Assignment 5 : Peaches Adventure

In assignment 5, you will (partially) build a game about peaches. The game has players moving about in a world trying to find, eat and store peaches.

Each member in your group will have to implement two classes (a `Location` subclass and a `Player` subclass). Your team is free to add more to the game if you wish but that will not be necessary.

Locations

The world that the game takes place in consists of different locations. A location can have zero or more players in it, zero or more peaches in it, and it may have some behaviour of its own.

Each member will implement a location subclass.

1) A `Home` is the home location. There is only one home in each world. This is where several of the players will begin the game. Is is a place for players to drop off (to stockpile) peaches. The home keeps track of how many peaches each player brings back.

The home will also create a new `Helper` player to go help another player in distress (low health). It will give the helper the location of the player in distress and some peaches to give to the distressed player to eat.

2) A `PeachGrove` has a bunch of peach trees, zero or more peaches (from the trees) and perhaps some bees (that may sting a player). When a player enters a peach grove for the n-th time then n bees might sting that person each with a 50% probability. Each sting will result in 5 health points being lost to the player. If the player doesn't die from bee stings then they can eat and take away as many peaches as they are able (if there are that many available). A grove does not grow more peaches as the game progresses.

3) A `PeachPit` is a location with a large hole. When a player enters (falls into) the pit, they lose 1/2 of their health points. (Just use integer division here.)

A peach pit also remembers who has already fallen into the pit. When a player falls in for the second (or more) time, they are instantly transported to the `Home` location, lose 1/2 their health points and their turn ends.

There may initially be several peaches in the pit. A player that falls in for the first time may take them (to eat or carry).

4) A `BearsDen` has a bear in it. If a player offers a bear two or more peaches then the bear will leave the

player alone (do nothing). If the player does not give two or more peaches to the bear then the bear will attack the player resulting in 25 health points being deducted. The bear will remember each player that gave it peaches and will not take peaches again or attack that player again.

Players

Each group member will implement a player subclass.

1) A `PeachHunter` searches for peaches to bring back to the home location. A peach hunter can carry lots of peaches (as many as 100). The peach hunter should return to home to deposit its peaches once they have 50 or more peaches. A peach hunter will remember where peach groves are when they find them and go back to them to get more peaches until the grove runs out. If a hunter's health is less than 50 then it can only carry 25 peaches. It must then put any excess peaches in the current location if this happens.

2) A `PitFinder` looks for peach pits and remembers where they are (locations). Each time a piut finder finds a new pit it will go back home and report this to the home (when the player enters home). When a pit finder and peach hunter interact, a finder will reveal the location of a pit to the finder in exchange of 5 peaches. A finder with health less then 30 can only carry 10 peaches. If their health drops down to this (if they fall in a pit or attacked by a bear for example) they must drop any excess peaches in the current location.

3) A `SmartyPants` just wanders around the world trying to sell advice to others. A smarty pants is given knowledge of all peach pits, bear dens and peach groves in the world. When it interacts with peach hunters they will reveal a grove's location for 7 peaches; when they interact with pit finders they will reveal a pit or bears den for 6 peaches; and when they interact with a thief they will steal all the theirs peaches (instead of the thief stealing theirs). There is only one smarty pants in the world. If a hunter or finder do not have enought peaches then no transaction is made.

A smarty pants is fair to peach hunters and pit finders. They will not reveal the same location twice to the same player (unless they already revealed all of the locations).

4) A `Helper` is created and dispatched from the home location whenever a player (pit finder or peach hunter) asks for help. They bring a bunch of peaches to the player (giving them to the player when they interact with them). Once a helper helps someone they go back home (and do nothing for the rest of the game). A helper will not give any peaches to a bear.

5) A `PeachThief` that will try to steal peaches from other players, as well as pick up peaches it finds along the way. When it interacts with a hunter or pit finder, it will repeatedly steal peaches (with probability 75%) until it fails to steal one (or the other player runs out of peaches). A thief will immediately eat the first peach successfully stolen in any interaction.

At any time (during the play method) a player can eat a peach they have. This will rejuvenate them by adding some health equal to the ripeness of the peach. So if a peach has ripeness 10 then eating it will give the player 10 health points. However, if the peach is bad, then it will deduct the value of the ripeness from their health instead.

If a player's health ever drops below 10 they can call home for help. The `Home` loction will dispatch a `Helper` player to send some peaches to eat.

Peaches Adventure

The provided `PeachesGame` program is a simple starting point for the game. Essentially, it lets each player in the world have their turn (play method) and repeats until the home location has accumulated enuogh peaches.

Additions

You are free to make additions to the "game". For example, you can add a human player if you wish.

Project

The project aspect of this work will involve presenting demos of the code. Each team should come prepared with several "programs" that each demonstrate one small aspect of the game.

For example, one demo might just have a small world with a single player to illustarte that the player behaves as designed. You might start with a peach hunter and a peach grove and demo how it finds the grove, takes peaches back to home and then goes back to the grove. You might then create a smarty pants player exactly where the hunter is to show that they interact woth each other properly. You have the freedom to construct and manipulate the state of the game to illustrate the desired proprty. Each demo should illustarte one or two behavours (bit not much more).

Hints

Look at the specification and incrementally add behaviour to your classes. You do not need to get everything working at once. Build up your classes one behaviour at a time.