Nitin Satishbabu

ID: 1152910

Class Descriptions

1) Die Class

Single Responsibility of class:
Handling functions of die, represents a die with a specified number of sides and provides a method to roll it, returning a random number.

Instance Variables:
@sides: Number of sides on the die (integer). Used to determine the range of possible roll outcomes.

Public methods:
a) def initialize(sides):
Responsibility: Initializes the die with a specified number of sides.
Arguments: sides (integer) - number of sides on the die.

b) def roll:
Responsibility: Returns a random integer between 1 and the number of sides.
Arguments: None

2) Player Class

Single Responsibility of class:
Represents a player in the game, allowing them to roll dice, remove certain rolls, calculate points, keep track of their score and whether the player is in the game or not.

Instance Variables:
@nameofplayer: Stores the player's name (string).
@dice: Array of Die objects representing the dice the player rolls.
@score: Tracks the player's total score (integer).
@out: Tracks whether the player is still in the game (Boolean).

Public methods:

a)def initialize(name, dice_count, sides):
Responsibility: Initializes the player with a name, a set number of dice, and dice sides.
Arguments: name (string) - player name, dice_count (integer) - number of dice, sides (integer) - sides on each die.

b)def name:
Responsibility: Returns the player's name.
Arguments: None

c)def roll_dice:
Responsibility: Rolls all the player's dice and returns the result.
Arguments: None.

d)def remove_dead_dice(rolls):
Responsibility: Removes dice that rolled a 2 or 5 from the player's dice set.
Arguments: rolls (array of integers) of the dices.

e)def calculate_points(rolls):
Responsibility: Calculates the score from the current dice rolls and updates the player's total score.
Arguments: rolls (array of integers) of the dices.

f)def any_remaining_dice:
Responsibility: Checks if the player has any remaining dice.
Arguments: None.

g)def total_score:
Responsibility: Returns the player's total score.
Arguments: None.

3) DropDeadGame Class

Single Responsibility of class:
Manages the overall game, including players, dice rolls, scoring, and game end conditions.

Instance Variables:
@players: Array of Player objects, representing all the players in the game.

@dice_count: The number of dice each player rolls (integer).
@sides: The number of sides on each die (integer).
@final_scores_hash: A hash storing players' names as keys and their final scores as values.

Public methods:
a) def initialize(sides, dice_count, player_count):
Responsibility: Initializes the game with a set number of sides, dice, and players.
Arguments: sides (integer) - sides per die, dice_count (integer) - dice per player, player_count (integer) - total players.

b) def play_round:
Responsibility: Simulates a round where each player rolls dice, calculates scores, removes dead dice, and checks if players are out of dice.
Arguments: None

c) def roll_dice(player):
Responsibility: Rolls dice for a given player.
Arguments: player (Player object) the player who is rolling.

d) def game_over:
Responsibility: Checks if the game is over (all players are out) and declares the winner.
Arguments: None

4) AutoDropDead Class

Single Responsibility of class:
Automates the whole process of a DropDead game and only focuses on this aspect.
Instance Variables:
None

Public methods:
a) def play_game(sides, dice_count, player_count):
Responsibility: Creates an instance of DropDeadGame with the arguments.
Arguments:
sides: The number of sides each die should have.
dice_count: The number of dice each player should have.
player_count: The number of players participating in the game.

Sequence Diagrams

Diagram 1

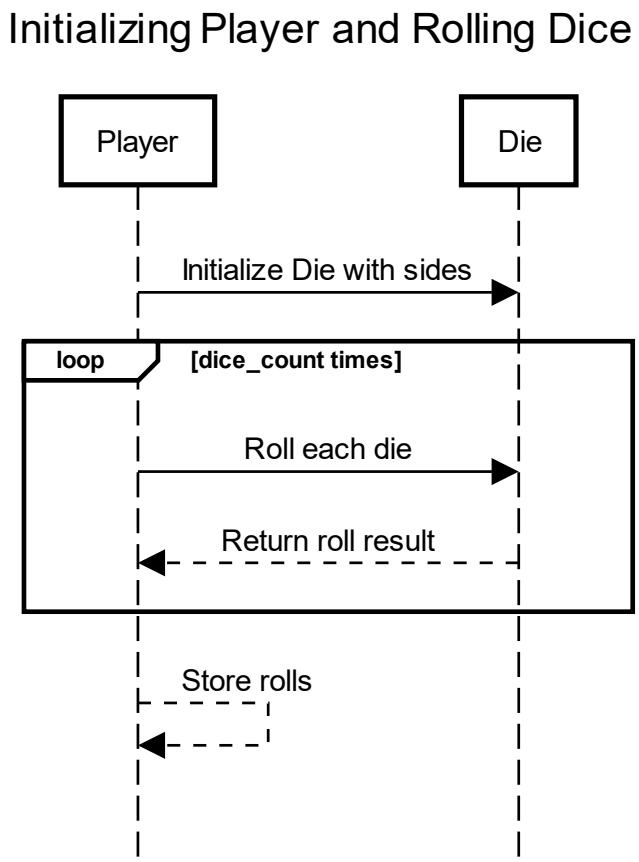


Diagram 2

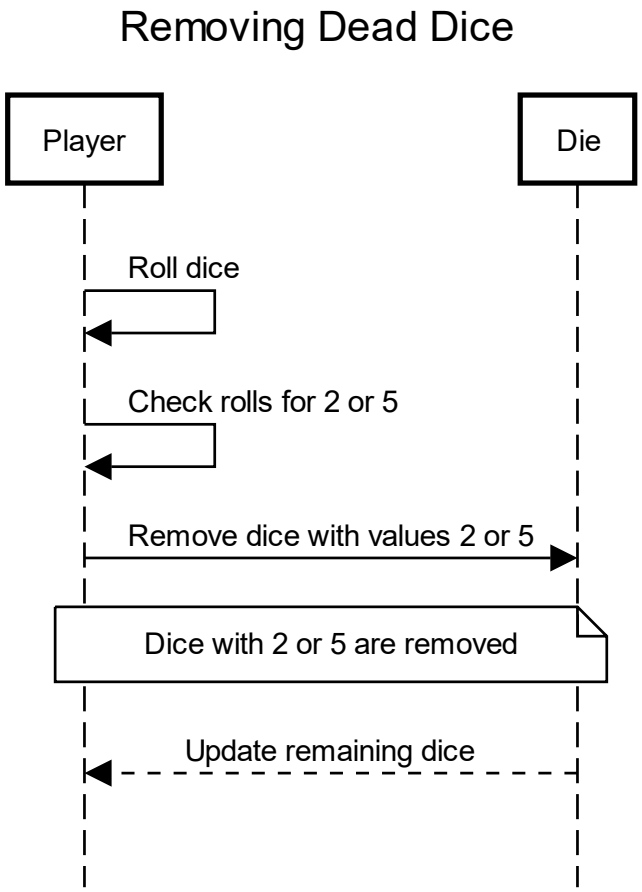
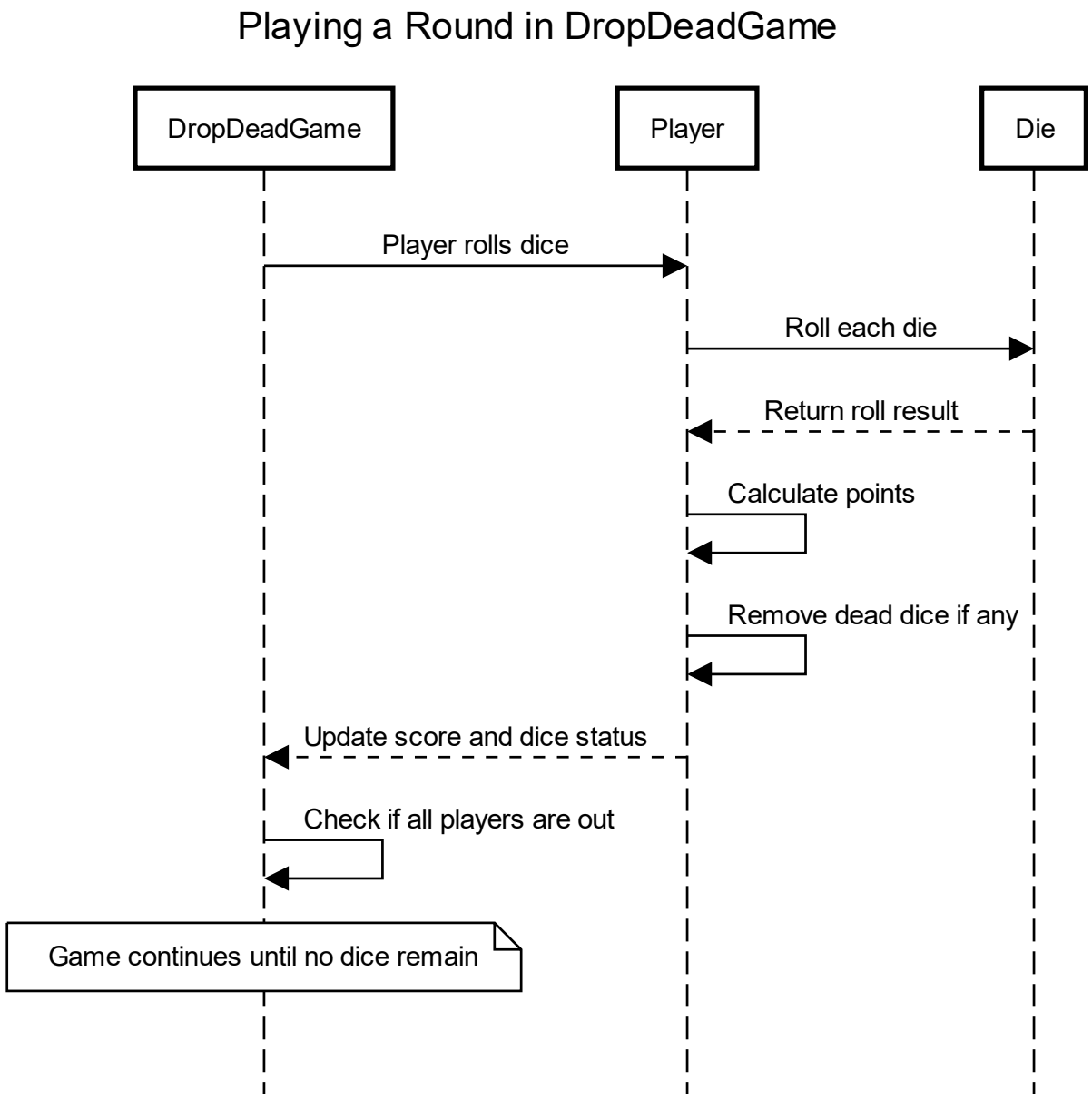

Removing Dead Dice
Player
Die
Roll dice
Check rolls for 2 or 5
Remove dice with values 2 or 5
Dice with 2 or 5 are removed
Update remaining dice

Diagram 3


Playing a Round in DropDeadGame
DropDeadGame
Player
Die
Player rolls dice
Roll each die
Return roll result
Calculate points
Remove dead dice if any
Update score and dice status
Check if all players are out
Game continues until no dice remain