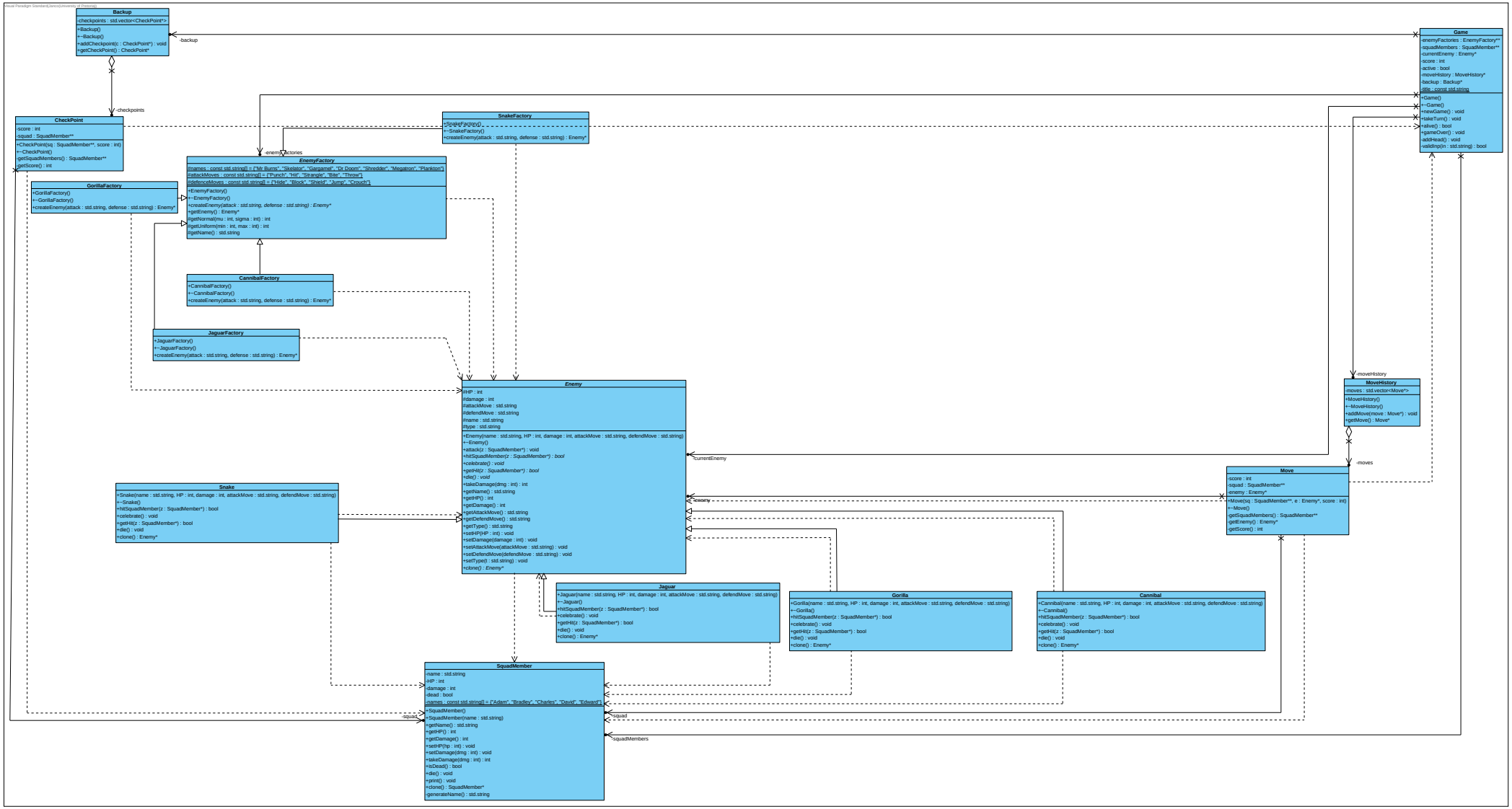
Backup
-checkpoints : std.vector<CheckPoint*>
+Backup()
+~Backup()
+addCheckpoint(c : CheckPoint*) : void
+getCheckPoint() : CheckPoint*
-backup
Game
-enemyFactories : EnemyFactory**
-squadMembers : SquadMember**
-currentEnemy : Enemy*
-score : int
-active : bool
-moveHistory : MoveHistory*
-backup : Backup*
-title : const std.string
+Game()
+~Game()
+newGame() : void
+takeTurn() : void
+alive() : bool
+gameOver() : void
-addHead() : void
-validInp(in : std.string) : bool
-checkpoints
CheckPoint
-score : int
-squad : SquadMember**
+CheckPoint(sq : SquadMember**, score : int)
+~CheckPoint()
-getSquadMembers() : SquadMember**
-getScore() : int
SnakeFactory
+SnakeFactory()
+~SnakeFactory()
+createEnemy(attack : std.string, defense : std.string) : Enemy*
-enemyFactories
EnemyFactory
#names : const std.string[] = {"Mr Burns", "Skelator", "Gargamel", "Dr Doom", "Shredder", "Megatron", "Plankton"}
#attackMoves : const std.string[] = {"Punch", "Hit", "Strangle", "Bite", "Throw"}
#defenceMoves : const std.string[] = {"Hide", "Block", "Shield", "Jump", "Crouch"}
+EnemyFactory()
+~EnemyFactory()
+createEnemy(attack : std.string, defense : std.string) : Enemy*
+getEnemy() : Enemy*
#getNormal(mu : int, sigma : int) : int
#getUniform(min : int, max : int) : int
#getName() : std.string
GorillaFactory
+GorillaFactory()
+~GorillaFactory()
+createEnemy(attack : std.string, defense : std.string) : Enemy*
CannibalFactory
+CannibalFactory()
+~CannibalFactory()
+createEnemy(attack : std.string, defense : std.string) : Enemy*
JaguarFactory
+JaguarFactory()
+~JaguarFactory()
+createEnemy(attack : std.string, defense : std.string) : Enemy*
-moveHistory
MoveHistory
-moves : std.vector<Move*>
+MoveHistory()
+~MoveHistory()
+addMove(move : Move*) : void
+getMove() : Move*
-moves
Enemy
#HP : int
#damage : int
#attackMove : std.string
#defendMove : std.string
#name : std.string
#type : std.string
+Enemy(name : std.string, HP : int, damage : int, attackMove : std.string, defendMove : std.string)
+~Enemy()
+attack(z : SquadMember*) : void
+hitSquadMember(z : SquadMember*) : bool
+celebrate() : void
+getHit(z : SquadMember*) : bool
+die() : void
+takeDamage(dmg : int) : int
+getName() : std.string
+getHP() : int
+getDamage() : int
+getAttackMove() : std.string
+getDefendMove() : std.string
+getType() : std.string
+setHP(HP : int) : void
+setDamage(damage : int) : void
+setAttackMove(attackMove : std.string) : void
+setDefendMove(defendMove : std.string) : void
+setType() : std.string) : void
+clone() : Enemy*
-currentEnemy
-enemy
Move
-score : int
-squad : SquadMember**
-enemy : Enemy*
+Move(sq : SquadMember**, e : Enemy*, score : int)
+~Move()
-getSquadMembers() : SquadMember**
-getEnemy() : Enemy*
-getScore() : int
Snake
+Snake(name : std.string, HP : int, damage : int, attackMove : std.string, defendMove : std.string)
+~Snake()
+hitSquadMember(z : SquadMember*) : bool
+celebrate() : void
+getHit(z : SquadMember*) : bool
+die() : void
+clone() : Enemy*
Jaguar
+Jaguar(name : std.string, HP : int, damage : int, attackMove : std.string, defendMove : std.string)
+~Jaguar()
+hitSquadMember(z : SquadMember*) : bool
+celebrate() : void
+getHit(z : SquadMember*) : bool
+die() : void
+clone() : Enemy*
Gorilla
+Gorilla(name : std.string, HP : int, damage : int, attackMove : std.string, defendMove : std.string)
+~Gorilla()
+hitSquadMember(z : SquadMember*) : bool
+celebrate() : void
+getHit(z : SquadMember*) : bool
+die() : void
+clone() : Enemy*
Cannibal
+Cannibal(name : std.string, HP : int, damage : int, attackMove : std.string, defendMove : std.string)
+~Cannibal()
+hitSquadMember(z : SquadMember*) : bool
+celebrate() : void
+getHit(z : SquadMember*) : bool
+die() : void
+clone() : Enemy*
SquadMember
-name : std.string
-HP : int
-damage : int
-dead : bool
-names : const std.string[] = {"Adam", "Bradley", "Charles", "David", "Edward"}
+SquadMember()
+SquadMember(name : std.string)
+getName() : std.string
+getHP() : int
+getDamage() : int
+setHP(hp : int) : void
+setDamage(dmg : int) : void
+takeDamage(dmg : int) : int
+isDead() : bool
+die() : void
+print() : void
+clone() : SquadMember*
-generateName() : std.string
-squad
-squad
-squadMembers