



Fakultät für Ingenieurwissenschaften, Informatik und Psychologie Institut für Softwaretechnik und Programmiersprachen

Softwaregrundprojekt Meilenstein 6

Softwaregrundprojekt an der Universität Ulm

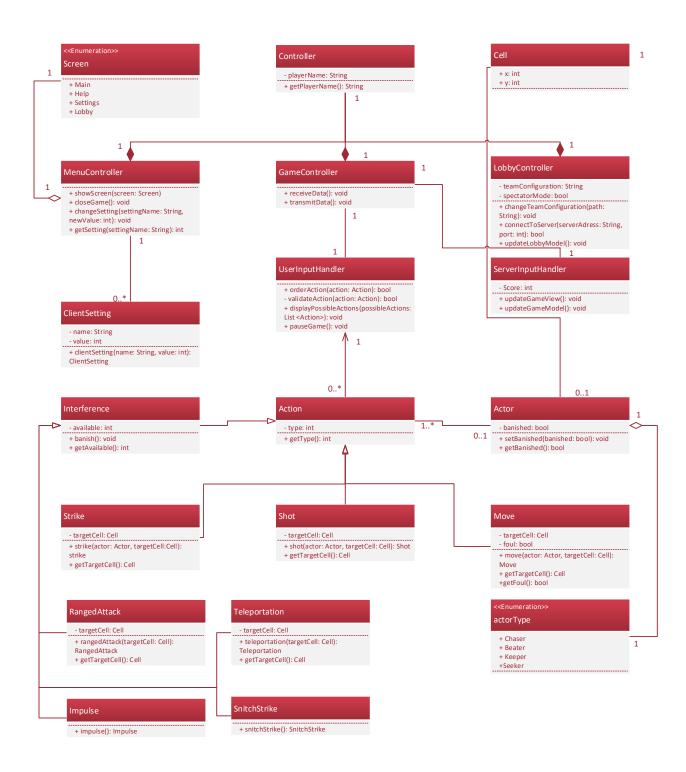
Vorgelegt von:			
Gruppe 10			
_			
Dozent:			
Florian Ege			

Betreuer: Stefanos Mytilineos

2019

Inhaltsverzeichnis

1 Controller



Im Folgenden werden die eingetragenen Methoden erklärt.

Name	Vorbedingungen	Nachbedingungen	Erklärung	
MenuController				
closeGame	-	Positive Antwort in einem Popup-Fenster	Beendet die Anwen- dung	
changeSetting	ClientSetting mit dem angegebe- nen settingName existiert	Der Wert von ne- wValue wird akzep- tiert	Ändert das Attribut value des ClientSetting mit dem Attribut name, der dem Parameter settingName entspricht, auf den Wert des Parameters newValue.	
getSetting	ClientSetting mit dem angegebe- nen settingName existiert	-	Liefert den derzeitigen Wert des Attributs value des ClientSetting, des- sen Attribut name mit dem Parameter settin- gName übereinstimmt.	
GameController				
receiveData	-	-	Empfängt Daten vom JSON-Parser und über- gibt sie dem ServerIn- putHandler	
transmitData	-	-	Empfängt Daten vom UserInputHandler und sendet sie über den JSON-Parser und den Kommunikator zum Server	
LobbyController				
changeTeamConfiguration	-	Parameter path ist ein gültiger Pfad zu einer Team- Konfigurations- Datei	Ändert den Wert von teamConfiguration, in dem der Pfad zu der zu verwendenden Team-Konfigurations-Datei gespeichert ist.	

connectToServer	Parameter server- Adress und port sind ungleich Null	-	Veranlasst über den JSON-Parser den Kommunikator dazu, eine Verbindung mit dem in den Parametern spezifiziertem Server aufzubauen. Gibt true zurück, wenn der Verbindungsaufbau erfolgreich war, ansonsten false.	
UserInputHandler				
validateAction	-	-	Fordert vom Partie- Model eine Liste aller gültigen Actions an und gibt true zurück wenn die als Parameter über- gebene Action darin enthalten ist, ansonsten false.	
orderAction	validateAction gibt für die als Parame- ter übergebene Ac- tion true zurück.	-	Übergibt die auszuführende Aktion an den GameController. Ruft die Methode validateAction auf und gibt deren Rückgabewert zurück.	
displayPossibleActions	-	-	Fordert vom Partie- Model eine Liste aller möglichen Aktionen an und gibt sie an die Partie-View weiter, um sie dem Spieler anzuzeigen.	
Interference				
banish	-	-	Verringert das Attribut available um eins.	

Die Funktionalen Anforderungen werden den Methoden folgendermaßen zugeteilt:

Funktionale Anforderungen	Methoden		
FA7	Shot::shot		
	Shot::getTargetCell		
FA21	Shot::shot		
FA26	Move::move		
FA27	Shot::shot		
FA28	Strike::strike		
FA30	Move::move		
FA31	RangedAttack::rangedAttack		
	Teleportation::teleportation		
	Impulse::impulse		
	SnitchStrike::snitchStrike		
	Inteference::banish		
FA32	Teleportation::teleportation		
FA33	RangedAttack::rangedAttack		
FA34	Impulse::impulse		
FA35	SnitchStrike::snitchStrike		
FA36	Inteference::banish		
	Actor::setBanished		
FA37	Actor::setBanished		
	Move::move		
FA38 - FA42	Move::move		
FA48	Actor::setBanished		
FA54	LobbyController::changeTeamConfiguratio		
	LobbyController::updateLobbyModel		
FA55	GameController::receiveData		
	GameController::transmitData		
	LobbyController::connectToServer		
FA56	GameController::receiveData		
	GameController::transmitData		
FA60-FA61	MenuController::showScreen		
FA67	UserInputHandler::orderActions		
	UserInputHandler::validateActions		
	MenuController::showScreen		
	MenuController::changeSetting		
	LobbyController::changeTeamConfiguration		
	LobbyController::connectToServer		
FA69	UserInputHandler::pauseGame		