**CODE STRUCTURE**

The Maze Runner project is organized into a set of classes, each with a specific role in the game's functionality. The classes are organized within the de/tum/cit/ase/maze directory. Here is an overview of the class hierarchy and organization:

PACKAGE GAME\_COMPONENTS

**GameEntities:** This class is a fundamental component of the Maze Runner game. It represents any interactive object or character within the game world. This class provides a basic structure for handling the position, texture, and rendering of entities.

**Block:** Represents the maze's individual blocks, affecting the layout and navigability of the game environment.

**BlockType:** Enum representing the different types of blocks in a maze runner game, Each enum constant represents a distinct type of block within the game

**Bullet:** The Bullet class represents a projectile in the Maze Runner game, allowing the player to shoot and eliminate enemies or deactivate traps. This class extends the GameEntities class and introduces specific behavior for bullets.

**Enemy:** Abstracts the behavior and properties of enemies in the game, presenting challenges and obstacles for the player to overcome. It inherits the class GameEntities. It is the superclass for Ghost1 and Ghost2.

**Ghost1:** Represents a specific enemy type with unique behaviors, adding variety and depth to the game's challenge. Ghost 1 enemy uses the A\* pathfinding algorithm to navigate the game map.

**Ghost2:** Similar to Ghost1, this class likely represents another enemy type, further diversifying the gameplay experience. But unlike Ghost1, Ghost2 activates when the player is in proximity and follows the player.

**Key:** this class would handle collectible key that the player needs to gather to use complete the game.

**Maps:** Responsible for generating or managing the game's maps, directly influencing the level design and adding all the game entities .

**Player:** The Player class represents the main player character in the Maze Runner game. It handles various aspects of the player's behavior, including movement, animation, shooting, and collision detection with the game environment.

**Traps:** Manages in-game hazards that add an additional layer of challenge, requiring the player to avoid or finish them.

PACKAGE PATHFINDING

**A\_Star**: Implements the A\* pathfinding algorithm, potentially used for enemy movement or navigation.

**Node**: Represents the nodes within the A\* pathfinding algorithm, crucial for calculating paths in the game. This class is used by the A\* algorithm to store the cost values and parent node for each node in the path

PACKAGE SCREENS

GameOverScreen: This class represents the screen displayed when the game is over. It handles the display of the game over message, the player's score, and the time taken. Additionally, it provides a button to return to the main menu.

GameScreen: The GameScreen class is responsible for rendering the gameplay screen in the Maze Runner game. It handles the game logic, user interface (UI), and rendering of game elements. The class includes functionality for updating game entities, handling user input, detecting collisions, managing UI elements, and controlling the flow of the game.

MazeRunnerGame: The MazeRunnerGame class is the core of the Maze Runner game. It manages different screens, such as the menu, game, victory, and game-over screens. Additionally, it handles the global resources, including the SpriteBatch, Skin for UI, and character animations. The class is responsible for switching between screens, loading games from map files, and cleaning up resources when the game is disposed.

MenuScreen: Displays a UI to Start, Load or Quit the game.

WinScreen: Displays the score and winning message when the player wins the game.   
  
  
PACKAGE UTILITIES

BoundingBox: Handles collision detection, a fundamental aspect of game physics and interactions between entities.

Camera: Manages the game's camera system, affecting the player's view and experience of the game world.

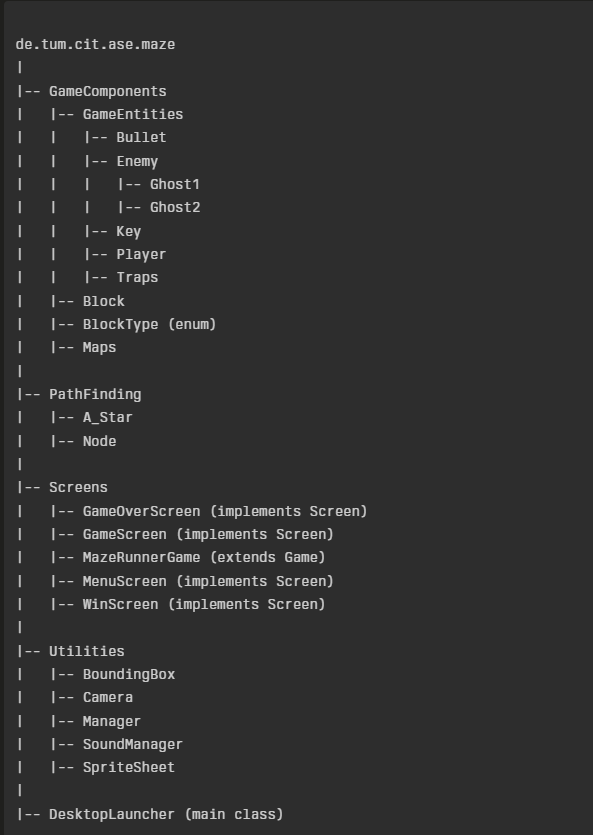
Manager: This class serves as a singleton manager for various utility classes in the game, here in SoundManager. It ensures that only one instance of each utility class is created and provides a global point of access to them.

SoundManager: Manages audio, including sound effects and music, contributing to the game's atmosphere and player feedback.

SpriteSheet: Handles sprite sheets for animations, affecting the visual representation and aesthetics of game entities.

Class Hierarchy

**CLASS HIERARCHY**



### **MAZE RUNNER INSTRUCTION MANUAL**

### Introduction

Maze Runner is an exciting game that challenges players to navigate through mazes, avoid enemies, and collect keys to advance. This manual provides an overview of the game's features and instructions on how to play.

### Game Objective

The goal of Maze Runner is to navigate through mazes, collect keys, and reach the exit while avoiding enemies and traps. Players must use strategy and quick reflexes to succeed.

### Game Screens

Maze Runner features the following screens:

* Menu Screen: The main menu where players can start a new game or access game settings.
* Game Screen: The primary gameplay screen where players navigate mazes, collect keys, and avoid enemies.
* Win Screen: The screen displayed when the player successfully completes a level.
* Game Over Screen: The screen displayed when the player loses the game.

### Player Controls

* Use the arrow keys to move the player character up, down, left, and right.
* Press the spacebar to fire bullets at enemies.

### Game Elements

Maze Runner includes the following elements:

* Player: The character controlled by the player.
* Enemies: Ghost1 and Ghost2 are enemies that patrol the mazes.
* Blocks: These are the maze walls that the player and enemies must navigate around.
* Keys: Players must collect keys to unlock the exit.
* Traps: Hazards that can harm the player.
* Bullets: Ammunition used by the player to destroy enemies.

### Game Mechanics

* Players must navigate through mazes, collect all keys, and reach the exit to win.
* Enemies will attempt to catch the player, so be cautious and avoid them.
* Bullets can be used to destroy enemies, but use them wisely as ammunition is limited.

How to Start and Play the Game

* The User has to run the main class named DesktopLauncher to load the game.
* The User clicks on Play button to start the game .
* OR the user can click on the Load button that opens up a NativeFileChooser with five different levels of maps to load based on difficulty.
* The User can click the Esc key to pause the game.
* The User can click on the Menu button to go back to Main Menu.
* The User can click on Exit button to quit the game.

**HAPPY GAMING, AND MAY YOU REACH THE EXIT WITH THE SATISFACTION OF A TRUE MAZE RUNNER CHAMPION!!!!!!**