```
@startuml
' Configuration
skinparam classAttributeIconSize 0
scale 0.5

package com.arena.game.core {
class Core {
   - static Core core
   - static boolean _isEnteringTick
   - Map<ActionEnum, IMessageHandler> handlers
   - PriorityBlockingQueue<Message> messageQueue
   - ScheduledExecutorService scheduler

   - Core()
   - static Core getInstance() : Core
   - void receive(Message message)
   - void processMessages()
   - void handleMessage(Message message)
   - void sendGameState()
   - void retryLater(Message message)
   - void shutdown()
 }
}

Core "1" --> "1" Server : utilise
Core "1" --> "1" JavaWebSocket : utilise
Core "1" --> "1" Logger : utilise
Core "1" --> "0..*" Game : utilise
Core "1" --> "0..1" GsonWorker : utilise


package com.arena.game.entity.building{
class Inhibitor {
   - String skinAnimationIdle
   - String skinAnimationForSpawnHold
   - long skinAnimationDurationForSpawnHold
   - String skinAnimationForSpawn
   - long skinAnimationDurationForSpawn
   - String skinAnimationForDeath
   - long skinAnimationDurationForDeath
   - String skinAnimationForDeathHold
```

```
    - long skinAnimationDurationForDeathHold

    - Inhibitor(String id, int team)
    - String getSkinAnimationForIdle() : String
    - long getSkinAnimationDurationForSpawnHold() : long
    - String getSkinAnimationForSpawnHold() : String
    - long getSkinAnimationDurationForSpawn() : long
    - String getSkinAnimationForSpawn() : String
    - long getSkinAnimationDurationForDeath() : long
    - String getSkinAnimationForDeath() : String
    - long getSkinAnimationDurationForDeathHold() : long
    - String getSkinAnimationForDeathHold() : String
    - void die()
 }
}

Inhibitor "1" --> "1" Vector3f : attribut
Inhibitor "1" --> "0..1" Player : utilise
Inhibitor "1" --> "1" Game : utilise


package com.arena.game.entity.building{
 class Nexus {
    - String skinAnimationIdle
    - String skinAnimationForSpawnHold
    - long skinAnimationDurationForSpawnHold
    - String skinAnimationForSpawn
    - long skinAnimationDurationForSpawn
    - String skinAnimationForDeath
    - long skinAnimationDurationForDeath
    - String skinAnimationForDeathHold
    - long skinAnimationDurationForDeathHold

    - Nexus(String id, int team)
    - String getSkinAnimationForIdle() : String
    - long getSkinAnimationDurationForSpawnHold() : long
    - String getSkinAnimationForSpawnHold() : String
    - long getSkinAnimationDurationForSpawn() : long
    - String getSkinAnimationForSpawn() : String
    - long getSkinAnimationDurationForDeath() : long
    - String getSkinAnimationForDeath() : String
```

```
   - long getSkinAnimationDurationForDeathHold() : long
   - String getSkinAnimationForDeathHold() : String
   - void die()
 }
}

Nexus "1" --> "1" Vector3f : attribut
Nexus "1" --> "0..1" Player : utilise
Nexus "1" --> "1" Game : utilise


package com.arena.game.entity.building{
 class Tower {
   - Tower(String id, int team)
   - void die()
 }
}

Tower "1" --> "1" Vector3f : attribut
Tower "1" --> "1" Player : possesseur
Tower "1" --> "1" Game : utilise


package com.arena.game.entity.building{
 class TowerDead {
   - TowerDead(String id, int team)
 }
}

TowerDead "1" --> "1" Tower : référence
TowerDead "1" --> "1" Game : utilise


package com.arena.game.entity.champion{
 class Garen {
   - skinAnimationForIdle : String
   - skinAnimationForRun : String
   - skinAnimationForQ : String
   - skinAnimationDurationForQ : long
   - skinAnimationForW : String
   - skinAnimationDurationForW : long
   - skinAnimationForE : String
   - skinAnimationDurationForE : long
   - skinAnimationForR : String
   - skinAnimationDurationForR : long
   - skinAnimationForDeath : String
   - skinAnimationDurationForDeath : long
   - qDamage : int
   - wShield : int
   - eDamage : int
   - rDamage : int
   - Garen(String id, int team)
 }
}

Garen "1" --> "1" Game : joue
Garen "1" --> "0..*" Ability : possède
Garen "1" --> "1" Player : contrôlé_par


package com.arena.game.entity{
 abstract class Entity {
   - id : String
   - generalId : String
   - Entity(String id)
 }
}

Entity "1" --> "0..1" Nexus : positionné_sur
Entity "1" --> "0..1" Tower : défense
Entity "1" --> "0..1" Inhibitor : affecte
Entity "1" --> "1" Game : appartient_à


package com.arena.game.entity{
 class EntityCollider {
   - enabled : boolean
   - EntityCollider()
   - setEnabled(enabled : boolean) : void
   - isEnabled() : boolean
 }
}
```

```
EntityCollider "1" --> "1" Entity : collision_avec
EntityCollider "0..*" --> "1" Game : utilise

package com.arena.game.entity{
 class EntityNavMeshAgent {
   - enabled : boolean
   - EntityNavMeshAgent()
   - setEnabled(enabled : boolean) : void
   - isEnabled() : boolean
 }
}

EntityNavMeshAgent "1" --> "1" Entity : contrôle
EntityNavMeshAgent "1" --> "0..1" Nexus : cible
EntityNavMeshAgent "1" --> "0..1" Tower : cible
EntityNavMeshAgent "1" --> "0..1" Inhibitor : cible
EntityNavMeshAgent "1" --> "1" Game : appartient_à


package com.arena.game.entity{
 class EntityPositions {
   - BLUE_SPAWN : Position
   - RED_SPAWN : Position
   - CENTER_SPAWN : Position
   - BLUE_TOWERS : Map<String, EntityInit>
   - RED_TOWERS : Map<String, EntityInit>
   - BLUE_INHIBITORS : Map<String, EntityInit>
   - RED_INHIBITORS : Map<String, EntityInit>
   - BLUE_NEXUS : Map<String, EntityInit>
   - RED_NEXUS : Map<String, EntityInit>
 }
}

EntityPositions "1" --> "1" Entity : positionne
EntityPositions "1" --> "1" Game : appartient_à


package com.arena.game.entity{
 class EntityRigidbody {
   - boolean isKinematic
 }
}

EntityRigidbody "1" --> "1" Entity : rigidbody
EntityRigidbody "1" --> "1" Game : appartient_à


package com.arena.game.entity{
 class EntityTransform {
   - float scale
 }
}

EntityTransform "1" --> "1" Entity : transform
EntityTransform "1" --> "1" Game : appartient_à


package com.arena.game.entity{
 interface ILiving {
 }
}

ILiving <|.. Player : implémente
ILiving <|.. Tower : implémente
ILiving <|.. TowerDead : implémente
ILiving <|.. Nexus : implémente


package com.arena.game.entity{
 class LivingEntity {
  - int health
  - int maxHealth
  - int armor
  - int magicResist
  - int attackDamage
  - int abilityPower
  - boolean moving
  - boolean hasArrived
  - boolean skinAnimationLocked
  - boolean attackable
  - boolean entityLocked
  - boolean entityCastLocked
  - boolean entityMoveLocked
  - float moveSpeed
  - float rotationY
  - float posX
  - float posZ
  - float posY
  - float posSkinX
  - float posSkinZ
  - float posSkinY
  - float skinScale
  - float posXDesired
  - float posZDesired
  - float posYDesired
  - float skinAnimationSpeed
  - float skinAnimationBaseSpeed
  - String name
  - String skinAnimation
  - int team
  - long cooldownQStart
  - long cooldownWStart
  - long cooldownEStart
  - long cooldownRStart
  - long cooldownQEnd
  - long cooldownWEnd
  - long cooldownEEnd
  - long cooldownREnd
  - long cooldownQMs
  - long cooldownWMs
  - long cooldownEMs
  - long cooldownRMs
  - EntityCollider collider
  - EntityNavMeshAgent navMeshAgent
  - EntityRigidbody rigidbody
  - EntityTransform transform
  - Collection<String> nextObjective
  - static ScheduledExecutorService scheduler
 }
}

LivingEntity "1" --> "1" Entity : hérite_de
LivingEntity "1" --> "1" ILiving : implémente
LivingEntity "1" --> "0..1" BuffManager : possède
LivingEntity "1" --> "1" Game : appartient_à


package com.arena.game.entity {

 interface ILivingEntityCast {
  +int getQTotalDamage()
  +int getWTotalShield()
  +int getETotalDamage()
  +int getRTotalDamage()
  +Zone getQZone()
  +Zone getWZone()
  +Zone getEZone()
  +Zone getRZone()
  +void useQ()
  +void useW()
  +void useE()
  +void useR()
  +void setCooldownQStart(long)
  +long getCooldownQStart()
  +void setCooldownWStart(long)
  +long getCooldownWStart()
  +void setCooldownEStart(long)
  +long getCooldownEStart()
  +void setCooldownRStart(long)
  +long getCooldownRStart()
  +void setCooldownQEnd(long)
  +long getCooldownQEnd()
  +void setCooldownWEnd(long)
  +long getCooldownWEnd()
  +void setCooldownEEnd(long)
  +long getCooldownEEnd()
  +void setCooldownREnd(long)
  +long getCooldownREnd()
  +void setCooldownQMs(long)
  +long getCooldownQMs()
  +void setCooldownWMs(long)
  +long getCooldownWMs()
  +void setCooldownEMs(long)
  +long getCooldownEMs()
  +void setCooldownRMs(long)
  +long getCooldownRMs()
 }

 interface ILivingEntityLock {
  +void lockEntity(boolean)
  +boolean isLocked()
  +void lockEntityCast(boolean)
  +boolean isCastLocked()
  +void lockEntityMove(boolean)
  +boolean isMoveLocked()
  +void lockSkinAnimation(boolean)
  +boolean isSkinAnimationLocked()
 }

 interface ILivingEntityPos {
  +float getPosX()
  +void setPosX(float)
  +float getPosZ()
  +void setPosZ(float)
  +float getPosY()
  +void setPosY(float)
  +float getPosXDesired()
  +void setPosXDesired(float)
  +float getPosZDesired()
  +void setPosZDesired(float)
  +float getPosYDesired()
  +void setPosYDesired(float)
  +void setRotationY(float)
  +float getRotationY()
 }

 interface ILivingEntitySkin {
  +String getSkinAnimation()
  +void setSkinAnimation(String)
  +float getSkinAnimationBaseSpeed()
  +void setSkinAnimationBaseSpeed(float)
  +float getSkinAnimationSpeed()
  +void setSkinAnimationSpeed(float)
  +String getSkinAnimationForRunning()
```

```
    +String getSkinAnimationForIdle()
    +String getSkinAnimationForQ()
    +String getSkinAnimationForW()
    +String getSkinAnimationForE()
    +String getSkinAnimationForR()
    +String getSkinAnimationForDeath()
    +String getSkinAnimationForDeathHold()
    +String getSkinAnimationForSpawn()
    +String getSkinAnimationForSpawnHold()
    +long getSkinAnimationDurationForQ()
    +long getSkinAnimationDurationForW()
    +long getSkinAnimationDurationForE()
    +long getSkinAnimationDurationForR()
    +long getSkinAnimationDurationForDeath()
    +long getSkinAnimationDurationForDeathHold()
    +long getSkinAnimationDurationForSpawn()
    +long getSkinAnimationDurationForSpawnHold()
    +float getPosSkinX()
    +void setPosSkinX(float)
    +float getPosSkinZ()
    +void setPosSkinZ(float)
    +float getPosSkinY()
    +void setPosSkinY(float)
    +float getSkinScale()
    +void setSkinScale(float)
  }
}


package com.arena.game.entity {

  abstract class LivingEntityLock extends Entity implements ILivingEntityLock {
    -boolean entityLocked
    -boolean skinAnimationLocked
    -boolean entityCastLocked
    -boolean entityMoveLocked
  }

  abstract class LivingEntityPos extends LivingEntityLock implements ILivingEntityPos {
    -float posX, posY, posZ
    -float posXDesired, posYDesired, posZDesired
```

```
   -float rotationY
  }

  abstract class LivingEntitySkin extends LivingEntityCast implements ILivingEntitySkin {
   -float skinAnimationSpeed
   -float skinAnimationBaseSpeed
   -float posSkinX, posSkinY, posSkinZ
   -float skinScale
   -String skinAnimation
  }
}

package com.arena.game.entity {

  ILivingEntityLock <|.. LivingEntityLock
  ILivingEntityPos <|.. LivingEntityPos
  ILivingEntitySkin <|.. LivingEntitySkin

  LivingEntityLock --|> Entity
  LivingEntityPos --|> LivingEntityLock
  LivingEntitySkin --|> LivingEntityCast
  ILivingEntityCast <|.. LivingEntityCast
}


package com.arena.game.handler{
 class CastEHandler {
  - void handle(Message message)
 }
}

CastEHandler "1" --> "0..*" CastEvent : gère
CastEHandler "1" --> "1" Game : appartient_à
CastEHandler "1" --> "0..*" Player : utilise


package com.arena.game.handler{
 class CastQHandler {
  - void handle(Message message)
 }
}
```

```
CastQHandler "1" --> "0..*" CastEvent : gère
CastQHandler "1" --> "1" Game : appartient_à
CastQHandler "1" --> "0..*" Player : utilise


package com.arena.game.handler{
 class CastRHandler {
  - void handle(Message message)
 }
}

CastRHandler "1" --> "0..*" CastEvent : gère
CastRHandler "1" --> "1" Game : appartient_à
CastRHandler "1" --> "0..*" Player : utilise


package com.arena.game.handler{
 class CastWHandler {
  + void handle(Message message)
 }
}

CastWHandler "1" --> "0..*" CastEvent : gère
CastWHandler "1" --> "1" Game : appartient_à
CastWHandler "1" --> "0..*" Player : utilise


package com.arena.game.handler{
 class CloseGameHandler {
  + void handle(Message message)
 }
}

CloseGameHandler "1" --> "1" Game : gère
CloseGameHandler "1" --> "0..*" Player : notifie


package com.arena.game.handler{
 class CreateGameHandler {
  + void handle(Message message)
 }
}

CreateGameHandler "1" --> "1" Server : utilise
CreateGameHandler "1" --> "0..*" Player : crée
CreateGameHandler "1" --> "1" Game : crée


package com.arena.game.handler{
 interface IMessageHandler {
  + void handle(Message message)
 }
}

IMessageHandler "0..*" --> "1" Message : traite
IMessageHandler "0..*" --> "1" Player : concerne


package com.arena.game.handler {
  class JoinHandler {
    + void handle(Message message)
  }
}

JoinHandler "1" --> "1" Server : utilise
JoinHandler "1" --> "1" Game : rejoint
JoinHandler "1" --> "1" Player : gère
JoinHandler "1" --> "0..*" GamePlayer : associe


package com.arena.game.handler {
  class PlayerStateUpdateHandler {
    + void handle(Message message)
  }
}

PlayerStateUpdateHandler "1" --> "1" Server : utilise
PlayerStateUpdateHandler "1" --> "1" Game : utilise
PlayerStateUpdateHandler "1" --> "1" Player : met à jour
PlayerStateUpdateHandler "1" --> "1" IMessageHandler : implémente
```

```
package com.arena.game.utils {
  class EntityInit {
    - Position position
    - boolean isAttackable
    - Collection<String> nextObjectiveId
  }
}

EntityInit "1" --> "1" Server : utilise
EntityInit "1" --> "1" Game : initialise
EntityInit "1" --> "0..*" Entity : crée


package com.arena.game.utils {
  class Position {
    - Vector3f pos
    - float rotY
  }
}

Position "1" --> "0..*" Entity : positionne
Position "1" --> "1" Vector3f : utilise (coordonnées)


package com.arena.game {
 class Game {
    - GameNameEnum gameNameEnum
    - GameStatusEnum gameStatusEnum
    - ConcurrentHashMap<String, LivingEntity> livingEntities
    - ConcurrentHashMap<String, Player> players
 }
}

Game "1" --> "0..*" Player : contient
Game "1" --> "0..*" Team : contient
Game "1" --> "1" Server : appartient
Game "1" --> "0..*" GameMap : utilise
Game "1" --> "0..*" GameEvent : génère
Game "1" --> "0..*" GameStatistic : collecte
Game "1" --> "0..*" GameRule : applique
Game "1" --> "0..1" CloseGameHandler : utilise
Game "1" --> "0..*" Nexus : contient
Game "1" --> "0..*" Tower : contient
Game "1" --> "0..*" TowerDead : contient
Game "1" --> "0..*" Inhibitor : contient


package com.arena.game {
 class GameManager {
   - int managingTeam
 }
}

GameManager "1" --> "5" Game : gère
GameManager "1" --> "0..*" Team : manipule
GameManager "1" --> "0..1" Player : contrôle
GameManager "1" --> "0..*" GameEvent : traite
GameManager "1" --> "0..1" Server : communique


package com.arena.game.zone {
 interface Zone {
   + boolean isInZone(LivingEntity attacker, LivingEntity target)
 }
}

Zone "1" --> "0..*" Player : contient
Zone "1" --> "0..*" GameObject : contient
Zone "1" --> "0..1" GameMap : appartient
Zone "1" --> "0..*" EventListener : utilise


package com.arena.game.zone {
 class ZoneCircle {
   - float radius
 }
}

ZoneCircle "1" --> "1" Zone : spécialisation
ZoneCircle "1" --> "1" Vector3f : centre
ZoneCircle "1" --> "1" Float : rayon
```

```
package com.arena.game.zone {
 class ZoneCone {
   - float distance
   - float angleDeg
 }
}

ZoneCone "1" --> "1" Zone : spécialisation
ZoneCone "1" --> "1" Vector3f : direction
ZoneCone "1" --> "1" Float : angle
ZoneCone "1" --> "1" Float : distance



package com.arena.game.zone {
 class ZoneRectangle {
   - float width
   - float length
 }
}

ZoneRectangle "1" --> "1" Zone : spécialisation
ZoneRectangle "1" --> "1" Vector3f : position
ZoneRectangle "1" --> "1" Vector3f : size


package com.arena.game {
 enum GameNameEnum {
   - String gameName
 }
}

GameNameEnum "0..*" --> "1" Game : gameName


package com.arena.game {
 enum GameStatusEnum {
   - String gameStatus
 }
```

```
}

Game "1" --> "1" GameStatusEnum : status


package com.arena.network.message {
 class Message {
   - String uuid
   - ActionEnum action
   - GameNameEnum gameName
   - long timestamp
   - LivingEntity livingEntity
 }
}

Message "1" --> "1" Player : sender
Message "1" --> "1" Player : recipient
Message "0.." --> "1" Game : game
Message "0.." --> "1" ChatChannel : channel



package com.arena.network.response {
  interface IResponseSender {
   + void sendResponse(Response response, boolean silent)
   + void sendGameResponse(Response response, GameNameEnum gameName,
boolean silent)
   + void sendUuidResponse(String uuid, Response response, boolean silent)
 }
}

IResponseSender "1" --> "0..*" Response : envoie


package com.arena.network.response {
 class Response {
   - String _uuid
   - ResponseEnum _reponse
   - GameNameEnum _gameName
   - String _ability
   - String _text
   - String _notify
   - long _timestamp
   - Collection<LivingEntity> _livingEntities
 }
}

Response "1" --> "1" IResponseSender : envoyeur
Response "0..*" --> "1" Message : messages


package com.arena.network.response {
 class ResponseService {
   - static IResponseSender responseSender
 }
}

ResponseService "1" --> "0..*" Response : gère

package com.arena.network {
 class JavaWebSocket {
   - int port
   - static JavaWebSocket instance
   - final JsonService jsonService
   - ConcurrentHashMap<WebSocket, Player> webSocketToUuid
   - ConcurrentHashMap<Player, WebSocket> uuidToWebSocket

   - JavaWebSocket(int port)
 }
}

JavaWebSocket "1" --> "0..*" JavaWebSocketClient : gère
JavaWebSocket "1" --> "1" Server : utilise


package com.arena.network {
 class JavaWebSocketResponseSender {
   - WebSocket getConnByUuid(String uuid)
   - void sendToConn(WebSocket conn, Response response)
 }
}
```

```
JavaWebSocketResponseSender "1" --> "0..*" JavaWebSocketClient : envoie
JavaWebSocketResponseSender "1" --> "1" JavaWebSocket : utilise


package com.arena.player {
 enum ActionEnum {
   - Login
   - CreateGame
   - Join
   - CloseGame
   - CastQ
   - CastW
   - CastE
   - CastR
   - PlayerStateUpdate

   - action : String
   - getAction() : String
 }
}


package com.arena.player {
 class Player {
   - uuid : String
   - getUuid() : String
 }
}

Player "1" --> "0.." Game : participe
Player "1" --> "0.." Team : membre
Player "1" --> "0.." Inventory : possède
Player "1" --> "0..1" Session : utilise
Player "1" --> "0.." Achievement : obtient
Player "1" --> "0.." ChatMessage : envoie
Player "1" --> "0..1" Statistics : possède
Player "1" --> "0.." Role : attribué
Player "1" --> "0..1" Profile : contient
Player "1" --> "0..*" Match : joue
```

```
package com.arena.player {
 enum ResponseEnum {
   - Info
   - Logged
   - GameCreated
   - GameAlreadyExists
   - GamesLimitReached
   - Joined
   - PlayerAlreadyInGame
   - GameClosed
   - GameNotFound
   - GameState
   - YourEntityIs

   - getResponse() : String
 }
}


package com.arena.server {
 class Server {
   - static Server instance
   - ConcurrentHashMap<String, Player> players
   - ArrayList<Game> games
   - static final int MAX_GAMES
   - void createNexusInhibitorAndTowers(Game game)
   - void createEntities(Game game, Map<String, EntityInit> map, String type, int team)
 }
}

Server "1" --> "0.." Player : gère
Server "1" --> "0..1" JavaWebSocket : websocketServer
Server "1" --> "0.." Game : hotes
Server "1" --> "0..1" Logger : logger
Server "1" --> "0..*" ResponseService : responseServices


package com.arena.utils.json {
 class ServerManager {
 }
}
```

```
ServerManager "1" --> "0.." Server : manages
ServerManager "1" --> "0..1" Logger : logger
ServerManager "1" --> "0.." Player : players
ServerManager "1" --> "0..*" Game : games


package com.arena.utils.json {
 interface IJson {
 }
}

IJson "1" --> "0..*" GsonWorker : utilise


package com.arena.utils.json {
 class JsonService {
  - static IJson worker
 }
}

GsonWorker "1" --> "0.." JsonService : utilise
JsonService "1" --> "0.." IJson : utilise


package com.arena.utils {
 class TimeUtil {
 }
}

TimeUtil "1" --> "0.." Logger : usedBy
TimeUtil "1" --> "0.." Game : usedBy
TimeUtil "1" --> "0..*" Player : usedBy


package com.arena.utils {
 class Vector2f {
  - float x
  - float y
 }
}
```

```
Vector2f "1" --> "0.." Zone : utilisé
Vector2f "1" --> "0.." Player : position
Vector2f "1" --> "0.." Tower : position
Vector2f "1" --> "0.." Inhibitor : position
Vector2f "1" --> "0..*" Nexus : position


package com.arena.utils {
 class Vector3f {
  - float x
  - float y
  - float z
 }
}




package com.arena.utils.logger {
 class Logger {
   - static final ConcurrentLinkedQueue<String> LOG_QUEUE
   - static final int MAX_BUFFER_SIZE
   - static void enqueueLog(String level, String message, String customBefore)
   - static void enqueueLog(String level, String message)
 }
}

Logger "1" --> "1" LogWriter : utilise
Logger "1" --> "1" TimeUtil : utilise
Logger "1" --> "0..1" GameNameEnum : utilise
Logger "1" --> "0..*" JavaWebSocket : log
Logger "1" --> "0..*" Server : log
Logger "1" --> "0..*" Player : log
Logger "1" --> "0..*" Game : log


package com.arena.utils.logger {
 class LogWriter {
   - static final Semaphore LOG_SEMAPHORE
   - static final String LOG_FILE_PATH
   - static void processLogQueue(ConcurrentLinkedQueue<String> logQueue)
 }
}

LogWriter "1" --> "1" Logger : utilise
LogWriter "1" --> "1" File : utilise
LogWriter "1" --> "1" BufferedWriter : utilise
LogWriter "1" --> "1" Semaphore : attribut
LogWriter "1" --> "1" ConcurrentLinkedQueue : utilise



package com.arena {
 class Main {
 }
}

Main "1" --> "1" JavaWebSocket : utilise
Main "1" --> "1" Server : utilise
Main "1" --> "1" Logger : utilise
Main "1" --> "1" GsonWorker : utilise
```