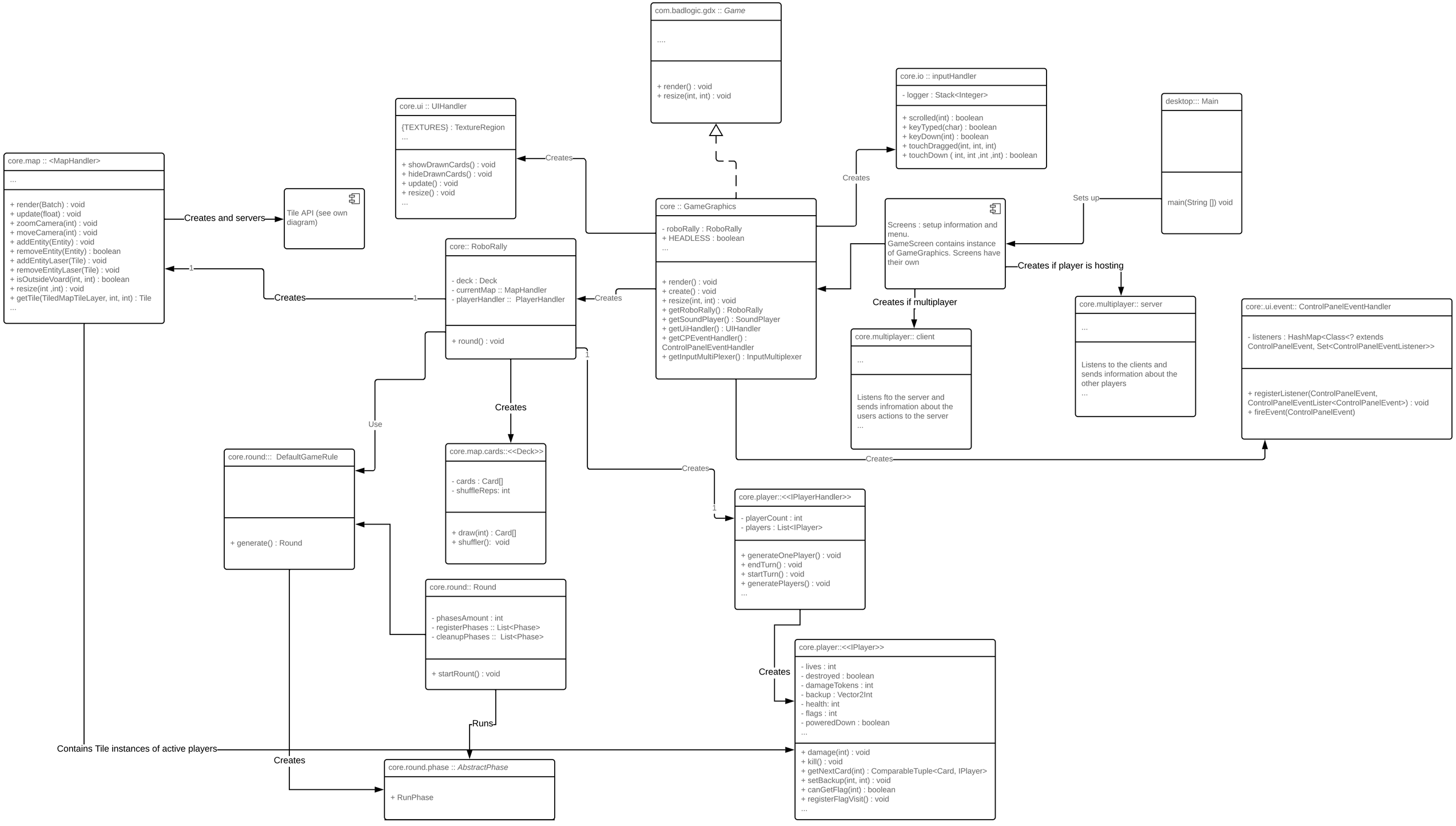
com.badlogic.gdx :: Game
....
+ render() : void
+ resize(int, int) : void
core.io :: inputHandler
- logger : Stack<Integer>
+ scrolled(int) : boolean
+ keyTyped(char) : boolean
+ keyDown(int) : boolean
+ touchDragged(int, int, int)
+ touchDown (int, int ,int ,int) : boolean
desktop::: Main
main(String []) void
core.ui :: UIHandler
{TEXTURES} : TextureRegion
...
+ showDrawnCards() : void
+ hideDrawnCards() : void
+ update() : void
+ resize() : void
...
Creates
Creates
Sets up
core.map :: <MapHandler>
...
+ render(Batch) : void
+ update(float) : void
+ zoomCamera(int) : void
+ moveCamera(int) : void
+ addEntity(Entity) : void
+ removeEntity(Entity) : boolean
+ addEntityLaser(Tile) : void
+ removeEntityLaser(Tile) : void
+ isOutsideVoard(int, int) : boolean
+ resize(int ,int) : void
+ getTile(TiledMapTileLayer, int, int) : Tile
...
Creates and servers
Tile API (see own diagram)
core :: GameGraphics
- roboRally : RoboRally
+ HEADLESS : boolean
...
+ render() : void
+ create() : void
+ resize(int, int) : void
+ getRoboRally() : RoboRally
+ getSoundPlayer() : SoundPlayer
+ getUiHandler() : UIHandler
+ getCPEventHandler() : ControlPanelEventHandler
+ getInputMultiPlexer() : InputMultiplexer
Screens : setup information and menu.
GameScreen contains instance of GameGraphics. Screens have their own
Creates if player is hosting
1
Creates
1
Creates
core:: RoboRally
- deck : Deck
- currentMap :: MapHandler
- playerHandler :: PlayerHandler
+ round() : void
Creates if multiplayer
core.multiplayer:: server
...
Listens to the clients and sends information about the other players
...
core.multiplayer:: client
...
Listens fto the server and sends infromation about the users actions to the server
...
core:.ui.event:: ControlPanelEventHandler
- listeners : HashMap<Class<? extends ControlPanelEvent, Set<ControlPanelEventListener>>
+ registerListener(ControlPanelEvent, ControlPanelEventLister<ControlPanelEvent>) : void
+ fireEvent(ControlPanelEvent)
Creates
1
Use
Creates
Creates
core.round::: DefaultGameRule
+ generate() : Round
core.map.cards::<<Deck>>
- cards : Card[]
- shuffleReps: int
+ draw(int) : Card[]
+ shuffler(): void
core.player::<<IPlayerHandler>>
- playerCount : int
- players : List<IPlayer>
+ generateOnePlayer() : void
+ endTurn() : void
+ startTurn() : void
+ generatePlayers() : void
...
core.round:: Round
- phasesAmount : int
- registerPhases :: List<Phase>
- cleanupPhases :: List<Phase>
+ startRount() : void
Creates
core.player::<<IPlayer>>
- lives : int
- destroyed : boolean
- damageTokens : int
- backup : Vector2Int
- health: int
- flags : int
- poweredDown : boolean
...
+ damage(int) : void
+ kill() : void
+ getNextCard(int) : ComparableTuple<Card, IPlayer>
+ setBackup(int, int) : void
+ canGetFlag(int) : boolean
+ registerFlagVisit() : void
...
Runs
Contains Tile instances of active players
Creates
core.round.phase :: AbstractPhase
+ RunPhase