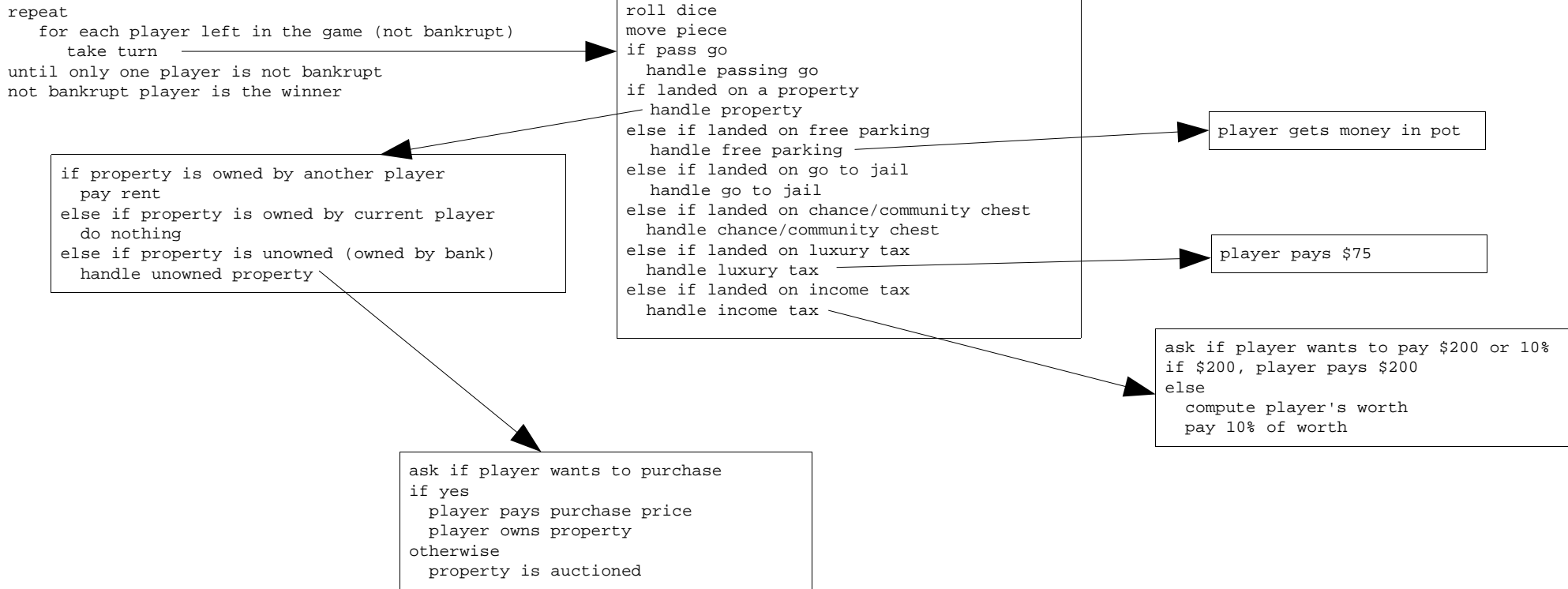


Things

<i>thing</i>	<i>properties/attributes</i>	<i>manipulations</i>	<i>notes</i>
money	<ul style="list-style-type: none"> amount 		maybe just store as an int
board	<ul style="list-style-type: none"> layout (what squares are where) position of players number of houses/hotels on each property amount of money in free parking pot 	<ul style="list-style-type: none"> move piece determine current position of a player 	think of physical game board, and what information it displays and what things are placed on it
player	<ul style="list-style-type: none"> amount of money is bankrupt? properties owned position on board associated piece/token 	<ul style="list-style-type: none"> retrieve amount of money adjust amount of money (add/subtract) take turn 	with the idea of the board storing player positions, that may not need to be stored with each player is it necessary to separately track whether or not the player is bankrupt, or is the amount of money enough?
pieces/tokens			
dice		<ul style="list-style-type: none"> roll determine if last roll was doubles determine value of last roll 	a pair of dice, since don't need an individual die for anything
property	<ul style="list-style-type: none"> name purchase price rent (at various levels) mortgage price is mortgaged? cost of houses/hotels number of houses/hotels 	<ul style="list-style-type: none"> determine rent 	
chance card community chest card		<ul style="list-style-type: none"> 	
deck of chance cards deck of community chest cards	<ul style="list-style-type: none"> the cards in the deck 	<ul style="list-style-type: none"> 	
bank	<ul style="list-style-type: none"> unsold properties number of unsold houses/hotels 	<ul style="list-style-type: none"> 	a logical concept more than a physical thing might be useful for keeping track of information such as the unsold properties, available houses/hotels, etc
game display		<ul style="list-style-type: none"> 	separate details of user interface/how to draw board on the screen from the actual information stored by the board to simplify and to make it easier to offer alternate user interfaces (e.g. graphics version and a text-only version)

Game Logic



Design Strategy – Data First!

- first
 - write down/obtain detailed specifications (game rules, etc) – a description of what the program is supposed to do
- then repeat as needed
 - identify the major things/concepts in the specifications – tend to be nouns, some or all will generally become classes
 - identify properties/attributes of the things/concepts – will generally become instance variables
 - identify ways in which the things/concepts are manipulated – will generally become methods
 - identify the information needed to carry out each operation – determines what class will contain the method and/or parameters to that method
 - write pseudocode to help identify things, attributes, operations – start with one-sentence description, then refine each step by repeatedly coming up with one-sentence descriptions
- write header files for classes
- refine pseudocode in terms of the classes/methods you've developed / implement