Lab 5 – Observations
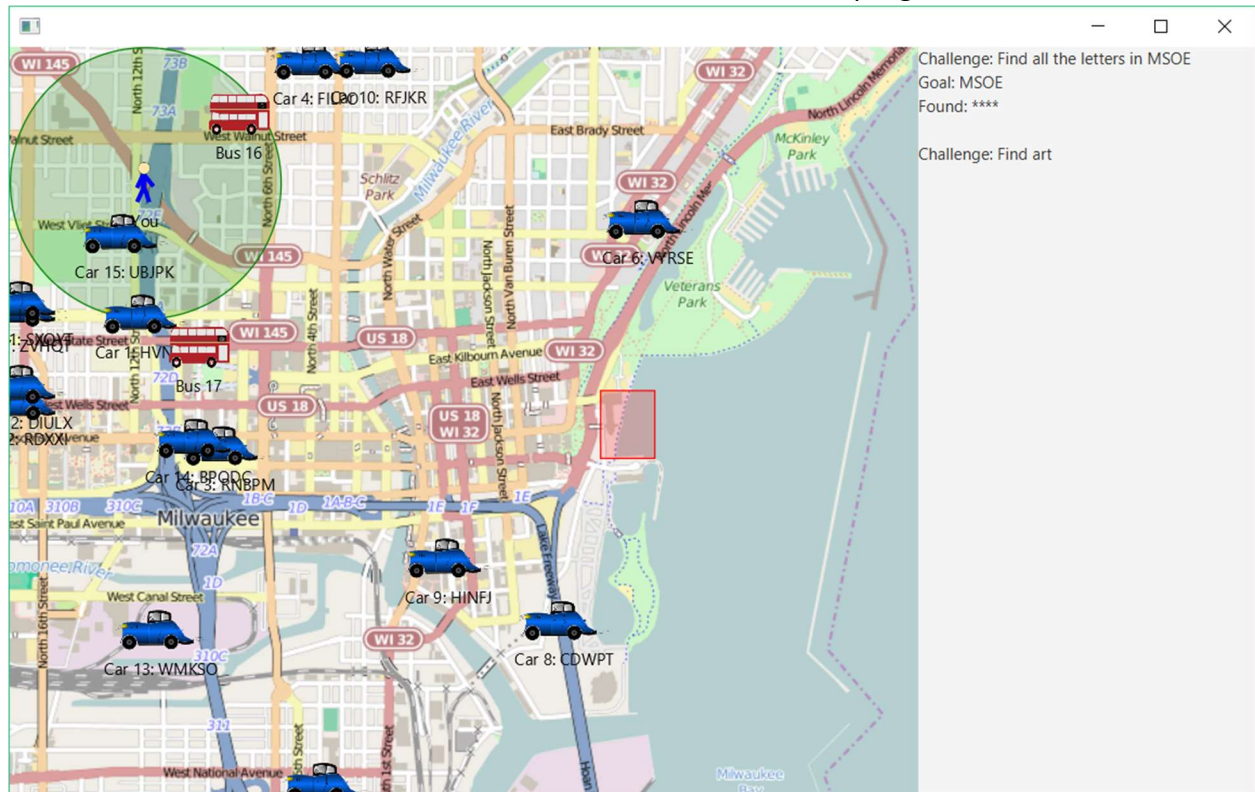
David Schulz

Screenshots

The first screenshot shows what the window looks like when the program first starts.

This image shows that the letters appear when the player tags cars with plates that have the necessary letters.

This image shows that the picture appears when the player tags the museum.

This image shows that the completed message will show when all letters are found.
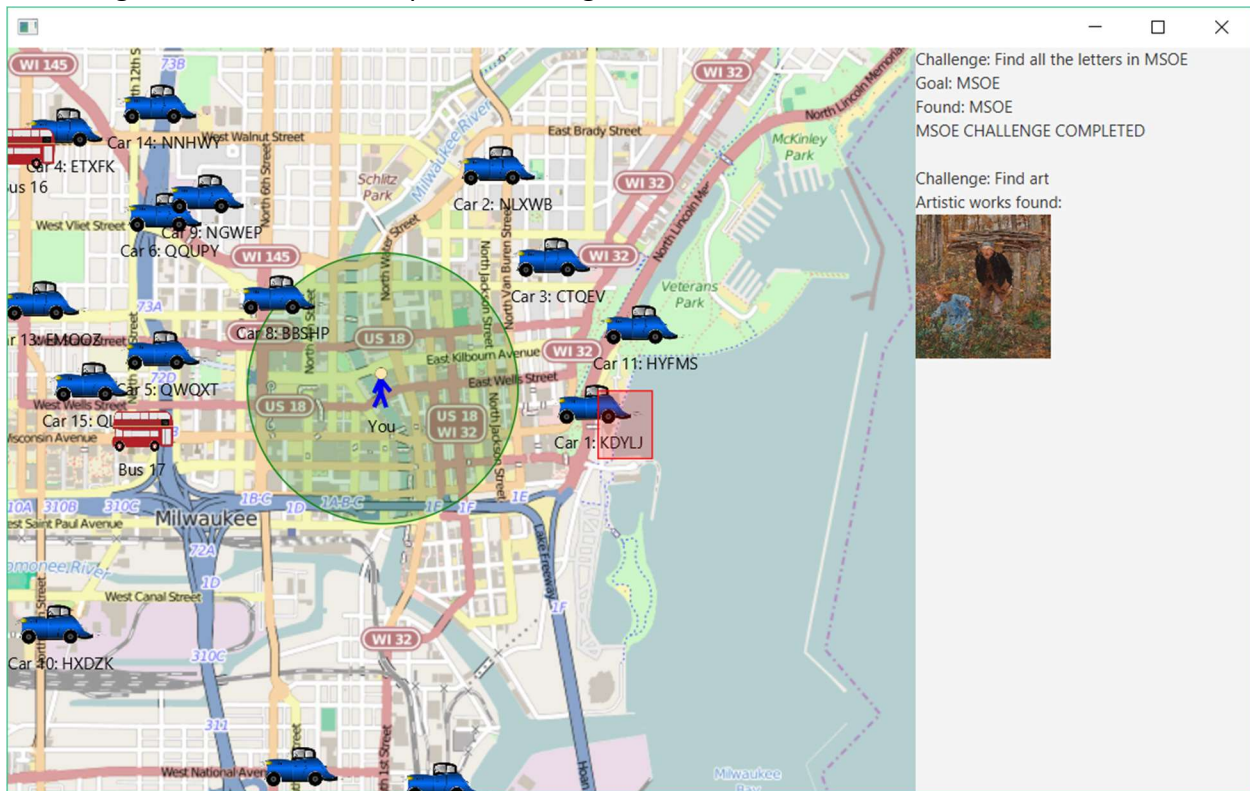
This image shows that the bus challenge will appear when the player tags a bus.
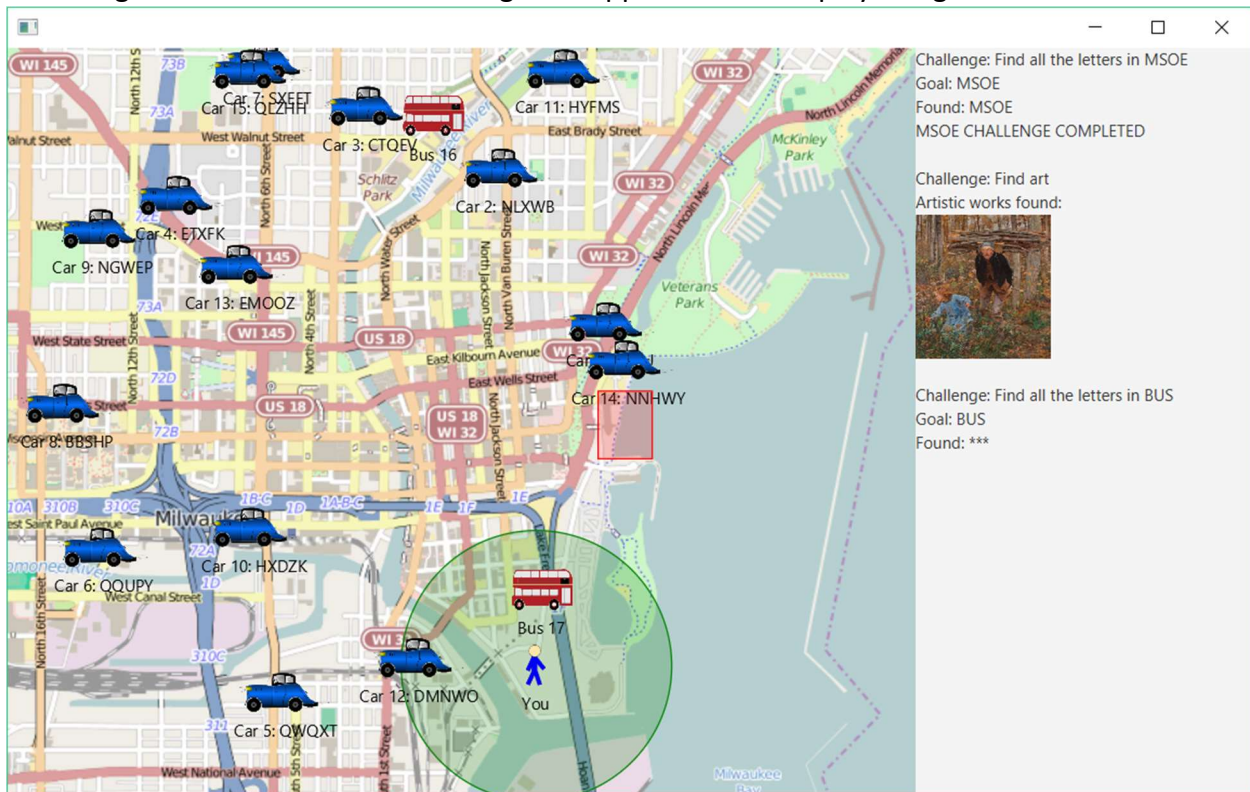
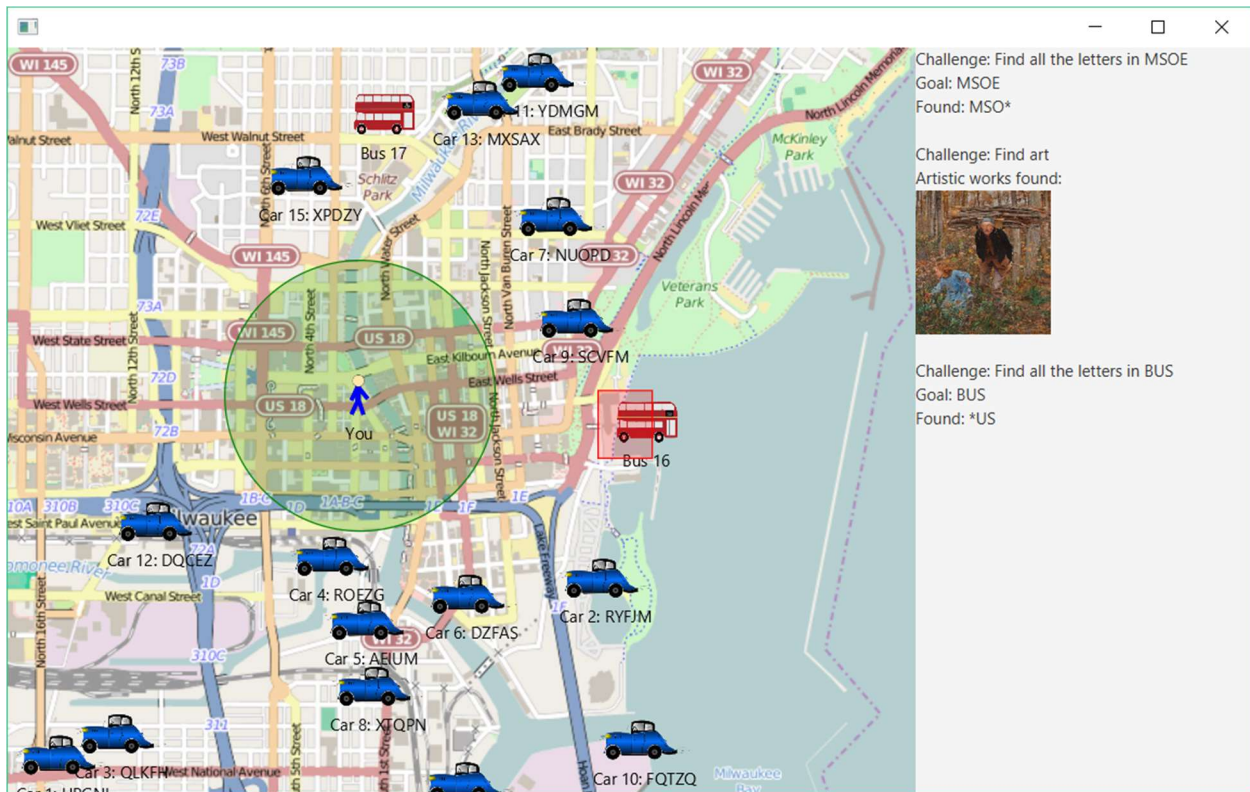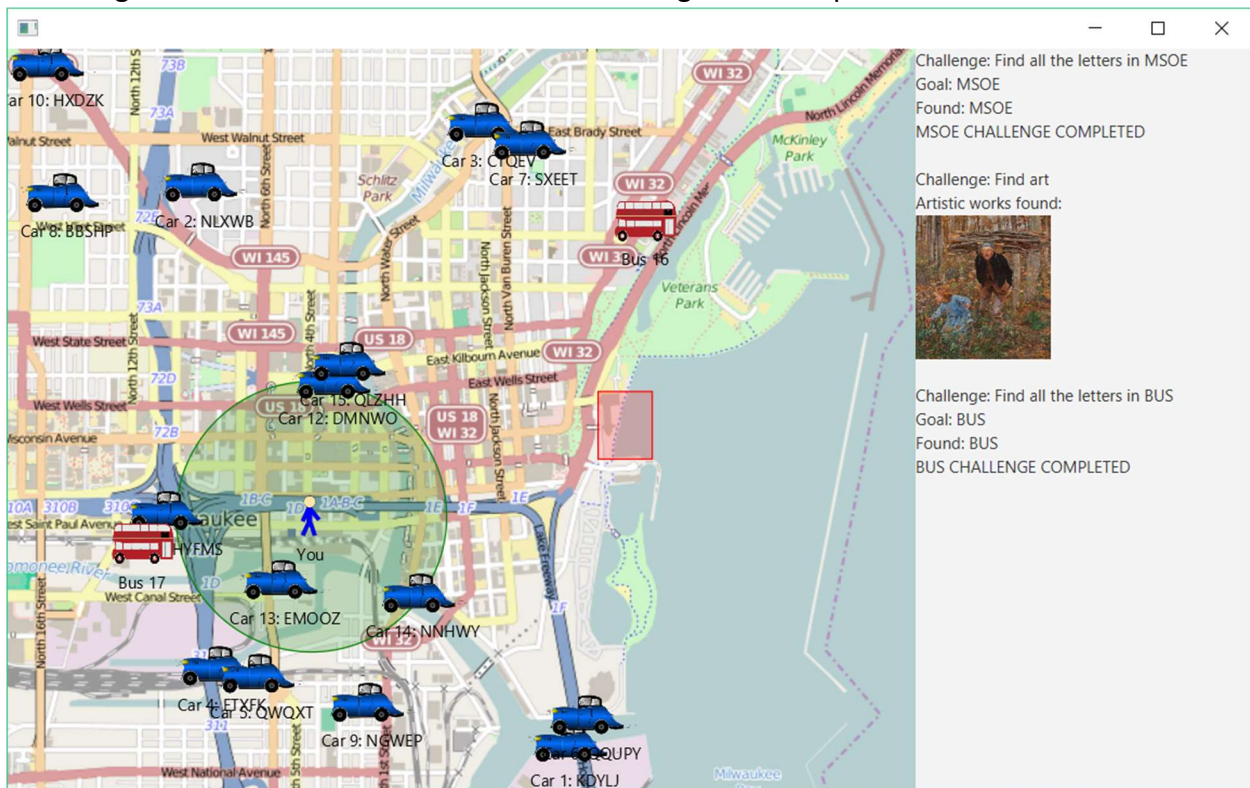This image shows what it looks like when the bus challenge is shown and some letters are found for both.

This image shows what it looks like when all challenges are completed.

UML Diagram

class Lab 5

«static»
Person::MouseGestures

- orgSceneX: double
- orgSceneY: double
- orgTranslateX: double
- orgTranslateY: double
- circleOnMousePressedEventHandler: EventHandler<MouseEvent> = (MouseEvent eve...
- circleOnMouseDraggedEventHandler: EventHandler<MouseEvent> = (MouseEvent eve...

- makeDraggable(Circle): void

«interface»
Taggable

+ isTagged(Point2D): boolean
+ taggedBy(MobileEntity): void

BusObserver

- initialized: boolean
- busPane: VBox
- busHit: boolean

+ BusObserver(Pane)
+ update(): void
+ setPlate(String): void

CarObserver

- carPane: VBox
- goal: String
- carFound: String
- initialized: boolean
- isComplete: boolean
- carFoundLabel: Label
- carPlate: String
- busFound: String
- isBusComplete: boolean

+ CarObserver(Pane, String)
+ update(): void
+ setPlate(String): void
+ busUpdate(): void

MobileEntity

+ HEIGHT: int = 30 {readOnly}
+ MAX_INITIAL_SPEED_PIXELS_PER_STEP: int = 2 {readOnly}
+ TAG_DISTANCE_PIXELS: int = 10 {readOnly}
+ TURNING_MARGIN: int = 10 {readOnly}
+ EAST_TURNING_MARGIN: int = 200 {readOnly}
+ MILLISECONDS_PER_STEP: double = 1000. / 30 {readOnly}
+ TEXT_DISPLAY_FUDGE_FACTOR: int = 5 {readOnly}
- cityMap: CityMap {readOnly}
- location: Point2D
- stepSize: Point2D
- iconView: ImageView {readOnly}
- instanceCount: int = 0
- name: String = "Entity "+ ++in...
- nameText: Text {readOnly}
observer: Observer

+ MobileEntity(CityMap, Image)
+ addToCityMap(): void
+ chooseRandomInBoundsPoint(): Point2D
+ setName(String): void
+ getName(): String
+ getLocation(): Point2D
- setPosition(): void
+ step(): void
# checkBounds(double, double, double, double): void
+ turnRight(): void
+ isTagged(Point2D): boolean
+ taggedBy(MobileEntity): void
+ toString(): String
+ attach(Observer): void
+ detach(Observer): void
+ notifyObservers(): void

CityMap *Application*

+ NUM_TOURISTS: int = 15 {readOnly}
+ MIN_CHALLENGES_WIDTH: int = 250 {readOnly}
+ NUM_BUSSES: int = 2 {readOnly}
- backgroundImage: Image = new Image(CityM...
- overlay: Pane = new Pane()
- backgroundView: ImageView = new ImageView(b...
- challengePane: Pane
- taggables: Collection<Taggable> = new ArrayList<>()
- mobileEntities: List<MobileEntity> = new ArrayList<>()
- museums: List<Museum> = new ArrayList<>()
- mainCharacter: MobileEntity = null

+ main(String[]): void
+ addNodeToMap(Node): void
+ addTaggableToMap(Taggable): void
+ addChallengeNode(Node): void
+ getMobileEntities(): List<MobileEntity>
+ getMuseums(): List<Museum>
+ taggedBy(MobileEntity): void
+ getWidth(): double
+ getHeight(): double
+ start(Stage): void
+ getMainCharacter(): MobileEntity
- addEntities(): void

«interface»
Observer

+ update(): void
+ setPlate(String): void

MuseumObserver

- artPane: VBox
- foundArt: boolean

+ MuseumObserver(Pane)
+ update(): void
+ setPlate(String): void

Museum

+ MUSEUM_WIDTH: int = 40 {readOnly}
+ MUSEUM_HEIGHT: int = 50 {readOnly}
+ MUSEUM_LEFT_CORNER: int = 500-64 {readOnly}
+ MUSEUM_TOP_CORNER: int = 250+3 {readOnly}
- area: Rectangle {readOnly}
- observer: Observer

+ Museum(CityMap)
+ isTagged(Point2D): boolean
+ taggedBy(MobileEntity): void
+ attach(Observer): void
+ detach(Observer): void
+ notifyObserver(): void
+ toString(): String

-museums
0..*

+observer

+mainCharacter

-cityMap

Bus

+ BUS_IMAGE: Image = new Image(CityM... {readOnly}
+ LEFT_EDGE: int = 100 {readOnly}
+ RIGHT_EDGE: int = 600 {readOnly}
+ TOP_EDGE: int = 50 {readOnly}
+ BOTTOM_EDGE: int = 400 {readOnly}

+ Bus(CityMap)
# step(): void

Car

- CAR_IMAGE: Image = new Image(CityM... {readOnly}
+ MEAN_DISTANCE_BEFORE_TURN: int = 100 {readOnly}
- PLATE_LENGTH: int = 5 {readOnly}
+ NUM_LETTERS: int = 26 {readOnly}
+ POSSIBLE_NUMBER_OF_PLATES: int = (int) Math.pow(... {readOnly}
- plates: Set<String> = new HashSet<>()
- random: Random = new Random()
- plate: String

+ Car(CityMap)
- findUniquePlate(): String
- generatePlate(): String
step(): void

Person

+ SMILEY_IMAGE: Image = new Image(CityM... {readOnly}
+ PERSON_MAX_SPEED: int = 2 {readOnly}
- goal: Goal

+ Person(CityMap, Goal)
# step(): void
+ notifyObservers(MobileEntity, CityMap): void

-goal

«static»
Person::Goal

+ RADIUS: int = 100 {readOnly}
- circularGoal: Circle {readOnly}

+ Goal(CityMap)
+ getTarget(): Point2D