Game	Application (it is what's run)
• Initialize Model, View, and Controller	• View • Model • Controller

View	
• Handle all aspects of game GUI. • Give easy access to important elements in the GUI.	• JavaFX elements • Model

ViewStart	
• Handle Splash Screen • Take player to the game	

ViewGameOver	
• Handle Game Over screen	

Model	
• Create the player • Give easy access to the player	

Ctrl

• Handle user input • Run all periodic/per frame actions of all classes • Random, but strategically, create all enemies. • Handle currency and tower construction	• Model • View • ViewObj • Fighter • Enemy • TowerEnum • EnemyEnum • Player

GameNoise

• Play music and sounds	• Task (threading) • JavaFX MediPlayer

ViewObj
Mob, Tower

• Create an image on the game screen • Allow for image transformations, i.e. translation and rotation • Allow for per frame actions	• View • Ctrl

ViewObj

Mob
Player, Enemy, Projectile

• Move. i.e. create smooth per frame translation	

Player	Mob
• Handle the different looking and moving directions. • Tie a combat object to the on-screen player	• Ctrl (moves Player) • Fighter

Tower	ViewObj
• Tie a combat object to the on-screen Tower	• Fighter

Enemy	Mob
• Tie a combat object to the on-screen enemy	• Fighter

Projectile	Mob
• Check for collisions with enemy Fighters. • Deal damage to enemy Fighters upon collision. • Despawn after specified time period.	

Fighter	PlayerFighter
• Handle all combat related actions, including: • Have health. • Attack with appropriate frequency. • Target enemies appropriately. • Take damage. • Die when health is too low. • Execute per frame actions	• Ctrl • Tower • Enemy • Projectile

PlayerFighter	
	Fighter
• Game over upon dying. • Shoot in direction of mouse	• Player