

1. Submit to JESTEC (12 Aug 2021)
2. First Revision (8 Sept 2021)
3. Submit Revised Paper (18 Sept 2021)
4. Accepted Submission (25 Sept 2021)
5. Paper Publish (05 Oct 2023)

1. Submit to JESTEC (12 Aug 2021)



Adam Mukharil Bachtiar

[<adam@email.unikom.ac.id>](mailto:adam@email.unikom.ac.id)

[JESTEC] Manuscript ID: JESTEC_16_5_40_2021_A M Bachtiar–Submission Received

1 message

Editorial Office [<jjestec@gmail.com>](mailto:jjestec@gmail.com)
Adam Mukharil Bachtiar [<adam@email.unikom.ac.id>](mailto:adam@email.unikom.ac.id)

12 August 2021 at 13:15To:

Dear Author,

Thank you very much for uploading the following manuscript to the submission system. One of our editors will be in touch with you soon.

Journal Name	Journal of Engineering, Science and Technology
Manuscript ID	JESTEC_16_5_40_2021_Adam Mukharil Bachtiar
Authors	Adam Mukharil Bachtiar, Dian Dharmayanti, Rakhmat Sabarudin
Title	Development of Class Library in Domain of Scrolling Shooter Game

If you have any questions, please do not hesitate to contact editorial office.

Kind regards,

Editorial Office
Journal of Engineering, Science and Technology
E-Mail: jjestec@gmail.com

*** This is an automatically generated email ***

Development of Class Library in Domain of Scrolling Shooter Game

Adam Mukharil Bachtiar*, Dian Dharmayanti, Rakhmat Sabarudin

Informatics Engineering, Faculty of Engineering and Computer Science, Universitas Komputer Indonesia, Jalan Dipatiukur
Nomor 112-116, Bandung, Jawa Barat, Indonesia

Corresponding Author adam@email.unikom.ac.id

Abstract. The purpose of this research is to develop a Class Library in the domain of Scrolling Shooter games. The built of this Class Library will be used to develop Scrolling Shooter games as the result of this research. The development of a Class Library is done through seven phases, Domain Analysis, Frozenspot Analysis, Hotspot Analysis, Modelling Class Library, Implementation Class Library, Unit Testing, and Integration Testing, and those will be the main discussion of this research. Furthermore, the conclusion of this research is expected to help game Programmers to develop Scrolling Shooter games, without having to make the whole system functional from scratch. This research also gives a contribution to software engineering concepts, namely: reusable code, abstraction, and class design.

1. Introduction

Based on data obtained from Newzoo (an intelligence market provider company in the global market), in 2017 the Indonesian gaming industry ranked 16th in the list of gaming markets in the world with several players reaching 43.7 million with potential income reaching around IDR. 11.9 trillion [1]. One of the popular game genres in Indonesia in January 2018 is Scrolling Shooter, it is noted that Scrolling Shooter won the top 10 on the Google Play Store in the January 2018 period, sorted by the Arcade game category. With this positive trend, game developers will automatically be interested in building a Scrolling Shooter game going forward.

Based on the results of an interview with one of Programmer which had been developed a Scrolling Shooter game, explained there is some obstacle during developing a Scrolling Shooter game was that the Programmer had to recreate the functional system from scratch. This will certainly lead to the duration of the development of the Scrolling Shooter game which resulted in the game not being released following the target time. Especially if you notice that the game industry is always following trends that require developers to be able to release games at the right time.

Based on these problems, it is necessary to build a Class Library to help programmers build Scrolling Shooter games. A Class Library is a group of classes that provide the basic functionalities of a domain and can be used directly by programmers. With the Class Library, programmers can reuse functional requirements without having to build them from scratch. Therefore, the Class Library that will be created in this study is expected to provide basic functionalities that are following the needs of programmers in building Scrolling Shooter games.

2. Method

This research belongs to the type of applied research using a quantitative approach. Applied research aims to find a solution to deal with the problems of society or business organizations [2]. While the quantitative approach is used because research starts from a hypothesis and is objective. The methodology used in this study refers to the journal "Development of Class Library in Domain of Scrolling Shooter Game" by adding the implementation stages as seen in Figure 1.

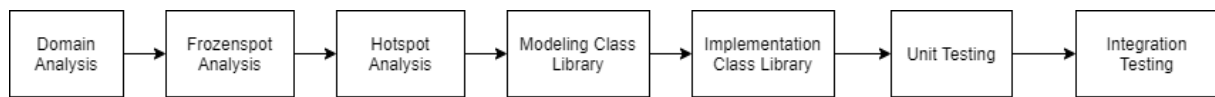


Figure 1. Research methodology in this research.

Here are the explanations of each stage:

1. Domain Analysis

Domain Analysis is performed to determine the characteristics of the Scrolling Shooter game domain. The characteristics obtained are used to determine the uniqueness of the Scrolling Shooter game domain with other game domains. As it is known that the characteristics of a domain, can be obtained from the domain knowledge. The domain knowledge can cover various aspects, ranging from electronic systems, banking to multimedia applications such as games [3]. In this research, existing game applications will be used as domain knowledge.

2. Frozenspot Analysis

Frozenspot Analysis is performed to determine the sliced function of several samples of existing Scrolling Shooter games. The slice will determine whether the function is mandatory or optional. In this research, Frozenspot Analysis is divided into two stages, namely Function Identification and Frozenspot Identification.

3. Hotspot Analysis

Hotspot Analysis is performed to identify and define hotspot points of the Scrolling Shooter game application domain. The hotspot is a collection of functions that are always present in the application domain and must not be removed, which will later represent functions that will be provided by the Class Library [4]. In this research, Hotspot Analysis is divided into two stages, namely Hotspot Identification and Hotspot Definition.

4. Modeling Class Library

Modeling Class library is performed to transform the hotspot into the object model. The object model serves to describe the design of objects and classes into diagrammatic form, using object-based modeling [5]. The results of the modeling can simplify the implementation process into the UML (Unified Modeling Language). In this research, the Class Library design consists of two stages, namely Class Design and Package Design.

5. Implementation Class Library

Implementation Class Library is a stage that is done to transform the object model into a programming language.

6. Unit Testing

Unit Testing stage is carried out to find out that each unit or component of the function in the Class Library can functionally work as expected [6]. A Unit Testing is done by creating several test cases of the method for each class of the Class Library which has been implemented.

7. Integration Testing

Integration Testing is done by directly testing each unit in the Class Library to create a prototype Scrolling Shooter game. Also, It tested directly by a Game Programmer.

3. Results and Discussion

All stages in the research methodology are explained specifically in this section.

1. Domain Analysis

Domain Analysis can be done by determining the source of information used to identify the characteristics of the application domain. There are several sources of information that can be used, including domain experts, textbooks, domain standards, and previous application samples [7]. There are four sample Scrolling Shooter game applications from the Google Play Store that will be used to assist the identification process. A characteristic of the Scrolling Shooter game that used for this research is

players generally play as pilot fighter planes, which are assigned to withstand from any enemy attacks [8]. The sample application consists of two types of vertical Scrolling Shooter games and two types of horizontal scrolling shooter games, which are selected based on the highest number of downloads and the highest rating on the Google Play Store in the January 2018 period. The four sample game applications will be given symbols to facilitate calling the application name in the identification process, which can be seen in Table 1.

Table 1. Four Sample Game Scrolling Shooter

Function	Game	Symbol
Space Shooter: Galaxy Attack	Vertical	G1
Galaxy Attack: Alien Shooter	Vertical	G2
Scrambler – Classic 80s Arcade Game	Horizontal	G3
R-TYPE	Horizontal	G4

Based on the identification results, four aspects become a review of the characteristics of the domain, namely the camera viewpoint, game presentation, character control and the basic mechanism of the game. Here are the results of identifying the characteristics obtained from the four sample Scrolling Shooter game applications:

a. Camera Viewpoint

In G1 and G2 games, the camera's perspective is taken from the top to down (top-down view), where only the top side of the object is seen by the player when playing the game. While in the G3 and G4 games the camera's perspective is taken from the side, where only the side of the object is seen by the player. Viewpoints of the four games.

b. Game Presentation

The presentation or display style used in the G1 and G2 games is vertical scrolling. Vertical Scrolling is a display style where the background of the game is seen moving automatically with a vertical orientation [9]. Meanwhile, in G3 and G4 games the presentation used is horizontal scrolling. Horizontal Scrolling is a display style where the background in the game moves automatically with a horizontal orientation.

c. Character Control

The four sample Scrolling Shooter game applications that are played show that the character controlled by the player can move with an eight-way mechanism (right, left, up, down, top right, bottom right, top left, and bottom left) on the x and y axes. The character movement controlled by the player is only limited to the screen area.

Also, there are two different mechanisms when players control character objects, namely control with a fixed mechanism and control with a relative mechanism. In fixed mechanism control, the screen position touched by the player has no distance from the character object being controlled. Whereas in the relative mechanism, there is a distance between the character object being controlled and the position of the screen being touched.

d. Basic Mechanic

In the four sample games that are played, players must be able to survive the pattern of attacks fired by the enemy to get the maximum score. To survive the enemy attack, players can destroy enemies by firing projectiles or avoiding incoming attacks. Also, there are power-ups that players can use to increase attack power.

The characteristics of the four sample Scrolling Shooter game applications will be used to identify the functions contained in the Scrolling Shooter game domain in the FrozenSpot Analysis stage.

2. Frozenspot Analysis

Frozenspot analysis is divided into two stages, namely Function Identification and Frozenspot Identification.

a. Functional Identification

This stage is done to identify the collection or set of functions that exist in the Scrolling Shooter game domain, based on four previous application samples. The resulting set of functions will be used to determine the function slices of the Scrolling Shooter game domain. There are nine functions of the results of identification that are as follows.

- 1)Character Control
- 2)Scrolling Background
- 3)Spawn Object
- 4)Destroy Object
- 5)Move Object
- 6)Shoot Object With Pattern
- 7)Change Attribute
- 8)Display Attribute
- 9)Save Data

The results of the function identification table show that there are nine sets of functions in the domain of the Scrolling Shooter game application that will be used as a reference in determining the function slices.

b. Frozenspot Identification

This stage is carried out to determine the sliced function of the four previous Scrolling Shooter game application samples. The set of functions that have been obtained, will be used as a reference in determining the function slices. Frozenspot Identification results from four sample Scrolling Shooter game applications based on nine functions that have been obtained can be seen in Table 2.

From the results of the Frozenspot Analysis, it can be seen that there are nine functions contained in the Scrolling Shooter game application domain. The nine functions are slices of the four sample Scrolling Shooter game applications G1, G2, G3, and G4. The results of the sliced function will be used to determine the hotspot that is done in the next stage.

Table 2. Frozenspot Identification Result

Function	Game
Character Control	G1, G2, G3, G4
Scrolling Background	G1, G2, G3, G4
Spawn Object	G1, G2, G3, G4
Destroy Object	G1, G2, G3, G4
Move Object	G1, G2, G3, G4
Shoot Object with Pattern	G1, G2, G3, G4
Change Attribute	G1, G2, G3, G4
Display Attribute	G1, G2, G3, G4
Save Data	G1, G2, G3, G4

3. Hotspot Analysis

Hotspot analysis is divided into two stages, namely Hotspot Identification and Hotspot Definition.

a. Hotspot Identification

This stage is carried out to identify functions that can be used as hotspots in the Scrolling Shooter game application domain. The sliced function obtained from the frozen-spot analysis results can determine whether the function is suitable as a hotspot or not. In this study, Hotspot Identification uses the Blackbox method where the function used as a hotspot is mandatory. Therefore, the hotspot

identification results show that there are nine mandatory functions used as hotspots, as shown in Table 2.

b. Defining Hotspot

This stage is carried out to define each function obtained from the Hotspot Identification results. Defining Hotspot is done by describing the behavior that can be performed by the function, using a hotspot card [10]. Then with the object model transformation rules, the hotspot card will be converted into object models in the form of class diagrams. The hotspot card from the Player Character Control function and the object model formed can be seen in Figure 2.

Hotspot Name CharacterControl Degree of Flexibility <input checked="" type="checkbox"/> <i>adaptation without restart</i> <input type="checkbox"/> <i>adaptation with by end user</i>
General Description This function is used to move the character object in eight directions, based on the input in the form of a touch from the player.
Examples of implementation, at least two different situations 1. Control of character objects using a fixed mechanism, where the character object will always move to follow the position of the screen touched by the player without any difference in distance between the position of the character object with the position of the screen touched. 2. Character object control uses a relative mechanism, where the character object will always move to follow the position of the screen touched by the player with the possibility of the difference in distance between the position of the character object and the position of the screen touched.

Figure 2. Hotspot Card Character Control

Based on the rules of the object model transformation, the hotspot design will be created by adding the hook method in a separate hook class. The results of the transformation of the object model of the hotspot Controlling Player Character can be seen in Figure 3.

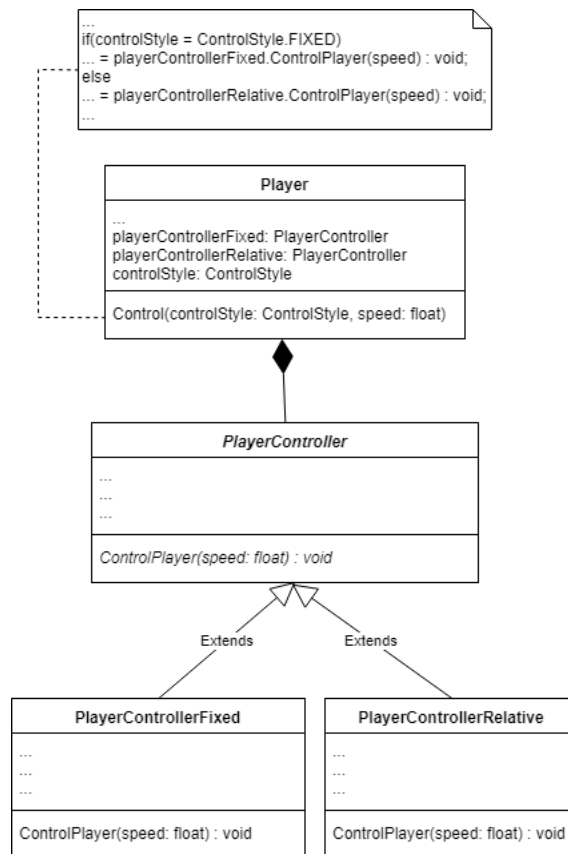


Figure 3. Object Transformation Model *Hotspot* Character Control

4. Modeling Class Library

Modeling class library design consists of two stages, namely class design and package design.

a. Class Modeling

This class diagram will illustrate the relationships between classes of hotspot Controlling Player Character, where the class contains a method used to move character objects with eight directions of movement according to player control. The class diagram of the hotspot controlling the player's character can be seen in Figure 4.

b. Package Modeling

This stage is done to describe the package diagram of the Class Library system that will be built. The package diagram created functions to find groups of related classes wrapped in one container or package. The results of the package design show that nine packages will be built in the Scrolling Shooter class library, as follows.

- a. ScrollingShooter2D.Player
- b. ScrollingShooter2D.Spawner
- c. ScrollingShooter2D.Shoot
- d. ScrollingShooter2D.DisplayUI
- e. ScrollingShooter2D.Destroyer
- f. ScrollingShooter2D.Data
- g. ScrollingShooter2D.Background
- h. ScrollingShooter2D.Attribute
- i. ScrollingShooter2D.Movement

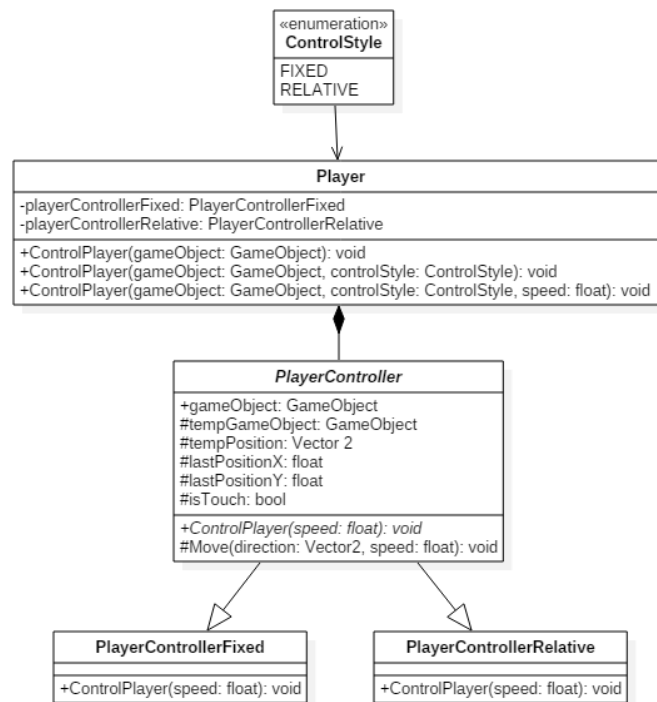


Figure 4. Class Diagram *Hotspot* Character Control

5. Implementation Class Library

Class Library will be implemented to C# (C Sharp) language. Naming identifier standards such as properties, classes and methods at the implementation stage will be adjusted to the C# language naming rules.

6. Unit Testing

Unit Testing is done by creating a test case method for each class Scrolling Shooter library function. The case test method is based on the design of the unit test scenario which can be seen in Table 3.

Table 3. Test Case Scenario

Class Test	Method Test	Status
Player	ControlPlayer()	Success
Background	ScrollBG()	Success
Spawner	Spawn()	Success
Data	Save()	Success
	Load()	Success
Destroyer	Destroy()	Success
DisplayUI	Display()	Success
Movement	Move()	Success
Shoot	Shoot()	Success
Player	ControlPlayer()	Success

The unit test results show that all methods in the Scrolling Shooter Class Library can function properly according to their functions.

7. Integration Testing

Integration testing will be tested directly by experienced programmers in building games using Unity. The prototype game display from the integration test results can be seen in Figure 5.



Figure 5. Prototype Game Scrolling Shooter

4. Conclusion

Based on all processes in this research, there are several conclusions, namely:

- a. Programmers can build Scrolling Shooter games without having to make the entire system functional from scratch using the Class Library. It is proven that the Class Library built in this research can produce a prototype of the Scrolling Shooter game.
- b. For further research, the sample of game application is expected to be able to use more than four application samples to know the characteristics that exist in the game domain more accurately. Also, it is hoped that the addition of other aspects will be used as a basis for conducting characteristics reviews. This is because the domain under study is a game application domain which certainly has its uniqueness when compared to other application domains.

References

- [1] Atanasova, N., Todorovski, L., Džeroski, S., & Kompore, B. (2006). Constructing a library of domain knowledge for automated modelling of aquatic ecosystems. *ecological modelling*, 194(1-3), 14-36.
- R. Maulana. "Tantangan dan Peluang Startup Game di Indonesia pada Tahun 2018,". Internet: <https://id.techinasia.com/tantangan-peluang-startup-game-indonesia-tahun2018>, Jan. 2, 2018 [Jan. 12, 2018].
- [2] Kothari, C. R. 2004. Research Methodology 2 rev.
- [3] Pressman, R. S. 2015. "Software Engineering—A Practitioner" s Approach", Mc Graw-Hill International Edition, 2010.
- [4] Bachtiar, A. M., & Bardansyah, A. 2017. Pembangunan Class Library untuk Domain Product Management di Aplikasi M-Commerce pada Android. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, 6(3), pp. 235-242.
- [5] Booch, G. (1994). Object-oriented analysis and design with applications. Number 2nd.
- [6] Jamalludin, H. Y., Winiati, S., & Riadi, I. 2018. Implementasi Test Driven Development Pada Pengembangan Aplikasi Android Untuk Mahasiswa Universitas Ahmad Dahlan. *Jurnal Ilmu Teknik Elektro Komputer dan Informatika (JITEKI)*, 4(1), pp. 43.
- [7] Bačíková, M. 2012. Domain analysis with reverse-engineering for gui feature models. In *POSTER 2012: 16th International Student Conference on Electrical Engineering*, 16, pp. 1-5).
- [8] Yuniarti, A., Wardanie, N. N., & Kuswardayan, I. 2018. Dynamic game balancing implementation using adaptive algorithm in mobile-based Safari Indonesia game. In *Journal of Physics: Conference Series*, 974(1), pp. 012049.
- [9] Lee, J. H., Karlova, N., Clarke, R. I., Thornton, K., & Perti, A. 2014. Facet analysis of video game genres. *IConference 2014 Proceedings*.
- [10] Mohamed, E. 1999. Fayad.; Douglas, C. Schmidt.; Ralph E. Johnson.: "Building Application Frameworks": Object-Oriented Foundations of Framework Design.

2. First Revision (8 Sept 2021)



Adam Mukharil Bachtiar
<adam@email.unikom.ac.id>

[JESTEC] Manuscript ID: JESTEC_2021_Adam Mukharil Bachtiar – Minor Revision

1 message

Editorial Office <jjestec@gmail.com>

8 September 2021 at 14:12

To: Adam Mukharil Bachtiar <adam@email.unikom.ac.id>

Dear Author,

Thank you for submitting the following manuscript to Journal of Engineering, Science and Technology:

Journal Name	Journal of Engineering, Science and Technology
Manuscript ID	JESTEC_16_5_40_2021_Adam Mukharil Bachtiar
Authors	Adam Mukharil Bachtiar, Dian Dharmayanti, Rakhmat Sabarudin
Title	Development of Class Library in Domain of Scrolling Shooter Game

It has been reviewed by experts in the field and we request that you make major revisions before it is processed further. Please find the comments from reviewer in the attached file.

Please revise the manuscript according to the reviewers' comments and upload the revised file within 10 days. Use the version of your manuscript found at the above link for your revisions, as the editorial office may have made formatting changes to your original submission.

If the reviewers have suggested that your manuscript should undergo extensive English editing, please address this during revision. We suggest that you have your manuscript checked by a native English-speaking colleague or use a professional English editing service.

Do not hesitate to contact us if you have any questions regarding the revision of your manuscript or if you need more time. We look forward to hearing from you soon.

Kind regards,

Editorial Office
Journal of Engineering, Science and Technology
E-Mail: jjestec@gmail.com

Development of Class Library in Domain of Scrolling Shooter Game

Adam Mukharil Bachtiar*, Dian Dharmayanti, Rakhmat Sabarudin

Informatics Engineering, Faculty of Engineering and Computer Science, Universitas Komputer Indonesia, Jalan Dipatiukur
Nomor 112-116, Bandung, Jawa Barat, Indonesia

Corresponding Author adam@email.unikom.ac.id

Abstract. The purpose of this research is to develop a Class Library in the domain of Scrolling Shooter games. The built of this Class Library will be used to develop Scrolling Shooter games as the result of this research. The development of a Class Library is done through seven phases, Domain Analysis, Frozenspot Analysis, Hotspot Analysis, Modelling Class Library, Implementation Class Library, Unit Testing, and Integration Testing, and those will be the main discussion of this research. Furthermore, the conclusion of this research is expected to help game Programmers to develop Scrolling Shooter games, without having to make the whole system functional from scratch. This research also gives a contribution to software engineering concepts, namely: reusable code, abstraction, and class design.

Commented [a1]: Add method in your abstract

1. Introduction

Based on data obtained from Newzoo (an intelligence market provider company in the global market), in 2017 the Indonesian gaming industry ranked 16th in the list of gaming markets in the world with several players reaching 43.7 million with potential income reaching around IDR. 11.9 trillion [1]. One of the popular game genres in Indonesia in January 2018 is Scrolling Shooter, it is noted that Scrolling Shooter won the top 10 on the Google Play Store in the January 2018 period, sorted by the Arcade game category. With this positive trend, game developers will automatically be interested in building a Scrolling Shooter game going forward.

Based on the results of an interview with one of Programmer which had been developed a Scrolling Shooter game, explained there is some obstacle during developing a Scrolling Shooter game was that the Programmer had to recreate the functional system from scratch. This will certainly lead to the duration of the development of the Scrolling Shooter game which resulted in the game not being released following the target time. Especially if you notice that the game industry is always following trends that require developers to be able to release games at the right time.

Based on these problems, it is necessary to build a Class Library to help programmers build Scrolling Shooter games. A Class Library is a group of classes that provide the basic functionalities of a domain and can be used directly by programmers. With the Class Library, programmers can reuse functional requirements without having to build them from scratch. Therefore, the Class Library that will be created in this study is expected to provide basic functionalities that are following the needs of programmers in building Scrolling Shooter games.

Commented [a2]: Make it into one paragraph, add more describe related to the title in your introduction

Commented [a3]: For all avoid to use future tense in your journal, such as will, would, shall

Commented [a4]: Add the previous study (state of the art) in paragraph 2

Commented [a5]: Add the purpose and method in the last paragraph of introduction

2. Method

This research belongs to the type of applied research using a quantitative approach. Applied research aims to find a solution to deal with the problems of society or business organizations [2]. While the quantitative approach is used because research starts from a hypothesis and is objective. The methodology used in this study refers to the journal "Development of Class Library in Domain of Scrolling Shooter Game" by adding the implementation stages as seen in Figure 1.

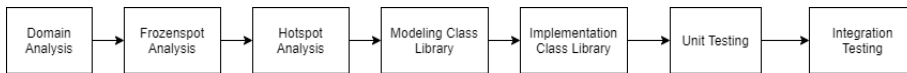


Figure 1. Research methodology in this research.

Here are the explanations of each stage:

1. Domain Analysis

Domain Analysis is performed to determine the characteristics of the Scrolling Shooter game domain. The characteristics obtained are used to determine the uniqueness of the Scrolling Shooter game domain with other game domains. As it is known that the characteristics of a domain, can be obtained from the domain knowledge. The domain knowledge can cover various aspects, ranging from electronic systems, banking to multimedia applications such as games [3]. In this research, existing game applications will be used as domain knowledge.

2. Frozenspot Analysis

Frozenspot Analysis is performed to determine the sliced function of several samples of existing Scrolling Shooter games. The slice will determine whether the function is mandatory or optional. In this research, Frozenspot Analysis is divided into two stages, namely Function Identification and Frozenspot Identification.

3. Hotspot Analysis

Hotspot Analysis is performed to identify and define hotspot points of the Scrolling Shooter game application domain. The hotspot is a collection of functions that are always present in the application domain and must not be removed, which will later represent functions that will be provided by the Class Library [4]. In this research, Hotspot Analysis is divided into two stages, namely Hotspot Identification and Hotspot Definition.

4. Modeling Class Library

Modeling Class library is performed to transform the hotspot into the object model. The object model serves to describe the design of objects and classes into diagrammatic form, using object-based modeling [5]. The results of the modeling can simplify the implementation process into the UML (Unified Modeling Language). In this research, the Class Library design consists of two stages, namely Class Design and Package Design.

5. Implementation Class Library

Implementation Class Library is a stage that is done to transform the object model into a programming language.

6. Unit Testing

Unit Testing stage is carried out to find out that each unit or component of the function in the Class Library can functionally work as expected [6]. A Unit Testing is done by creating several test cases of the method for each class of the Class Library which has been implemented.

7. Integration Testing

Integration Testing is done by directly testing each unit in the Class Library to create a prototype Scrolling Shooter game. Also, It tested directly by a Game Programmer.

Commented [a6]: Make it into narrative way

3. Results and Discussion

All stages in the research methodology are explained specifically in this section.

1. Domain Analysis

Domain Analysis can be done by determining the source of information used to identify the characteristics of the application domain. There are several sources of information that can be used, including domain experts, textbooks, domain standards, and previous application samples [7]. There are four sample Scrolling Shooter game applications from the Google Play Store that will be used to assist the identification process. A characteristic of the Scrolling Shooter game that used for this research is

players generally play as pilot fighter planes, which are assigned to withstand from any enemy attacks [8]. The sample application consists of two types of vertical Scrolling Shooter games and two types of horizontal scrolling shooter games, which are selected based on the highest number of downloads and the highest rating on the Google Play Store in the January 2018 period. The four sample game applications will be given symbols to facilitate calling the application name in the identification process, which can be seen in Table 1.

Table 1. Four Sample Game Scrolling Shooter

Function	Game	Symbol
Space Shooter: Galaxy Attack	Vertical	G1
Galaxy Attack: Alien Shooter	Vertical	G2
Scrambler – Classic 80s Arcade Game	Horizontal	G3
R-TYPE	Horizontal	G4

Commented [a7]: •for all table use JESTEC format

Based on the identification results, four aspects become a review of the characteristics of the domain, namely the camera viewpoint, game presentation, character control and the basic mechanism of the game. Here are the results of identifying the characteristics obtained from the four sample Scrolling Shooter game applications:

a. Camera Viewpoint

In G1 and G2 games, the camera's perspective is taken from the top to down (top-down view), where only the top side of the object is seen by the player when playing the game. While in the G3 and G4 games the camera's perspective is taken from the side, where only the side of the object is seen by the player. Viewpoints of the four games.

b. Game Presentation

The presentation or display style used in the G1 and G2 games is vertical scrolling. Vertical Scrolling is a display style where the background of the game is seen moving automatically with a vertical orientation [9]. Meanwhile, in G3 and G4 games the presentation used is horizontal scrolling. Horizontal Scrolling is a display style where the background in the game moves automatically with a horizontal orientation.

c. Character Control

The four sample Scrolling Shooter game applications that are played show that the character controlled by the player can move with an eight-way mechanism (right, left, up, down, top right, bottom right, top left, and bottom left) on the x and y axes. The character movement controlled by the player is only limited to the screen area.

Also, there are two different mechanisms when players control character objects, namely control with a fixed mechanism and control with a relative mechanism. In fixed mechanism control, the screen position touched by the player has no distance from the character object being controlled. Whereas in the relative mechanism, there is a distance between the character object being controlled and the position of the screen being touched.

Commented [a8]: Make it into one paragraph

d. Basic Mechanic

In the four sample games that are played, players must be able to survive the pattern of attacks fired by the enemy to get the maximum score. To survive the enemy attack, players can destroy enemies by firing projectiles or avoiding incoming attacks. Also, there are power-ups that players can use to increase attack power.

The characteristics of the four sample Scrolling Shooter game applications will be used to identify the functions contained in the Scrolling Shooter game domain in the Frozenspot Analysis stage.

Commented [a9]: Make it into one paragraph

2. Frozenspot Analysis

Frozenspot analysis is divided into two stages, namely Function Identification and Frozenspot Identification.

a. Functional Identification

This stage is done to identify the collection or set of functions that exist in the Scrolling Shooter game domain, based on four previous application samples. The resulting set of functions will be used to determine the function slices of the Scrolling Shooter game domain. There are nine functions of the results of identification that are as follows.

- 1) Character Control
- 2) Scrolling Background
- 3) Spawn Object
- 4) Destroy Object
- 5) Move Object
- 6) Shoot Object With Pattern
- 7) Change Attribute
- 8) Display Attribute
- 9) Save Data

The results of the function identification table show that there are nine sets of functions in the domain of the Scrolling Shooter game application that will be used as a reference in determining the function slices.

b. Frozenspot Identification

This stage is carried out to determine the sliced function of the four previous Scrolling Shooter game application samples. The set of functions that have been obtained, will be used as a reference in determining the function slices. Frozenspot Identification results from four sample Scrolling Shooter game applications based on nine functions that have been obtained can be seen in Table 2.

From the results of the Frozenspot Analysis, it can be seen that there are nine functions contained in the Scrolling Shooter game application domain. The nine functions are slices of the four sample Scrolling Shooter game applications G1, G2, G3, and G4. The results of the sliced function will be used to determine the hotspot that is done in the next stage.

Commented [a10]: Make it into one paragraph

Table 2. Frozenspot Identification Result

Function	Game
Character Control	G1, G2, G3, G4
Scrolling Background	G1, G2, G3, G4
Spawn Object	G1, G2, G3, G4
Destroy Object	G1, G2, G3, G4
Move Object	G1, G2, G3, G4
Shoot Object with Pattern	G1, G2, G3, G4
Change Attribute	G1, G2, G3, G4
Display Attribute	G1, G2, G3, G4
Save Data	G1, G2, G3, G4

3. Hotspot Analysis

Hotspot analysis is divided into two stages, namely Hotspot Identification and Hotspot Definition.

a. Hotspot Identification

This stage is carried out to identify functions that can be used as hotspots in the Scrolling Shooter game application domain. The sliced function obtained from the frozen-spot analysis results can determine whether the function is suitable as a hotspot or not. In this study, Hotspot Identification uses the Blackbox method where the function used as a hotspot is mandatory. Therefore, the hotspot

identification results show that there are nine mandatory functions used as hotspots, as shown in Table 2.

b. Defining Hotspot

This stage is carried out to define each function obtained from the Hotspot Identification results. Defining Hotspot is done by describing the behavior that can be performed by the function, using a hotspot card [10]. Then with the object model transformation rules, the hotspot card will be converted into object models in the form of class diagrams. The hotspot card from the Player Character Control function and the object model formed can be seen in Figure 2.

Hotspot Name Character Control Degree of Flexibility <input checked="" type="checkbox"/> <i>adaptation without restart</i> <input type="checkbox"/> <i>adaptation with by end user</i>
General Description This function is used to move the character object in eight directions, based on the input in the form of a touch from the player.
Examples of implementation, at least two different situations 1. Control of character objects using a fixed mechanism, where the character object will always move to follow the position of the screen touched by the player without any difference in distance between the position of the character object with the position of the screen touched. 2. Character object control uses a relative mechanism, where the character object will always move to follow the position of the screen touched by the player with the possibility of the difference in distance between the position of the character object and the position of the screen touched.

Figure 2. Hotspot Card Character Control

Based on the rules of the object model transformation, the hotspot design will be created by adding the hook method in a separate hook class. The results of the transformation of the object model of the hotspot Controlling Player Character can be seen in Figure 3.

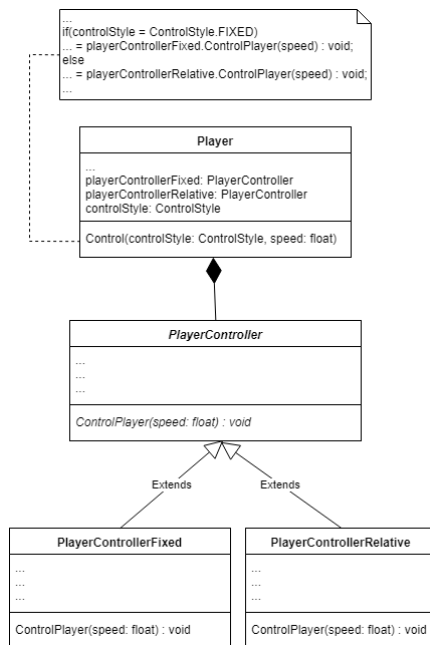


Figure 3. Object Transformation Model *Hotspot* Character Control

4. Modeling Class Library

Modeling class library design consists of two stages, namely class design and package design.

a. Class Modeling

This class diagram will illustrate the relationships between classes of hotspot Controlling Player Character, where the class contains a method used to move character objects with eight directions of movement according to player control. The class diagram of the hotspot controlling the player's character can be seen in Figure 4.

b. Package Modeling

This stage is done to describe the package diagram of the Class Library system that will be built. The package diagram created functions to find groups of related classes wrapped in one container or package. The results of the package design show that nine packages will be built in the Scrolling Shooter class library, as follows.

- a. ScrollingShooter2D.Player
- b. ScrollingShooter2D.Spawner
- c. ScrollingShooter2D.Shoot
- d. ScrollingShooter2D.DisplayUI
- e. ScrollingShooter2D.Destroyer
- f. ScrollingShooter2D.Data
- g. ScrollingShooter2D.Background
- h. ScrollingShooter2D.Attribute
- i. ScrollingShooter2D.Movement

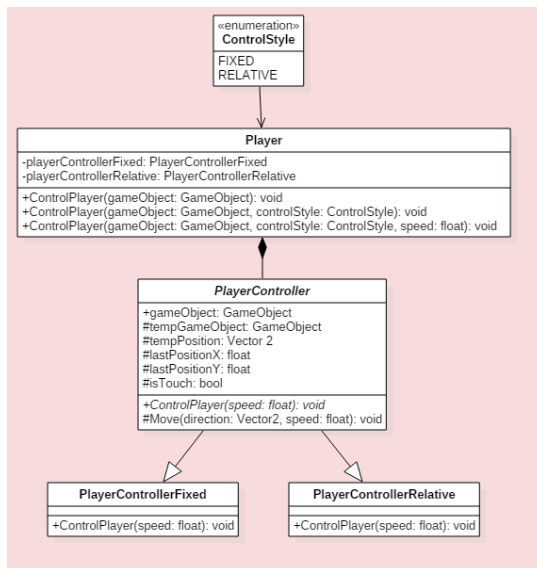


Figure 4. Class Diagram *Hotspot* Character Control

5. Implementation Class Library

Class Library will be implemented to C# (C Sharp) language. Naming identifier standards such as properties, classes and methods at the implementation stage will be adjusted to the C# language naming rules.

6. Unit Testing

Unit Testing is done by creating a test case method for each class Scrolling Shooter library function. The case test method is based on the design of the unit test scenario which can be seen in Table 3.

Table 3. Test Case Scenario

Class Test	Method Test	Status
Player	ControlPlayer()	Success
Background	ScrollBG()	Success
Spawner	Spawn()	Success
Data	Save()	Success
	Load()	Success
Destroyer	Destroy()	Success
DisplayUI	Display()	Success
Movement	Move()	Success
Shoot	Shoot()	Success
Player	ControlPlayer()	Success

The unit test results show that all methods in the Scrolling Shooter Class Library can function properly according to their functions.

7. Integration Testing

Integration testing will be tested directly by experienced programmers in building games using Unity. The prototype game display from the integration test results can be seen in Figure 5.]

Commented [a11]: Add more description before you show the figure



Figure 5. Prototype Game Scrolling Shooter

4. Conclusion

Based on all processes in this research, there are several conclusions, namely:

- a. Programmers can build Scrolling Shooter games without having to make the entire system functional from scratch using the Class Library. It is proven that the Class Library built in this research can produce a prototype of the Scrolling Shooter game.
- b. For further research, the sample of game application is expected to be able to use more than four application samples to know the characteristics that exist in the game domain more accurately. Also, it is hoped that the addition of other aspects will be used as a basis for conducting characteristics reviews. This is because the domain under study is a game application domain which certainly has its uniqueness when compared to other application domains.

Commented [a12]: Make it into narrative way

References

- [1] Atanasova, N., Todorovski, L., Džeroski, S., & Kompare, B. (2006). Constructing a library of domain knowledge for automated modelling of aquatic ecosystems. *ecological modelling*, 194(1-3), 14-36.
- R. Maulana. "Tantangan dan Peluang Startup Game di Indonesia pada Tahun 2018,". Internet: <https://id.techinasia.com/tantangan-peluang-startup-game-indonesia-tahun2018>, Jan. 2, 2018 [Jan. 12, 2018].
- [2] Kothari, C. R. 2004. *Research Methodology 2 rev.*
- [3] Pressman, R. S. 2015. "Software Engineering—A Practitioner's Approach", Mc Graw-Hill International Edition, 2010.
- [4] Bachtiar, A. M., & Bardansyah, A. 2017. Pembangunan Class Library untuk Domain Product Management di Aplikasi M-Commerce pada Android. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTEI)*, 6(3), pp. 235-242.
- [5] Booch, G. (1994). *Object-oriented analysis and design with applications*. Number 2nd.
- [6] Jamalludin, H. Y., Winiati, S., & Riadi, I. 2018. Implementasi Test Driven Development Pada Pengembangan Aplikasi Android Untuk Mahasiswa Universitas Ahmad Dahlan. *Jurnal Ilmu Teknik Elektro Komputer dan Informatika (JITEKI)*, 4(1), pp. 43.
- [7] Bačíková, M. 2012. Domain analysis with reverse-engineering for gui feature models. In *POSTER 2012: 16th International Student Conference on Electrical Engineering*, 16, pp. 1-5).
- [8] Yuniarti, A., Wardanie, N. N., & Kuswardayan, I. 2018. Dynamic game balancing implementation using adaptive algorithm in mobile-based Safari Indonesia game. In *Journal of Physics: Conference Series*, 974(1), pp. 012049.
- [9] Lee, J. H., Karlova, N., Clarke, R. I., Thornton, K., & Perti, A. 2014. Facet analysis of video game genres. *IConference 2014 Proceedings*.
- [10] Mohamed, E. 1999. Fayad.; Douglas, C. Schmidt.; Ralph E. Johnson. "Building Application Frameworks": *Object-Oriented Foundations of Framework Design*.

Commented [a13]: Change into international journals, and the latest one

Commented [a14]: Add volume, issue number and pages in your reference

Commented [a15]: Use JESTEC template

Commented [a16]: Use JESTEC format for your reference and the minimum should be 15 references

Commented [a17]: Change it to the latest one from international journal

3. Submit Revised Paper (18 Sept 2021)



Adam Mukharil Bachtiar
<adam@email.unikom.ac.id>

[JESTEC] Manuscript ID: JESTEC_16_5_40_2021_Adam Mukharil Bachtiar – Revised Version Received

1 message

Editorial Office <jjestec@gmail.com>

18 September 2021 at 11:24

To: Adam Mukharil Bachtiar <adam@email.unikom.ac.id>

Dear Author,

Thank you very much for resubmitting the modified version of the following manuscript:

Journal Name	Journal of Engineering, Science and Technology
Manuscript ID	JESTEC_16_5_40_2021_Adam Mukharil Bachtiar
Authors	Adam Mukharil Bachtiar, Dian Dharmayanti, Rakhmat Sabarudin
Title	Development of Class Library in Domain of Scrolling Shooter Game

A member of the editorial office will be in touch with you soon regarding progress of the manuscript.

Kind regards,

Editorial Office
Journal of Engineering, Science and Technology
E-Mail: jjestec@gmail.com

*** This is an automatically generated email ***

DEVELOPMENT OF CLASS LIBRARY IN DOMAIN OF SCROLLING SHOOTER GAME

**ADAM MUKHARIL BACHTIAR*, DIAN DHARMAYANTI, RAKHMAT
SABARUDIN**

Informatics Engineering, Faculty of Engineering and Computer Science,
Universitas Komputer Indonesia,
Jalan DipatiUkur 112-114, Bandung 40163, Jawa Barat, Indonesia,
*Corresponding Author: adam@email.unikom.ac.id

Abstract

The purpose of this research is to develop a Class Library in the domain of Scrolling Shooter games. The built of this Class Library will be used to develop Scrolling Shooter games as the result of this research. In software engineering, there is a reuse concept that is used to reuse functional systems. One application of this concept is the class library. With the class library, programmers can directly call the functions they need without building them from scratch. This study also used the class library concept for the domain of the shooter scrolling game case. The development of a Class Library is done through seven phases, Domain Analysis, Frozenspot Analysis, Hotspot Analysis, Modelling Class Library, Implementation Class Library, Unit Testing, and Integration Testing, and those will be the main discussion of this research. Furthermore, this research's conclusion is expected to help game Programmers develop Scrolling Shooter games without making the whole system functional from scratch. This research also contributes to software engineering concepts, namely: reusable code, abstraction, and class design.

Keywords: Game Scrolling Shooter, Class Library, Domain Analysis,
Frozenspot Analysis, Hotspot Analysis

1. Introduction

Based on data obtained from Newzoo (an intelligence market provider company in the global market), in 2017, the Indonesian gaming industry ranked 16th in the list of gaming markets in the world with several players reaching 43.7 million with potential income reaching around IDR. 11.9 trillion [1]. One of the popular game genres in Indonesia in January 2018 is Scrolling Shooter; it is noted that Scrolling Shooter won the top 10 on the Google Play Store in the January 2018 period, sorted by the Arcade game category. With this positive trend, game developers will automatically be interested in building a Scrolling Shooter game going forward.

The interview results with one of the programmers found that the problem in building a scrolling shooter game was that the programmer had to rebuild the functional system from scratch [2]. It caused the long development of the Scrolling shooter game, resulting in the game not being released according to the target time [3]. If you pay attention, the game industry always follows trends that require developers to release games at the right time.

Based on these problems, it is necessary to build a Class Library to help programmers build Scrolling Shooter games. A Class Library is a group of classes that provide the basic functionalities and can be used directly by programmers. With the Class Library, programmers can reuse functional requirements without having to build them from scratch. Therefore, a class library was created in this study to provide basic functionalities in accordance with the needs of programmers in building scrolling shooter games. This class library is an application of the reuse concept in software development.

2. Method

This research belongs to the type of applied research using a quantitative approach. Applied research aims to find a solution to deal with society or business organizations [4]. Meanwhile, the quantitative approach is used because research starts from a hypothesis and is objective. The methodology used in this study refers to the journal "Development of Class Library in Domain of Scrolling Shooter Game" by adding the implementation stages as seen in Fig. 1.

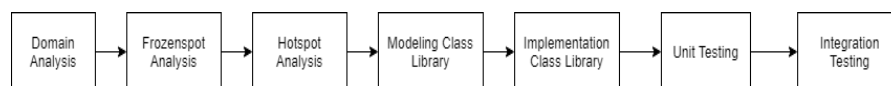


Fig. 1. Research methodology in this research.

This research methodology is that domain analysis was conducted to determine the characteristics of the scrolling shooter game application domain. The characteristics obtained are used to determine the scrolling shooter game domain's uniqueness from other game domains. Frozenspot analysis was carried out to determine the function slices of several samples of scrolling shooter game applications. Meanwhile, hotspot analysis is conducted to identify and define the points of flexibility (hotspots) of the scrolling shooter game application domain.

Class library design is done to model hotspots into a modeling language, namely UML (Unified Modeling Language). The results of this modeling can simplify the implementation process into a programming language. Class library implementation is a step taken to change the class library design model into a programming language. The unit testing phase is carried out to determine that each unit or function component in the class library can function as expected. Integration testing is done by directly testing each unit in the class library to create a scrolling shooter game prototype.

3. Results and Discussion

3.1. Domain Analysis

Domain Analysis can be done by determining the source of information used to identify the characteristics of the application domain. Several sources of information can be used, including domain experts, textbooks, domain standards, and previous application samples [5]. There are four sample Scrolling Shooter game applications from the Google Play Store that will assist the identification process. A characteristic of the Scrolling Shooter game used for this research is that players generally play as pilot fighter planes, which are assigned to withstand any enemy attacks [6]. The sample application consists of two types of vertical Scrolling Shooter games and two types of horizontal scrolling shooter games selected based on the highest number of downloads and the highest rating on the Google Play Store in the January 2018 period. The four sample game applications will be given symbols to facilitate calling the application name in the identification process, as shown in Table 1.

Table 1. Four Sample Game Scrolling Shooter.

Function	Game	Symbol
Space Shooter: Galaxy Attack	Vertical	G1
Galaxy Attack: Alien Shooter	Vertical	G2
Scrambler – Classic 80s Arcade Game	Horizontal	G3
R-TYPE	Horizontal	G4

Based on the identification results, four aspects become a review of the domain's characteristics: the camera viewpoint, game presentation, character control, and the game's basic mechanism. Here are the results of identifying the characteristics obtained from the four sample Scrolling Shooter game applications:

a. Camera Viewpoint

In G1 and G2 games, the camera's perspective is taken from the top to down (top-down view), where only the top side of the object is seen by the player when playing the game. While in the G3 and G4 games, the camera's perspective is taken from the side, where the player sees only the object's side. Viewpoints of the four games.

b. Game Presentation

The presentation or display style used in the G1 and G2 games is vertical scrolling. Vertical Scrolling is a display style where the game's background is seen moving automatically with a vertical orientation [7]. Meanwhile, in G3 and G4 games, the presentation used is horizontal scrolling. Horizontal Scrolling is a display style where the background in the game moves automatically with a horizontal orientation.

c. Character Control

The four sample Scrolling Shooter game applications that are played show that the player's character can move with an eight-way mechanism (right, left, up, down, top right, bottom right, top left, and bottom left) on the x and y axes. The character movement controlled by the player is only limited to the screen area. There are two different mechanisms for controlling character objects, namely control with a fixed mechanism and control with a relative mechanism. In fixed mechanism control, the player's screen position has no distance from the character object being controlled. Whereas in the relative mechanism, there is a distance between the character object being controlled and the screen's position being touched.

d. Basic Mechanic

In the four sample games that are played, players must survive the pattern of attacks fired by the enemy to get the maximum score. To survive the enemy attack, players can destroy enemies by firing projectiles or avoiding incoming attacks. Also, there are power-ups that players can use to increase attack power. The four sample Scrolling Shooter game applications' characteristics will be used to identify the functions contained in the Scrolling Shooter game domain in the Frozenspot Analysis stage.

3.2. Frozenspot Analysis

Frozenspot analysis is divided into two stages, namely Function Identification and Frozenspot Identification.

a. Functional Identification

This stage is done to identify the collection or set of functions in the Scrolling Shooter game domain, based on four previous application samples. The resulting set of functions will be used to determine the Scrolling Shooter game domain's function slices. Namely, there are nine functions: Character Control, Scrolling Background, Spawn Object, Destroy Object, Move Object, Shoot Object With Pattern, Change Attribute, Display Attribute, and Save Data. The function identification table results show that there are nine sets of functions in the Scrolling Shooter game application domain and are used as a reference in determining the function slices.

b. Frozenspot Identification

This stage is carried out to determine the four previous Scrolling Shooter game application samples' sliced function. The set of functions that have been obtained will be used as a reference in determining the function slices. Frozenspot Identification results from four sample Scrolling Shooter game applications based on nine functions that have been obtained can be seen in Table 2. From the Frozenspot Analysis results, it can be seen that there are nine

functions contained in the Scrolling Shooter game application domain. The nine functions are slices of the four sample Scrolling Shooter game applications G1, G2, G3, and G4. The sliced function results will be used to determine the hotspot that is done in the next stage.

Table 2. Frozen-spot Identification Result.

Function	Game
Character Control	G1, G2, G3, G4
Scrolling Background	G1, G2, G3, G4
Spawn Object	G1, G2, G3, G4
Destroy Object	G1, G2, G3, G4
Move Object	G1, G2, G3, G4
Shoot Object with Pattern	G1, G2, G3, G4
Change Attribute	G1, G2, G3, G4
Display Attribute	G1, G2, G3, G4
Save Data	G1, G2, G3, G4

3.3. Hotspot Analysis

Hotspot analysis is divided into two stages, namely Hotspot Identification and Hotspot Definition [8].

a. Hotspot Identification

This stage is carried out to identify functions that can be used as hotspots in the Scrolling Shooter game application domain. The sliced function obtained from the frozen-spot analysis results can determine whether the function is suitable as a hotspot or not. In this study, Hotspot Identification uses the Blackbox method, where the function used as a hotspot is mandatory. Therefore, the hotspot identification results show that there are nine mandatory functions used as hotspots.

b. Defining Hotspot

This stage is carried out to define each function obtained from the Hotspot Identification results. Defining Hotspot is done by describing the behavior performed by the function, using a hotspot card [9]. With the object model transformation rules, the hotspot card will be converted into object models in the form of class diagrams. The hotspot card from the Player Character Control function and the object model formed can be seen in Fig. 2.

<p>Hotspot Name</p> <p>Character Control</p> <p>Degree of Flexibility</p> <p><input checked="" type="checkbox"/> <i>adaptation without restart</i></p> <p><input type="checkbox"/> <i>adaptation with by end user</i></p>
<p>General Description</p> <p>This function is used to move the character object in eight directions, based on the input in the form of a touch from the player.</p>

- Examples of implementation, at least two different situations**
1. Control of character objects using a fixed mechanism, where the character object will always move to follow the position of the screen touched by the player without any difference in distance between the position of the character object with the position of the screen touched.
 2. Character object control uses a relative mechanism, where the character object will always move to follow the position of the screen touched by the player with the possibility of the difference in distance between the position of the character object and the position of the screen touched.

Fig. 2. Hotspot Card Character Control.

Based on the object model transformation rules, the hotspot design will be created by adding the hook method in a separate hook class. The transformation of the object model of the hotspot Controlling Player Character can be seen in Fig. 3.

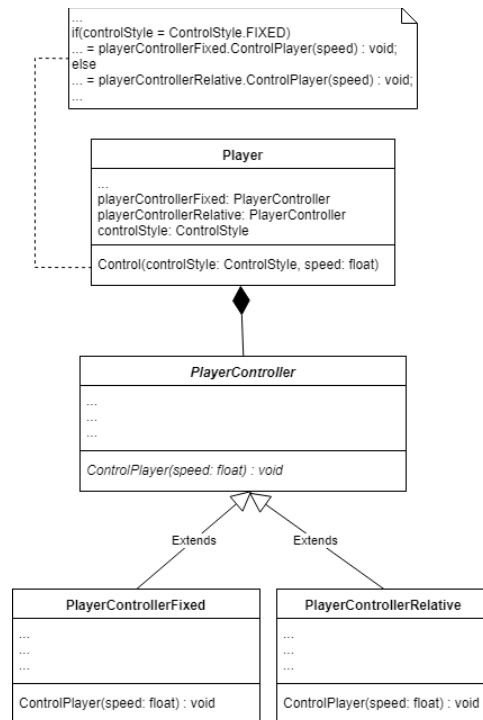


Fig. 3. Object transformation model hotspot character control.

3.4. Modeling Class Library

Modeling class library design consists of two stages, namely class design and package design.

a. Class Modeling

This class diagram will illustrate the relationships between classes of hotspot Controlling Player Character, where the class contains a method used to move character objects with eight directions of movement according to player control

[10]. The class diagram of the hotspot controlling the player's character can be seen in Fig. 4.

b. Package Modeling

This stage is done to describe the package diagram of the Class Library system that will be built. The package diagram created functions to find groups of related classes wrapped in one container or package. The package design results show that nine packages will be built in the Scrolling Shooter class library, as follows.

- ScrollingShooter2D.Player
- ScrollingShooter2D.Spawner
- ScrollingShooter2D.Shoot
- ScrollingShooter2D.DisplayUI
- ScrollingShooter2D.Destroyer
- ScrollingShooter2D.Data
- ScrollingShooter2D.Background
- ScrollingShooter2D.Attribute
- ScrollingShooter2D.Movement

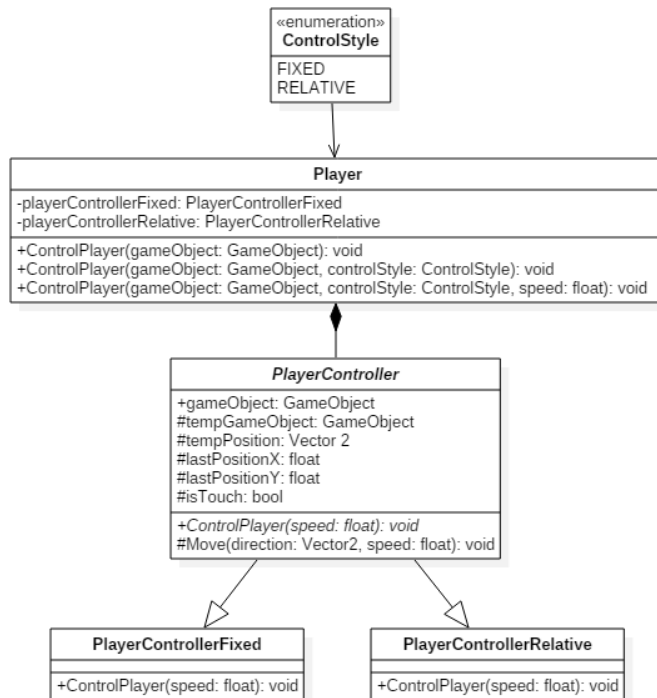


Fig. 4. Class diagram hotspot character control.

3.5. Implementation Class Library

Class Library will be implemented to C# (C Sharp) language [11]. Naming identifier standards such as properties, classes, and methods at the implementation stage will be adjusted to the C# language naming rules [12].

3.6. Unit Testing

Unit Testing is done by creating a test case method for each class Scrolling Shooter library function [13]. The case test method is based on the unit test scenario's design, which can be seen in Table 3.

Table 3. Test Case Scenario

<i>Class Test</i>	<i>Method Test</i>	<i>Status</i>
Player	ControlPlayer()	Success
Background	ScrollBG()	Success
Spawner	Spawn()	Success
Data	Save()	Success
	Load()	Success
Destroyer	Destroy()	Success
DisplayUI	Display()	Success
Movement	Move()	Success
Shoot	Shoot()	Success
Player	ControlPlayer()	Success

The unit test results show that all Scrolling Shooter Class Library methods can function properly according to their functions.

3.7. Integration Testing

After unit testing is complete, the next testing stage is integration testing [14, 15]. Integration testing is done by directly testing each unit in the class library to create a scrolling shooter game prototype. In this study, integration testing was tested directly by experienced programmers in building games using Unity [16]. The display of the game prototype from the results of integration testing can be seen in Fig. 5.



Fig. 5. Prototype Game Scrolling Shooter.

4. Conclusion

Based on the results of integration testing, it is concluded that programmers can build scrolling shooter games without having to make the entire system functional from scratch using the class library. It is proven that the class library built in this study can produce a scrolling shooter game prototype consisting of a vertical scrolling shooter game and a horizontal scrolling shooter game. It is because the class library that has been built has provided basic functions that can be reused by programmers to build scrolling shooter games.

References

1. Atanasova, N.; Todorovski, L.; Džeroski, S.; and Kompare, B. (2006). Constructing a library of domain knowledge for automated modelling of aquatic ecosystems. *Ecological Modelling*, 194(1-3), 14-36.
2. Klatte, R.; Kulisch, U.; Wiethoff, A.; and Rauch, M. (2012). *C-xsc: a c++ class library for extended scientific computing*. New York: Springer Science and Business Media.
3. Atmaja, P. W.; and Mandyartha, E. P. (2020). Difficulty curve-based procedural generation of scrolling shooter enemy formations. In *Journal of Physics: Conference Series*, Surabaya, Indonesia, 1569(2), 1-8.
4. Seaman, C. B. (1999). Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering*, 25(4), 557-572.
5. Kienle, H. M.; and Müller, H. A. (2010). Rigi—an environment for software reverse engineering, exploration, visualization, and redocumentation. *Science of Computer Programming*, 75(4), 247-263.
6. Mowlaei, S.; Schmidt, S.; Zadtootaghaj, S.; and Möller, S. (2018). Know your game: a bottom-up approach for gaming research. In *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, Cagliari, Italian., 1-3. IEEE.
7. Ştefan, A.; Stănescu, I. A.; and Baalsrud Hauge, J. M. (2016). Approaches to reengineering digital games. In *ASME 2016 International Design*

Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers Digital Collection. Charlotte, North Carolina, USA.

8. Pree, W. (2000). Hot-spot-driven framework development. *Framework*, Salzburg, Austria, 2(B1), 1-13.
9. Frameworks, B. A. (1999). *Object-oriented foundations of framework design.* United States: Wiley, 1st Edition.
10. Forina, M.; Oliveri, P.; Lanteri, S.; and Casale, M. (2008). Class-modeling techniques, classic and new, for old and new problems. *Chemometrics and Intelligent Laboratory Systems*, 93(2), 132-148.
11. Kokholm, N.; and Sestoft, P. (2006). The c5 generic collection library for c# and cli. the it university of copenhagen. Retrieved February 15, 2021, from <https://www.itu.dk/research/c5/latest/ITU-TR-2006-76.pdf>.
12. Schuller, D. (2011). *C# game programming: for serious game creation.* Canada: Nelson Education.
13. Sawant, A. A.; Bari, P. H.; and Chawan, P. M. (2012). Software testing techniques and strategies. *International Journal of Engineering Research and Applications (IJERA)*, 2(3), 980-986.
14. Bissi, W.; Neto, A. G. S. S.; and Emer, M. C. F. P. (2016). The effects of test driven development on internal quality, external quality and productivity: a systematic review. *Information and Software Technology*, 74, 45-54.
15. Ali, S.; Briand, L. C.; Rehman, M. J. U.; Asghar, H.; Iqbal, M. Z. Z.; and Nadeem, A. (2007). A state-based approach to integration testing based on uml models. *Information and Software Technology*, 49(11-12), 1087-1106.
16. Wazir, H. K.; and Annaz, F. Y. (2014). Using unity for 3d object orientation in a virtual environment. *5th Brunei International Conference on Engineering and Technology (BICET 2014)*. Bandar Seri Begawan, Brunei, 1-6.

4. Accepted Submission (25 Sept 2021)



Adam Mukharil Bachtiar
<adam@email.unikom.ac.id>

[JESTEC] Manuscript ID: JESTEC_16_5_40_2021_Adam Mukharil Bachtiar – Accepted

1 message

Editorial Office <jjestec@gmail.com>
To: Adam Mukharil Bachtiar <adam@email.unikom.ac.id>

25 September 2021 at 11:05

Dear Author,

We are pleased to inform you that the following paper has been officially accepted for publication:

Journal Name	Journal of Engineering, Science and Technology
Manuscript ID	JESTEC_16_5_40_2021_Adam Mukharil Bachtiar
Authors	Adam Mukharil Bachtiar, Dian Dharmayanti, Rakhmat Sabarudin
Title	Development of Class Library in Domain of Scrolling Shooter Game

We will now make the final preparations for publication, then return the manuscript to you for your approval.

If, however, extensive English edits are required to your manuscript, we will need to return the paper requesting improvements throughout.

Kind regards,

Editorial Office
Journal of Engineering, Science and Technology
E-Mail: jjestec@gmail.com

5. Paper Publish (05 Oct 2023)



Adam Mukharil Bachtiar
<adam@email.unikom.ac.id>

[JESTEC] Manuscript ID: JESTEC_16_5_40_2021_Adam Mukharil Bachtiar – Published Online

1 message

Editorial Office <jjestec@gmail.com>

5 October 2021 at 16:11

To: Adam Mukharil Bachtiar <adam@email.unikom.ac.id>

Dear Author,

We are pleased to inform you that your article:

Journal Name	Journal of Engineering, Science and Technology
Manuscript ID	JESTEC_16_5_40_2021_Adam Mukharil Bachtiar
Authors	Adam Mukharil Bachtiar, Dian Dharmayanti, Rakhmat Sabarudin
Title	Development of Class Library in Domain of Scrolling Shooter Game

is available online:

https://jestec.taylors.edu.my/Vol%2016%20Issue%205%20October%202021/16_5_40.pdf

Please take a moment to check that everything is correct. You can reply to this email if there is a problem. Note that at this stage we will not accept further changes to the manuscript text.

Thank you for choosing our journal to publish your work, we look forward to receiving further contributions from your research group in the future.

Kind regards,

Editorial Office
Journal of Engineering, Science and Technology
E-Mail: jjestec@gmail.com