OOAD Project 6
Chelsea Stockberger and Bella Barbieri

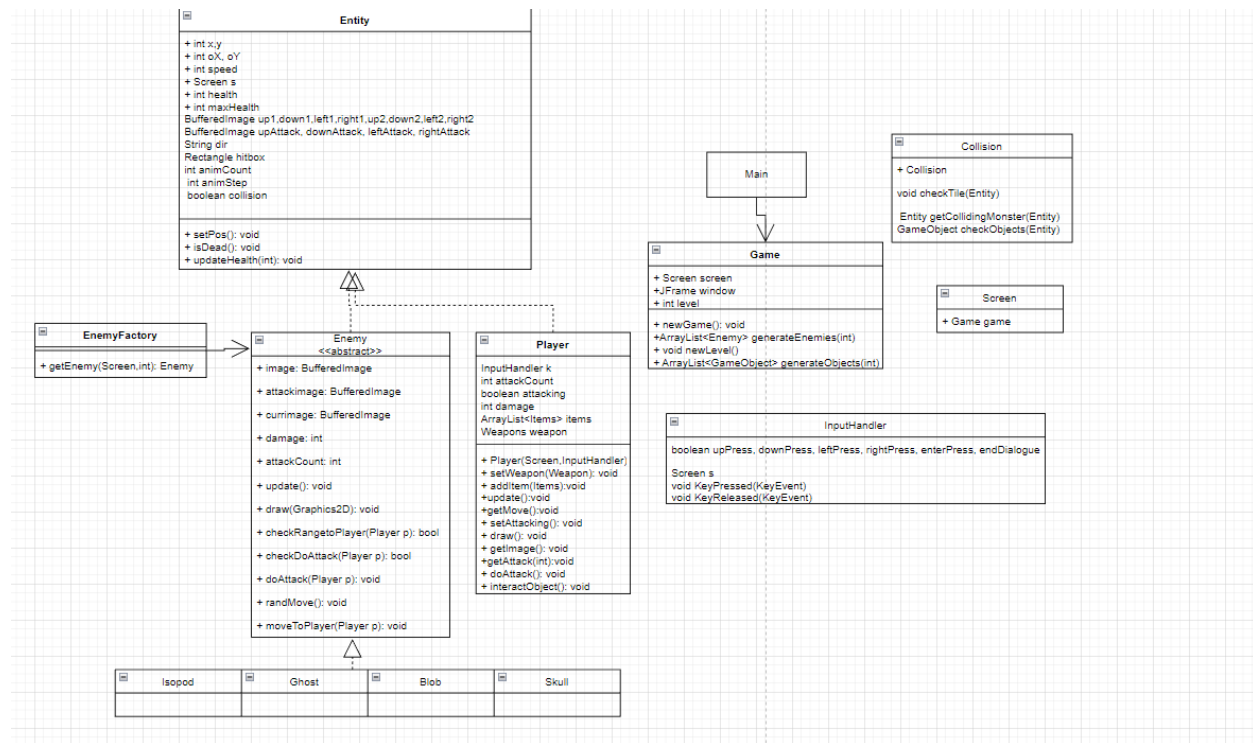**Status Summary**

Work Done:
	Game progress:
		- Tilemap loads
		- Start menu loads
		- Custom pixel art loaded and utilized
		- Movement and collision implemented
		- Enemy movement and collision implemented
		- System to attack and destroy enemies implemented
		- System for enemies to attack and deal damage to player implemented
		- Once all enemies destroyed, portal appears
		- Once portal entered, new enemies loaded in
		- Player can interact with objects and obtain weapon or item from them
		- Objects become non collision and change graphic
		- Dialogue shows up when player interacts with object, with text that can be scrolled through and then closed once finished
	Objects created:
- Map: holds all Tiles and the map displayed on screen
- Tile: holds if tile can be collided with and tile image
- Screen: controls main game thread and draws on entities
- Enemy: contains each enemy type data and image, inherits Entity
- Player: holds player data and image, inherits Entity
- InputHandler: takes user key input
- Entity: holds entity information such as health, etc
- Game: controls starting of new game, generating enemies, new level
- EnemyFactory and TileFactory: for creating Enemy/Tile objects using factory pattern
- Audio: Holds audio file to loop
- GameObject: For interactable objects, chest and mushroom
- Weapon: Holds all weapons
- Item: holds items the player can obtain
- StartMenu: handles the swing start menu for before the game

Class Diagram

Entity

+ int x,y
+ int oX, oY
+ int speed
+ Screen s
+ int health
+ int maxHealth
BufferedImage up1,down1,left1,right1,up2,down2,left2,right2
BufferedImage upAttack, downAttack, leftAttack, rightAttack
String dir
Rectangle hitbox
int animCount
int animStep
boolean collision

+ setPos(): void
+ isDead(): void
+ updateHealth(int): void

Main

Collision

+ Collision

void checkTile(Entity)

Entity getCollidingMonster(Entity)
GameObject checkObjects(Entity)

Game

+ Screen screen
+JFrame window
+ int level

+ newGame(): void
+ArrayList<Enemy> generateEnemies(int)
+ void newLevel()
+ ArrayList<GameObject> generateObjects(int)

Screen

+ Game game

EnemyFactory

+ getEnemy(Screen,int): Enemy

Enemy
<>

+ image: BufferedImage

+ attackimage: BufferedImage

+ currimage: BufferedImage

+ damage: int

+ attackCount: int

+ update(): void

+ draw(Graphics2D): void

+ checkRangetoPlayer(Player p): bool

+ checkDoAttack(Player p): bool

+ doAttack(Player p): void

+ randMove(): void

+ moveToPlayer(Player p): void

Player

InputHandler k
int attackCount
boolean attacking
int damage
ArrayList<Items> items
Weapons weapon

+ Player(Screen,InputHandler)
+ setWeapon(Weapon): void
+ addItem(Items):void
+update():void
+getMove():void
+ setAttacking(): void
+ draw(): void
+ getImage(): void
+getAttack(int):void
+ doAttack(): void
+ interactObject(): void

InputHandler

boolean upPress, downPress, leftPress, rightPress, enterPress, endDialogue

Screen s
void KeyPressed(KeyEvent)
void KeyReleased(KeyEvent)

Isopod

Ghost

Blob

Skull

Plan for Next Iteration
-    Implement game-end circumstance
     -    Add final boss at Level 10, ends once defeated
     -    Game-end once player dies
-    Implement NPCs to buy weapons or potions from
-    Add save-game feature, holding player health, level, and equipped items making user able to load up game even after closing entire program
-    Add more tilemaps