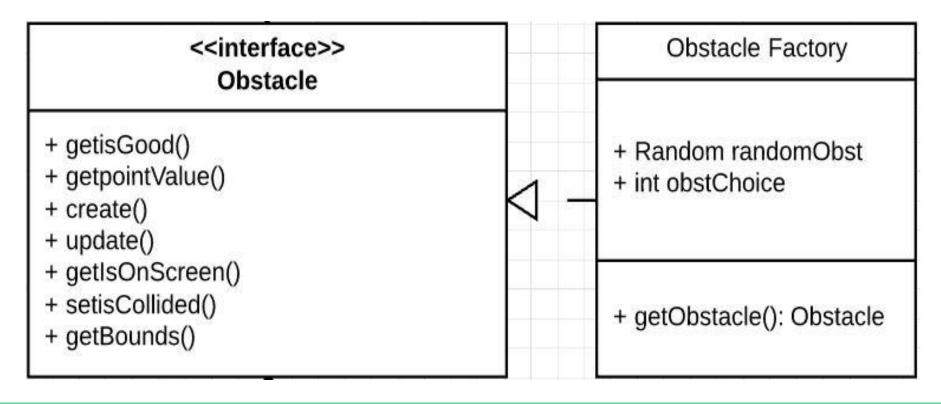
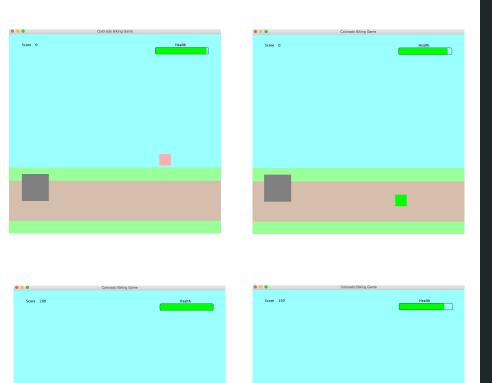
Colorado Biking Game

Brandon Barrett, Brandon Jacquez, Isabella Figueroa, Dilara Madinger

Demo Video

Factory Design Pattern

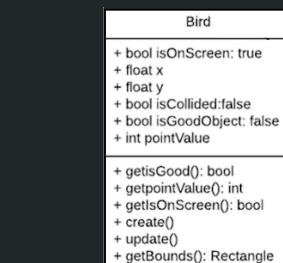




Obstacle Factory
Randomly
Creates Different
Obstacles.

Each has:

- Paint Function
- Update Function
- Point Values



+ setisCollided()

Biker + bool isOnScreen: true + float x + float y + bool isCollided: false + float jump_height + float yVelocity + int original_height + int duck_height + bool isGoodObject: false + int pointValue + getisGood(): bool

+ getpointValue(): int

+ create()

+ update()

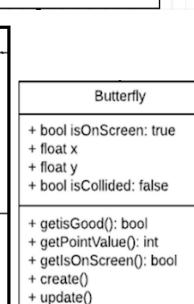
+ jump()

+ duck()

+ setidCollided()

+ getIsOnScreen(): bool

+ getBounds(): Rectangle



+ getBounds()

+ setisCollided()

<<interface>>

Obstacle

+ getisGood()

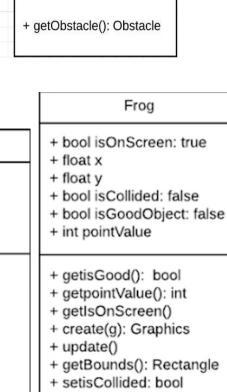
+ create()

+ update()

+ getpointValue()

+ getIsOnScreen() + setisCollided()

+ getBounds()



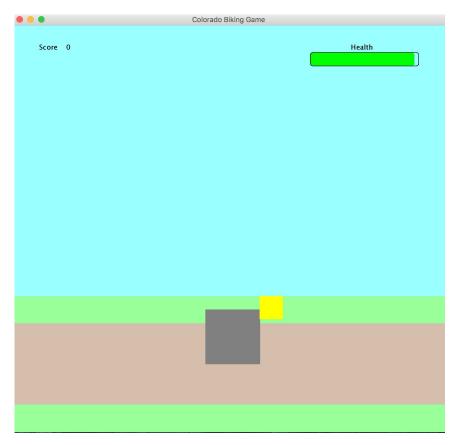
Obstacle Factory

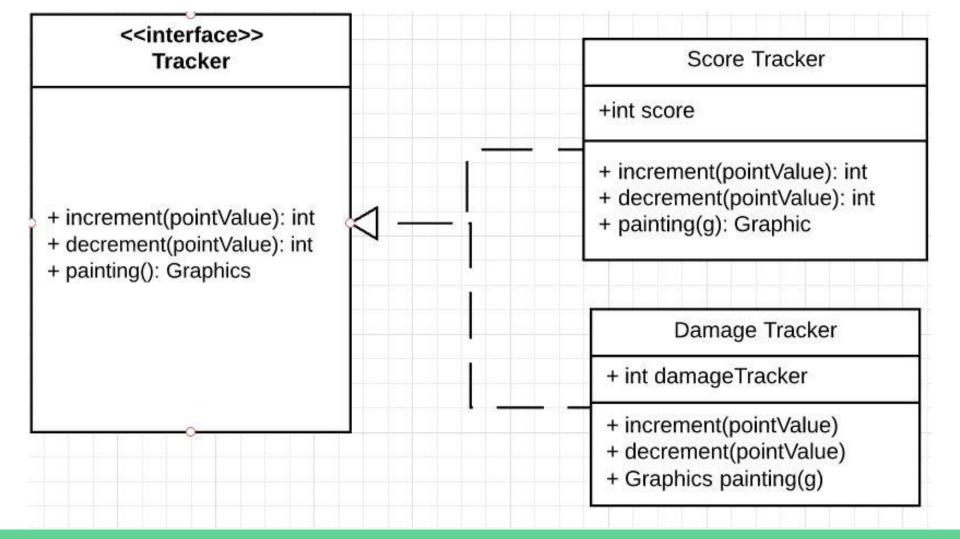
+ Random randomObst

+ int obstChoice

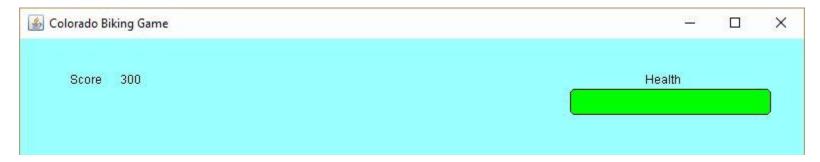
Use Case: Collision Detection

- Use getBounds method in obstacles classes to make a rectangle object with dimensions
- Have Sprite class call intersects()
 in game loop to detect a collision
 continuously.
- To avoid multiple deductions to score, we destroy the obstacle after hit and immediately create a new obstacle





Use Case "Trackers"



Score: decrements with frogs, ducks, and a fellow bike.

Increments with butterflies.

Health: decrements with frogs, ducks, and a fellow bike.