

Monopoly Design

November 3, 2008

Candidate Classes

These are the candidate classes that we developed in class considering the sections of the rules labeled: the play, buying property and paying rent.

GameController

- Responsibilities:
 - Figure out who has the highest roll
 - **Keep track of players**
 - **Know whose turn it is**
- Collaborators:
 - Player
 - Board

Dice

- Responsibilities:
 - throw and return a number
- Collaborators:

Property

- Responsibilities:
- Collaborators:
 - Player
 - Bank / banker
 - Title deed card
 - House
 - Hotel

Token

- Responsibilities:
 - **move to a specific place**
 - **move some number of spaces**
 - know my position
- Collaborators:
 - Board space
 - **Board**

Player

- Responsibilities:
 - **move token to go**
 - **play a turn**
 - buy
 - pay rent
 - pay taxes
 - pass go
 - return dice total
 - keep track of last dice total
- Collaborators:
 - Token
 - Money
 - Dice
 - Property

Board space

- Responsibilities:
- Collaborators:
 - Board space GO

Banker

- Responsibilities:
 - Sell at auction
- Collaborators:
 - Player
 - Bank
 - TitleDeedCard
 - Property

Board

- Responsibilities:
 - **Know the order of board spaces**
 - **Determine which space is a particular distance after another space**
 - **Be immutable**
- Collaborators:

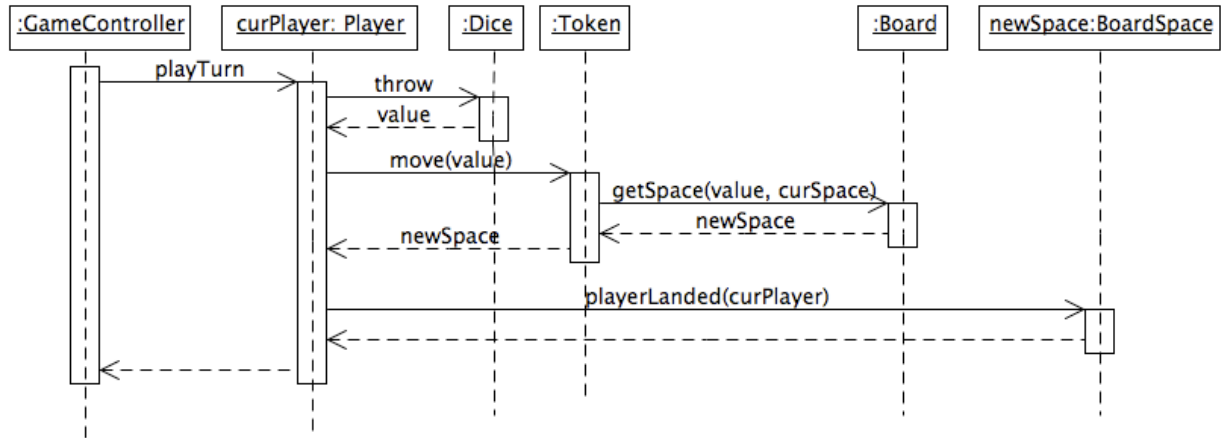
Other candidate classes for our design

- Title deed card
- Bank
- Color group
- Rent
- Auction
- House
- Hotel
- Money
- Owner
- Board space GO

Some other comments on our design

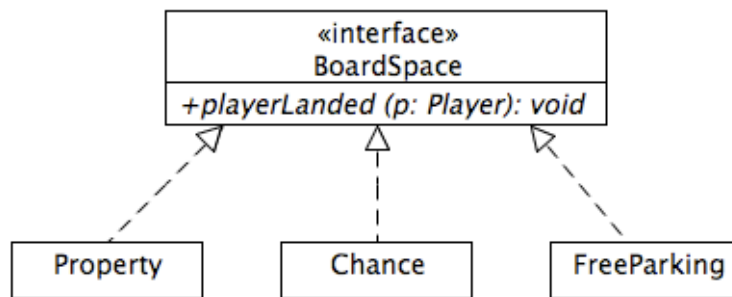
- A buyer is a transient role that a player may have
- Owner is a relationship between a player and property
- Should house and hotel be combined into a single class?

Sequence Diagram for the Player's playTurn method.



There is more to a turn than this diagram shows - such as buying houses/hotels, trading with other players, etc. Also missing are the details about what playerLanded does. This will vary a lot depending on the type of square the player landed on. For example, an unowned property vs. an owned property vs. chance vs. free parking. As the class diagram below shows, we will rely on polymorphism to carry out the details appropriate to the type of square.

Class Diagram showing BoardSpace Hierarchy



There would be other types implementing BoardSpace as well.