







DifficultySelectionScreen

- private ImageButton easyBtn
- private ImageButton normalBtn
- private ImageButton hardBtn
- private ImageButton exitBtn
- private ImageButton createButton(String assetPath)

MainMenuScreen

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

OptionScreen

- private ImageButton exitBtn
- private ImageButton scenarioBtn
- private ImageButton endlessBtn
- private ImageButton createButton(String assetPath)

StartScreen

- private PlayScreen pausedScreen

WorldContactListener

- 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
o private Viewport viewport
o private MainGame game
o private PlayScreen screen
o private Boolean scenarioComplete
o private integer score
o public String timeStr
o public Table table
o public Integer repPoints = 3
o private Integer numOrders
o private Integer powerUpTime = 31
o private Boolean powerUp = false
o private ArrayList<String> powerUps = new ArrayList<String>()
o private String currentPowerUp
o private Random randomizer = new Random()
o private Boolean freezeTime = false
o private Boolean isPaused = false
o private Boolean chefAvailable = true
o private Boolean choppingBoardAvailable = true;
o private Boolean ovenAvailable = true
o private Boolean panAvailable = true
o private final static Integer CHEF_PRICE = 400
  private final static Integer OVEN_PRICE = 400
o private final static Integer BOARD_PRICE = 100
o 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
o public int V_HEIGHT = 160
o public int TILE_SIZE = 16
o public float PPM = 100
o public SpriteBatch batch

● public void create ()
● public void render()
● public void dispose()

```

playScreen

```
private HUD hud
private TiledMap map
private Chef chef1
private Chef chef2
private Chef chef3
public ArrayList<Chef> chefs = new ArratList<Chef>()
private Chef controlledChef
private Chef lastChef = chef2
public ArrayList<Order> ordersArray = new ArrayList<Order>()
public PlateStation plateStation
public Boolean scenarioComplete
public Boolean createdOrder
public static float trayX
public static float trayY
private float timeSeconds = 0f
private float timeSecondsCount = 0f
private int orderCount
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()
```