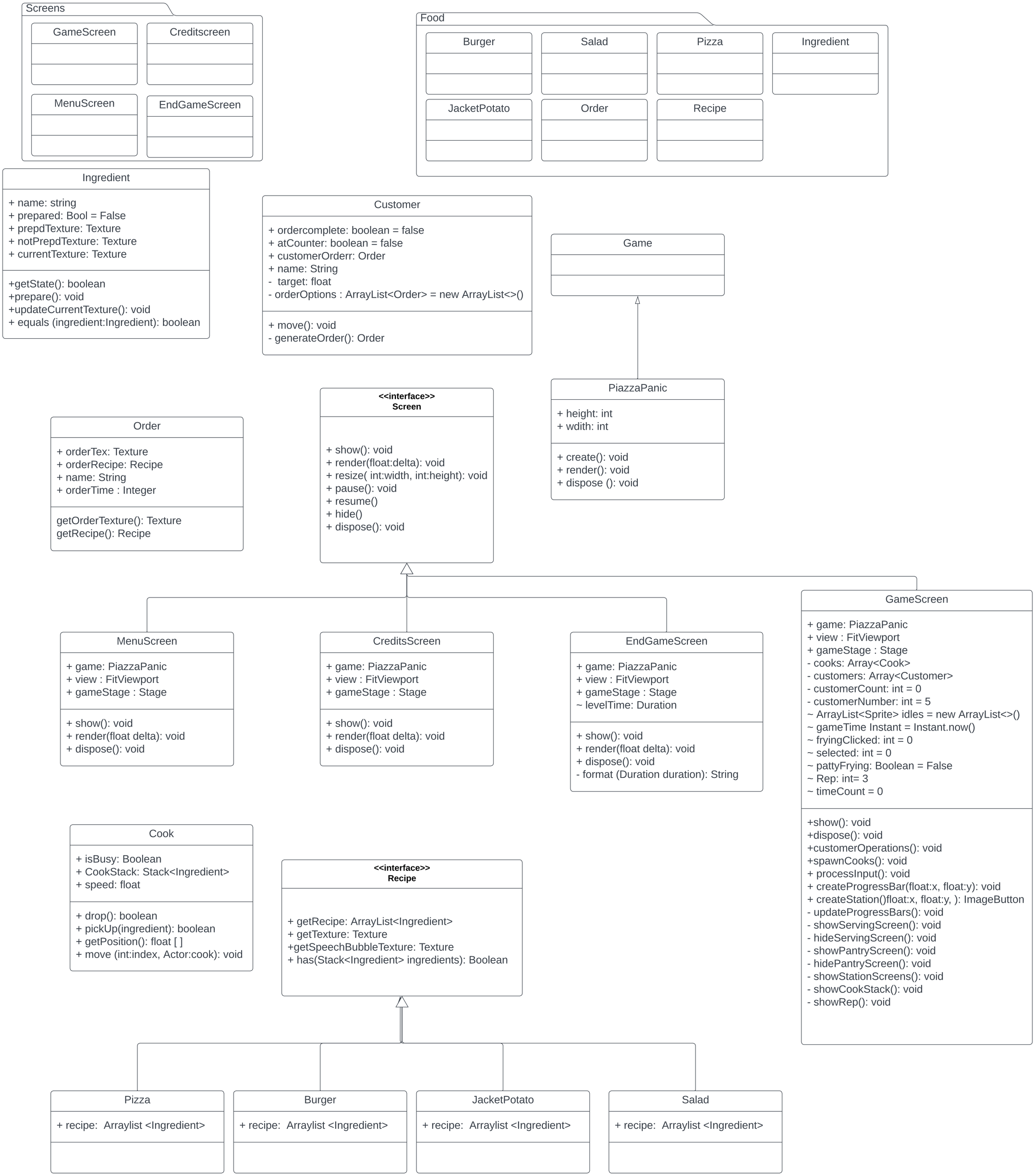
Screens
GameScreen
Creditscreen
MenuScreen
EndGameScreen
Food
Burger
Salad
Pizza
Ingredient
JacketPotato
Order
Recipe
Ingredient
+ name: string
+ prepared: Bool = False
+ prepdTexture: Texture
+ notPrepdTexture: Texture
+ currentTexture: Texture
+getState(): boolean
+prepare(): void
+updateCurrentTexture(): void
+ equals (ingredient:Ingredient): boolean
Customer
+ ordercomplete: boolean = false
+ atCounter: boolean = false
+ customerOrderr: Order
+ name: String
- target: float
- orderOptions : ArrayList<Order> = new ArrayList<>()
+ move(): void
- generateOrder(): Order
Game
PiazzaPanic
+ height: int
+ wdith: int
+ create(): void
+ render(): void
+ dispose (): void
Order
+ orderTex: Texture
+ orderRecipe: Recipe
+ name: String
+ orderTime : Integer
getOrderTexture(): Texture
getRecipe(): Recipe
<<interface>>
Screen
+ show(): void
+ render(float:delta): void
+ resize(int:width, int:height): void
+ pause(): void
+ resume()
+ hide()
+ dispose(): void
MenuScreen
+ game: PiazzaPanic
+ view : FitViewport
+ gameStage : Stage
+ show(): void
+ render(float delta): void
+ dispose(): void
CreditsScreen
+ game: PiazzaPanic
+ view : FitViewport
+ gameStage : Stage
+ show(): void
+ render(float delta): void
+ dispose(): void
EndGameScreen
+ game: PiazzaPanic
+ view : FitViewport
+ gameStage : Stage
~ levelTime: Duration
+ show(): void
+ render(float delta): void
+ dispose(): void
- format (Duration duration): String
GameScreen
+ game: PiazzaPanic
+ view : FitViewport
+ gameStage : Stage
- cooks: Array<Cook>
- customers: Array<Customer>
- customerCount: int = 0
- customerNumber: int = 5
~ ArrayList<Sprite> idles = new ArrayList<>()
~ gameTime Instant = Instant.now()
~ fryingClicked: int = 0
~ selected: int = 0
~ pattyFrying: Boolean = False
~ Rep: int= 3
~ timeCount = 0
+show(): void
+dispose(): void
+customerOperations(): void
+spawnCooks(): void
+ processInput(): void
+ createProgressBar(float:x, float:y): void
+ createStation()float:x, float:y,): ImageButton
- updateProgressBars(): void
- showServingScreen(): void
- hideServingScreen(): void
- showPantryScreen(): void
- hidePantryScreen(): void
- showStationScreens(): void
- showCookStack(): void
- showRep(): void
Cook
+ isBusy: Boolean
+ CookStack: Stack<Ingredient>
+ speed: float
+ drop(): boolean
+ pickUp(ingredient): boolean
+ getPosition(): float [ ]
+ move (int:index, Actor:cook): void
<<interface>>
Recipe
+ getRecipe: ArrayList<Ingredient>
+ getTexture: Texture
+getSpeechBubbleTexture: Texture
+ has(Stack<Ingredient> ingredients): Boolean
Pizza
+ recipe: Arraylist <Ingredient>
Burger
+ recipe: Arraylist <Ingredient>
JacketPotato
+ recipe: Arraylist <Ingredient>
Salad
+ recipe: Arraylist <Ingredient>