Domains API

By Jon “DarkJaslo”

Welcome! This is the API for Domains’ players. There are a few available functions that can be useful to know about:

Any public function in GameInfo.hh:

- cols() : returns the number of columns
- rows() : returns the number of rows
- round() : returns the current round
- points(int pl) : returns the points player *pl* has
- square(Position p) : returns the square in position *p* of the board
- unit(int uid) : returns the unit with id = *uid*
- unit(Position p) : returns the unit in position *p* of the board
- bubble(int bid) : returns the bubble with id = *bid*
- bubble(Position p) : returns the bubble in position *p* of the board
- posOk(Position p) : returns true if *p* is within the board
- randomNumber(int l, int r) : returns a random number in [l,r]
- randomPermutation() : returns a random permutation of {0,1,2,3}
- units(int pl) : returns the units owned by player *pl*
- numberOfUnits() : returns the number of units belonging to each player
- maxNumberOfUnits() : returns the maximum number of units

Player.hh:

- me() : returns your player’s id

PlayerOrders.hh:

- move(int unitId, Direction dir) : orders unit *unitId* to move in direction *dir*
- attack(int unitId, Direction dir) : orders unit *unitId* to attack in direction *dir*
- ability(int unitId) : orders unit *unitId* to use the ability

Utility.hh (also includes *Order* and *Matrix*, which may be useful to you):

1. Directions are the basic type to issue orders, handle positions and more:
    - null, up, down, left, right, UL (up-left), UR (up-right), DL (down-left), DR (down-right)
    - operator << (can be printed)

2. Positions are a very basic type that can be used by the player. To see all details, look at its definition in Utility.hh:
    - operators ==, !=, <
    - operators +, += (with a Direction)
    - operators +, +=, -, -= (with a Position)
    - to(Position p): if *p* is adjacent (diagonal counts) to this Position, returns the direction from this to *p* (null Direction in all other cases)
    - operator << (can be printed)
3. Units have some more methods:
    - id() : returns its id
    - player() : returns the player that owns it
    - position() : returns its position
    - upgraded() : returns whether it is upgraded or not
    - energy() : returns its energy
4. Bubbles:
    - roundsToPop() : returns the rounds to pop, -1 if it hasn't been attacked
    - player() : returns which player's colour this bubble is
    - position() : returns its position
5. Squares store the information of a position:
    - pos() : returns its position
    - painted() : returns whether it is painted or not
    - painter() : returns the id of the painter player
    - drawn() : returns whether it is drawn or not
    - drawer() : returns the id of the drawer player
    - unitDrawer() : returns the id of the drawer unit
    - border() : returns whether this square separates domains or not
    - ability() : returns whether this square is affected by an ability
    - roundsUntilAbilityEnd() : returns the number of rounds until the ability affecting this square ends
    - empty() : returns whether this square is empty
    - hasUnit() : returns whether this square has a unit
    - unit() : if and only if it has a unit, returns such unit
    - hasBonus() : returns whether this square has a bonus
    - bonus() : if and only if it has a bonus, returns such bonus
    - hasBubble() : returns whether this square has a bubble
    - bubble() : if and only if it has a bubble, returns such bubble