Robot Dodge

UML Diagram

Robot Dodge Nathan Bell | May 6, 2020

Program Bullet PlaverLives + bulletbitmap: Bitmap + Ouit: bool << auto property>> life: int - HeartsX: int <<auto property>> - HeartsY: int <<auto property>> + X: Double + Y: Double + HandleInput(player: Player): void + Life: int + bulletcollision: Bool + Endgame(playerlives: PlayerLives, rbd: RobotDodge): void - Velocity: Vector2D + Main <<static>> PlayerLives(playerwindow: Window) + Target(player; Player); void + Drawn: void + CollidedWith(other:Robot): bool + Update(player: Player): void + Draw(): void + Alive(): bool Player + X: Double <<auto property>> + Y: Double <<auto property>> + shoot: bool <<false>> + hasFired: bool <<false>> RobotDodge Robot player: Player _PlayerBitmap: Bitmap playerlives: PlayerLives - gamewindow: Window - Robots: List<Robot> speed: int + X: Double << auto property>> + Y: Double <-auto property>> + Y: Double <-auto property>> + CollisionCircle: Circle <-auto property>> - Velocity: Vector2D <<auto property>> + Width: int + Height: int + Score: int

+ RandomRobot(_gamewindow: Window): Robot

+ Draw(playerwindow: Window): void

+ Update(): void

+ RobotDodge(w: Window, p: Player) + BulletInteractions(bullet: Bullet) void

+ Collision(): void + PrintScore(_gamewindow: Window, tmr: Timer): void

+ Player (playerwindow: Window)

+ Draw(): void

+ StayOnWindow(playerwindow: Window): void + HandleInput() + Collidedwith(other: Robot): bool

+ MoveUp(): void + MoveDown(): void + MoveRight(): void

+ MoveLeft(): void

+ MainColor: Color

+ Width: int

+ Height: int

+ Robot(gamewindow: Window, player: Player)

+ Update(): void + IsOffScreen(screen: Window): bool + Draw(): void

When adding the bullets to the program, I came into many issues. The first was managing to drawing the correct bitmap for the side of the person that it was being shot on. This turned out to be a simple if statement based around the mouse's 'X' value.

I used the same concept for how the robot moves to make the functionality for shooting a bullet. Using two Point2D's, I could then create a velocity Vector2D.

The next issue that I ran into was drawing the bullet. I would either have an issue where the bullet would draw but not move from the player, or the bullet would only draw momentarily and then disappear like it wasn't a part of the while loop that is in program.

After fixing that issue, I then created the ability for bullets to destroy the robots. Upon a bullet hitting a robot, both the bullet and robot will be removed, similar to how a robot would disappear upon hitting the player.