば，ninja のように素材ファイルの割合が 40%を超えているものもあれば，roundcubeemail のように 10%を切るものもあった．以上のことから，含まれる素材ファイルの割合によらず，どちらのドメインにおいてもこうした特徴を持つ開発者が存在し得ると考えられる．

また，加えて得られた知見として，どちらのドメインにおいても，コードファイルと素材ファイル双方に一定の割合でコミットを行っている開発者の存在が確認された．こうした特徴を持つ開発者は双方のドメインにおいて 25%程度存在しており，同様に双方のドメインにおける類似点と考えられる．

6.3 研究設問 3

ゲーム開発プロジェクトと GUI アプリケーション開発プロジェクトの間では，テストが書かれているリポジトリの割合がほとんど変わらないという結果となった．また，どちらのソフトウェア開発の場合においても，全体のリポジトリ数に対しテストが書かれているリポジトリの割合が 20%程度に留まった．この事から，ゲーム開発プロジェクトと GUI アプリケーション開発プロジェクトは，どちらも同様に単体テストが書かれにくい傾向にあると考えられる．

Murphy-Hill らの研究ではゲーム開発プロジェクトで自動テストが書かれにくい理由として，ゲームではユーザーインターフェースがソフトウェアと切り分けられない特徴を持つ事を挙げている．しかし，この特徴は，同様に特徴的な入力デバイスを使用する事から GUI アプリケーション開発プロジェクトにも見られる可能性がある．以上の事から GUI アプリケーション開発プロジェクトにおいても自動テストが書かれにくくなっていると考えられ，ゲーム開発プロジェクトとの類似性を示唆している．

なお，Murphy-Hill らの研究はゲーム開発プロジェクトにおいてテストがあまり書かれない理由の一つに，ゲームが持つ複雑性を挙げている．この複雑性とは，ゲームのソースコードに含まれるオブジェクトに関し，オブジェクトの持つ状態数が非常に多くなる事や，オブジェクト間のやりとりが頻繁に行われる事を指す．こうした特徴は通常，GUI アプリケーションでは見られない特徴であると考えられる．しかし，本研究の結果ではゲーム開発プロジェクトの方が GUI アプリケーション開発プロジェクトよりもテストが書かれにくいというような差異は確認されなかった．これは，ゲームの持つ単体テストがあまり書かれないという性質において，他の問題よりも上記で述べたインターフェースの抱える問題に主な要因がある事を示唆している．

7. 妥当性への脅威

7.1 内的妥当性

内的妥当性の一つ目は，ファイル分類の妥当性である．本研究は Pascarella らの手法 [2] を参考にリポジトリ上のファイルを分類したが，十分に分類できているとは言えない点が存在する．例えば xml ファイルのような様々な役割を持ち得るファイルに関しては，プロジェクト内で何らかの役割を担っているにも関わらずその他カテゴリに入れざるを得ない状況があった．

二つ目は，開発者分類の妥当性である．今回は全てのファイルにおけるコミット履歴を利用して開発者を分類した．しかし，詳細にコミットの内容を精査している訳では無い．その為，各コミットにおける修正行数の違いや，そのプロジェクトが抱える様々な経緯によって特定の履歴が消失している可能性を考慮できていない．

7.2 外的妥当性

外的妥当性の一つ目は，収集したサンプル数の妥当性である．今回は 44 件の GUI アプリケーション開発プロジェクトと 65 件のゲーム開発プロジェクトを収集したが，この数は今回の結論を一般的に主張するには不足しているといえる．

二つ目は，今回収集したプロジェクトが全てオープンソースで提供されているプロジェクトであり，オープンソースプロジェクト特有の文化が含まれている可能性がある事である．例えば，オープンソースなゲームプロジェクトの場合，リポジトリではコードファイルのみを公開し，その他の素材ファイルに関してはリポジトリに含めない運用をしているプロジェクトが存在する．以上の理由から，オープンソースプロジェクトとして提供されていないプロジェクトに対してこの結論を適用する事は難しい可能性がある．また，サンプルの取得に OpenHub と GitHub というサービスを利用したが，それぞれのサービスに含まれるサンプルについて，そのサービス固有の文化を含む可能性がある．その為，これらのサービスを利用せずにリポジトリを管理しているプロジェクトについて今回の研究結果を適用することは難しい可能性がある．

7.3 構成概念妥当性

構成概念妥当性として，テストコード検出の妥当性が挙げられる．今回はPythonを用いて書かれたコードが存在するリポジトリにのみ対象を絞り，標準モジュールであるunittestを用いてテストを行っているリポジトリのみを検出している．その為，unittest以外のモジュールを用いて単体テストを行っているようなリポジトリが存在したとしても，今回の調査ではテストコードが存在しないリポジトリとして扱われている．

8. 結言

8.1 本研究のまとめ

本研究の目的は，ゲーム開発プロジェクトとGUIアプリケーション開発プロジェクトを比較することで，ゲーム開発と非ゲームソフトウェア開発との間に類似点が存在する可能性を明らかにすることである．64件のゲーム開発プロジェクトと44件のGUIアプリケーション開発プロジェクトのリポジトリに対し様々な調査をし，その結果，各ドメイン間の差異と類似点を発見した．

差異としては，リポジトリに含まれる素材ファイルの割合の違いが挙げられる．ゲーム開発プロジェクトはGUIアプリケーション開発プロジェクトと比較しても，より多くの素材ファイルが含まれていた．

また，本研究の結果は，差異だけではなく，類似点も示している．その一つは，開発者の特徴に関するものである．どちらのドメインにおいても，コードファイルと素材ファイルの両方にコミットしている開発者や，素材ファイルにより多くコミットする開発者が確認された．

他にも，テストコードを含むリポジトリが少ないという類似点が確認された．これは，Murphy-Hillらがゲーム開発において指摘しているユーザーインターフェースが分離できないという特徴が，GUIアプリケーション開発にも同様に当てはまる事が理由として考えられる．

上記の結果は，調査対象のドメインを絞る事で，ゲーム開発と非ゲームソフトウェア開発プロジェクトとの間にも類似点が発見できる事を示している．このことから，ゲーム開発と非ゲームソフトウェア開発との差異を前提とした過去の研究において，疑問の余地があるといえる．

8.2 今後の課題

本研究における今後の展開には，二つの方針が考えられる．

一つ目は，本研究で明らかとなったゲーム開発プロジェクトとGUIアプリケーション開発プロジェクトの類似点についてより深く追及する方針である．例えば，テストコードがあまり書かれない点について，それぞれのプロジェクトに関わる開発者

がどのような認識を持っているかを調査する．他にも，双方の共通点から，各ドメインに跨るようなテストコード生成ツールの開発といった研究も考えられる．

二つ目は，今回対象としたゲームおよびGUIアプリケーション以外のドメインについて詳細に調査する方針である．本研究のようにデータベースやCUIアプリケーションといった非ゲームソフトウェア開発とゲーム開発を比較する事は．興味深い類似点を発見する可能性がある．

謝辞

本研究を行うにあたり，研究設問の設定や研究に対する姿勢，本報告書の作成に至るまで，全ての面で丁寧なご指導を頂きました，本学情報工学・人間科学系水野修教授，崔恩瀞助教に厚く御礼申し上げます．本報告書執筆にあたり貴重な助言を多数頂きました，本学設計工学専攻近藤将成先輩，西浦生成先輩，本学情報工学専攻塩津拓真君，國領正真君，情報工学課程 杉浦智基君，山本凱君をはじめとする，ソフトウェア工学研究室の皆さん，学生生活を通じて著者の支えとなった家族や友人に深く感謝致します．

参考文献

[1] T.Z. Emerson Murphy-Hill and N. Nagappan, “Cowboys, ankle sprains, and keepers of quality: How is video game development different from software development?,” ICSE, pp.1–11, May 2014.

[2] L. Pascarella, F. Palomba, M.D. Penta, and A. Bacchelli, “How is video game development different from software development in open source?,” MSR, p.392–402, March 2018.

[3] C. Politowski, D.D. Vargas, L.M. Fontoura, and A. Foletto, “Software Engineering Processes in Game Development: a Survey about Developers’ Experiences,” in Proceedings of SBGames 2016, pp.154–161, July 2016.

[4] C. Politowski, L. Fontoura, F. Petrillo, and Y.-G. Guéhéneuc, “Learning from the past: a process recommendation system for video game projects using postmortems experiences,” 04 2018.

[5] S. Aleem, L. Capretz, and F. Ahmed, “Game development software engineering process life cycle: A systematic review,” Journal of Software Engineering Research and Development, vol.4, pp.1–30, Dec. 2016.

[6] A. Schweda, D. Winkler, S. Biffl, and J. Musil, “A survey on the state of the practice in video game software development,” 03 2010.

[7] R. Al-azawi, A. Ayesh, and M.A. Obaidy, “Towards agent-based agile approach for game development methodology,” 2014 World Congress on Computer Applications and Information Systems (WCCAIS), pp.1–6, Jan. 2014.

[8] D. Callele, E. Neufeld, and K. Schneider, “Requirements engineering and the creative process in the video game industry,” 13th IEEE International Conference on Requirements Engineering (RE’05), pp.240–250, 2005.

[9] C. Politowski, L. Fontoura, F. Petrillo, Y.-G. Gueh´ eneuc., “Are the old days gone?: A survey on actual software engineering processes in video game industry,” 2016.

[10] A. Godoy and E.F. Barbosa, "Game-Scrum: An Approach to Agile Game Development," SBCGames, pp.292–295, 2010.

[11] C.M. Kanode and H.M. Haddad, "Software engineering challenges in game development," 2009 Sixth International Conference on Information Technology: New Generations, pp.260–265, April 2009.

[12] V. Schetinger, C. Souza, L.M. Fontoura, and C.T. Pozzer, "User stories as actives for game development," SBGames 2011, pp.1–4, 2011.

[13] T. Tschang, "Videogames as interactive experiential products and their manner of development," International Journal of Innovation Management (ijim), vol.09, pp.103–131, 03 2005.

[14] A. Ampatzoglou and I. Stamelos, "Software engineering research for computer games: A systematic review," Information and Software Technology, vol.52, pp.888–901, 09 2010.

[15] F. Petrillo, M. Pimenta, F. Trindade, and C. Dietrich, "What went wrong? A survey of problems in game development," Computers in Entertainment, vol.7, no.1, pp.1–22, 2009.

[16] F. Petrillo, M. Pimenta, F. Trindade, and C. Dietrich, "Houston, we have a problem...: A survey of actual problems in computer games development," Proceedings of the 2008 ACM Symposium on Applied Computing, p.707–711, SAC '08, Association for Computing Machinery, New York, NY, USA, 2008.

[17] F. Petrillo and M. Pimenta, "Is agility out there? agile practices in game development," Proceedings of the 28th ACM International Conference on Design of Communication, p.9–15, SIGDOC '10, Association for Computing Machinery, New York, NY, USA, 2010.

[18] B. Duck, "Openhub," 2019-01-14. https://www.openhub.net.

[19] GitHub, "Github," 2019-01-14. https://github.com.

[20] C. Bird, N. Nagappan, B. Murphy, H. Gall, and P. Devanbu, "Don't touch my code! examining the effects of ownership on software quality," Proceedings of the 19th ACM

SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering, p.4–14, ESEC/FSE '11, Association for Computing Machinery, New York, NY, USA, 2011.