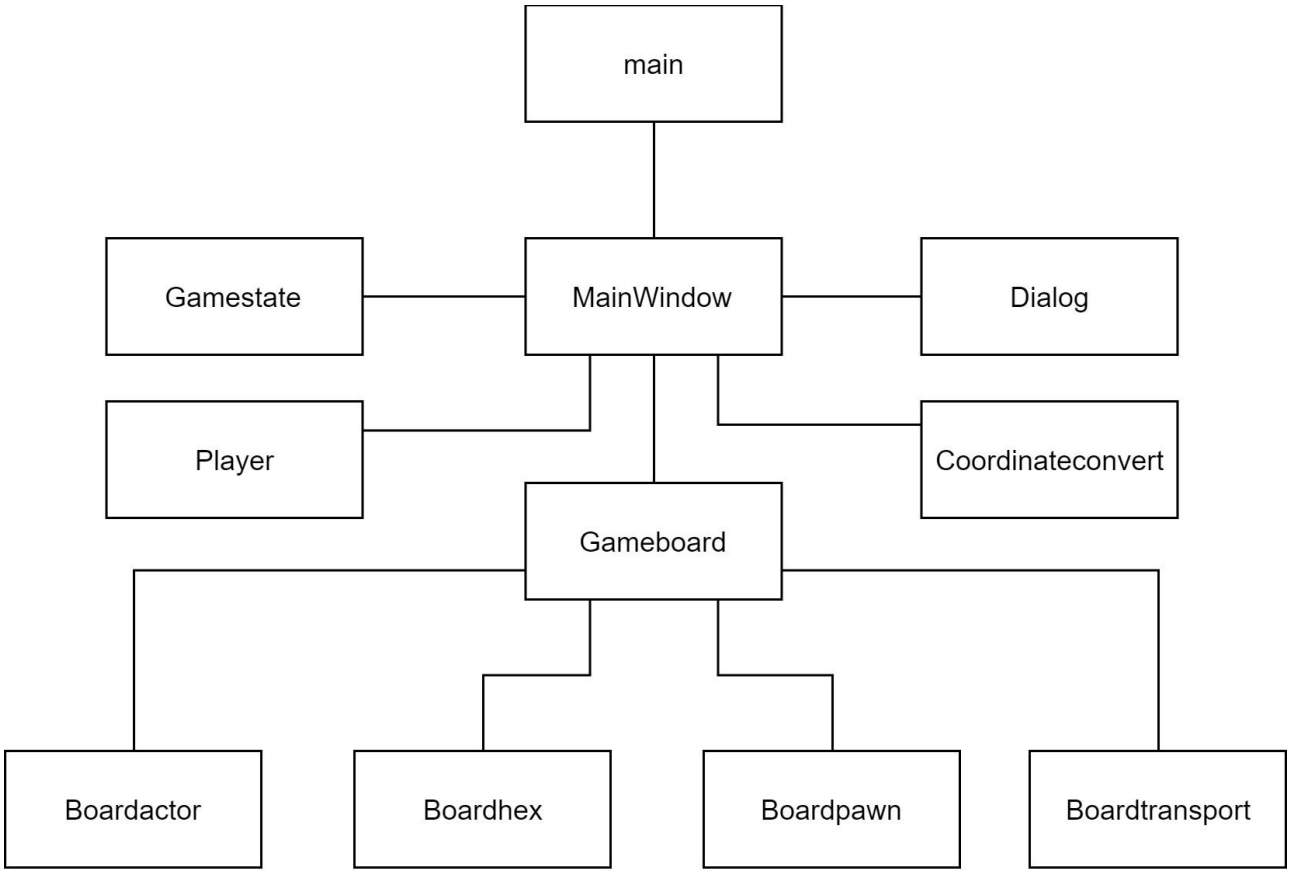
TIE-02402 Ohjelmointi 3: Tekniikat
Syksy 2018
Jaakko Ekman, Saku Jussila

Course project documentation

Software structure



Class responsibilities

BoardActor

- Graphical representation of actor on board
- Stores invidual actor data

BoardHex

- Graphical representation of hex on board and base for other graphics
- Call other classes' functions based on what is dropped to hex
- Stores hex data

BoardPawn

- Graphical representation of pawn on board
- Store pawn data

BoardTransport

- Graphical representation of transport on board
- Move pawns when transport is moved

coordinateconvert

- Convert cube coordinates to axial coordinates and vice versa

Dialog

- Gets player count from input and starts game

GameBoard

- Calls other graphic classes' functions to draw gameboard
- Stores all graphic objects

GameState

- Keeps track of the current state of the game
- Gives statistics for main window

MainWindow

- Initializes game
- Draws scene and updates statistics

Player

- Stores player data

Program function

- Dialog asks player count and sends it to mainwindow.
- Mainwindow initializes game.
- Movement phase: User drags pawn to boardhex which checks type of the object and calls for corresponding functions from gamerunner and mainwindow.
- Sinking phase: User clicks hex tile which flips it and adds actor/transport to boardhex by calling functions from gamerunner and mainwindow.
- Spinning phase: User spins wheel to get spin result. This result is transmitted to hex when actor/transport is dragged to it and it calls for corresponding functions from gamerunner and mainwindow.
- Gamephase is changed by clicking "Next gamephase" button or when tile is flipped. This also updates gamestate.
- Game ends when there are no more pawns left on board and gamestate relays winner to game message.

Extra features

- Visualised wheel
  - Implemented as QMovie in MainWindow method spinWheelMovie
- Animated wheel spin
  - Implemented as QMovie in MainWindow methods spinWheelMovie and spinWheel
- Scrollable game board
  - Implemented as bigger QGraphicsScene behind smaller view window

Division of work

Agreed:

Saku Jussila: Gameboard class and graphical interface

Jaakko Ekman: Gamestate and Player classes

Actual:

Saku Jussila: Gameboard, Coordinateconvert, Dialog, animated wheel spin, all graphical classes

Jaakko Ekman: Gamestate/tests for it, Player, gamephasing and some graphics for statistics

User manual

Movement phase: Drag pawn to another hex to move it. To move pawn into transport drag it inside of it. Transport is moved by dragging it and to get pawn out of it you have to drag it into another hex. When you don't want to move anymore, click "Next gamephase".

Sinking phase: Click boardhex to sink it and spawn actor/transport to board.

Spinning phase: Click "Spin wheel" to get spin result. Then move actor by dragging it or click "Next phase" to change turn to next player.

Rules:

- Goal is to get pawns to purple coral hexes on the corners of the gameboard. Player gets 10 points for reaching goal. Game ends when there are no more pawns on gameboard
- Maximum of 3 players
- Player has 3 actions for movement phase. Moving on land takes one action per hex and 3 on water.
- 3 pawns allowed per hex and player can't move through full hex.
- Only one actor and one transport allowed per hex.
- Hex tiles must be sunk in following order: beach (yellow), forest (green), mountain (light grey) and lastly peak tile (dark grey).
- Coral tiles (purple) cannot be sunk.
- Boat (brown box) has capacity of three pawns, dolphin (grey box) can carry one.
- Boat and dolphin take one action per hex to move
- Shark (grey hex) removes swimmers from hex

- Kraken (red hex) destroys boats, pawns are moved to water
- Seamunster (green hex) destroys boats and removes pawns
- Vortex destroys everything in neighbouring tiles
- Player has one spin per spinning phase. Spin result tells what actor player can move and how much.
- Player with the most points at the end wins

Tile layout: layout for our game should be 11 rounds of hexes. Here is an example layout used by us when testing the game:

```
{
  "Common": [
    { "name": "Peak"    , "layers": 1 },
    { "name": "Mountain", "layers": 2 },
    { "name": "Forest"  , "layers": 2 },
    { "name": "Beach"   , "layers": 2 },
    { "name": "Water"   , "layers": 3 },
    { "name": "Coral"   , "layers": 1 }
  ]
}
```

Bugs and missing features

We could not find any.