Final Version c-----LeaderboardUI LeaderboardData - RESOURCE_FILE_URL : URL - <u>frameHeight</u>: int - EXTERNAL_SAVE_PATH : Path - <u>frameWidth</u> : int class to handle - LEADERBOARD : Map<Integer, Integer> - <u>optionalSpacing</u> : int reading and - NUMBER_OF_PLAYERS : int Classes are clearly distinguishable in the diagram, where each extension is + <u>setFrameHeight(newFrameHeight:int)</u>: void class to handle UI writing into the + LeaderboardData(resourceFileUrl : URL, numberOfPlayers : int) : LeaderboardData + <u>setFrameWidth(newFraneWidth:int)</u>: void assigned with a distcint color. With newly added classes background color related component - initializeLeaderboard(): void + <u>setOptionalSpacing(newOptionalSpacing : int)</u> : void Stats.txt file and modified methods/attributes as well as anyt new relation are indicated by - loadLeaderboard:() void + <u>createLeaderboardFrame()</u> : JFrame for the the same distinct color. - saveLeaderboardToFile(): void + <u>createLeaderboardPanel(table : JTable)</u> : JPanel + getLeaderboard(): Map<Integer, Integer> + <u>createStartGameButton(frame : JFrame, onStartGame : Runnable)</u> : JButton leaderboard + getExternalSavePath(): Path + getLeaderboardFrame(Runnable onStartGame, DefaultTableModel leaderboardTableModel) : JFrame Required Extension (Zeus): In Blue + showLeaderBoard(leaderboardTableModel : DefaultTableModel) : JFrame + updateScore() : void Pick From List Extension (Dynamic board and mutliplayer): In Red + setInitialDataTable(leaderboardTableModel : DefaultTableModel) : void + <u>getLeaderboardButton(leaderboardTableModel : DefaultTableModel)</u>: JButton Self defined Extension (Leaderboard): In Green Santorini GameController - WORLD: World ADERBOARD_TABLE_MODEL : DefaultTableModel - currentPlayer: Player ADERBOARD_DATA : LeaderboardData - selectedWorker: Worker main(String[] args) :void GameManager(numberOfPlayers : int, fileName : URL) - - - -- GOD_CARD_LABEL: JLabel etLeaderboardTableModel() : DefaultTableModel - STATUS_LABEL: JLabel pdateScore(playerId : int, scoreToAdd : int) : void - GOD_ABILITY_DISABLE_BUTTON: JButton nitializeOrLoad(): void <<enumeration>> startGame (onStartGame : Runnable) : void Timing + GameController(World world, List<Player> playerList, int numOfWorkers, List<GodCard> godCards, JFrame frame): GameController + nextPlayer(): void AFTER_MOVE, Added new Timing - updateGodCardDisplay(): void AFTER_BUILD, BEFORE_BUILD - attachSelectWorkerListeners(): void BEFORE_BUILD MultiplayerUtils - onWorkerSelected(Location loc): void Refactored # attachMoveListeners(Location from): void attachBuildListener to - onMove(Location from, Location to): void MultiplayerUtils(): void handle before Build timing + attachBuildListeners(Location from): void + askPlayerNumber(minPlayer: int, maxPlayer: int): int logic - onBuild(Location from, Location at): void Artemis - clearAllListeners(): void + updateStatus(String text): void <<abstract>> + checkLosingPlayer(): void GodCard + Artemis(imagePath: *URL*): Artemis - activateTiming: Timing - godName: String - imagePath: URL - PLAYERS: List<Player> Demeter - BOARD: Board + GodCard(timing: Timing, godName: String, imagePath: *URL*): GodCard - AVAILABLE_GOD_CARD: Map<Integer, GodCard> + getActivateTiming(): Timing GameComponentUI - godCardId: int + getName(): String currentPlayerIndex: int + Demeter(imagePath: *URL*): Demeter + getImagePath(): URL · <u>locationButtonWidth</u>: int - <u>locationButtonHeight</u>: int + performAbility(from: Location, to: Location, selectedWorker: Worker, godDisableAbilityButton: + World(boardWidth: int, boardHeight: int): World JButton, board: Board, gameController: GameController): boolean + addGodCards(godCard: GodCard): void Refactored to + activateAbility(from: Location, to: Location, board: Board, gameController: GameController): + <u>createLocationButton(</u>): JButton + getBoard(): Board handle - — — + <u>createBoardPanel(</u>width: int, height: int): JPanel + getPlayerAmount(): int multiplayer New God Card Zeu + <u>createGameFrame(windowWidth: int, windowHeight: int):</u> JFrame + deactivateAbility(from: Location, to: Location, board: Board, gameController: GameController): + getPlayers(): List<Player> creation logic With Build below itself + <u>setLocationButtonWidth(newLocationButtonWidth: int): void</u> + randomlyAssignGodCards(): void logic in performAbility & + getLocationButtonWidth(): int Zeus(imagePath: *URL*): Zeus randomlyAssignWorker(numOfWorkers: int): void activateAbility function + <u>setLocationButtonHeight(newLocationButtonHeight: int)</u>: void + initializeGame(players: List<Player>, numOfWorkers: int, godCards: List<GodCard>): void + getLocationButtonHeight(): int + processMove(from: Location, to: Location): void + processBuild(from: Location, to: Location): void + nextPlayer(): Player + removePlayer(player: Player): void + isGameOver(): boolean Player c - - - + - - - - -<<enumeration>> - ID: int Capability - ICON_PATH: URL - workers: List<Worker> MOVE_TO_FLOOR, - godCard: GodCard MOVE_TO_TOWER_1, MOVE_TO_TOWER_2, ------ \longrightarrow MOVE_TO_TOWER_3, + Player(int id, *URL* iconPath): Player Board + getID(): int - MIN_SIZE: int + getICON_PATH(): URL Added Config - BOARD_PANEL: JPanel + getGodCard(): GodCard To grant worker BUILD_BELOW_WORKER_ON_TOWER file to board to - SCROLL_PANE: JScrollPane ability to build + setGodCard(GodCard godCard): void - BOARD_LOCATIONS: Map<Position, Location> + addWorker(worker: Worker): void - WORKER_LOCATION_CONTROLLER: WorkerLocationController + getWorkers(): List<Worker> Board(configFilePath : String, numberOfPlayers: int): Board belongs to getBoardPanel(): JPanel WorkerLocationController getScrollPane(): JScrollPane CapabilityController + addExitFrom(from: Location, position: Position): void Worker - WORKER_LOCATION_MAP: Map<Worker, Location> createLocation(position: Position): void Floor - LOCATION_WORKER_MAP: Map<Location, Worker> - capabilities: Set<Enum<?>> - OWNER: Player at(position: Position): Location - ICON: ImageIcon getWorkerAt(location: Location): Worker + WorkerLocationController(): WorkerLocationController + CapabilityController(): CapabilityController - CAPABILITIES: CapabilityController moveWorker(worker: Worker, location: Location): void + addWorker(worker: Worker, location: Location): void + addCapability(capability: Enum<?>): void keep track of— + canWorkerTravel(Worker : worker): boolean # addWorker(worker: Worker, location: Location): void + removeWorker(worker: Worker): void + addCapability(capabilities: List<Enum<?>>): void + buildUp(Location : location): void + Worker(player: Player): Worker removeWorker(Worker: worker): void + moveWorker(worker: Worker, location: Location): void + removeCapability(capibility: Enum<?>): void + toString(): String + getIcon(): ImageIcon + isWorkerAt(location: Location): boolean + isWorkerAt(location: Location): boolean + removeCapability(capibility: List<Enum<?>>): void + getPlayer(): Player locationOfWorker(worker: Worker): Location + isWorkerExists(worker: Worker): boolean + hasCapability(capability: Enum<?>): boolean + getCapabilites(): CapabilityController refreshBoardUI(): void + getWorkerAt(location: Location): Worker + getCapabilities(): Set<Enum<?>> Tower1 + locationOfWorker(worker: Worker): Location Ground # capabilityHolds : List<Enum<?>> + canWorkerTravel(Worker : worker): boolean Location # lostCapability : List<Enum<?>> + buildUp(Location : location): void # hasDome: boolean - LOCATION_BUTTON: JButton + toString(): String # level : int MoveAction - BOARD: Board - POSITION: Position toLocation: Location + getCapabilities(): List<Enum<?>> - ground: Ground o..* - exits: List<Location> + getLostCapabilities: List<Enum<?>> Tower2 + MoveAction(toLocation: Location): MoveAction + buildUp(Location location): void + canWorkerTravel(soruceLocation: Location): boolean + execute(worker: Worker, board: Board, from + Location(board: Board, position: Position, ground: Ground): Location + canWorkerBuild(soruceLocation: Location): boolean + Tower2() + hasWorker(): boolean + addCap(Worker: worker): void + canWorkerTravel(Worker : worker): <<abstract>> + removeCap(Worker: worker): void + addWorker(worker: worker): void DoNothingAction Action + setGround(ground: Ground): void + buildUp(Location : location): void + getLevel() : int + getGround(): Ground modified Position + toString() : String + canWorkerTravel(worker: Worker): boolean canWorkerBuild to + execute(worker: Worker, board: Board, from + canWorkerBuild(worker: Worker): boolean Added getter + X: int handle the Zeus god execute(worker: Worker, board: Board, from Location): void + Y: int for getting + addExit(exit: Location): void ocation): void card logic Added action location Ground dome + getExits(): List<Location> Tower3 from to execute method + Position(x: int, y: int): Position BuildAction boolean + getBoard(): Board paramter to handle + equals(object: Object): boolean + getLocationButton(): JButton - toLocation: Location worker only wins when + Tower3() + hashCode(): int + getWorker(): Worker level attribute to move up to Tower 3 canorkerTravel(Worker: worker): boolean + hashCode(): int keep track of tower + BuildAction(toLocation: Location): BuildAction + buildUp(Location : location): void height (logic for only wins when xecute(worker: Worker, board: Board, from + toString() : String moved up)