Carl Stevenson, Krishna Patel, Michael Bates, Troy Wynn

CS334

Wumpus CRC cards

* Class
  + player
* Responsibility
  + Stores information about the player character. Manages the action of the player character. Manages the movement of the player character. Has an empty function to pull the pin of a grenade.
* Collaboration
  + container class
  + game class
  + playercommand class
  + pickupcommand class
  + dropcommand class
  + inventory command class
  + quitcommand class
  + shootcommand class
  + throwcommand class
  + cavern class
  + cave class
  + gameobject class
  + gameio class
  + tunnel class
  + Random class (Java standard)
* Class
  + carlplayer
* Responsibility
  + Stores information about an ai player.
  + Can check if it is next to an entity.
  + Can check if it is next to a pit.
  + Has a function to act.
  + Has a function to check if it should shoot an arrow.
    - Uses an old function that no longer exists in player class
  + Has a function to shoot an arrow.
* Collaboration
  + player class
  + carlMap class
  + cavern class
  + cave class
  + game class
  + tunnel class
  + gameio class
  + Random class (Java standard)
* Class
  + wumpus
* Responsibility
  + Stores information about the wumpus.
  + Has a function to act in a finite state machine.
  + Has function for each of the states: snore, sleepy, roar, wander
  + Can move towards the player.
  + Contains a pathfinding function for use in moving towards the player.
  + Has a function to move to a specific cave.
  + Has a class defined within the wumpus class file called PathQueue
    - Stores the paths to look through
    - Contains the functions neccesary to have a queue for use in the pathfinding function of the wumpus
      * isEmpty
      * queueing and dequeuing functions
      * size
      * findstart
* Collaboration
  + Random class (Java standard)
  + gameio class
  + game class
  + player class
  + cavern class
  + cave class
  + tunnel class
  + PathQueue class
  + gameobject class
  + PathQueue class does not collaborate with any other class.
* Class
  + tunnel
* Responsibility
  + Stores information about the paths between caves, and whether it has a pit or not.
  + Can set the destination of the tunnel
  + Has functions to return the information in the class
    - return where the tunnel leads to
    - return if it as a pit or not
* Collaboration
  + Does not collaborate with another class.
* Class
  + playercommand
* Responsibility
  + Interprets dynamic player commands such as pick up, drop, inventory, quit, shoot and throw things.
* Collaboration
  + player class
  + pickupcommand class (child class)
  + dropcommand class (child class)
  + inventorycommand class (child class)
  + quitcommand class (child class)
  + shootcommand class (child class)
  + throwcommand class (child class)
* Class
  + pickupcommand
* Responsibility
  + Sets commandletter p as pickup option to playercommand class
  + Gets location of player from cave class
  + Checks size of contents
  + Sends message to gameio class
  + Picks up item that is in cave, but not player itself and adds to player’s inventory.
  + Removes picked up item from the cave
* Collaboration
  + player class
  + playercommand class (super class)
  + gameio class
  + container class
  + cave class
  + gameobject class
  + game class
* Class
  + dropcommand
* Responsibility
  + Sets commandletter r as drop option to playercommand class
  + Gets location of player from cave class
  + Checks size of contents
  + Sends message to gameio class
  + Removes item from players inventory
  + Adds removed item from the players inventory to cave
* Collaboration
  + player class
  + playercommand class (super class)
  + gameio class
  + container class
  + cave class
  + gameobject class
  + game class
* Class
  + inventorycommand
* Responsibility
  + Sets commandletter i as inventory option to playercommand class
  + Gets location of player from cave class
  + Checks size of contents
  + Sends message to gameio class about what items the player has.
* Collaboration
  + player class
  + playercommand class (super class)
  + gameio class
  + container class
  + cave class
  + gameobject class
  + game class
* Class
  + quitcommand
* Responsibility
  + Sets commandletter q as quit option to playercommand class
  + Exits from the game if player chooses option q.
* Collaboration
  + player class
  + playercommand class (supper class)
* Class
  + shootcommand
* Responsibility
  + Sets commandletter a as drop option to playercommand class
  + Checks which direction to shoot the arrow at (n, e, s, w, u, d)
  + Gets location of player from cave class
  + Sends message to gameio class about whether the player has an arrow or not.
  + Checks whether tunnel exists or not
  + Gets arroweffect from gameio class
  + Checks whether wumpus is in the destination cave or not.
  + If wumpus is not in the desitination cave number of arrows is decreased
  + wumpus is moved to different cave.
* Collaboration
  + player class
  + playercommand class (supper class)
  + gameio class
  + cave class
  + game class
  + wumpus class
  + tunnel class
* Class
  + throwcommand
* Responsibility
  + Sets commandletter t as throw option to playercommand class
  + Checks content size and sends message to gameio class about whether player has anything to throw or not.
  + Gets the name of item to be thrown to gameobject class
  + Checks which direction to throw item at (n, e, s, w, u, d)
  + Gets the command of direction from gameio class
  + Gets location of player from cave class
  + Checks whether tunnel exists or not
  + Checks whether wumpus is in the destination cave or not.
  + Gets wumpuskill effect from gameio class to kill the wumpus with thrown object
  + If throwing object is grenade, then grenadeeffect is been called from gameio class
  + Then the thrown object is moved from the destination cave
* Collaboration
  + player class
  + playercommand class (supper class)
  + gameio class
  + cave class
  + game class
  + wumpus class
  + tunnel class
* Class
  + threestateobject
* Responsibility
  + Create object with three states, set current state to first state, and set the name of the object to the state name (not used).
* Collaboration
  + Gameobject class
* Class
  + grenade
* Responsibility
  + Create grenade with pin and timer in **cave** class.
  + Return whether or not pin was pulled (not used).
* Collaboration
  + Gameobject class
  + Gameio class
  + Cave class
  + Player class
* Class
  + torch
* Responsibility
  + Create torch with a count of 5 in **game** class.
  + Send message to **Cave Class** telling if the torch is lit or not.
  + Add 5 to count of torch in **cave** class.
* Collaboration
  + Gameobject class
  + Game class
  + Cave class
* Class
  + wumpusfinder
* Responsibility
  + Wumpusfinder object created in **game** class.
  + Sends message to itself if neighboring caves have a player.
  + Gets called from **wumpus** class, outputs text stating what direction the wumpus is in.
* Collaboration
  + Gameobject class
  + Game Class
  + Wumpus class
  + Wumpusfinder class
* Class
  + gameobject
* Responsibility
  + Creating gameobjects in **cave, dropcommand, game, grenade, pickupcommand, player, threestateobject, throwcommand, torch, wumpus,** and **wumpusfinder** classes.
* Collaboration
  + Object Class
  + Cave class
  + Dropcommand class
  + Game class
  + Container Class
  + Grenade Class
  + Pickupcommand class
  + Player class
  + Throwcommand class
  + Threestateobject Class
  + Torch Class
  + Wumpus Class
  + Wumpusfinder Class
* Class
  + Cave
* Responsibility
  + Used to create cave objects in **carlRoom**, **carlplayer**, **cavern**, **dropcommand**, **game**, **inventorycommand**, **pickupcommand**, **player**, **shootcommand**, **throwcommand**, **wumpus**, and **wumpusfinder** classes.
  + Returns a random tunnel to **carlMap** and **game** classes.
  + Prints warnings for any information the player should know (not used).
  + Sends message to **cavern** class with information about requested cave.
  + Receives request from **cavern** class to output information about cave.
  + Outputs description of cave (only called within class).
* Collaboration
  + Container class
  + CarlRoom class
  + Carlplayer class
  + Cavern class
  + Inventorycommand class
  + Pickupcommand class
  + Player class
  + Shootcommand class
  + Throwcommand class
  + Wumpus class
  + Wumpusfinder class
* Class
  + Cavern
* Responsibility
  + Creates cavern object in **game** class.
  + Reads in each cave and adds to cavern object (used within class).
  + Outputs information about each cave in cavern (not used).
* Collaboration
  + Game class
* Class
  + Container
* Responsibility
  + Inserts objects in container in **cave**, **game**, **play**, and **wumpus** classes.
  + Removes objects from container in **dropcommand, pickupcommand, player, throwcommand, wumpus,** and **wumpusfinder** classes.
  + Checks if container has object in **carlRoom, carlplayer, cave, game, player, shootcommand, throwcommand,** and **wumpusfinder** classes.
  + Gets an object by its name (not used).
* Collaboration
  + Cave class
  + Cavern class
  + Player class
  + Game class
  + Wumpus class
  + Pickup command class
  + Throwcommand class
  + Wumpusfinder class
  + Dropcommand class
  + CarlRoom class
  + Shootcommand class
  + Carlplayer class
* Class
  + gameio
* Responsibility
  + Handles all input from users and all the output to the command line.
* Collaboration
  + wumpus class
  + player class
  + gameobject class
  + tunnel class
* Class
  + game
* Responsibility
  + Keep track of all the player information.
  + Keep track of the information of the cave.
  + Keep track of all the wumpus information.
* Collaboration
  + player class
  + cave class
  + cavern class
  + tunnel class
  + wumpus class
  + gameio class
* Class
  + carlRoom
* Responsibility
  + Keep track of the information about a single room in the caver
* Collaboration
  + cave class
  + game class
  + cavern class
  + tunnel class
* Class
  + carlMap
* Responsibility
  + To keep a record of the rooms explored by the AIplayer.
  + To help predict which movements would be most effective path for several scenerios.
* Collaboration
  + carlRoom class
  + cave class
  + tunnel class
  + gameio class