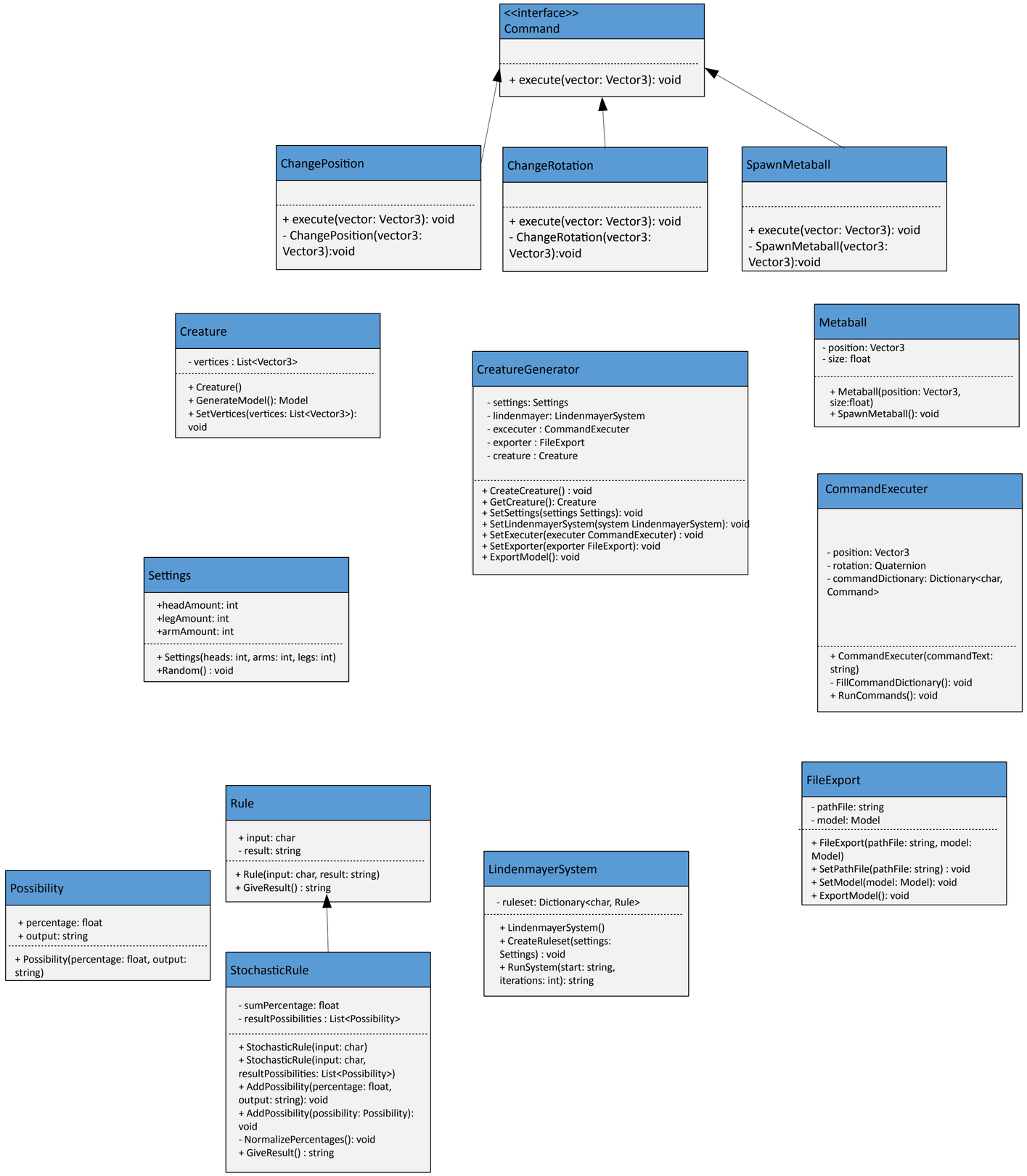
<<interface>>
Command
+ execute(vector: Vector3): void
ChangePosition
+ execute(vector: Vector3): void
- ChangePosition(vector3: Vector3):void
ChangeRotation
+ execute(vector: Vector3): void
- ChangeRotation(vector3: Vector3):void
SpawnMetaball
+ execute(vector: Vector3): void
- SpawnMetaball(vector3: Vector3):void
Creature
- vertices : List<Vector3>
+ Creature()
+ GenerateModel(): Model
+ SetVertices(vertices: List<Vector3>): void
CreatureGenerator
- settings: Settings
- lindenmayer: LindenmayerSystem
- excecuter : CommandExecuter
- exporter : FileExport
- creature : Creature
+ CreateCreature() : void
+ GetCreature(): Creature
+ SetSettings(settings Settings): void
+ SetLindenmayerSystem(system LindenmayerSystem): void
+ SetExecuter(executer CommandExecuter) : void
+ SetExporter(exporter FileExport): void
+ ExportModel(): void
Metaball
- position: Vector3
- size: float
+ Metaball(position: Vector3, size:float)
+ SpawnMetaball(): void
Settings
+headAmount: int
+legAmount: int
+armAmount: int
+ Settings(heads: int, arms: int, legs: int)
+Random() : void
CommandExecuter
- position: Vector3
- rotation: Quaternion
- commandDictionary: Dictionary<char, Command>
+ CommandExecuter(commandText: string)
- FillCommandDictionary(): void
+ RunCommands(): void
Rule
+ input: char
- result: string
+ Rule(input: char, result: string)
+ GiveResult() : string
Possibility
+ percentage: float
+ output: string
+ Possibility(percentage: float, output: string)
StochasticRule
- sumPercentage: float
- resultPossibilities : List<Possibility>
+ StochasticRule(input: char)
+ StochasticRule(input: char, resultPossibilities: List<Possibility>)
+ AddPossibility(percentage: float, output: string): void
+ AddPossibility(possibility: Possibility): void
- NormalizePercentages(): void
+ GiveResult() : string
LindenmayerSystem
- ruleset: Dictionary<char, Rule>
+ LindenmayerSystem()
+ CreateRuleset(settings: Settings) : void
+ RunSystem(start: string, iterations: int): string
FileExport
- pathFile: string
- model: Model
+ FileExport(pathFile: string, model: Model)
+ SetPathFile(pathFile: string) : void
+ SetModel(model: Model): void
+ ExportModel(): void