Model

SnakeView

+ startButton : HtmlElement
+ user : String
+ password : String

+ update(SnakeGame model, List<Map> highscores) : void
+ generateField(SnakeGame model) : void
+ showHighscore(SnakeGame model, List<Map> highscores) : void
+ closeForm() : void
+ warn(String message) : void

view 1

SnakeGameController

+ snakeSpeed : Duration <<CONST>>
+ miceSpeed : Duration <<CONST>>
+ gamekeyCheck : Duration <<CONST>>
+ acceleration : Float <<CONST>>
+ gamekeyHost : String << CONST>>
+ gamekeyPort : Integer <<CONST>>
+ gameId : String <<CONST>>
+ gameSecret : String <<CONST>>

+ snakeTrigger : Timer
+ miceTrigger : Timer
+ gamekeyTrigger : Timer

- getHighscores() : Future<List>
- gameOver() : void
- newGame() : void
- moveMice() : void
- moveSnake() : void
- increaseSnakeSpeed() : void

gamekey 1

GameKey

+ gameId : String
- secret : String
+ uri : Uri

+ GameKey(String host, int port, String gameid, String secret)

+ registerUser(String n, String p) : Future<Map> { Map encodes JSON }
+ getUser(String id, String p) : Future<Map> { Map encodes JSON }
+ authenticate() : Future<Bool>
+ getUserId(String n) : Future<String>
+ listUsers() : Future<List<Map>> { Map encodes JSON }
+ getStates() : Future<List<Map>> { Map encodes JSON }
+ storeState(String uid, Map state) : Future<bool> { Map encodes JSON }

game 1

Snake

- dr : int { row direction of snake head }
- dc : int { column direction of snake head }
+ tangled : bool
+ onField : bool
+ notOnField : bool
+ length : int
+ head : Map { Map encodes first body element }
+ tail : Map { Map encodes last body element }
+ body : List<Map> { Map encodes snake body elements }

+ Snake.on(SnakeGame model)
+ move() : void
+ headUp() : void
+ headDown() : void
+ headLeft() : void

+ snake 1

- game 1

SnakeGame

+ size : int { Size of quadratic game field }
+ level : int { not used so far }
+ stopped : bool
+ running : bool
+ gameOver : bool
+ field : List<List<Symbol>> { game field }
+ miceCounter : int

+ increaseMiceCounter(int n) : void
+ moveSnake() : void
+ moveMice() : void
+ addMouse() : void

- game 1

+ mice 1..*

Mouse

+ row : int
+ col : int
- dr : int { row direction }
- dc : int { column direction }
+ pos : Map { Map encodes Mouse position }

+ Mouse.staticOn(SnakeGame g, int row, int col) { creates a non moving mouse }
+ Mouse.movingOn(SnakeGame g, int row, int col) { creates a moving mouse with random direction }
+ move() : void