Team Swaggerz a.k.a "Based Gods to the Max"

Andrew Filipski a.k.a "Da Bawse"

Thomas Moore a.k.a "T-Dog RPG"

Jon Shippling a.k.a "Don't mess with him"

Akshay Karnawat a.k.a "Bomb Squad 2.0"

Agenda

- Reverse Engineering
- Metrics
- Patterns
- Refactoring
- Patterns Used
- Implementation
- Questions?

Reverse Engineering

- Global Design missing relationship lines
  - Level Builder -> Command Interpreter
  - Game -> Level Builder
- Some relationships unused
  - Game never interacts with Board
- high coupling low cohesion
- Limited and likely unintentional usage of patterns, deviations from standard pattern usage exist
- Use of metrics alerted us to problem areas in the code that the design did not convey by itself

Metrics

Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
Number of Overridden Methods (avg/max per	2	0.083	0.276	1	/SE362_Combat/combat/PlayerManager.java	
Number of Attributes (avg/max per type)	68	2.833	3.091	11	/SE362_Combat/combat/KeyDialog.java	
Number of Children (avg/max per type)	3	0.125	0.599	3	/SE362_Combat/combat/Sprite.java	
Number of Classes (avg/max per packageFragr	24	24	0	24	/SE362_Combat/combat	
Method Lines of Code (avg/max per method)	966	7.209	14.263	106	/SE362_Combat/combat/Bullet.java	conflict
Number of Methods (avg/max per type)	128	5.333	3.738	13	/SE362_Combat/combat/DirectionalImage.java	
Nested Block Depth (avg/max per method)		1.313	0.786	5	/SE362_Combat/combat/Board.java	pretick
Depth of Inheritance Tree (avg/max per type)		2.375	1.867	6	/SE362_Combat/combat/KeyDialog.java	
Number of Packages	1					
Afferent Coupling (avg/max per packageFragn		0	0	0	/SE362_Combat/combat	
Number of Interfaces (avg/max per packageFra	2	2	0	2	/SE362_Combat/combat	
McCabe Cyclomatic Complexity (avg/max per		2.134	3.032	25	/SE362_Combat/combat/Bullet.java	conflict
Total Lines of Code	1539					
Instability (avg/max per packageFragment)		1	0	1	/SE362_Combat/combat	
Number of Parameters (avg/max per method)		0.709	1.021	9	/SE362_Combat/combat/PlayerManager.java	PlayerManager
Lack of Cohesion of Methods (avg/max per typ		0.221	0.276	0.81	/SE362_Combat/combat/Game.java	
Efferent Coupling (avg/max per packageFragm		0	0	0	/SE362_Combat/combat	
Number of Static Methods (avg/max per type)	6	0.25	0.829	4	/SE362_Combat/combat/ImmutableList.java	
Normalized Distance (avg/max per packageFra		0.115	0	0.115	/SE362_Combat/combat	
Abstractness (avg/max per packageFragment)		0.115	0	0.115	/SE362_Combat/combat	
Specialization Index (avg/max per type)		0.022	0.074	0.286	/SE362_Combat/combat/PlayerManager.java	
Weighted methods per Class (avg/max per typ	286	11.917	11.456	43	/SE362_Combat/combat/Bullet.java	
Number of Static Attributes (avg/max per type	25	1.042	2.263	11	/SE362_Combat/combat/DirectionalImage.java	

Metric Analyze

- At the start of the project we ran the metrics and got an idea of what we were dealing with.
- Order of numbers (Max, Mean, SD)
- Bullet Conflict has a very high cyclomatic complexity (25,2.134,3.032)
  - Also, method lines of code is tremendous (106,7.209,14.263)
- Player Manager has too many parameters in the constructor (9,.709, 1.021)
- Player Manager overloads many things SpecIndex (.286, .022, .074)
- DirectionalImage has many static attributes (11, 1.042, 0.276)
- Game is not very cohesive (.81, .221, .276)

Metric Analyze (cont.)

- The method pretick() in Board has many nested blocks (5, 1.313, .786)
- After looking and reading through the code, we have decided that the nested blocks are used correctly and used for a good purpose.

Patterns (Them)

- Through the comments on the code and the code itself we found that the team has attempted to use some patterns
- Proxy (DirectionalImage) : Stand-in for Sprites until they are loaded, requests to change Sprite appearance passes through this class

- Iterator (Board, ImmutableList) : ImmutableList allows a client to iterate forward through its content, Board uses this to check Sprites for conflicts

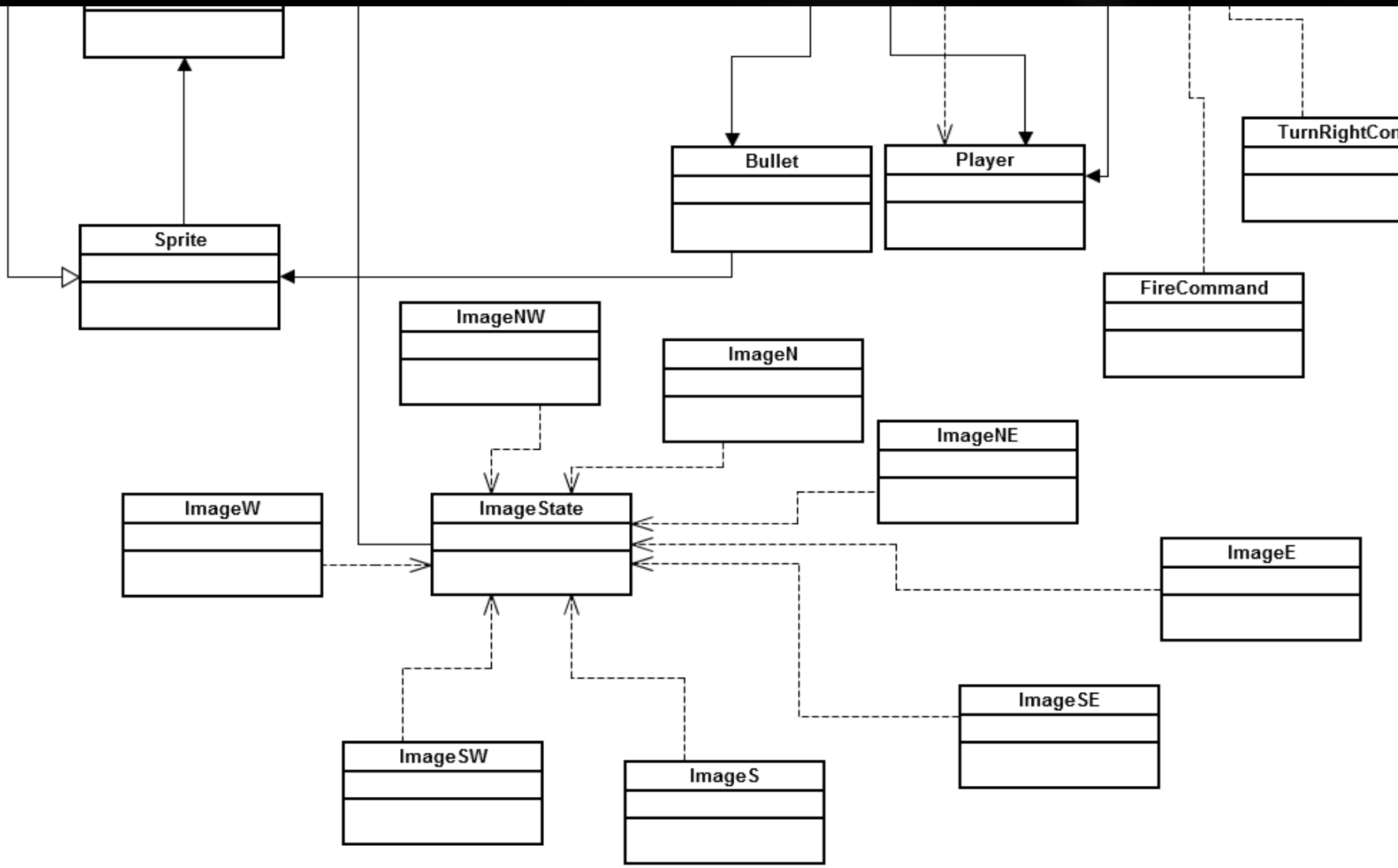
- Command (CommandInterpreter) : Stores commands to be used by the PlayerManager. No use of Command classes to encapsulate requests

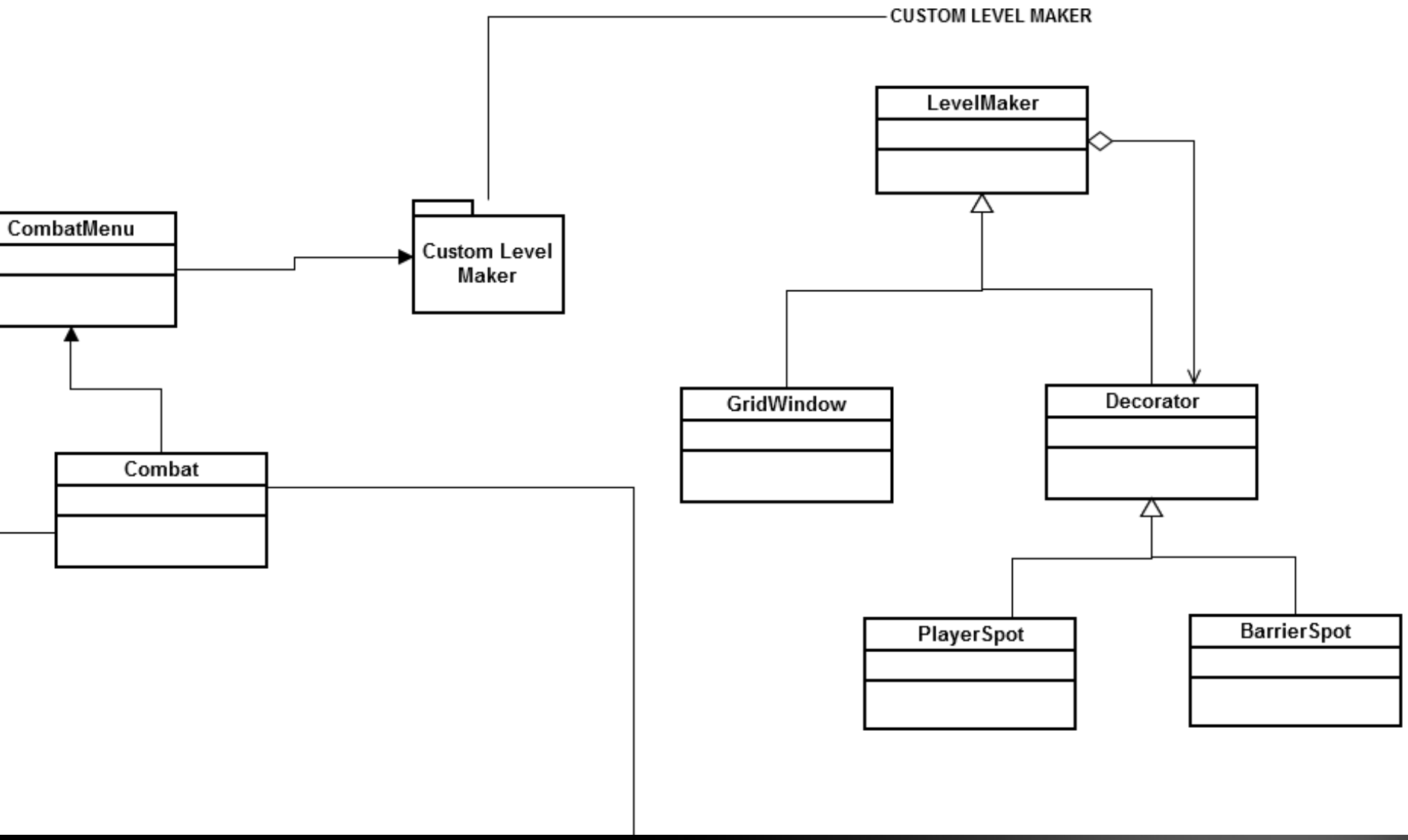
Refactoring

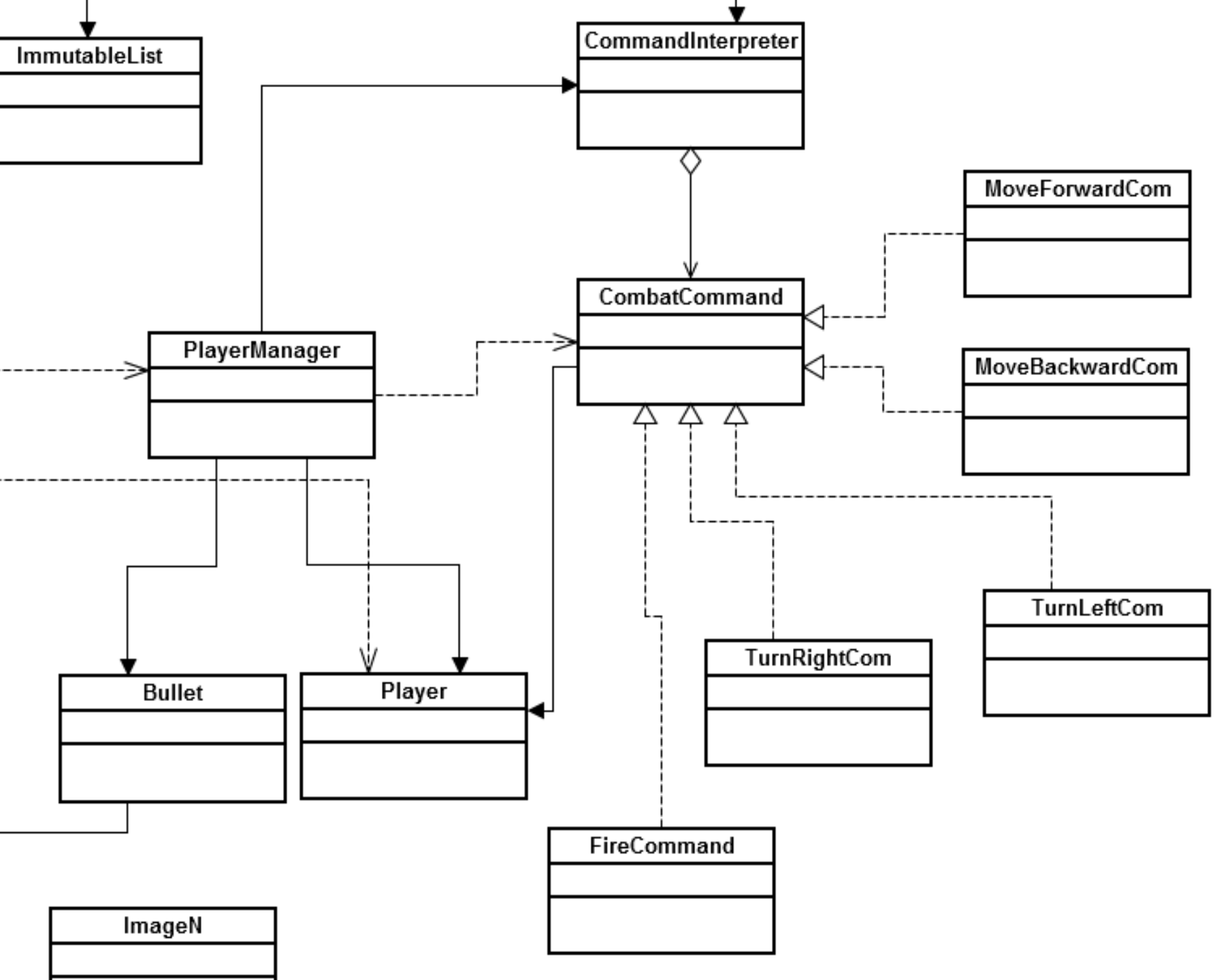
- Broke apart the Game class, separated some of its responsibilities into other classes
- Refactored the conflict() method in Bullet to reduce its length and Cyclomatic Complexity
  - With sufficient refactoring the MLOC could drop by over 50 and the MCC could drop by 16
- Reworked CommandInterpreter to fit better into a Command Pattern
- Deleted Thing since it is unused, Powerups were not a requirement
- PlayerManager constructor takes an array instead of 5 integers now, reduces number of parameters by 4

Patterns Used

- State
  - Direction and movement behavior of the tank
- Decorator
  - GUI level creator
- Command
  - Part of refactored design, use of command objects to issue commands to Players based on keys pressed


Bullet
Player
TurnRightCon
Sprite
FireCommand
ImageNW
ImageN
ImageNE
ImageW
ImageState
ImageE
ImageSE
ImageSW
ImageS


CUSTOM LEVEL MAKER
LevelMaker
CombatMenu
Custom Level Maker
GridWindow
Decorator
Combat
PlayerSpot
BarrierSpot


ImmutableList
CommandInterpreter
MoveForwardCom
CombatCommand
PlayerManager
MoveBackwardCom
TurnLeftCom
TurnRightCom
Bullet
Player
FireCommand
ImageN

Implementation

- As a group we decided to implement the third requirement
  - Allow the users to select the keyboard inputs they want to use during game play.

Questions?

ask now or forever hold your peace