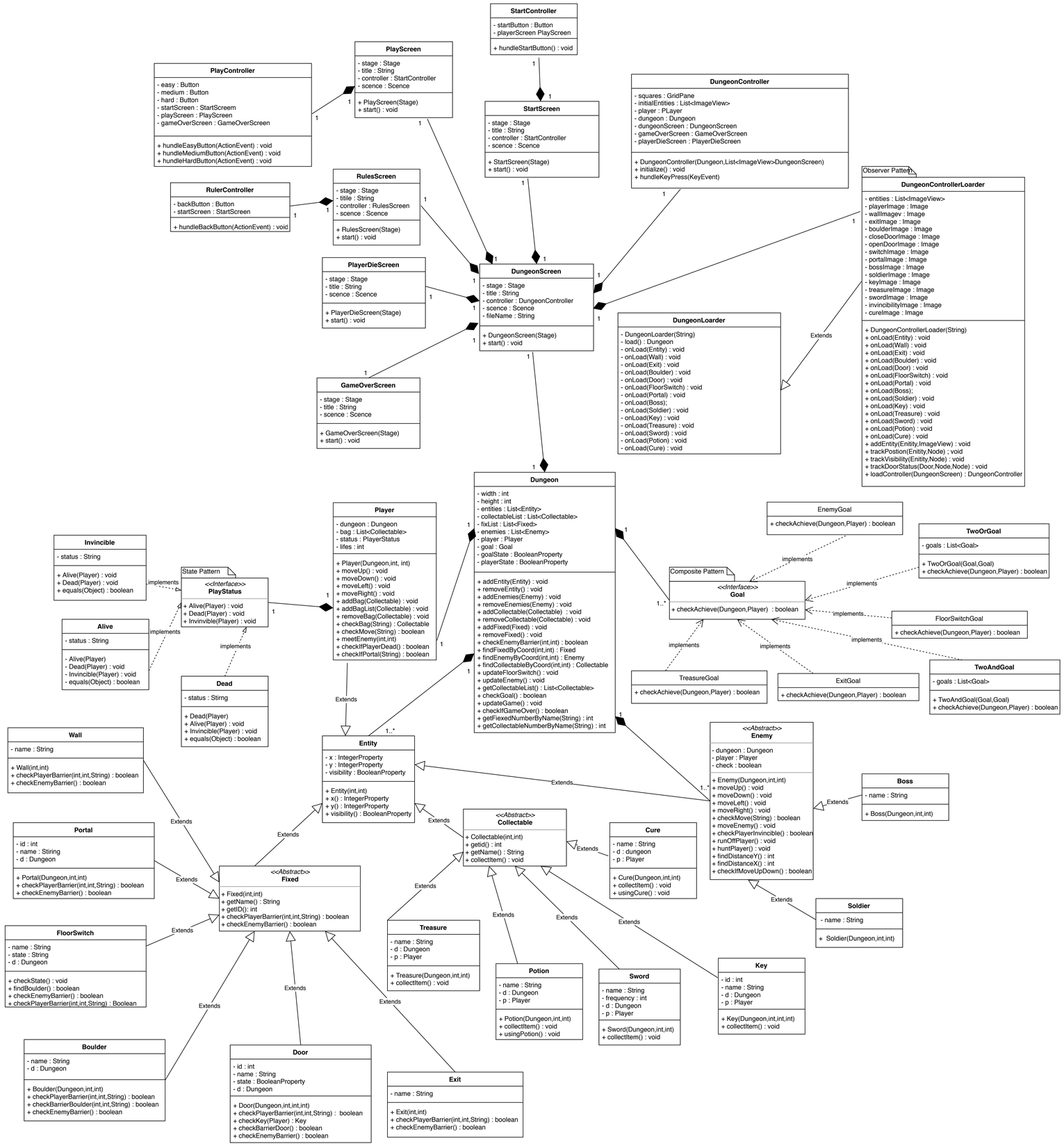
StartController
- startButton : Button
- playerScreen PlayScreen
+ hundleStartButton() : void
PlayScreen
- stage : Stage
- title : String
- controller : StartController
- scence : Scence
+ PlayScreen(Stage)
+ start() : void
PlayController
- easy : Button
- medium : Button
- hard : Button
- startScreen : StartScreem
- playScreen : PlayScreen
- gameOverScreen : GameOverScreen
+ hundleEasyButton(ActionEvent) : void
+ hundleMediumButton(ActionEvent) : void
+ hundleHardButton(ActionEvent) : void
StartScreen
- stage : Stage
- title : String
- controller : StartController
- scence : Scence
+ StartScreen(Stage)
+ start() : void
DungeonController
- squares : GridPane
- initialEntities : List<ImageView>
- player : PLayer
- dungeon : Dungeon
- dungeonScreen : DungeonScreen
- gameOverScreen : GameOverScreen
- playerDieScreen : PlayerDieScreen
+ DungeonController(Dungeon,List<ImageView>DungeonScreen)
+ initialize() : void
+ hundleKeyPress(KeyEvent)
RulerController
- backButton : Button
- startScreen : StartScreen
+ hundleBackButton(ActionEvent) : void
RulesScreen
- stage : Stage
- title : String
- controller : RulesScreen
- scence : Scence
+ RulesScreen(Stage)
+ start() : void
Observer Pattern
DungeonControllerLoarder
- entities : List<ImageView>
- playerImage : Image
- wallImagev : Image
- exitImage : Image
- boulderImage : Image
- closeDoorImage : Image
- openDoorImage : Image
- switchImage : Image
- portalImage : Image
- bossImage : Image
- soldierImage : Image
- keyImage : Image
- treasureImage : Image
- swordImage : Image
- invincibilityImage : Image
- cureImage : Image
+ DungeonControllerLoader(String)
+ onLoad(Entity) : void
+ onLoad(Wall) : void
+ onLoad(Exit) : void
+ onLoad(Boulder) : void
+ onLoad(Door) : void
+ onLoad(FloorSwitch) : void
+ onLoad(Portal) : void
+ onLoad(Boss);
+ onLoad(Soldier) : void
+ onLoad(Key) : void
+ onLoad(Treasure) : void
+ onLoad(Sword) : void
+ onLoad(Potion) : void
+ onLoad(Cure) : void
+ addEntity(Enitity,ImageView) : void
+ trackPostion(Entity,Node) ; void
+ trackVisibility(Entity,Node) : void
+ trackDoorStatus(Door,Node,Node) : void
+ loadController(DungeonScreen) : DungeonController
PlayerDieScreen
- stage : Stage
- title : String
- scence : Scence
+ PlayerDieScreen(Stage)
+ start() : void
DungeonScreen
- stage : Stage
- title : String
- controller : DungeonController
- scence : Scence
- fileName : String
+ DungeonScreen(Stage)
+ start() : void
DungeonLoarder
- DungeonLoarder(String)
- load() : Dungeon
- onLoad(Entity) : void
- onLoad(Wall) : void
- onLoad(Exit) : void
- onLoad(Boulder) : void
- onLoad(Door) : void
- onLoad(FloorSwitch) : void
- onLoad(Portal) : void
- onLoad(Boss);
- onLoad(Soldier) : void
- onLoad(Key) : void
- onLoad(Treasure) : void
- onLoad(Sword) : void
- onLoad(Potion) : void
- onLoad(Cure) : void
Extends
GameOverScreen
- stage : Stage
- title : String
- scence : Scence
+ GameOverScreen(Stage)
+ start() : void
Dungeon
- width : int
- height : int
- entities : List<Entity>
- collectableList : List<Collectable>
- fixList : List<Fixed>
- enemies : List<Enemy>
- player : Player
- goal : Goal
- goalState : BooleanProperty
- playerState : BooleanProperty
+ addEntity(Entity) : void
+ removeEntity() : void
+ addEnemies(Enemy) : void
+ removeEnemies(Enemy) : void
+ addCollectable(Collectable) : void
+ removeCollectable(Collectable) : void
+ addFixed(Fixed) : void
+ removeFixed() : void
+ checkEnemyBarrier(int,int) : boolean
+ findFixedByCoord(int,int) : Fixed
+ findEnemyByCoord(int,int) : Enemy
+ findCollectableByCoord(int,int) : Collectable
+ updateFloorSwitch() : void
+ updateEnemy() : void
+ getCollectableList() : List<Collectable>
+ checkGoal() : boolean
+ updateGame() : void
+ checkIfGameOver() : boolean
+ getFiexedNumberByName(String) : int
+ getCollectableNumberByName(String) : int
Player
- dungeon : Dungeon
- bag : List<Collectable>
- status : PlayerStatus
- lifes : int
+ Player(Dungeon,int, int)
+ moveUp() : void
+ moveDown() : void
+ moveLeft() : void
+ moveRight() : void
+ addBag(Collectable) : void
+ addBagList(Collectable) : void
+ removeBag(Collectable) : void
+ checkBag(String) : Collectable
+ checkMove(String) : boolean
+ meetEnemy(int,int)
+ checkIfPlayerDead() : boolean
+ checkIfPortal(String) : boolean
Invincible
- status : String
+ Alive(Player) : void
+ Dead(Player) : void
+ equals(Object) : boolean
implements
State Pattern
<<Interface>>
PlayStatus
+ Alive(Player) : void
+ Dead(Player) : void
+ Invinvible(Player) : void
Alive
- status : String
- Alive(Player)
- Dead(Player) : void
- Invincible(Player) : void
- equals(Object) : boolean
Dead
- status : Stirng
+ Dead(Player)
+ Alive(Player) : void
+ Invincible(Player) : void
+ equals(Object) : boolean
Extends
EnemyGoal
+ checkAchieve(Dungeon,Player) : boolean
TwoOrGoal
- goals : List<Goal>
+ TwoOrGoal(Goal,Goal)
+ checkAchieve(Dungeon,Player) : boolean
Composite Pattern
<<Interface>>
Goal
+ checkAchieve(Dungeon,Player) : boolean
FloorSwitchGoal
+ checkAchieve(Dungeon,Player) : boolean
TreasureGoal
+ checkAchieve(Dungeon,Player) : boolean
ExitGoal
+ checkAchieve(Dungeon,Player) : boolean
TwoAndGoal
- goals : List<Goal>
+ TwoAndGoal(Goal,Goal)
+ checkAchieve(Dungeon,Player) : boolean
Wall
- name : String
+ Wall(int,int)
+ checkPlayerBarrier(int,int,String) : boolean
+ checkEnemyBarrier() : boolean
Entity
- x : IntegerProperty
- y : IntegerProperty
- visibility : BooleanProperty
+ Entity(int,int)
+ x() : IntegerProperty
+ y() : IntegerProperty
+ visibility() : BooleanProperty
<<Abstract>>
Enemy
- dungeon : Dungeon
- player : Player
- check : boolean
+ Enemy(Dungeon,int,int)
+ moveUp() : void
+ moveDown() : void
+ moveLeft() : void
+ moveRight() : void
+ checkMove(String) : boolean
+ moveEnemy() : void
+ checkPlayerInvincible() : boolean
+ runOffPlayer() : void
+ huntPlayer() : void
+ findDistanceY() : int
+ findDistanceX() : int
+ checkIfMoveUpDown() : boolean
Boss
- name : String
+ Boss(Dungeon,int,int)
Portal
- id : int
- name : Stiring
- d : Dungeon
+ Portal(Dungeon,int,int)
+ checkPlayerBarrier(int,int,String) : boolean
+ checkEnemyBarrier() : boolean
<<Abstract>>
Fixed
+ Fixed(int,int)
+ getName() : String
+ getID(): int
+ checkPlayerBarrier(int,int,String) : boolean
+ checkEnemyBarrier() : boolean
<<Abstract>>
Collectable
+ Collectable(int,int)
+ getid() : int
+ getName() : String
+ collectItem() : void
Cure
- name : String
- d : dungeon
- p : Player
+ Cure(Dungeon,int,int)
+ collectItem() : void
+ usingCure() : void
Soldier
- name : String
+ Soldier(Dungeon,int,int)
FloorSwitch
- name : String
- state : String
- d : Dungeon
+ checkState() : void
+ findBoulder() : boolean
+ checkEnemyBarrier() : boolean
+ checkPlayerBarrier(int,int,String) : Boolean
Treasure
- name : String
- d : Dungeon
- p : Player
+ Treasure(Dungeon,int,int)
+ collectItem() : void
Potion
- name : String
- d : Dungeon
- p : Player
+ Potion(Dungeon,int,int)
+ collectItem() : void
+ usingPotion() : void
Sword
- name : String
- frequency : int
- d : Dungeon
- p : Player
+ Sword(Dungeon,int,int)
+ collectItem() : void
Key
- id : int
- name : String
- d : Dungeon
- p : Player
+ Key(Dungeon,int,int,int)
+ collectItem() : void
Boulder
- name : String
- d : Dungeon
+ Boulder(Dungeon,int,int)
+ checkPlayerBarrier(int,int,String) : boolean
+ checkBarrierBoulder(int,int,String) : boolean
+ checkEnemyBarrier() : boolean
Door
- id : int
- name : String
- state : BooleanProperty
- d : Dungeon
+ Door(Dungeon,int,int,int)
+ checkPlayerBarrier(int,int,String) : boolean
+ checkKey(Player) : Key
+ checkBarrierDoor() : boolean
+ checkEnemyBarrier() : boolean
Exit
- name : String
+ Exit(int,int)
+ checkPlayerBarrier(int,int,String) : boolean
+ checkEnemyBarrier() : boolean