Ludo

https://github.com/Hoverbear/seng-271-group-project
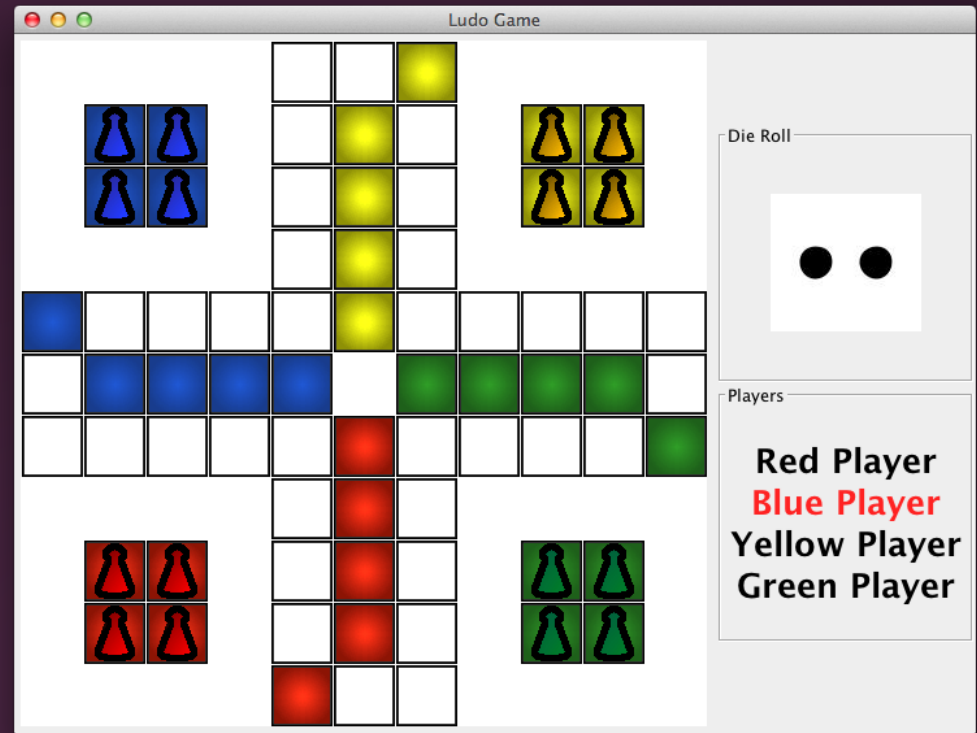
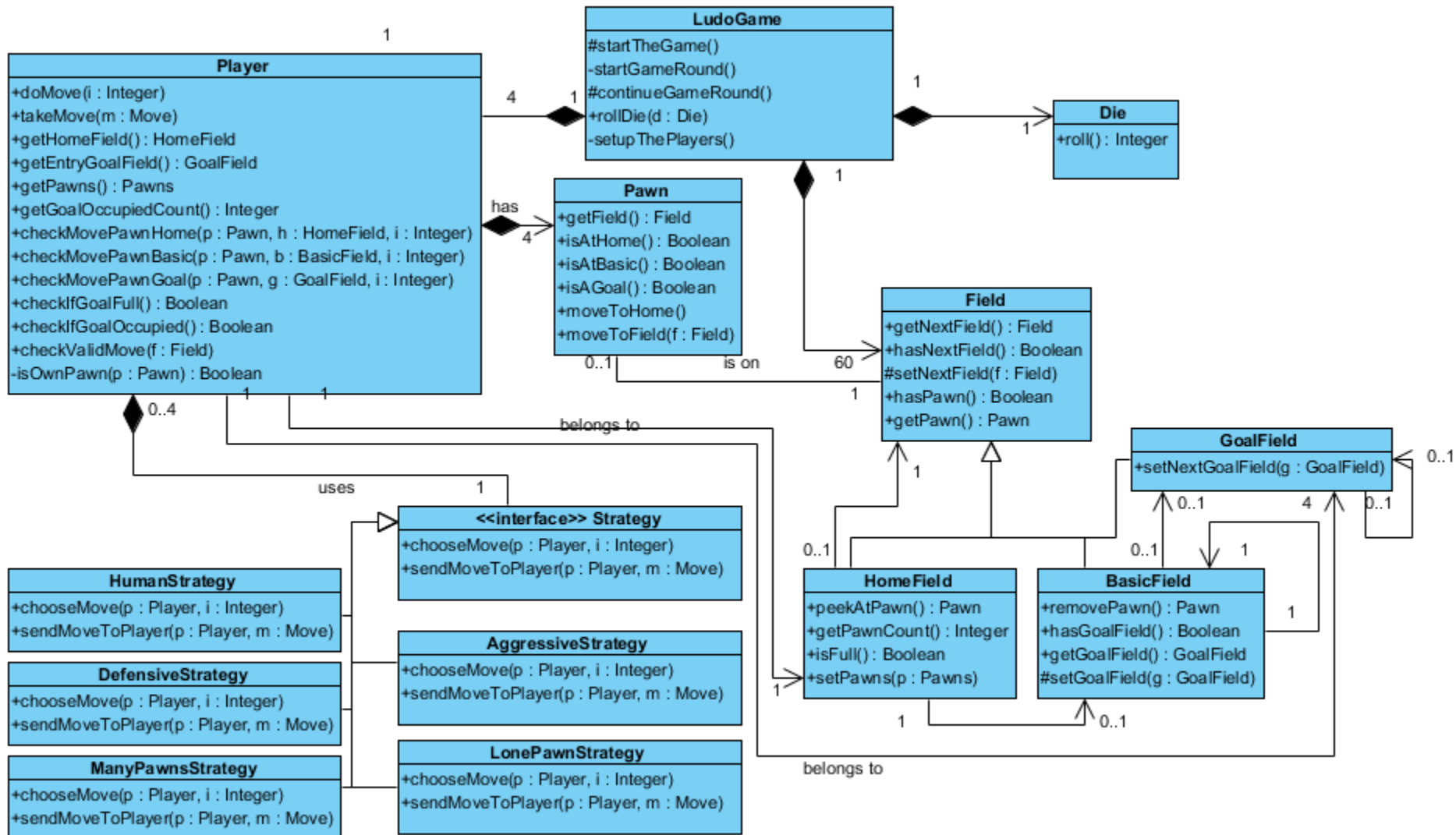*Daniel Faulkner (@maxentropic), Andrew Hobden (@Hoverbear), and Scott Byrne (@mrgoboom)*

UML

Pros

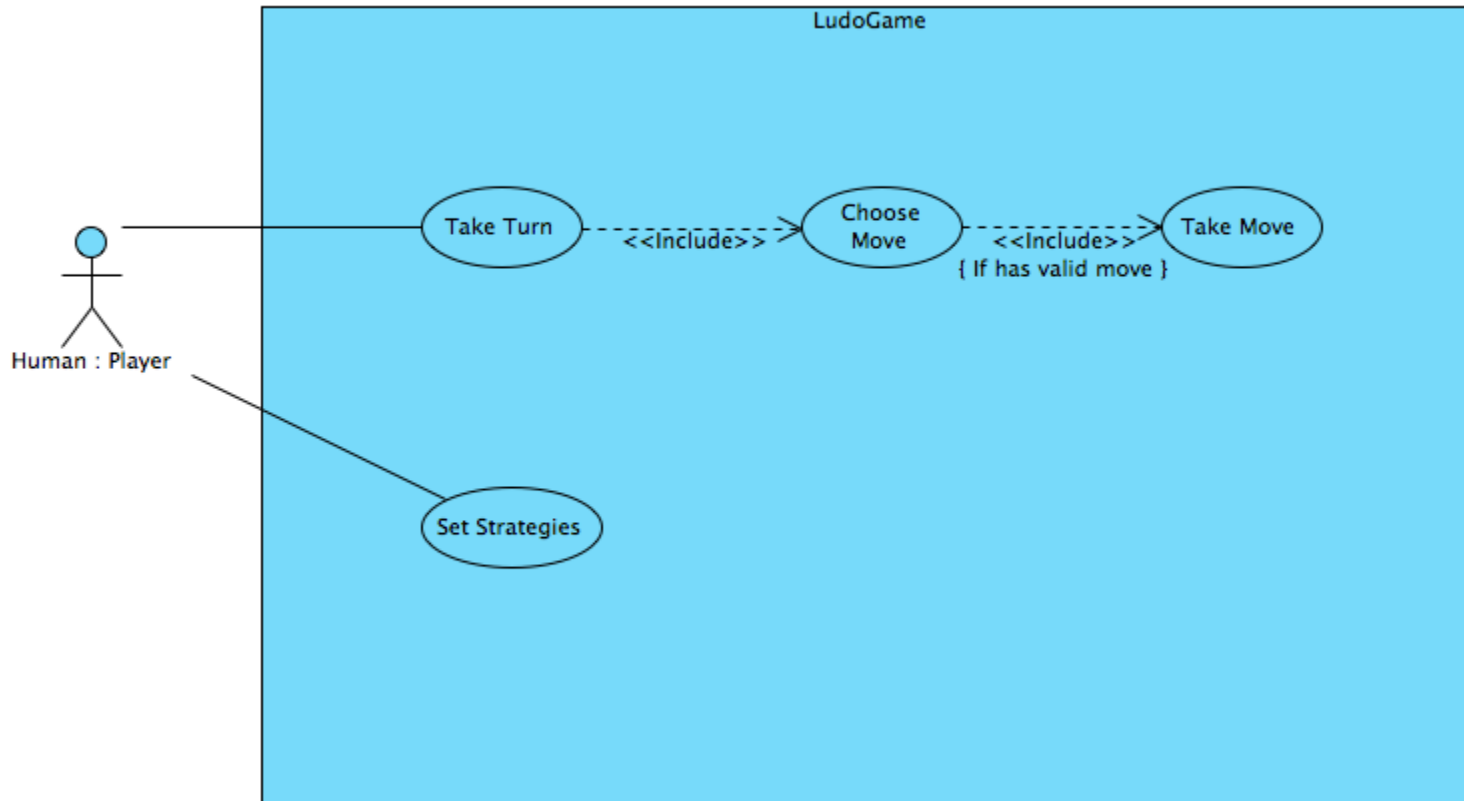- UML provided a nice way to explain to each other what we pictured.

Cons

- Took **a lot** of time.
- Really annoying to collaboratively edit diagrams.
- Not very iterative (Slightly due to project design.)
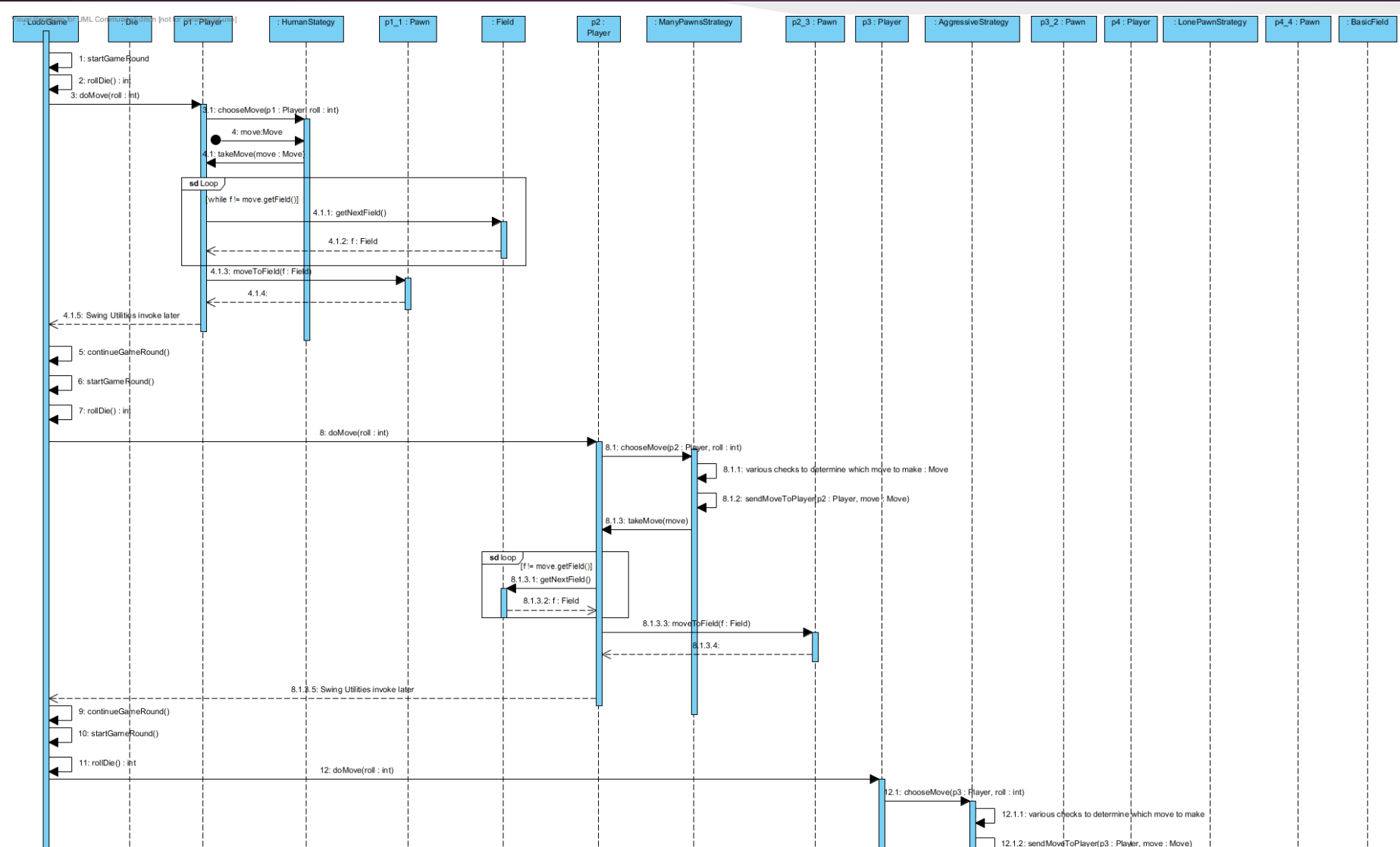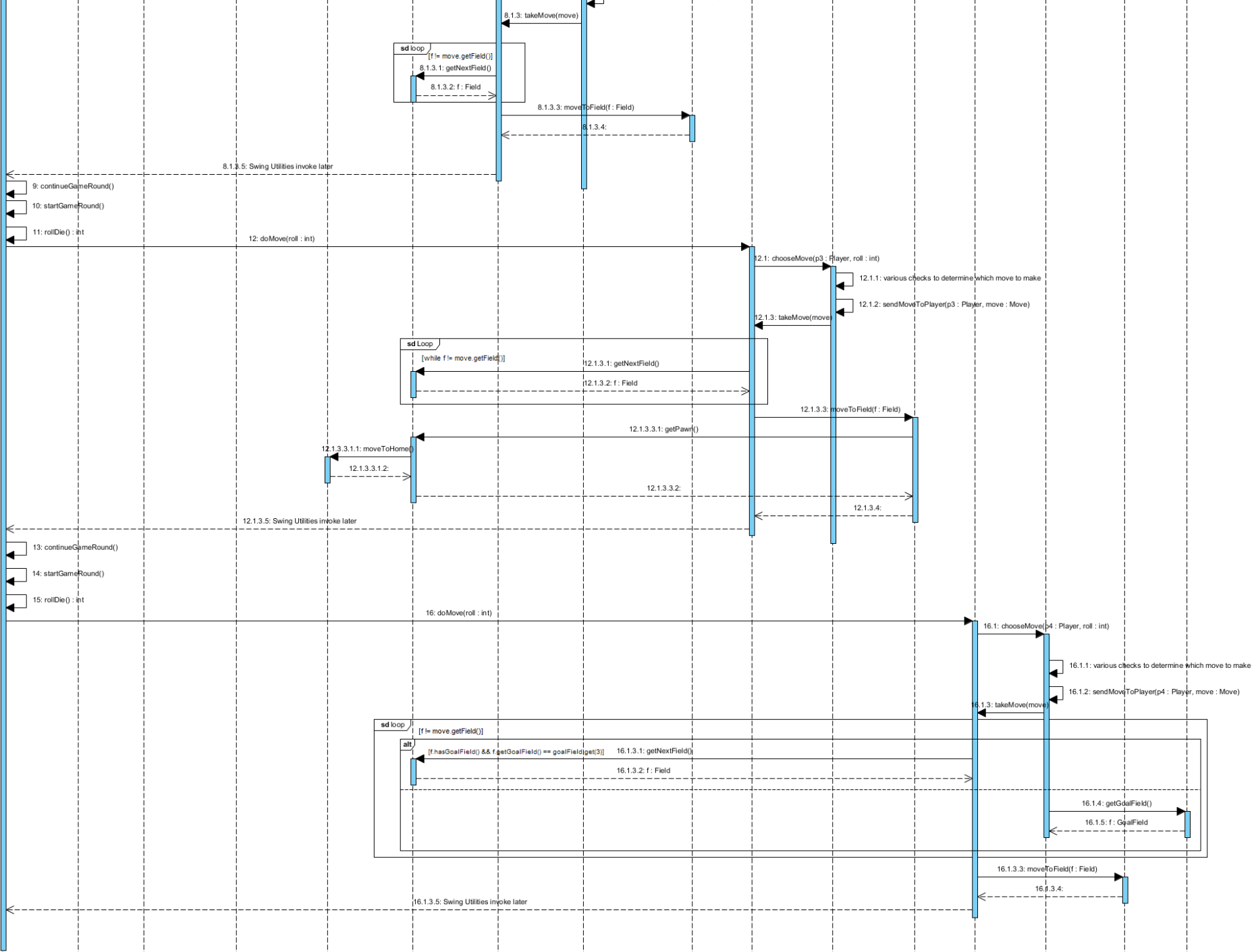
Class Diagram

Use Case Diagram

Nice and simple! Players really only choose their given move (if they have one)
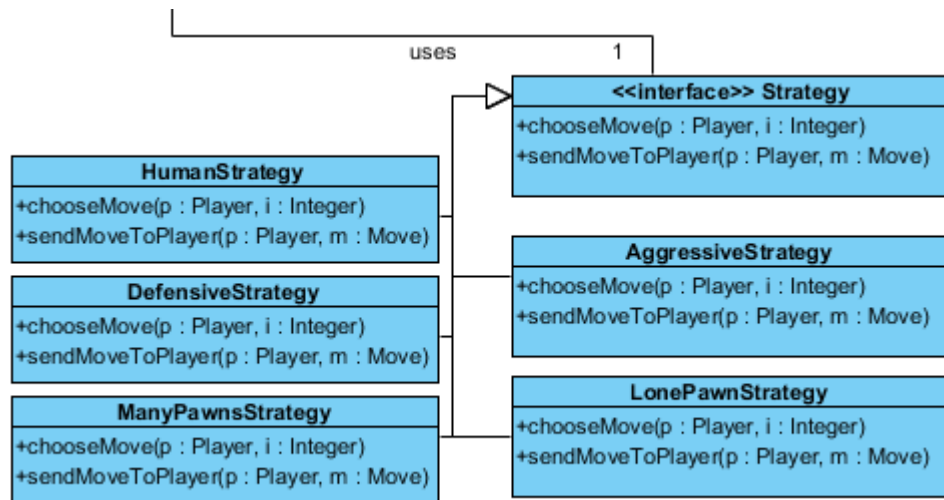
Sequence Diagram

8.1.3: takeMove(move)

sd loop
[f != move.getField()]
8.1.3.1: getNextField()
8.1.3.2: f : Field
8.1.3.3: moveToField(f : Field)
8.1.3.4:
8.1.3.5: Swing Utilities invoke later

9: continueGameRound()
10: startGameRound()
11: rollDie() : int
12: doMove(roll : int)

12.1: chooseMove(p3 : Player, roll : int)
12.1.1: various checks to determine which move to make
12.1.2: sendMoveToPlayer(p3 : Player, move : Move)
12.1.3: takeMove(move)

sd Loop
[while f != move.getField()]
12.1.3.1: getNextField()
12.1.3.2: f : Field
12.1.3.3: moveToField(f : Field)
12.1.3.3.1: getPawn()
12.1.3.3.1.1: moveToHome()
12.1.3.3.1.2:
12.1.3.3.2:
12.1.3.4:
12.1.3.5: Swing Utilities invoke later

13: continueGameRound()
14: startGameRound()
15: rollDie() : int
16: doMove(roll : int)

16.1: chooseMove(p4 : Player, roll : int)
16.1.1: various checks to determine which move to make
16.1.2: sendMoveToPlayer(p4 : Player, move : Move)
16.1.3: takeMove(move)

sd loop
[f != move.getField()]
alt
[f.hasGoalField() && f.getGoalField() == goalField.get(3)]
16.1.3.1: getNextField()
16.1.3.2: f : Field
16.1.4: getGoalField()
16.1.5: f : GoalField

16.1.3.3: moveToField(f : Field)
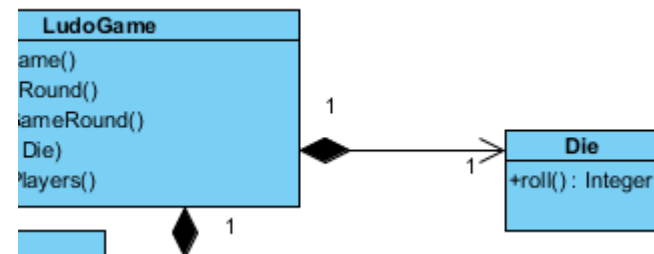16.1.3.4:
16.1.3.5: Swing Utilities invoke later

Design Patterns

Used

- Singleton (Die)
- Strategies (Player Strategy)

Strategy

Singleton

Design Patterns

Pros

- Kept code DRY (Don't repeat yourself)
- Kept code smarter.

Cons

- We should have found more to integrate.
- … Nothing else really!

What We Learned…

- UML didn't really help (except for identifying some design patterns we wouldn't have.)
- Swing (Threads, mostly) is annoying.
- Design Patterns are the best thing since scoping.
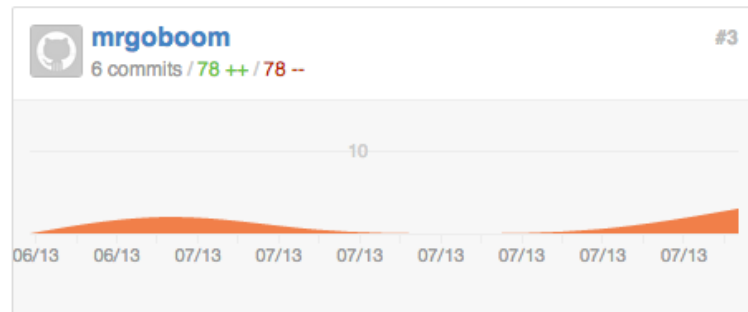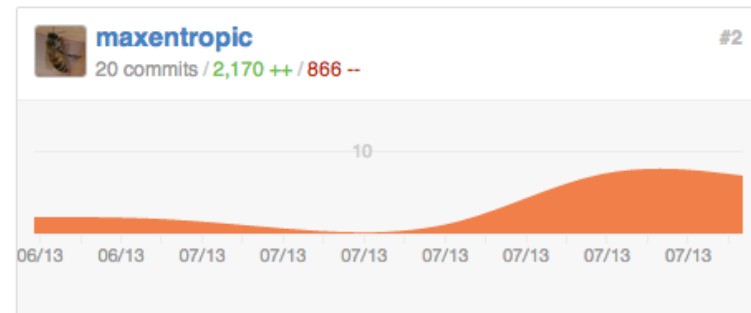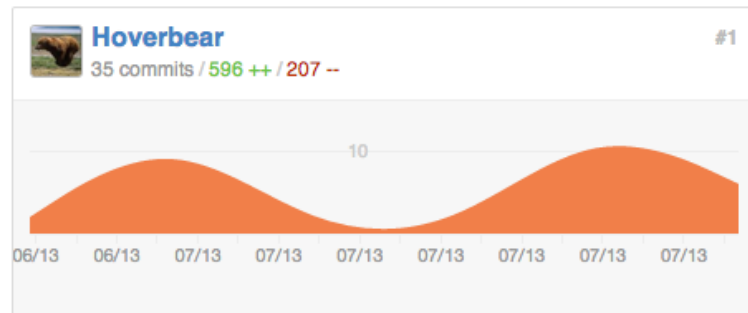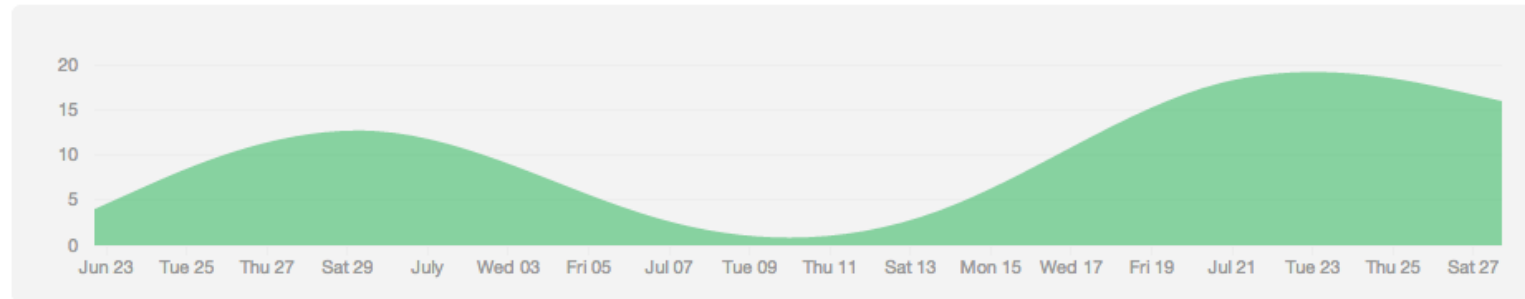- Github is **amazing** for collaborative coding.
- … And Analytics!

Code Frequency
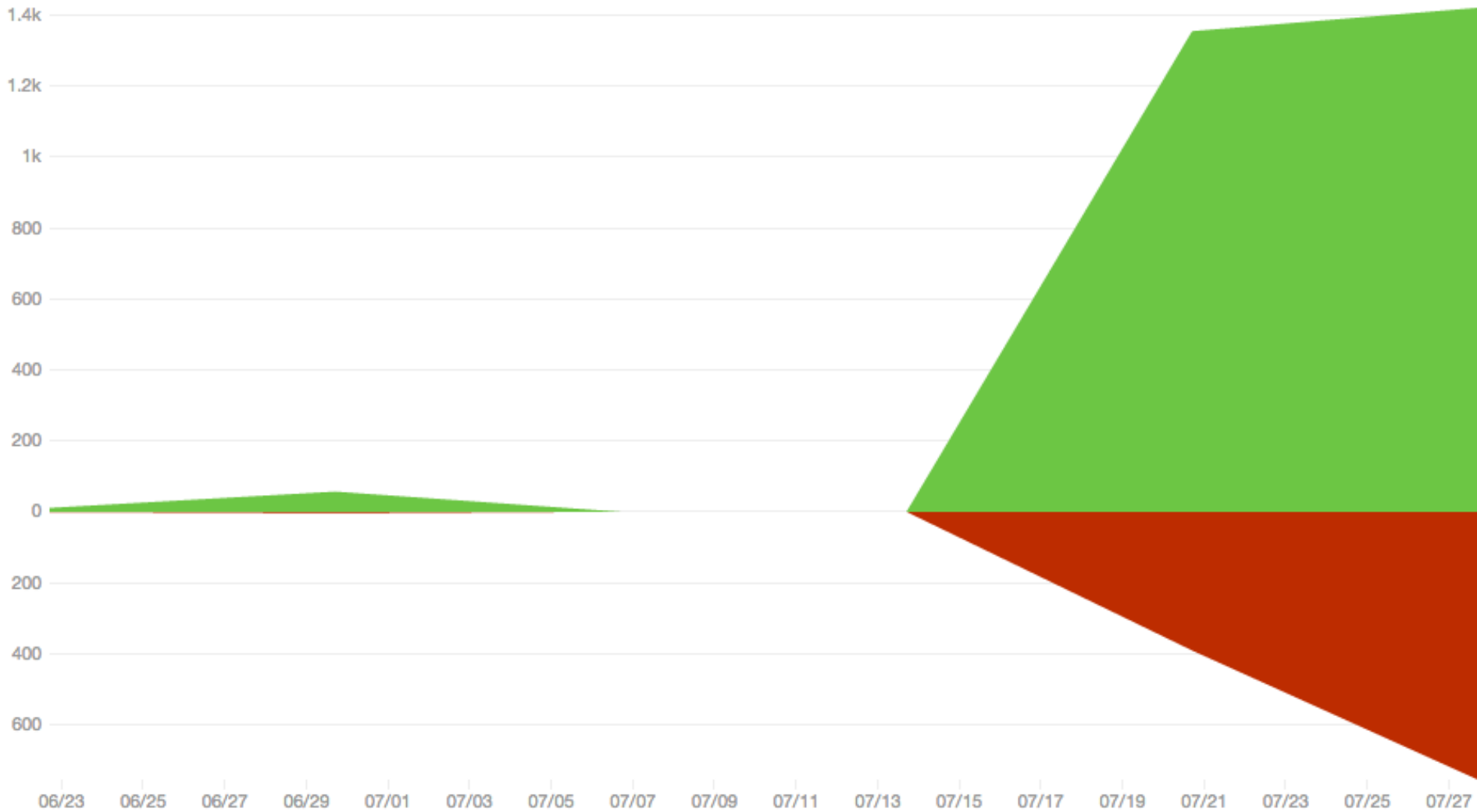
Activity Over Time

June 22nd 2013 - July 27th 2013
Commits to master, excluding merge commits

Hoverbear #1
35 commits / 596 ++ / 207 --

maxentropic #2
20 commits / 2,170 ++ / 866 --

mrgoboom #3
6 commits / 78 ++ / 78 --

When We Code
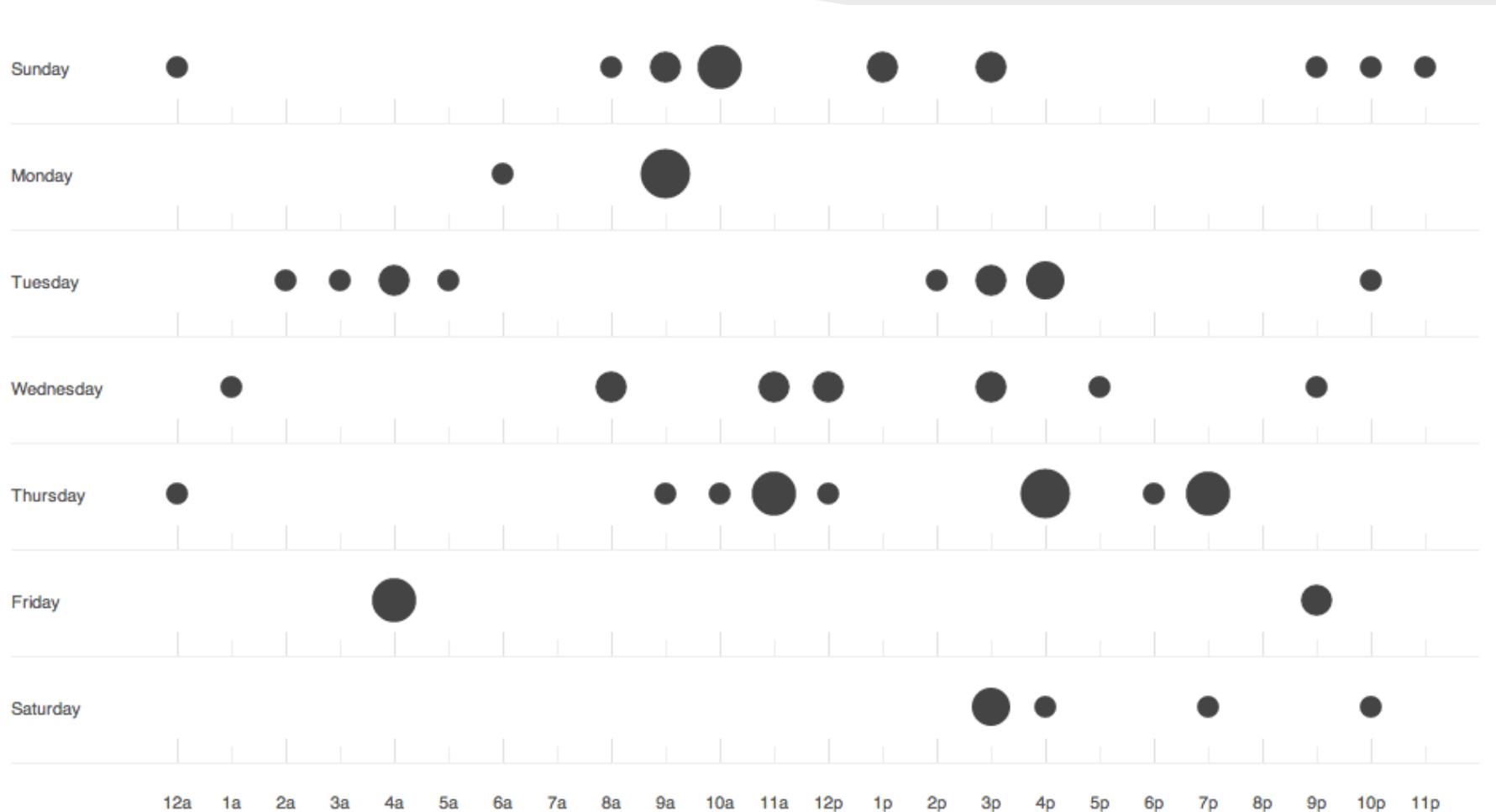
Who the heck codes on Saturday?

Late Nights

SHA: aee28abb474b8477d37554a361bd0286b07dab0f

Added the awesome strategy selector into branch
Authored by **Daniel Faulkner** on 2013-07-30 at 5:11 AM.
Also removed some extra code

⊞ Showing **1 changed file** with **42 additions** and **51 deletions**.

▼ Seng271_LudoGame/src/LudoGame.java   93 ■■■■☐

```
...   ...   @@ -13,6 +13,7 @@ import javax.swing.BorderFactory;
13    13    import javax.swing.BoxLayout;
```

Early Mornings

SHA: 2837058b91445d2d75c7e23e6ff7aeee6f1be651

Fixed file paths for #6.
Authored by **Hoverbear** on 2013-07-22 at 6:30 AM.
However this might introduce incompatabilities.

Showing **2 changed files** with **6 additions** and **6 deletions**.

▼ .gitignore   2

... | ... | @@ -1,5 +1,5 @@
1 | 1 | *.class