System Design Artifacts

Systems

Here is a breakdown of the systems that encompass the game itself

* Action System – for handling any given action of the game?
* Artificial Intelligence System – for handling the AI of the game
* Entity System – for handling any entities within our game
  + Players themselves
  + Tiles
  + Meeples
  + Tile Stack
  + Board
* Board System – for handling the overall game itself
* Network System – for handling the networking we will have to implement later
  + Sender
  + Receiver
  + Parser
  + Adapter/port
* Scoring System – for handling the game scoring both in time and at end of game
* UI system – for handling UI stuff?

Architecture

Also mentioned in class BCE (ECB) – entity, control, boundary pattern/”Clean” architecture/hexagonal architecture

In particular, the following table illustrates our actors, boundaries, controllers, and entities as per the application TigerZone.

|  |  |  |  |
| --- | --- | --- | --- |
| **Actors (External Entities)** | **Boundary Objects** | **Controllers** | **Entities** |
| * TCP Server * Players * Computer AI | * FIT * JUnit * Net * CLI * GUI | * Scoring handler * Interaction handler * Tile handler * Board handler * AI handler | * Tiles * Board * Meeple * Terrain * Players * Tile stack |

Actor (External Entities)

* TCP Server
* Players
* Computer AI

Boundary Objects

* FIT
* JUnit
* Net
* CLI
* GUI

Controllers

* Scoring handler
  + Checks after each tile placed
  + Adds score to players
  + Run at end of game for remaining meeples
* Interaction handler
  + Assign player order
  + Give tile to players for board
  + Call turn scorer
  + Switch player
  + Check out of tiles
  + Check invalid placement
  + Execute end game
  + Call total scorer
  + Get player scores and return winner
* Tile handler
  + Creates tiles
  + Rotates tiles
* Board handler
  + Return state of board
  + Return valid tile placements
  + Add tile
  + Check tile placement
  + Add meeple
  + Return meeple
* AI handler

Entities

Objective: The entity system is responsible for managing all of the different types of entities within the game. These entities include:

* Tiles
  + ID for type
  + Coordinate location on the board
  + Multiplicity
  + Orientation
  + 3 points on each Edge (# of, or list of)
  + 1 point incenter
  + Points have Cities, Farmland, roads, monasteries, crossroads
* Farmland (Terrain)
  + Collective ID
  + Individual ID
  + List of connected terrain
  + Meeples (array)
  + Points it connects to
  + Neighboring cities
* Roads (Terrain)
  + Collective ID
  + Individual ID
  + List of connected terrain
  + Meeples (array)
  + Points it connects to
  + End road
* Monasteries (Terrain)
  + Collective ID
  + Individual ID
  + List of connected terrain
  + Meeples (array)
  + Points it connects to
  + Surrounding tiles
* Cities (Terrain)
  + Collective ID
  + Individual ID
  + List of connected terrain
  + Meeples (array)
  + Points it connects to
  + Shield
  + Completed
* Board
  + Number of tiles used
  + Board itself (2D array)
  + Tiles
  + Tile locations
* Meeple
  + Player
  + Type (thief, farmer, knight, monk)
  + If it is in use (Boolean)
  + Location/placement on a terrain
  + Its location/placement on the board
* Players
  + Indicator for if they are player1 or player2
  + Clock countdown
  + Score
  + Available meeples
* Tile stack
  + Number of remaining tiles
  + Stack of tiles in randomized order
  + Multiplicity of tiles
  + Tiles themselves
  + Alert when out