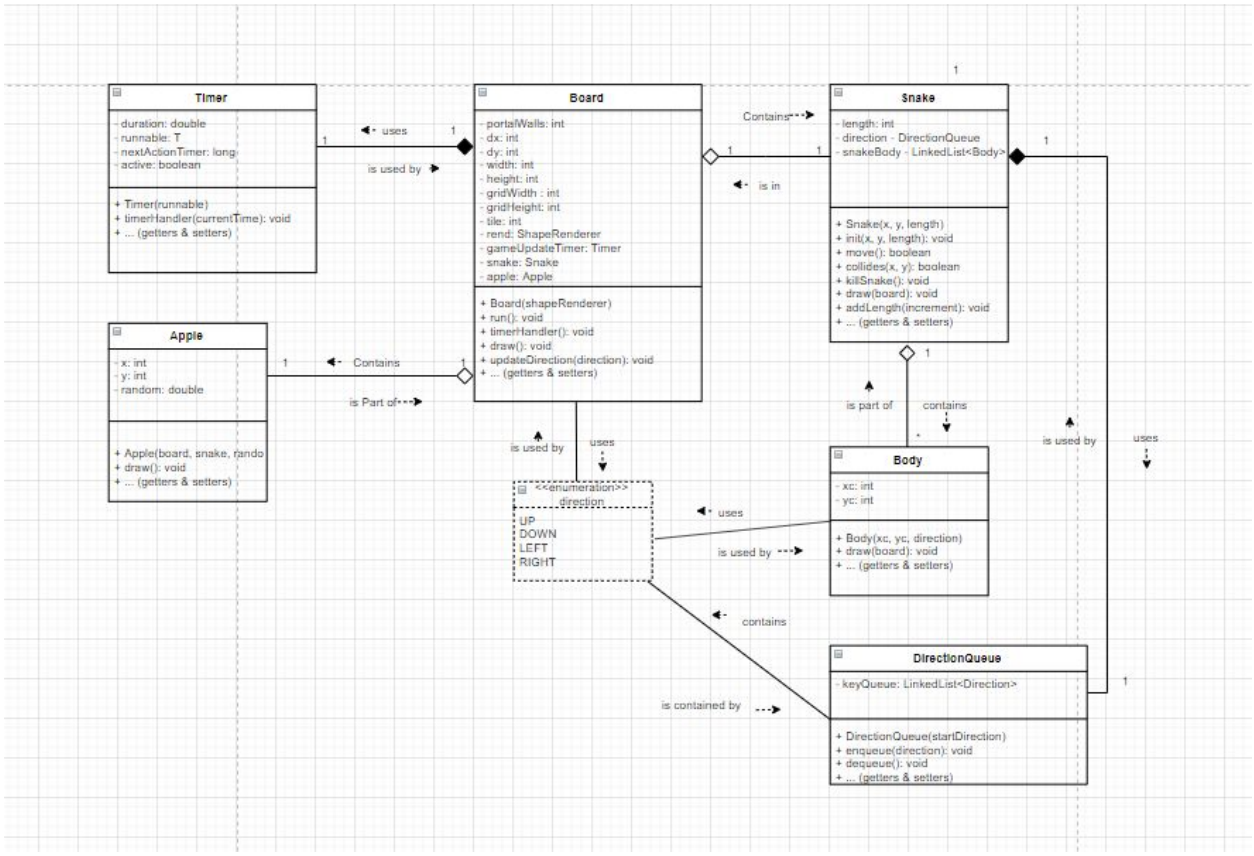
Class diagram explanation:

For creating the game we opted to go with the LibGdx library as it simplified the process of drawing and rendering objects on the screen, allowing us to focus more on the actual game logic. The main logic class within the game is the **Board** class. It handles all the basic variables such as boardsize, tilesize, and other features such as the boolean portalWalls (which determines if the snake dies when it hits a wall). Besides handling basic variables, the Board class always contains a **Snake**, **Apple** and **Timer** object within it. At the initialization of the board, a new **Snake** is created and the player is given control of the snake.
The **Snake** class houses the snake's length, snake's current direction, and the snakes body. The class also contains the logic for collision checking, killing the snake, and has methods for moving and drawing the snake's body parts on the game board.
Each snake class contains linked-list made of a minor class called, **Body**. Within the class there are two variables for representing the coordinates of the snake's body part, the class also contains methods for drawing the snake's body parts individually on the board.
The Snake also houses a **DirectionQueue**, which mainly exists to take user inputs in at a regular pace, and to transfer those user inputs back to the snake class.
The **Apple** class represents the snacks a snake can eat within a game. Within the apple class there are draw function to draw the apple on the board, as well as an input function to check if the randomly generated position of the apple does not interfere with the Snake and is within the board.
Finally there is the Timer Class which gets invoked every **frame update** by the Board class, and decides whether it has waited enough for the next **game update** (which moves the snake forward)
These are the classes that represent the base logic of our game. We chose to explain these classes as they are critical to the functioning of the game, or are critical to the function of the core logic classes.


Timer
- duration: double
- runnable: T
- nextActionTimer: long
- active: boolean
+ Timer(runnable)
+ timerHandler(currentTime): void
+ ... (getters & setters)
uses
is used by
Board
- portalWalls: int
- dx: int
- dy: int
- width: int
- height: int
- gridWidth : int
- gridHeight: int
- tile: int
- rend: ShapeRenderer
- gameUpdateTimer: Timer
- snake: Snake
- apple: Apple
+ Board(shapeRenderer)
+ run(): void
+ timerHandler(): void
+ draw(): void
+ updateDirection(direction): void
+ ... (getters & setters)
Contains
is in
Snake
- length: int
- direction - DirectionQueue
- snakeBody - LinkedList<Body>
+ Snake(x, y, length)
+ init(x, y, length): void
+ move(): boolean
+ collides(x, y): boolean
+ killSnake(): void
+ draw(board): void
+ addLength(increment): void
+ ... (getters & setters)
Apple
- x: int
- y: int
- random: double
+ Apple(board, snake, rando
+ draw(): void
+ ... (getters & setters)
Contains
is Part of
is part of
contains
is used by
uses
Body
- xc: int
- yc: int
+ Body(xc, yc, direction)
+ draw(board): void
+ ... (getters & setters)
<<enumeration>>
direction
UP
DOWN
LEFT
RIGHT
uses
is used by
contains
is contained by
DirectionQueue
- keyQueue: LinkedList<Direction>
+ DirectionQueue(startDirection)
+ enqueue(direction): void
+ dequeue(): void
+ ... (getters & setters)