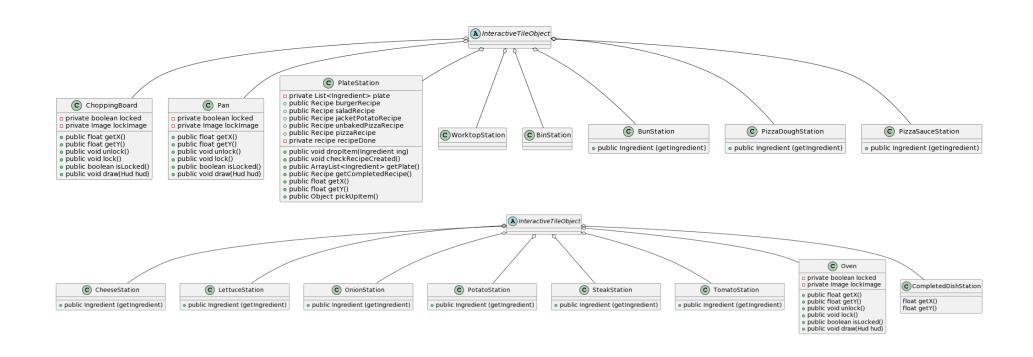


# C Order

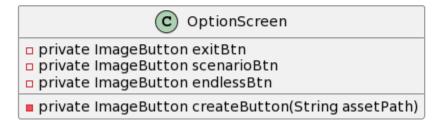
- private Dish burgerDish
- private Dish saladDish
- private Dish jacketPotatoDish
- private Dish pizzaDish
- private ArrayList<Dish> menu
- private Random randomizer
- o public int orderTime = 0
- public ArrayList<Dish> dishes = new ArrayList<Dish>()
- public boolean isComplete()
- public void create(float x, float y, SpriteBatch batch)
- public boolean completeDish(Recipe recipe)



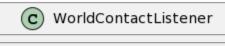


private ImageButton createButton(String assetPath)

MainMenuScreen private ImageButton exitBtn private ImageButton startBtn private ImageButton resumeBtn private preferences save private ImageButton createButton(String assetPath)



C StartScreen private PlayScreen pausedScreen



- public void beginContact(Contact contact) public void endContact(Contact contact)

B2WorldCreator private int choppingBoardCount = 0 private int panCount = 0 private int ovenCount = 0 private int unlockedChoppingBoards = 0 private int unlockedPans = 0 private int unlockedOvens = 0

# C HUD o public Stage stage private Viewport viewport private MainGame game private PlayScreen screen private Boolean scenarioComplete private integer score o public String timeStr o public Table table o public Integer repPoints = 3 private Integer numOrders private Integer powerUpTime = 31 private Boolean powerUp = false private ArrayList<String> powerUps = new ArrayList<String>() private String currentPowerUp private Random randomizer = new Random() private Boolean freezeTime = false private Boolean isPaused = false private Boolean chefAvailable = true private Boolean choppingBoardAvailable = true; private Boolean ovenAvailable = true private Boolean panAvailable = true private final static Integer CHEF PRICE = 400 private final static Integer OVEN PRICE = 400 private final static Integer BOARD PRICE = 100 private final static Integer PAN PRICE = 200 public void updateTime(Boolean scenarioCoplete, Order currentOrder) public void updateScore(Boolean scenarioComplete. Integer expectedTime. Integer multiplier. Order currentOrder) public void updateOrder( Boolean scenarioComplete, Integer orderNum) public void generatePowerUp() public String getPowerup() public void freezeTime() public void unfreezeTime() public void setNumOrders(Integer num) public void dispose() public void createProgressBar(float x, float y, Chef chef, float duration) public void updateProgressBards() public void pause() public void showShop() public void hideShop() public boolean isPaused() public void save(boolean quit)

### (C) MainGame

- o public int V WIDTH = 160
- public int V HEIGHT = 160
- public int TILE SIZE = 16
- o public float PPM = 100
- o public SpriteBatch batch
- public void create ()
- public void render()
- public void dispose()

### c playScreen private HUD hud private TiledMap map private Chef chef1 private Chef chef2 private Chef chef3 o public ArrayList<Chef> chefs = new ArratList<Chef>() private Chef controlledChef private Chef lastChef = chef2 o public ArrayList<Order> ordersArray = new ArrayList<Order>() o public PlateStation plateStation o public Boolean scenarioComplete o public Boolean createdOrder o public static float trayX o public static float trayY private float timeSeconds = 0f private float timeSecondsCount = 0f private int orderCount o public static int orderTime = 40 private float chefSpeedMultiplier = 1f private int moneyMultiplier = 1 private float cookSpeedMultiplier = 1f private boolean endless = false private int dishAmount = 1 private Preferences saving = Gdx.app.getPreferences("userData") private boolean resume private Integer currentOrderNum = 1 private ArrayList<ChoppingBoard> boards = new ArrayList<ChoppingBoard>() private ArrayList<Pan> pans = new ArrayList<Pan>() private ArrayList<Oven> ovens = new ArrayList<Oven>() public void addChoppingBoard(ChoppingBoard board) public ArrayList<ChoppingBoard> getChoppingBoards() public void addPan(Pan pan) • public ArrayList<Pan> getPans() public void addOven(Oven oven) public ArrayList<Oven> getOvens() public void handleInput(float dt) public void update(float dt) public void createResumedOrder() public void createOrder() public void updateOrder() public void render(float delta) private void activatePowerUp() private void disablePowerUps() public void resize(int width, int height) public void pause() public void resume()