



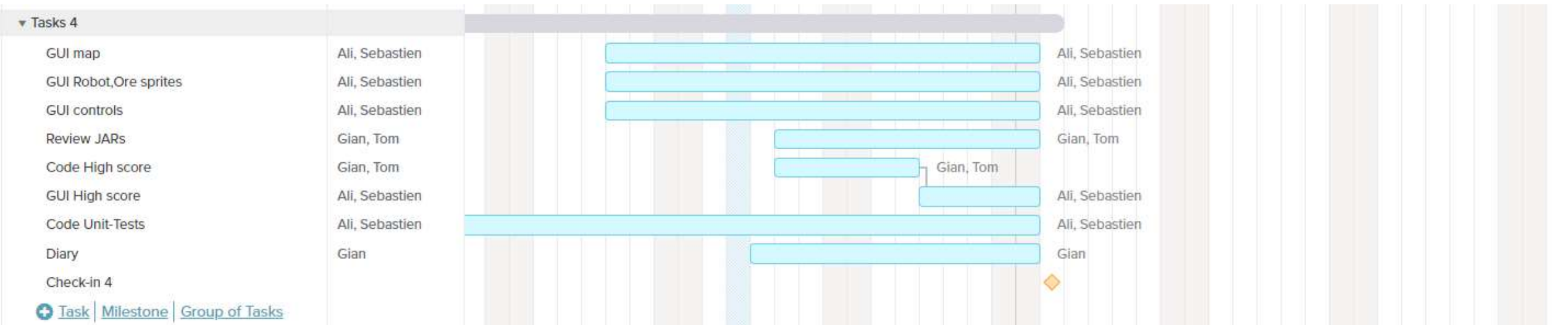
Ore Rush!

By TASG Force

Progress Report

Time Line:

- Game-Logic Basics
- GUI Basics
- Game Objects
- GUI Basics



Progress Report

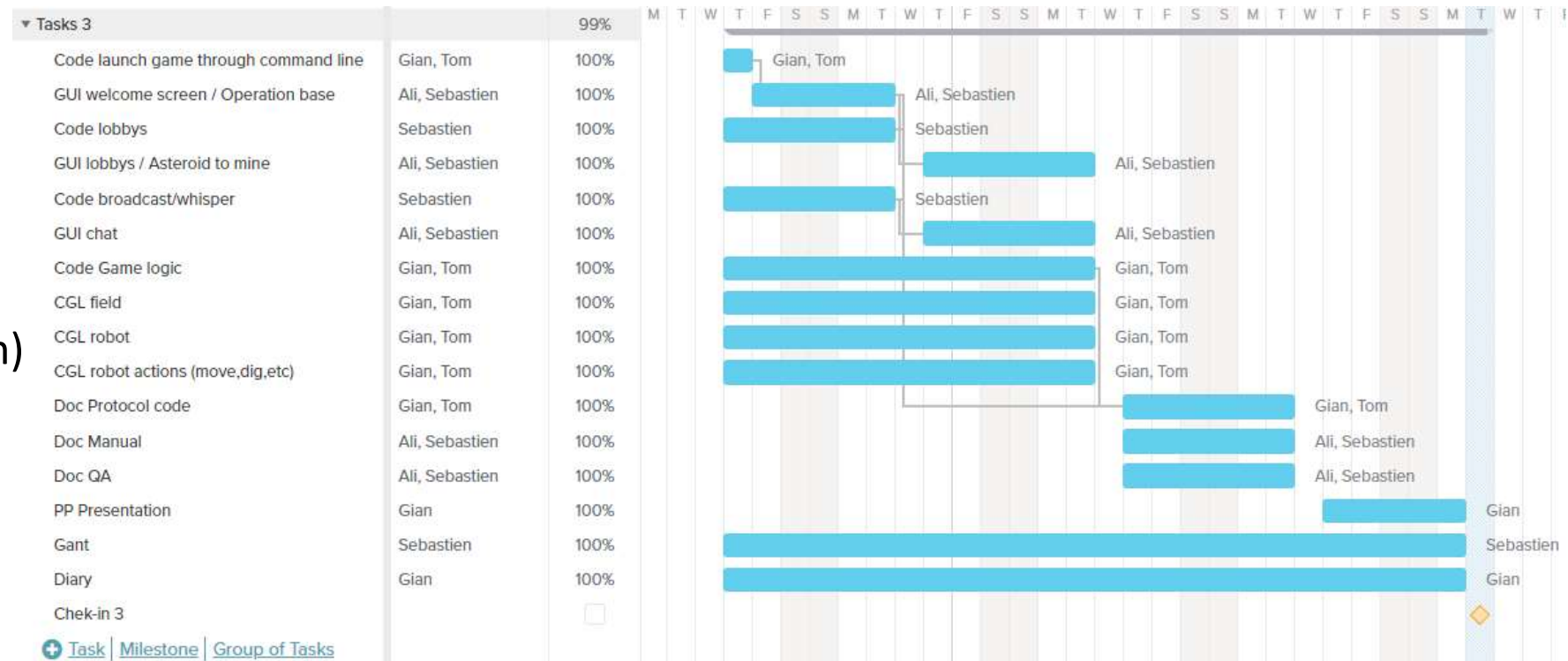
Time Line: Milestone 3

- Big Picture Goals:

- Game-Logic Basics
- GUI Basics
- Lobbys

- Responsibilities:

- Game Objects (Gian)
- Game Logic/Rules (Tom)
- GUI (Sebastien, Ali)
- Lobbys (Sebastien)
- QA (Ali)

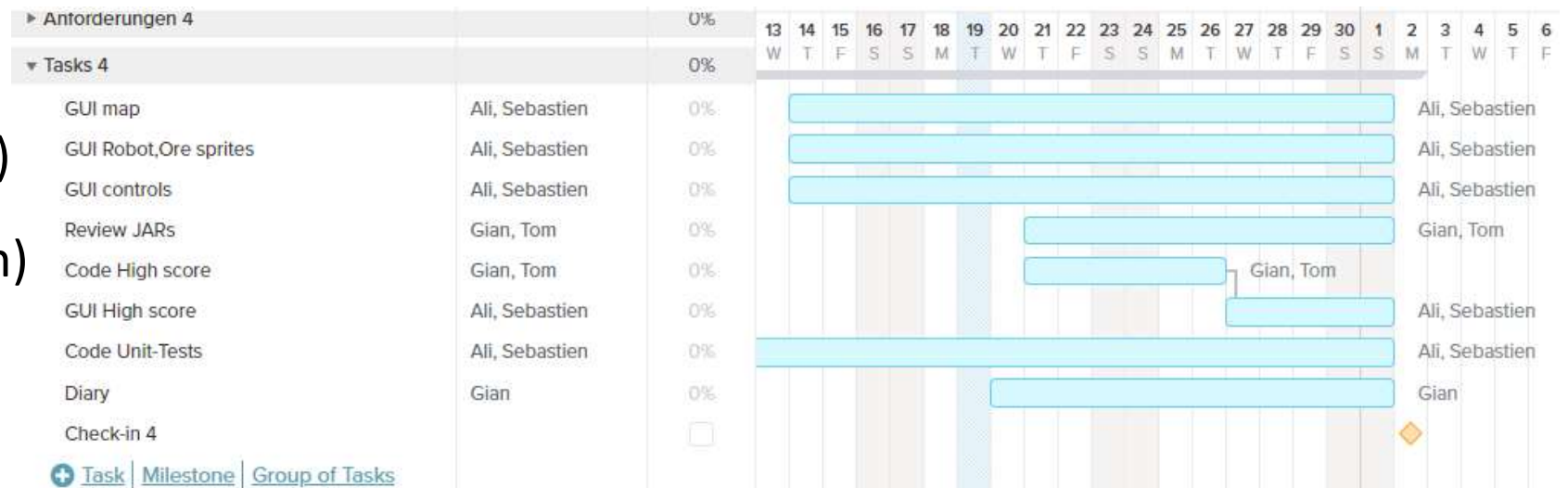


Progress Report

Time Line: Milestone 4

- Big Picture Goals:
 - Polish Game Mechanics
 - GUI
 - Art

- Responsibilities:
 - Highscore (Gian, Tom)
 - Polish Game Mechanics (Gian, Tom)
 - GUI (Sebastien, Ali)
 - Unit-Tests



Progress Report

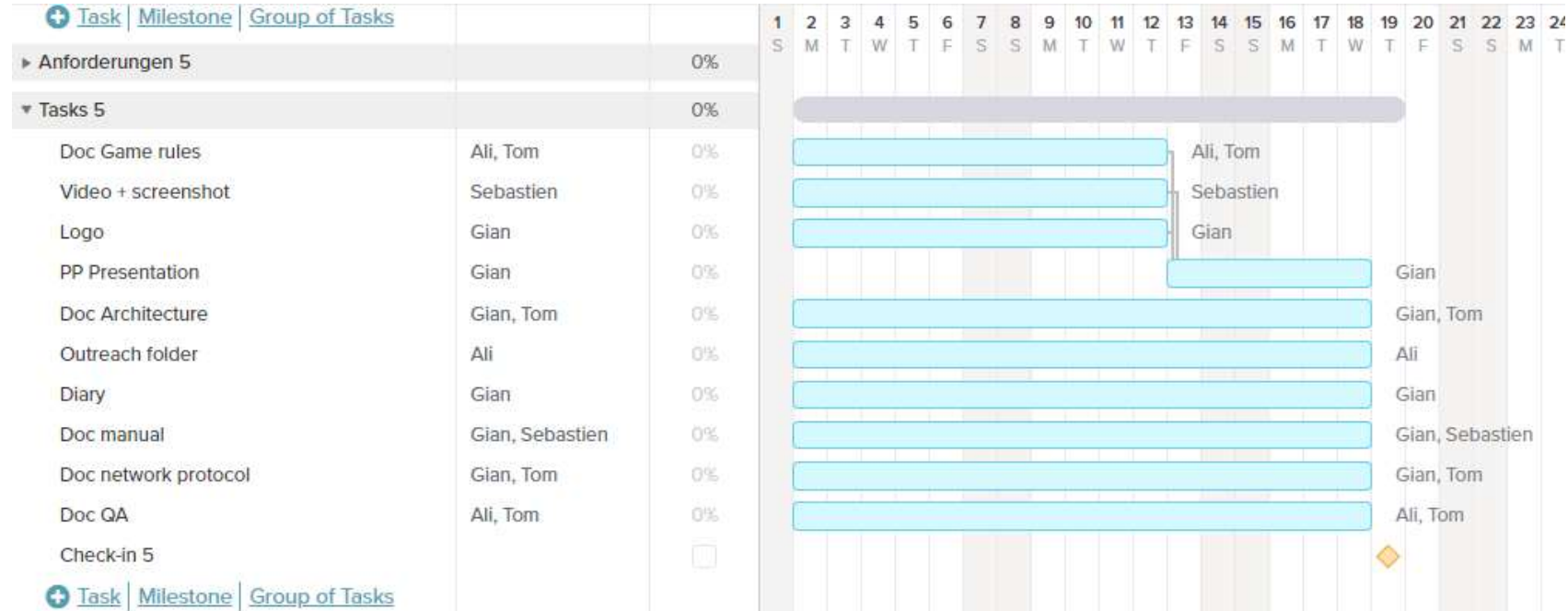
Time Line: Milestone 5

- Big Picture Goals:

- External Documents
- Game Presentation (Logo, Video, ect)

- Responsibilities:

- Logo (Gian)
- Video + Sceenshots (Sebastien)
- Game rules Doc (Sebastien, Ali)
- Architecture Doc (Tom)
- QA Doc (Ali)



Progress Report

Probleme

- Einige Systeme wurden komplett neu gestaltet.
 - ✓ Zuerst die Idee mit jemandem besprechen: Gibt es eine bessere Lösung?
 - ✓ Besseres Absprechen: Was wird verlangt?
- Probleme bei der Präsentation
 - ✓ Zusammen alles nochmals testen & vorbereiten.
 - ✓ Absprechen: Wer ist für was zuständig?

QA

Metrics

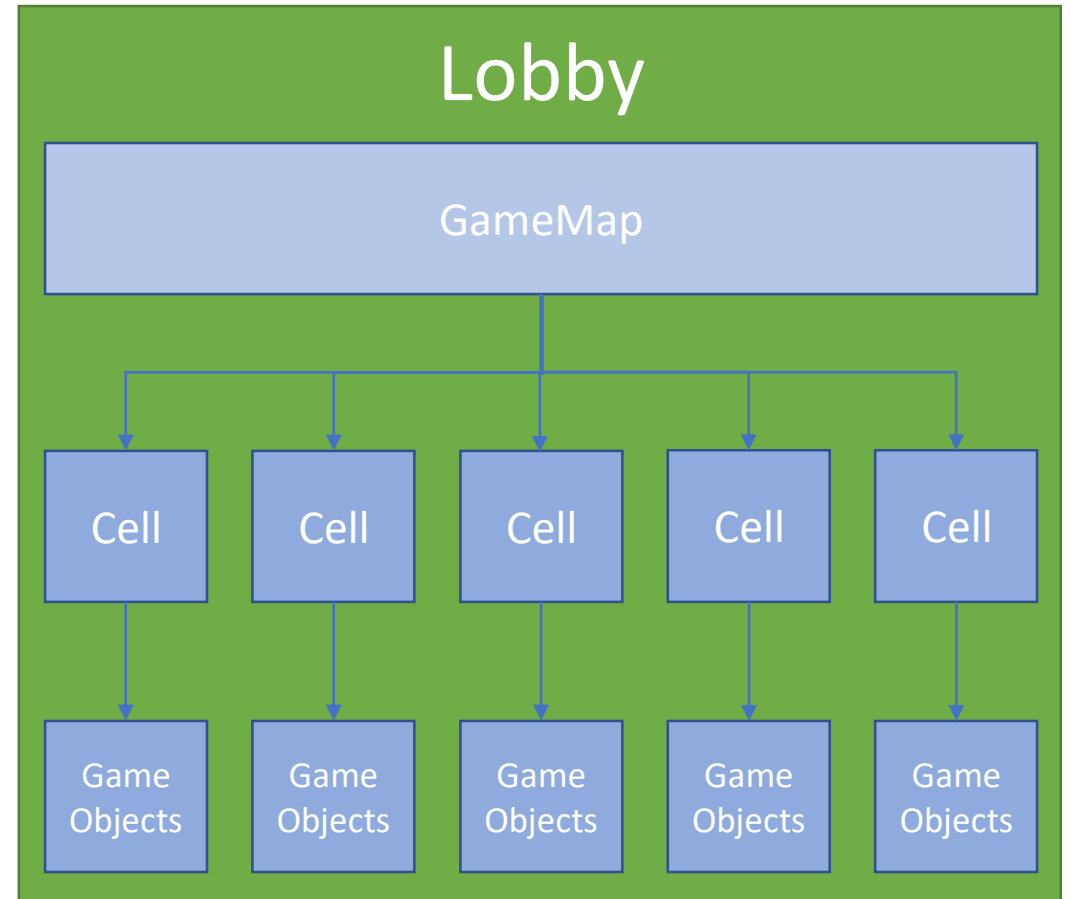
- Lines of Code
 - Mit MetricsReloaded bestimmen
 - Max 60 Character pro Zeile
 - Max 200 Zeilen pro Methode
 - Max 400 Zeilen pro Klasse
- Tight Class Cohesion
 - Bestimmt die Komplexität einer Klasse
 - $\text{Min} < \text{Average} < \text{Max}$
- Javadoc lines per method
 - Beschreibt wie lange eine JavaDoc sein soll
 - 0: Keine JavaDoc
 - <30 : Nicht genügend für komplexe Methoden
 - >70 : Zuviel für einfachere Methoden

Method metrics	Class metrics	Interface metrics	Package metrics	Module metrics
class	CLOC	JLOC	LOC	
game.datastructures.GameMap	58	49	346	
game.server.ClientThread	56	47	248	
game.client.Client	17	9	172	
game.datastructures.Robot	34	31	127	
game.gui.LobbyController	14	10	112	
game.packet.AbstractPacket	33	30	110	
game.server.Lobby	15	14	106	
game.gui.StartMenuController	3	0	102	
game.server.ServerSettings	15	10	96	
game.packet.packets.Move	4	4	94	
game.packet.PacketHandler	12	12	79	
game.datastructures.Cell	0	0	69	
game.client.InputStreamThread	8	3	66	
game.packet.packets.Connect	16	10	66	
game.packet.packets.ChatLobby	14	12	64	
game.Test	6	0	63	
game.datastructures.Radar	24	24	63	
game.datastructures.Trap	24	24	63	
game.packet.packets.Update	26	11	63	
game.client.LobbyInClient	4	4	62	
game.packet.packets.Nickname	13	10	60	
game.server.Server	3	0	60	
game.packet.packets.Broadcast	12	12	58	
game.packet.packets.Whisper	5	3	58	
game.packet.packets.LeaveLobby	2	0	57	
game.packet.packets.Chat	12	12	56	
game.packet.packets.JoinLobby	2	0	50	
game.server.PingThread	9	5	50	
game.Main	0	0	44	
game.datastructures.Ore	6	6	43	
game.packet.packets.Awake	13	13	43	
game.packet.packets.Close	11	11	42	

Rules to Code

Gamestate

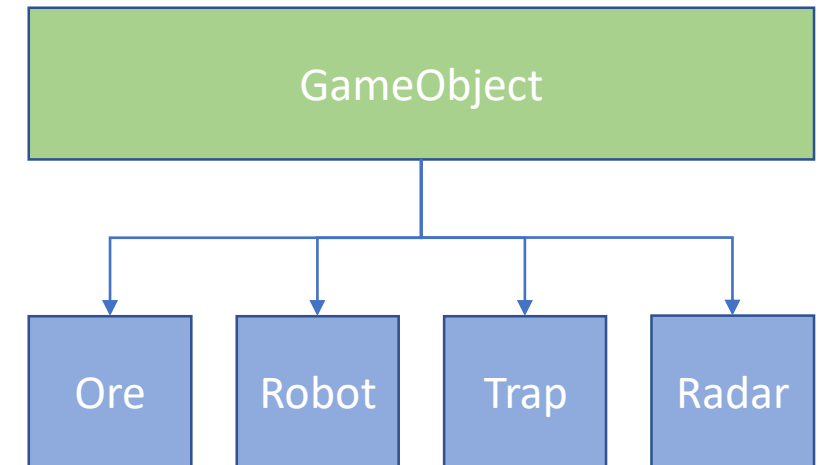
- Das Spielfeld wird durch eine Instanz der «GameMap» Klasse dargestellt.
- Jedes Feld des Spielfeld wird durch eine Instanz der «Cell» Klasse dargestellt
- Die Objekte im Spiel werden durch «GameObjects» dargestellt



Rules to Code

Game Objects

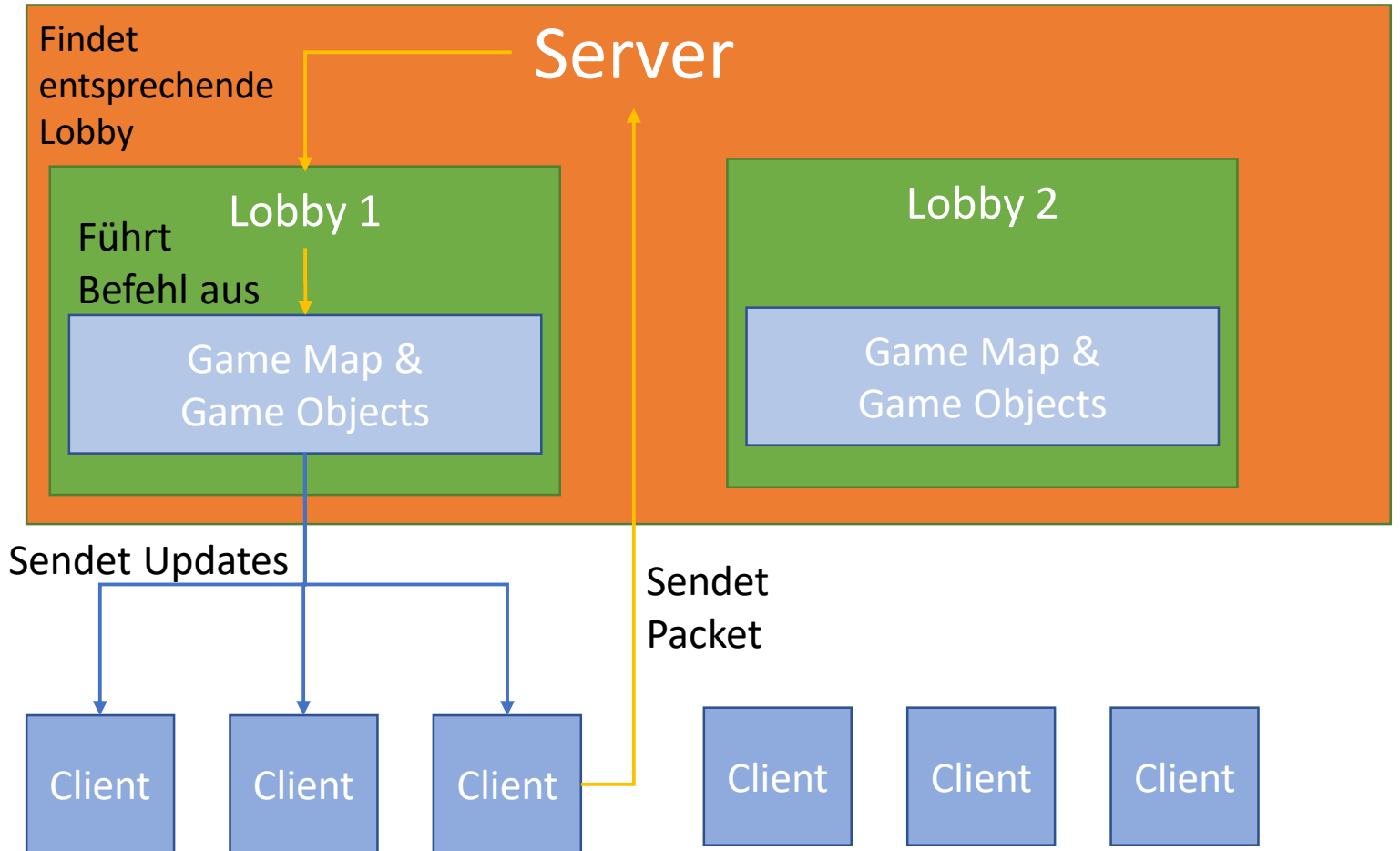
- Spiel Objekte erweitern das «GameObject» Interface
- Allgemeine Methoden im interface
- Spezifische Methoden in den Klassen.



Rules to Code

Game Rules

- Befehle werden per Packets vom Client zum Server gesendet.
- Packets werden der entsprechenden Lobby zugeteilt.
- Packets werden im Server decodiert und befehle ausgeführt.
- Server sendet Update an Clients



Technology

- Tools:
 - MetricsReloaded von IntelliJ -> Berechnet einige Metrics
 - Discord -> Kommunikation, Protokoll
 - Code with me -> Pair programming
- Processes:
 - Abstrakte Klassen -> Modulares System
 - Model view client Architektur -> GUI konstruieren.

Demo

End

Vielen Dank für ihre Aufmerksamkeit.

Fragen?

Questions?