AZURITE: A MOBILE GAME USING SPATIAL HASHING ALGORITHM

A Thesis
Presented to the Faculty of the
College of Computer Studies
Lyceum of the Philippines University

In Partial Fulfillment
Of the Requirements for the Degree
Bachelor of Science in Computer Science
Specialized in Game Development

by:

Glen A. De Torres
Tagumpay R. Quizon II
Samuel James M. Velasquez

March 2017

APPROVAL SHEET

In partial fulfilment of the requirements for the degree Bachelor of Science in Computer Science (Specialized in Game Development), this thesis entitled **"Azurite: A Mobile Game using Spatial Hashing Algorithm"** has been prepared and submitted by Glen A. De Torres, Tagumpay R. Quizon II, and Samuel James M. Velasquez and is hereby recommended for oral examination.

Roselie B. Alday, PhD. Cand.
Adviser

Defended in an oral examination before a duly constituted panel with a grade of _____.

Mischelle A. Esguerra, MIS
Chairman

Janice E. Velasquez, MSCS
Member

Elaine Joy J. Ilao, MAITE
Member

Chona D. Andal, MAEd
Member

Accepted in partial fulfilment of the requirements for the degree Bachelor of Science in Computer Science.

Roselie B. Alday, PhD. Cand.

Dean, College of Computer Studies

ACKNOWLEDGMENT

First of all, the developers would like to give recognition to their adviser Prof. Rose B. Alday for the unending guidance, support and spirit to never give up.

Aside from their Adviser, they would like to thank the rest of their panelists: Prof. Mischelle Esguerra, Prof. Elaine Ilao, and Prof. Janice Velasquez for their critical responses and encouragement that helped them a lot. Lastly, they would like to thank their families, for giving them unwavering support and guidance.

Glen A. De Torres

Tagumpay R. Quizon II

Samuel James M. Velasquez

DEDICATION

The developers dedicate this thesis to their family and friends, who supported them in their endeavours and for guiding them throughout the years. The developers will never forget the things they have done for them to develop the knowledge they have today.

Glen A. De Torres

Tagumpay R. Quizon II

Samuel James M. Velasquez

Table of Contents

List of Figures

AZURITE: A MOBILE GAME USING SPATIAL HASHING ALGORITHM

De Torres, Glen A.
Lyceum of the Philippines
University Batangas
Capitol Site, Batangas City
glendetorres.ccs@gmail.com
0000-0002-4252-9311

Quizon, Tagumpay II R.
Lyceum of the Philippines
University Batangas
Capitol Site, Batangas City
tagumpayquizon.ccs@gmail.com
0000-0001-6818-7498

Velasquez, Samuel James M.
Lyceum of the Philippines
University Batangas
Capitol Site, Batangas City
samueljames.velasquez@gmail.com
0000-0001-5804-8478

Abstract

Azurite is a game developed using spatial hashing algorithm and Unity Game Engine that lets one control an original character, fight enemy AIs, collect coins and navigate various stages within the game. It was developed using C# programming language which Unity natively incorporates and can be ported to any platform including Android.

Keywords: *AI, Spatial Hashing Algorithm, Android, Unity*

1. Introduction

Azurite is a 2D platformer game that takes place in the distant future in which the user can control the character Bloo, a robot set on a journey to acquire fragments of azurite and travel in distant worlds to find the azurite stones. The game is specifically designed to take one into the role of a character that lets the player travel different areas of the in-game world to find various fragments of azurite that is spread throughout the world. Precision, Hand-Eye reflexes, skills and a bit of luck are needed to survive the game's basic learning curve. The goal of the game is to find azurite fragments to bring it to his homeworld.

The game was created on Unity Game Engine that takes advantage of the C# programming language which is one of the most popular programming languages when it comes to games especially in 2D.

Azurite uses the spatial hashing algorithm, a process by which a 3D or 2D domain space is projected into a 1D hash table.

1.1 ***Objectives***

1. To develop a game that provides entertainment to a player while teaching one on how to decide better for acquiring the desired goal.
2. To use Spatial Hashing Algorithm for a faster collision detection in the game.
3. To develop a game that uses C# programming language.

2. Review of Literature

2.1 What is a Computer Game?

According to Technopedia (n.d.), a personal computer game (PC game) is a video game that is played on a personal computer rather than on a console. The game is controlled using PC input devices such as the keyboard, mouse, joystick, etc. PC games can be played with or without an Internet connection, and have been available since the introduction of personal computers. A large number of games are available for the PC platform.

2.2 Hardware and Software Requirements

This game can be played on an android device running android KitKat and up on a screen size of 4-5 inches. This game does not need a camera and could run without one. Azurite is a 2D platformer which does not really require too much graphical prowess of the latest hardware on the market.

2.3 What is a Mobile Game?

According to Technopedia (n.d.), mobile games are games designed for mobile devices, such as smartphones, feature phones, pocket PCs, personal digital assistants (PDA), tablet PCs and portable media players. Mobile games range from basic (like Snake on older Nokia phones) to sophisticated (3D and augmented reality games).
Today's mobile phones - particularly smartphones - have a wide range of connectivity features, including infrared, Bluetooth, Wi-Fi and 3G. These technologies facilitate wireless multiplayer games with two or more players.

2.4 What is Game Genre?

As stated by Hanna (n.d.), Games are often classified into genres, which purport to define games in terms of having a common style or set of characteristics, e.g. as defined in terms of perspective, gameplay, interaction, objective, etc.

2.4.1 Example of Game Genres

Adventure games

As said by Hanna (n.d.), typically, the player is the protagonist

of a story and in order to progress must solve puzzles. The puzzles can often involve manipulating and interacting with in-game objects, characters, and others.
Ex. Zork, King's Quest

Action games

Consistent with Hanna (n.d.), a number of other action-oriented genres can be broadly classified as belonging to this genre. Action games are typified by fast-paced events and movement which often have to be performed reflexively.
Ex. Pong, Space Invaders

Action-adventure games

On the word of Hanna (n.d.), action-adventure games can be described in terms of a blend of the characteristics associated with both adventure and action games, i.e. often involving both exploration and puzzle solving alongside fast-paced action sequences.
Ex. Legend of Zelda, Metroid Prime 2

Platformers

According to Hanna (n.d.), this genre often requires the protagonist to run and jump between surfaces (i.e. platforms) whilst avoiding game objects and the detrimental effects of gravity. Traditionally, platform games were side-on 2D in perspective and very popular on earlier gaming platforms.
Ex. Super Mario Bros, Prince of Persia

Fighting games

In keeping with Hanna (n.d.), In fighting games the player typically fights other players or the computer in some form of one-on-one combat.
Ex. Tekken, Mortal Kombat

First-person shooter (FPS) games

According to Hanna (n.d.), first-person shooter games where the player is "behind the eyes" of the game character in a first-person perspective. Although a number of FPS games also support third-person views.
Ex. DOOM, Far Cry

Rhythm games (music games)

In line with Hanna (n.d.), rhythm games require the player to undertake some action (e.g. follow a sequence of movement or develop specific rhythms) in response to some stimulus (often music).
Ex. Dance Dance Revolution, DJ Max Technika

Puzzle games

As said by Hanna (n.d.), puzzle games often require the player to solve puzzles or problems and can involve the exercise of logic, memory, pattern matching, reaction time, etc.
Ex. Tetris, Minesweeper

Role playing games (RPGs)

According to Hanna (n.d.), this originally started out as video games based on pen and pencil games like Dungeons and Dragons. A fantasy theme is often retained. Often characterised in terms of providing the player with flexibility in terms of character development, problem resolution, etc.
Ex. Final Fantasy, Elder Scrolls

Oblivion

Azurite is a 2D Platformer game that lets the player control Bloo, a robot that seeks the azurite stone on different worlds to acquire its fragments as a power source for its home planet.

2.5 Principles of Gameplay Design

According to Allmer (2009), there are 13 Basic Principles of Gameplay Design.

2.5.1 Direction

1. Focal Point

As stated by Allmer (2009), never allow the players to guess what they should focus on. At the same time, always allow secondary subject matter, but it is the designer's job to clearly provide the primary focus at all times. This applies to both visual and visceral aspects of gameplay.

2. Anticipation

Consistent with Allmer (2009), time is needed to inform the player that something is about to happen. Always factor in Anticipation when designing and implementing events and behaviors.

3. Announce Change

On the word of Allmer (2009), communicate all changes to the player. This short step occurs between Anticipation and the event itself. The important part to remember is maintaining a hierarchy of notable changes. A good rule of thumb is degree of rarity. If a change occurs a hundred times in an hour, the announcement may not be required. However, if the change occurs five times throughout the entire game experience, a number of visual cues could be needed. This principle is so obvious, it can be taken for granted and sometimes overlooked. Be diligent in knowing what changes the player should be aware of at the correct time and on the correct event.

2.5.2 Behavior

4. Believable Events and Behavior

According to Allmer (2009), every event or behavior must occur according to the logic and expectations of the player. Every action, reaction, results, emotion and conveyance must satisfy the players' subconscious acceptance test.

5. Overlapping Events and Behavior

As said by Allmer (2009), dynamism is lost if only one change occurs at a time. Discover the right amount of events to occur at any given moment of time.

6. Physics

On the word of Allmer (2009), the player's primary logic operates within the known possibilities of physics. Keep in mind gravity, weight, mass, density, force, buoyancy, elasticity, etc. Use this as the starting point, but do not be limited by it.

7. Sound

According to Allmer (2009), ask yourself, "What sound does it make when an action happens?" "Is the sound appropriate?" "Is the sound necessary?" "Does it benefit the experience or hinder it?" If players close their eyes, the sound alone should still achieve the desired effect.

2.5.3 Progression

8. Pacing

As said by Allmer (2009), keep in mind the desired sense of urgency, the rate in which events occur, the level of concentration required and how often events are being repeated. Spread out the moments of high concentration, mix up the sense of urgency, and change things wherever possible to achieve the proper effect.

2.5.4 Environment

9. Spacing

On the word of Allmer (2009), understand how much space is available both on-screen and in-world, recognize the spatial relationship between elements and take into account the effects of modifying those spaces.

2.5.5 Method

10. Linear Design versus Component Breakdown

In keeping with Allmer (2009), linear design involves solving challenges as they come. All solutions and possibilities hold the same institutional value. Focus can be lost with this method, but it provides creative and spontaneous solutions. Component Breakdown involves systemic categorization and forming a logical hierarchy of all solutions. This method can restrict innovation but preserves clarity of primary design objectives.

2.5.6 Foundation

11. Player

As said by Allmer (2009), how does the player interact with everything that has been designed? More than just device input, address how the player contributes to the experience. If it is a good idea and the player is able to convey it correctly but the player is not into it, change it or scrap it.

12. Communication

As stated by Allmer (2009), ss the appropriate team member correctly aware of the objective? Are the appropriate developers clear on the solution? If it is a good idea but one cannot communicate it correctly, it might as well be a bad idea because it is very likely to be received as such.

13. Appeal

In relation to Allmer (2009), when addressing anyone, ask yourself, "Does this draw the audience in?" This applies to (but is not limited to) the player, the spectator, your fellow developers, the publisher, and their marketing team. If it is not a

good idea, there is no need to continue until it becomes a good idea or is replaced by something better.

2.6 Algorithm

According to MacDonald (2009), a spatial hash is a 2 or 3 dimensional extension of the hash table, which should be familiar to the user from the standard library or algorithms book/course of his or her choice.

The basic idea of a hash table is that one takes a piece of data (the 'key'), runs it through some function (the 'hash function') to produce a new value (the 'hash'), and then uses the hash as an index into a set of slots ('buckets').

To store an object in a hash table, the user runs the key through the hash function, and stores the object in the bucket referenced by the hash. To find an object, one runs the key through the hash function, and looks in the bucket referenced by the hash.

According to The Mind of Conkerjo (2009)

Spatial hashing is a process by which a 3D or 2D domain space is projected into a 1D hash table.

Spatial hashing can be used to detect collision in 2D space.

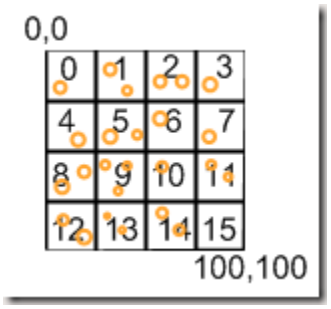

Figure 1. Buckets with game objects

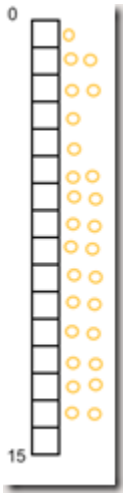

Figure 2. 1D Hash Table of Figure 1.

2.7 What is Collision Detection

As stated by Techopedia (n.d.), collision detection algorithmically calculates impact time by identifying two or more object intersection points. Collision detection is also a virtual interface that determines user and object distance for collision prevention. Collision detection is a key 3-D component associated with robotics, video games and physical simulation.

2.8 2D vs 3D Games

In games, 3D Games are essentially harder to make than 2D

Games. Programing in a 3D game uses math that is more complex for rendering, physics, collision, etc. than 2D. Also, animation is more difficult and more complicated in 3D. Animation in 2D is just full of frames with possibly different positions for each frame. With 3D, the developer needs to deal with separate animation assets, bones, skinning, etc.

3D Assets are more complex in which the user is needed for him to familiarize or a certain expertise in using assets and when doing the asset processing. It can make the game either crash or buggy. Unlike in 2D assets, it is simply, partly easy to do for most beginners since it can be done with only one person because of its simplicity. Finally, for 3D Design, it takes a lot of time to work on design of stages, characters (unless the developer would be using pre-made assets) because it is not a “flat” art of 2D.

2.9 What is Azurite?

Azurite is an android puzzle platformer that encompasses upon the spatial hashing algorithm that the developers are using. It is notable that azurite is a 2D game to make the algorithm more efficient. The algorithm contains a “bucket” that can process different types of variables without colliding with other items. According to Minerals.net, Azurite is actually a soft, deep blue copper mineral produced by weathering of copper ore deposits. It is also known as Chessylite after the type locality at Chessy-les-Mines near Lyon, France. Azurite Mineral is supposed to power machines for humans to use in the game. The player is able to navigate stages using the playable robot character named “bloo” in which was drawn digitally.

2.10 Methods

2.10.1 How does the algorithm work?

Many Games require Spatial Hashing. They are mainly used for collision detection and for rendering algorithm in 2D games. Spatial Hashing starts as one cell called a “Bucket” it checks items on a bucket and will not collide with other buckets. A Spatial Hashing algorithm works where each bucket processes every item inside it then will process the next bucket. This process is repeated until every bucket is hashed. Generally, this limit is imposed due to storage considerations or to limit processing time or due to the resolution of the output device. A pixel is the smallest subsection of the Spatial Hashing Algorithm. In a Spatial Hashing, it represents as a 2D table where each Bucket can store items.

2.10.2 How does the game work?

The researchers used spatial hashing algorithm in this game to use objects like the blocks that the player walks on or an invisible wall that limits the stage in an area useable to the user or consumable objects like azurite fragments. Also each object

contains pixels inside a bucket in order to form a “land” where the character can move and then add a collision detection mechanism where Spatial Hashing is capable of doing. The goal of this game in using Spatial Hashing algorithm is to separately make use of buckets to efficiently put items on within the game for collision detection and rendering to make the game smoother and simpler to walk on knowing there is still land to walk on. If not, the player would not most likely fall because of the invisible wall that is set in the game and is also used to make enemies aware of the player and the whole stage, as well as the stage’s environment.

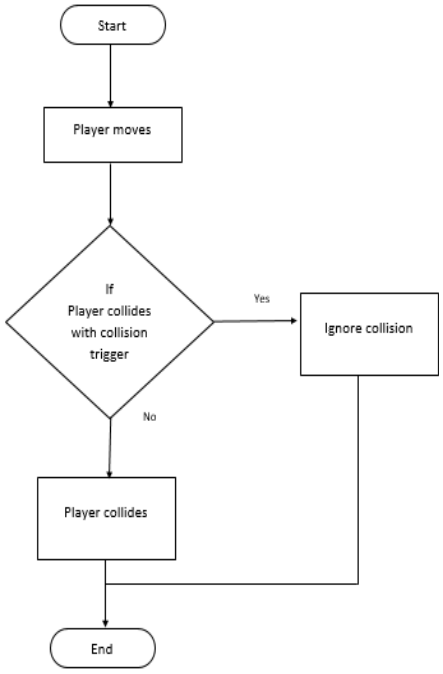

Figure 3. Collision trigger

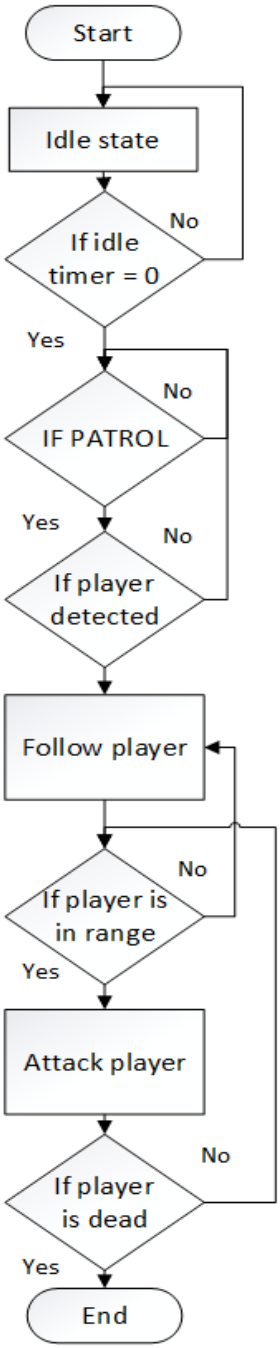

Figure 4. AI flowchart

2.10.3 What principles will be used?

The programming language used by the developers is C#, which is a general purpose, high-level programming language developed Microsoft Corporation. C# is an

Object-Oriented language similar to Java and C++ but modernized with added features to provide in software engineering.

The programming environment used is Unity. It contains a workspace that contains windows for scene, animation, game, and hierarchy which are commonly used in development. The developers installed Android studio to use Java SDK and JDK so they can build for android devices.

For the developers to accomplish the research, they tried and used different kinds and genres of games as basis for Azurite's development. The Developers tested the integrity of the collision detection of their game. Different kinds of objects and AI were used to further development and testing of Azurite. The researchers decided to use spatial hashing algorithm which is prominent in several games. A spatial hash is a 2 or 3 dimensional extension of the hash table, which should be familiar to the user from the standard library or algorithms book/course of his or her choice. The developers of Azurite all agreed to make the game easy to understand and they would like to make the game child-friendly to cater to a wide variety of audiences. One of their game's foundations would most likely appeal to a large section of audiences of all ages. As earlier emphasized, the game aims to deliver lessons for the player to know what is right and wrong in a physical way.

3.0 Results and Discussions

The Developers tested the accuracy of the collision detection by adjusting the hitboxes of objects, especially the enemy AI. The developers added an AI that would take damage every time the player attacks it. The AI would take 10 damages to the given health points of an enemy. After the enemy AI dies, it would drops coins for the player to collect. The implementation of the collision detection became evident during the final process of development of the AI.

4.0 Conclusion

Overall, the game is considered to be at a playable state. Factors that can affect the game would be the variety of features that can be put into the game. Modern entertainment features like lootable items or multiplayer, that can be a significant idea to put into the game. Azurite is a game that can be updated in the future to further improve the game.

5.0 Recommendation

The developers would recommend to add more maps, different types of enemies and features that would make the game more entertaining to a wide variety of audience. Keeping the storyboard and the idea but making the game using 3D visuals would also be advantageous.

6. References

Allmer, Matt. (2009). “The 13 Basic Principles of Gameplay Design”. Retrieved from http://www.gamasutra.com/view/feature/132341/the_13_basic_principles_of_.php

Hanna, Philip. (n.d.). “Video Game Technologies”. Retrieved from http://www.di.ubi.pt/~agomes/tjv/teoricas/01-genres.pdf

MacDonald, Tristam. (2009). “Spatial Hashing”. Retrieved from https://www.gamedev.net/resources/_/technical/game-programming/spatial-hashing-r2697

Minerals.net. (n.d.). “THE MINERAL AZURITE”. Retrieved from http://www.minerals.net/mineral/azurite.aspx

Techopedia. (n.d.) “Definition - What does Collision Detection mean?”. Retrieved from https://www.techopedia.com/definition/4778/collision-detection

Techopedia. (n.d.) “Mobile Games”. Retrieved from https://www.techopedia.com/definition/24261/mobile-games

Techopedia. (n.d.). “Personal Computer Game (PC Game)”. Retrieved from https://www.techopedia.com/definition/31136/personal-computer-game-pc-game

The Mind of Conkerjo. (2009). “Spatial hashing implementation for fast 2D collisions”. Retrieved from https://conkerjo.wordpress.com/2009/06/13/spatial-hashing-implementation-for-fast-2d-collisions/

Appendices

1 Screenshot

Figure 5. Menu screen – This scene is about where the game usually starts when it is opened.

Figure 6. Stage name – This scene is about the name of the stage and lets the player click “next” to start the game

Figure 7. Player interface – This scene lets the player control the character Bloo; by which controls such as left and right buttons for horizontal movement, Jump for vertical movement, Sword Attack, Gun Attack, and Dash Button are included.

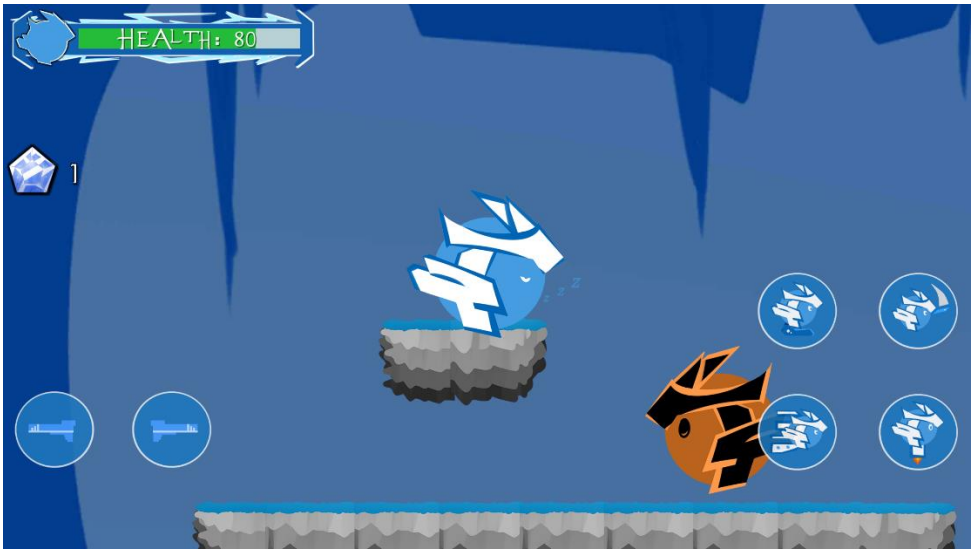

Figure 8. Enemy – the enemy is the alter ego of the main character which mirrors itself which adds depth to the name of the stage.

CURRICULUM VITAE

PERSONAL INFORMATION

Name: Glen A. De Torres

Age: 20

Gender: Male

Date of Birth: January 29, 1997

Address: Roxas Road Kumintang Ibaba, Batangas City

Contact Number: 09993677301

Email Address: glendetorres.ccs@gmail.com

Nationality: Filipino

FAMILY BACKGROUND

Father's Name: Gil C. De Torres **Contact Number:** 09237334731

Occupation: Driver

Mother's Name: Nancy A. De Torres **Contact Number:**

Occupation:

EDUCATIONAL BACKGROUND

Primary: Batangas City East Elementary School

Secondary: Batangas Christian School

Tertiary: Lyceum of the Philippines University – Batangas

CURRICULUM VITAE

PERSONAL INFORMATION

Name: Tagumpay R. Quizon II

Age: 20

Gender: Male

Date of Birth: January 11, 1997

Address: Balayan, Batangas

Contact Number: 09363913493

Email Address: tagumpayquizon.ccs@gmail.com

Nationality: Filipino

FAMILY BACKGROUND

Father's Name: Tagumpay G. Quizon **Contact Number:**

Occupation: Electrician

Mother's Name: Gleceria R. Quizon **Contact Number:** 09178251257

Occupation: Teacher

EDUCATIONAL BACKGROUND

Primary: Immaculate Concepcion College

Secondary: Immaculate Concepcion College

Tertiary: Lyceum of the Philippines University – Batangas

CURRICULUM VITAE

PERSONAL INFORMATION

Name: Samuel James M. Velasquez

Age: 19

Gender: Male

Date of Birth: June 3, 1997

Address: Gulod Labac, Batangas City

Contact Number: 09153873305

Email Address: sjmv_888@yahoo.com

Nationality: Filipino

FAMILY BACKGROUND

Father's Name: Gilbert S. Velasquez **Contact Number:**

Occupation: Seaman

Mother's Name: Lorna M. Velasquez **Contact Number:** 09275533144

Occupation: Housewife

EDUCATIONAL BACKGROUND

Primary: Batangas Christian School

Secondary: Batangas Christian School

Tertiary: Lyceum of the Philippines University – Batangas