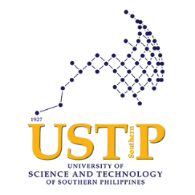
Logo, company name

Description automatically generated

University of Science and Technology of Southern Philippines

College of Information Technology and Computing

**IT214**

(Object Oriented Programming)

**Final Project**

**Team Spanish Latte Members:**

JV Paula Villaflor  
Jean Maverick Dela Cruz

Pauline Pacalda

Harvey Francis Magarin

Jelhoney Rollan

BSCS – CS2A

June -, 2024

**GAME 1: HIDDEN TREASURE GAME**

**Description/Overview**

Hidden Treasure is an exciting and suspenseful game inspired by classic treasure hunt games. The objective is simple yet thrilling: the player must choose from a series of boxes, one of which contains a hidden treasure.

The game begins with several boxes displayed on the screen. The player will click on the boxes one by one. If the player selects a box containing a bomb, the game ends immediately—it's game over. However, if the player successfully selects the box with the treasure, they win the game.

With each selection, the tension builds as the player narrows down their choices, hoping to avoid the bombs and find the coveted treasure. The random placement of the treasure and bombs in each game session ensures that every playthrough is unique, keeping players engaged and eager to try their luck again.

**Project Scope and Limitation**

**Scope:**

1. Gameplay Mechanics: The game involves a grid of boxes. The player must select boxes one by one to find the hidden treasure.
2. Winning Condition: The player wins by selecting the box that contains the treasure.
3. Losing Condition: The game ends if the player selects a box that contains a bomb.
4. Randomization: The placement of the treasure and bombs is randomized at the start of each game session to ensure variability and replayability.
5. User Interface: The game features an intuitive user interface with clearly identifiable boxes that the player can click to make their selections.
6. Feedback: Visual and audio cues indicate the outcome of each selection (e.g., treasure found, bomb hit).
7. Replayability: The game is designed for multiple playthroughs, with each session offering a new random configuration of the treasure and bombs.

**Limitations:**

1. Finite Choices: The number of boxes available is fixed for each game session, limiting the total number of selections the player can make.
2. Single Outcome per Playthrough: Each game session concludes with either a win or a loss, with no intermediate rewards or progress tracking.
3. No Difficulty Levels: The game does not feature adjustable difficulty settings; the challenge level remains consistent across all sessions.
4. Simple Graphics and Animations: The game prioritizes functionality over advanced graphics and animations, providing a basic visual experience.
5. Single-player Mode: The game is designed exclusively for single-player gameplay, with no options for multiplayer or competitive modes.
6. No Save Feature: Progress cannot be saved; each session must be completed in one sitting.
7. Limited Storyline: The game focuses on the core mechanic of selecting boxes and does not include an elaborate storyline or character development.
8. Potential Repetitiveness: While randomization adds variety, the fundamental gameplay loop may become repetitive over extended play periods.

**GAME 2: BAOBAO**

**Description/Overview**

The game is a 2D Baobao racing game set in the street of the philippines. Players control a race car and navigate through the streets, avoiding obstacles such as bumps, potholes, and even pedestrians crossing the road. The primary objective of the game is to last as long as possible without crashing into obstacles, with the score being determined by the distance covered by the player's car.

**Project Scope and Limitation**

**Scope**

1. Game Mechanics: Implement basic racing game mechanics such as acceleration, deceleration or braking, and steering.
2. Track Design: Design the imitation of the streets of the Philippines as the primary track for the game.
3. Obstacle Generation: Implement various obstacles like bumps, potholes, and pedestrians crossing the road to add challenges.
4. Scoring System: Develop a scoring system based on the distance covered by the player's car without crashing.
5. Object-Oriented Design: Utilize OOP principles to create modular, reusable, and maintainable code. Implement classes for the player's car, obstacles, track elements, and game manager.
6. User Interface: Create a simple and intuitive user interface displaying the score, distance covered, and any other relevant information.
7. Graphics and Sound: Incorporate basic graphics and sound effects to enhance the gaming experience.

**Limitation**

1. Graphics Complexity: Due to resource constraints, the game will have simple 2D graphics rather than advanced 3D graphics.
2. Limited Track Variety: The game will focus solely on the streets of the Philippines, limiting the variety of tracks available to the player.
3. Obstacle Variety: While there will be multiple types of obstacles, the variety may be limited compared to more complex racing games.
4. Advanced Features: Extensive customization options may not be included like car selecting options due to time and resource constraints.

**GAME 3: TRIAD OF TRUTH DUNGEON**

**Description/Overview**

You are a warrior that aims to find the truth hidden behind the mystery inside the dungeon. The dungeon is filled with troublesome puzzles and challenges. But you are not alone as monsters also go around this dungeon. You must find a way to beat these monsters with your power. The power of the ROCK of destiny, the PAPER of purification, and the SCISSORS of justice. Will you discover the truth?

**Project Scope and Limitation**

**Scope**

1. An RPG with a "choose a door" system, inside these doors the user will encounter either a puzzle that's java-related or a monster where they battle through rock, paper, and scissors like the original game. The user will have 3 lives and will only lose a life if they lose to a monster in a game of rock, paper, and scissors. If the user reaches 0 lives, they will be shown a game over screen and be sent back to the main menu.

**Limitation**

1. The player will only have to pick between doors 5 times before reaching the end of the game.
2. The interface will be made using jframe (or libGDX).
3. The in-game combat relies on RNG.