

# **VIRTUAL REALITY VEHICLE SIMULATOR PHASE 1**

## **FINAL PROJECT REPORT**

by

Dr. Orion Lawlor and Ben Talotta  
University of Alaska Fairbanks

Sponsorship  
PacTrans & University of Alaska Fairbanks

for

Pacific Northwest Transportation Consortium (PacTrans)  
USDOT University Transportation Center for Federal Region 10  
University of Washington  
More Hall 112, Box 352700  
Seattle, WA 98195-2700

In cooperation with U.S. Department of Transportation,  
Office of the Assistant Secretary for Research and Technology (OST-R)



## **DISCLAIMER**

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation's University Transportation Centers Program, in the interest of information exchange. The Pacific Northwest Transportation Consortium, the U.S. Government and matching sponsor assume no liability for the contents or use thereof.

**TECHNICAL REPORT DOCUMENTATION PAGE**

<b>1. Report No.</b>		<b>2. Government Accession No.</b>		<b>3. Recipient's Catalog No.</b>	
<b>4. Title and Subtitle</b> Virtual Reality Vehicle Simulator Phase 1				<b>5. Report Date</b> 2021-12-15	
				<b>6. Performing Organization Code</b>	
<b>7. Author(s) and Affiliations</b> Dr. Orion Lawlor and Ben Talotta University of Alaska Fairbanks				<b>8. Performing Organization Report No.</b>	
<b>9. Performing Organization Name and Address</b> PacTrans Pacific Northwest Transportation Consortium University Transportation Center for Federal Region 10 University of Washington More Hall 112 Seattle, WA 98195-2700				<b>10. Work Unit No. (TRAIS)</b>	
				<b>11. Contract or Grant No.</b> DTRT13-G-UTC40	
<b>12. Sponsoring Organization Name and Address</b> United States Department of Transportation Research and Innovative Technology Administration 1200 New Jersey Avenue, SE Washington, DC 20590				<b>13. Type of Report and Period Covered</b> Draft Final, 8/16/2019 - 2/15/2022	
				<b>14. Sponsoring Agency Code</b>	
<b>15. Supplementary Notes</b> Report uploaded to: <a href="http://www.pactrans.org">www.pactrans.org</a>					
<b>16. Abstract</b> Modern game simulation technology can put people in the driver's seat virtually, so they can safely experience <b>accidents</b> like skids and rollovers, and walk through various hazardous scenarios to understand and visualize the physical forces involved in accident scenarios. This project's goal is to provide a single integrated hardware/software platform capable of training operators of <b>offroad vehicles</b> such as All-Terrain Vehicles (ATVs). Because the people that would benefit most from this training, outdoor-oriented young people, are the least likely to seek it out, we " <b>gamify</b> " the training experience by building an attractive outdoor world, and putting interesting challenges in it.					
<b>17. Key Words</b> All terrain vehicles, Virtual reality, Simulation, Driver education				<b>18. Distribution Statement</b>	
<b>19. Security Classification (of this report)</b> Unclassified.		<b>20. Security Classification (of this page)</b> Unclassified.		<b>21. No. of Pages</b> 11	<b>22. Price</b> N/A

## SI\* (MODERN METRIC) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS				
Symbol	When You Know	Multiply By	To Find	Symbol
<b>LENGTH</b>				
in	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
<b>AREA</b>				
in <sup>2</sup>	square inches	645.2	square millimeters	mm <sup>2</sup>
ft <sup>2</sup>	square feet	0.093	square meters	m <sup>2</sup>
yd <sup>2</sup>	square yard	0.836	square meters	m <sup>2</sup>
ac	acres	0.405	hectares	ha
mi <sup>2</sup>	square miles	2.59	square kilometers	km <sup>2</sup>
<b>VOLUME</b>				
fl oz	fluid ounces	29.57	milliliters	mL
gal	gallons	3.785	liters	L
ft <sup>3</sup>	cubic feet	0.028	cubic meters	m <sup>3</sup>
yd <sup>3</sup>	cubic yards	0.765	cubic meters	m <sup>3</sup>
NOTE: volumes greater than 1000 L shall be shown in m <sup>3</sup>				
<b>MASS</b>				
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2000 lb)	0.907	megagrams (or "metric ton")	Mg (or "t")
<b>TEMPERATURE (exact degrees)</b>				
°F	Fahrenheit	5 (F-32)/9 or (F-32)/1.8	Celsius	°C
<b>ILLUMINATION</b>				
fc	foot-candles	10.76	lux	lx
fl	foot-Lamberts	3.426	candela/m <sup>2</sup>	cd/m <sup>2</sup>
<b>FORCE and PRESSURE or STRESS</b>				
lbf	poundforce	4.45	newtons	N
lbf/in <sup>2</sup>	poundforce per square inch	6.89	kilopascals	kPa
APPROXIMATE CONVERSIONS FROM SI UNITS				
Symbol	When You Know	Multiply By	To Find	Symbol
<b>LENGTH</b>				
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft
m	meters	1.09	yards	yd
km	kilometers	0.621	miles	mi
<b>AREA</b>				
mm <sup>2</sup>	square millimeters	0.0016	square inches	in <sup>2</sup>
m <sup>2</sup>	square meters	10.764	square feet	ft <sup>2</sup>
m <sup>2</sup>	square meters	1.195	square yards	yd <sup>2</sup>
ha	hectares	2.47	acres	ac
km <sup>2</sup>	square kilometers	0.386	square miles	mi <sup>2</sup>
<b>VOLUME</b>				
mL	milliliters	0.034	fluid ounces	fl oz
L	liters	0.264	gallons	gal
m <sup>3</sup>	cubic meters	35.314	cubic feet	ft <sup>3</sup>
m <sup>3</sup>	cubic meters	1.307	cubic yards	yd <sup>3</sup>
<b>MASS</b>				
g	grams	0.035	ounces	oz
kg	kilograms	2.202	pounds	lb
Mg (or "t")	megagrams (or "metric ton")	1.103	short tons (2000 lb)	T
<b>TEMPERATURE (exact degrees)</b>				
°C	Celsius	1.8C+32	Fahrenheit	°F
<b>ILLUMINATION</b>				
lx	lux	0.0929	foot-candles	fc
cd/m <sup>2</sup>	candela/m <sup>2</sup>	0.2919	foot-Lamberts	fl
<b>FORCE and PRESSURE or STRESS</b>				
N	newtons	0.225	poundforce	lbf
kPa	kilopascals	0.145	poundforce per square inch	lbf/in <sup>2</sup>
*SI is the symbol for the International System of Units. Appropriate rounding should be made to comply with Section 4 of ASTM E380. (Revised March 2003)				

## TABLE OF CONTENTS

Disclaimer.....	i
Technical Report Documentation Page .....	ii
SI* (Modern Metric) Conversion Factors.....	iii
List of Figures .....	v
List of Tables .....	vi
Acknowledgments.....	vii
Executive Summary.....	1
CHAPTER 1. Virtual Reality Vehicle Simulator .....	2
1.1. Introduction .....	2
1.2. Literature Review.....	2
1.3. Vehicle Simulation Approach.....	2
1.3.1. Tire Simulation .....	3
1.3.2. Force Visualization .....	5
1.3.3. Rewind after Accidents .....	6
1.3.4. Simulation Graphics .....	6
1.4. Driver Training Scenarios .....	6
1.4.1. Scenario: Drive to the lake .....	8
1.4.2. Scenario: Park in the truckbed.....	8
1.4.3. Scenario: Take friends up the mountain.....	8
1.4.4. Scenario: Jump the house .....	9
1.4.5. Scenario: Mountain pass.....	9
1.4.6. Scenario: Racing.....	9
1.5. Initial Findings .....	10
1.5.1. WebGL Deployment.....	10
1.5.2. VR Deployment Challenges.....	10
1.6. Conclusions and Future Work.....	11
CHAPTER 2. References .....	13

## LIST OF FIGURES

Figure 1: Our vehicle simulator, after beginning the “Park in the truckbed” scenario .....	1
Figure 2: Keyboard controls for the WebGL or desktop version of the simulator. ....	3
Figure 3: An early prototype of a VR input device modeled after ATV handlebars.....	3
Figure 4: Glowing lines show how each wheel is modelled as a spherical collider on a suspension. ....	4
Figure 5: Current force on each wheel is shown in red, and net force on the vehicle in green.....	5
Figure 6: Early CAD design for a "bucking bronco" style tilt-pitch ATV simulator .....	11

## LIST OF TABLES

Table 1: Six driver training scenarios .....	7
--	---

## ACKNOWLEDGMENTS

Major software contributions have been made, in chronological order, by:

- Dr. Orion Lawlor
- Dr. Jonathan Metzgar
- Kim Fairbanks
- Curtis Fortenberry
- Chris McClure
- Jacob Guenther
- Dylan Palmieri
- Tailon Russell
- Gregory Talotta
- Ben Talotta

We sincerely thank all our developers, many of whom volunteered to work on this project as part of their computer science capstone course. We also thank our many play testers, who freely gave their time to help improve our software.



## EXECUTIVE SUMMARY

Modern game simulation technology can put people in the driver's seat virtually, so they can safely experience accidents like skids and rollovers, and walk through various hazardous scenarios to understand and visualize the physical forces involved in accident scenarios.

This project's goal is to provide a single integrated hardware/software platform capable of training operators of offroad vehicles such as All-Terrain Vehicles (ATVs). Due to the fact that the people that would benefit most from this training, outdoor-oriented young people, are the least likely to seek it out, we "gamify" the training experience by building an attractive outdoor world and putting interesting challenges in it.

ATVs in Alaska, like in the rest of America, are used for recreation, hunting, and fishing. However, in rural Alaska especially off the road system, ATVs are also used for everyday transportation. Hundreds of Alaskans have died on ATVs in the past few decades, the second-highest ATV fatality rate in the nation (CPSC, 2017).

Most offroad accidents involve losing control due to vehicle dynamic forces. To help people understand and visualize these forces, our simulator shows the forces on the ATV tires as they act against the vehicle's inertia while driving.



Figure 1: Our vehicle simulator, after beginning the "Park in the truckbed" scenario

## CHAPTER 1. VIRTUAL REALITY VEHICLE SIMULATOR

### 1.1. Introduction

This project's goal is to build a software simulator platform capable of training operators of offroad vehicles such as All-Terrain Vehicles (ATVs).

Despite accounting for 17% of America's land area, Alaska has less than 0.4% of America's lane-miles of highway. Although Alaska's major cities have paved highways, most of rural Alaska's transport is non-traditional, with long haul travel occurring by airplane or boat, and local trips taken via ATV or snowmobile. Further, Alaska's trails are often irregularly maintained, terrain and weather can be challenging, and unpredictable events such as wild animal encounters are common. ATVs in Alaska, like in the rest of America, are used for recreation, hunting, and fishing. However, in rural Alaska, especially off the road system, ATVs are also used for everyday transportation. Because of these factors, over 200 Alaskans have died on ATVs since 1980, the second-highest ATV fatality rate in the nation (CPSC, 2017).

### 1.2. Literature Review

Long-term research has indicated that simulator-based driver training can reduce unsafe driving, as indicated by a 5% lower driving infraction rate for students exposed to a few hours of simulator-based training compared to conventional lecture-type driver training (Hirsch & Bellavance, 2017). A history of interactive driving simulation experiences has been summarized by Bouchner (Bouchner, 2016).

A huge variety of simulators have been developed over the past decades, ranging from simple onscreen simulations to immersive 6DOF motion simulators. Unlike most prior work, we target off-road operation, where speeds are low but single-vehicle accidents are common due to hazards from speed, terrain, and vehicle dynamics.

### 1.3. Vehicle Simulation Approach

Our vehicle simulator is built using the Unity game engine (Unity, 2021), which supports integrated physics and high-quality graphical effects on a wide range of platforms, from WebGL that can run in a web browser through VR applications on high-end gaming hardware. Crucially, the same simulation and graphics assets can be applied to a variety of scripted and open-world interactions, and can be reused and repurposed for different training scenarios and situations.

Unity allows the same graphical content, scenes, and scenarios to be run on a variety of computers and operating systems:

- Unity supports **VR** via the SteamVR, Oculus, or Hololens interfaces, which mainly target Windows PCs.
- Unity supports **WebGL** to allow content to run inside a web browser, and it runs well in Chrome or Firefox, and is listed as supporting Edge and Safari. Running WebGL requires no downloads or installation, so we recommend it for wide distribution of our content.
- Unity supports packaging the content for a variety of platforms including Windows (7+), MacOS (10.12+), Linux (16.04+), Android (4.1+), iOS (10+), PlayStation 4+, Xbox One+, Stadia, and Nintendo Switch.

For the WebGL or desktop versions, the keyboard controls are shown below.

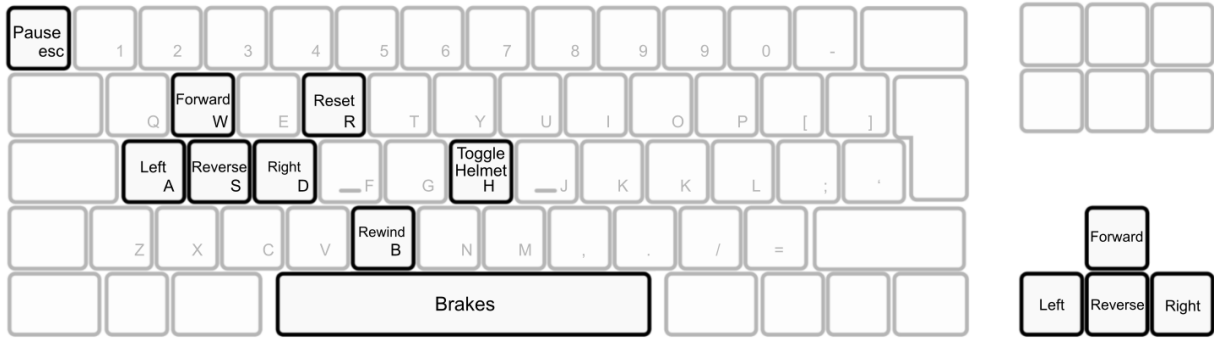


Figure 2: Keyboard controls for the WebGL or desktop version of the simulator.

For the VR version, we use the Vive hand mounted to a steel frame, which allows the right controller's analog trigger to act as a thumb throttle, and the central controller's analog trigger to be linked to the brake cable. The relative rotation of the two controllers determines the angle of the handlebars for steering, and can be extracted from the Vive's pose tracking of the controllers. This VR control scheme allows analog inputs, making driving smoother than the on/off keyboard control system used for the WebGL version.

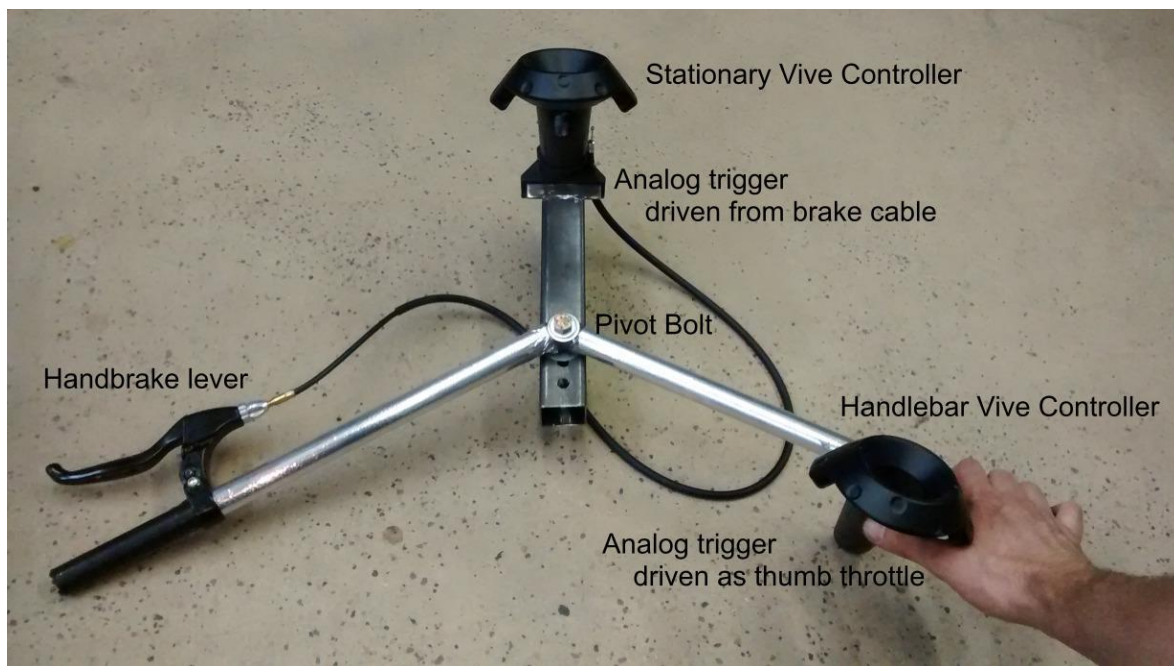


Figure 3: An early prototype of a VR input device modeled after ATV handlebars

### 1.3.1. Tire Simulation

Interactive on-road vehicle simulators often use a very simple approximation of the tire road interaction, such as a simple raycast downwards to find the tire-road contact patch. This is done because at highway speeds, tires are often spinning at tens of revolutions per second, so at a default Unity simulation physics timestep of 50Hz it is difficult to track the moving contact patch accurately. We found the standard Unity WheelCollider, while producing reasonable results for smooth surfaces, on unlevel

terrain such as rocks produced jerky and visibly nonphysical results—it is designed for high speed use on paved surfaces only.

At the lower speeds common in off-road travel, and particularly in the rough trails common in rural Alaska, we were able to use a rolling sphere collider to approximate the tire-terrain interaction for each wheel, which gave much more reasonable behavior when driving over rough surfaces. This also allows us to accelerate by making the motor apply torque to the tires, and brake by slowing down the tires, which results in more accurate vehicle motion even in exceptional circumstances, such as with some wheels off the ground.



Figure 4: Glowing lines show how each wheel is modelled as a spherical collider on a suspension.

Each wheel is simulated with fully independent suspension, implemented in Unity using a ConfigurableJoint per wheel allowing limited Y motion (suspension) and free X rotation (wheel rotation around the axle). Steering is implemented as a parent transform with a HingeJoint, and changing the axis of wheel rotation naturally changes the direction of travel of the vehicle. We have not yet added suspension features like sway bars, hence modeling a basic off-road vehicle configuration.

Terrain-tire friction forces are modeled using a simple static/dynamic friction model, which allows tires to begin to skid, particularly at high speeds. This seems to be accurate enough to capture effects such as steering into a skid to regain control, while still working on slopes and arbitrary surfaces. A more sophisticated terrain-tire model such as (Mancardi, 2020) would incorporate terramechanics effects such as soil compaction, but we have not yet investigated adding these. Incorporating off-road vs on-road tire friction might allow training on the hazards of transitioning between paved and unpaved driving.

This overall vehicle simulation approach has proven flexible enough to handle several off-road vehicle variants during testing, and we have actually used the same code to simulate a 6-wheel lunar mining robot, so the approach seems to be generic enough to be useful for multiple vehicles.

### 1.3.2. Force Visualization

A major goal of the simulation is to allow the operator to develop some intuition for how braking and steering inputs translate to vehicle motion and tire forces, and hence overall vehicle behavior like skids and rollovers.



Figure 5: Current force on each wheel is shown in red, and net force on the vehicle in green

The current dynamic force on each wheel is displayed in red, starting from the wheel's contact patch with the ground. Note how above, the vehicle is making a sharp left turn, so the contact patches are pushing back much harder from the right-side tires. Driving slightly faster or turning slightly more sharply will lift the left tires off the ground, the first step in an overturn or rollover accident.

These contact patch forces are extracted live, in realtime, directly from the simulation's ConfigurableJoint that attaches the simulated tire to the frame. The ConfigurableJoint's currentForce vector hence operates like a strain gauge on the axle. This means they incorporate live effects from both the physics and dynamics of the vehicle and environment, which we believe is a novel approach. We display this wheel net force in real time using a red 3D overlay vector, starting from the contact patch.

We display the net force on the vehicle as a green 3D overlay vector, starting from the center of mass, and oriented so it visually opposes the wheel forces. The objective here is to illustrate how a vehicle's inertia is opposed by the forces on the tire contact patches, which is what creates the turning forces that result in rollover accidents.

### **1.3.3. *Rewind after Accidents***

Poor driving in many offroad situations, particularly on slopes, will result in a rollover, which is one of the most life-threatening ATV accident types (Van Ee et al, 2014). We added an automatic "rewind" feature that after a rollover, rewinds time up to five seconds to give the operator a chance to see what went wrong, and try a different approach to prevent the accident.

The simulator detects a rollover by comparing the vehicle up vector against the world up vector, and if they're more than 90 degrees apart, the vehicle has rolled over. The time rewind keeps a rolling history buffer with the position, rotation, and velocity of each body component (chassis and each tire), and restores these if a rollover has been detected.

### **1.3.4. *Simulation Graphics***

We have invested significant effort in building a beautiful naturalistic world, since our target audience enjoys outdoor activities. However, for good performance on WebGL or VR hardware, we have had to limit the geometric complexity of objects. Foliage like trees and ground cover dominates the graphics budgets for polygons and fillrate unless these are kept very sparse, so we portray a high mountain area near the treeline, with a few sparse birch trees. We use a simple friendly cartoon style for the vehicles, again to stay in the polygon budget.

One way we mask the realism / rendering budget tradeoff is to use geometrically simple but richly textured natural surfaces, and we use the Unity terrain paint feature to provide a multi-layered ground texture that smoothly transitions between paths and biomes.


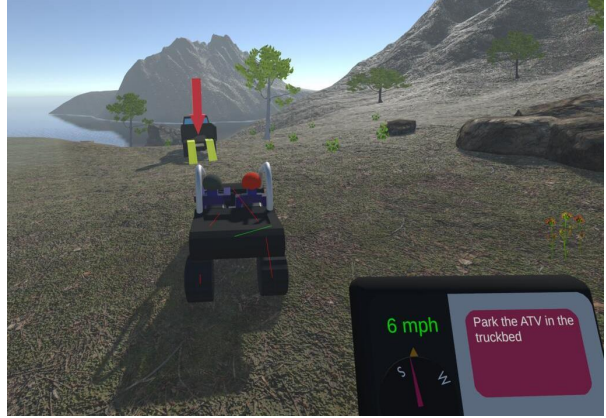



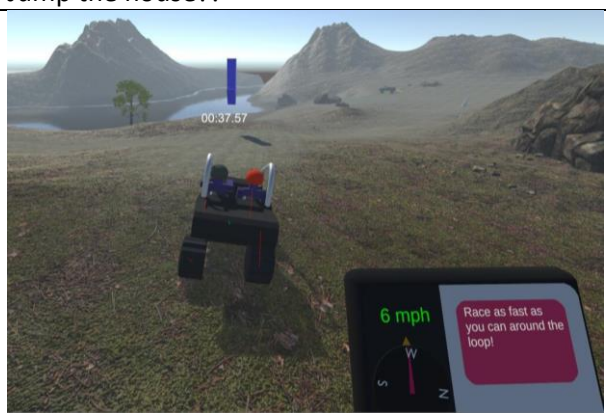
To keep the user oriented in the world, we provide a rotating compass on a virtual "smartphone" in the bottom right corner of the screen. This also displays a color-coded speed readout, and any messages for active scenarios.

We limit ourselves to a polygon budget of about 1 million triangles per frame, which results in good performance on recent hardware. There are about 700 batches, which reflects the many object types (terrain, water, vegetation, rocks) in the world.

## **1.4. Driver Training Scenarios**

We chose the driver training scenarios to match common accident types, such as overturns and collisions (CPSC, 2020), and passengers carried unsafely, as well as address common underlying causes such as excessive speed and lack of situational awareness.

Table 1: Six driver training scenarios

 <p>19 mph Drive to the lake</p>	 <p>6 mph Park the ATV in the truckbed</p>
 <p>8 mph Take your two friends up the mountain. Don't let your friend in the back fall out!</p>	 <p>17 mph Jump the house?! Looks fun!</p>
 <p>7 mph Navigate the mountain pass</p>	 <p>00:37.57 6 mph Race as fast as you can around the loop!</p>

We designed the simulation as an open world, so the user begins a scenario by choosing to drive through a floating white arrow, which are scattered around the map. Users can choose to complete the scenarios in any order. We then explain the scenario to the user via text message to their virtual phone, and display the scenario target with a floating red arrow, which also shows up in their compass.

#### **1.4.1. Scenario: Drive to the lake**

Scenario summary:

- Start location: just south of vehicle start point.
- End location: on the shore of the south lake.
- Intent: introduce users to the scenario system and vehicle dynamics.
- Challenges: the steep downslope before the lake shore can cause a skid.
- Skills developed: speed management, slope handling.

A recurring theme in these scenarios is that even simple tasks require care and attention while driving. This simple starter scenario, arranged to be the first one the user encounters, just requires the user to drive down to a lake shore. However, there is a steep slope near the end that makes it easy to lose control or end up in the lake, unless the driver keeps their speed under control.

#### **1.4.2. Scenario: Park in the truckbed**

Scenario summary:

- Start location: between the lake and the village.
- End location: inside the bed of a large truck.
- Intent: illustrate some hazards of driving a vehicle up into a truckbed.
- Challenges: avoiding driving off the ramps, overshooting the speed.
- Skills developed: precision steering, speed management.

The user is asked to drive up two ramps into the bed of a large truck, which can result in falling off the ramps unless the user steers carefully, or ramming the back of the pickup unless the user controls their speed.

#### **1.4.3. Scenario: Take friends up the mountain**

Scenario summary:

- Start location: at the base of the mountain, just southeast of the village.
- End location: up the mountain, at the start of the mountain pass.
- Intent: illustrate some hazards of unsecured passengers (in any vehicle).
- Challenges: minimizing vehicle tilts that will throw the passenger out.
- Skills developed: speed management, not accepting unsafe passengers.

This scenario requires the user to carry a third passenger unsafely in the bed of the UTV. Sharply accelerating, braking, or turning will throw the passenger out the back of the bed.

This is designed to, first, visibly demonstrate how easy it is for people to fall off the back of an ATV or UTV. We include a second passenger seated safely inside the UTV to illustrate how much safer this is



than clinging to the back. Second, losing a passenger forces the scenario to restart from the base of the mountain, which after a few tries becomes frustrating—why don't they get their own vehicle, or find a safe seat?

#### **1.4.4. Scenario: Jump the house**

Scenario summary:

- Start location: west of the village, by a ramp in front of a house.
- End location: just east of the village, beyond the house.
- Intent: illustrate the hazards of jumps.
- Challenges: the ramp isn't high enough to actually jump the house, but is high enough to easily roll the vehicle or destroy the house.
- Skills developed: not attempting stunts like jumps.

This scenario has a ramp in front of a small house, and a prompt to "Jump the house?!". It's clearly a bad idea to try to jump over a house, and trying this results in either bouncing off the side of the house, or demolishing the house. The only safe approach is to drive around the house and "Leave the tricks to professionals."

The objective of this scenario is to give users experience with rejecting bad suggestions.

#### **1.4.5. Scenario: Mountain pass**

Scenario summary:

- Start location: east of the village, up the mountain.
- End location: north of the village.
- Intent: illustrate some hazards of steep winding rocky paths.
- Challenges: the pass has several steep slopes and poorly banked turns.
- Skills developed: speed management, operating vehicle on cross slopes.

This scenario takes the user across a high mountain pass, including steep slopes and rocky terrain that makes it easy to lose control, particularly at high speed, but also enjoy some nice terrain and long views.

#### **1.4.6. Scenario: Racing**

Scenario summary:

- Start location: southeast corner, beyond the lake.
- End location: same as start location, after a series of waypoints.
- Intent: illustrate some hazards of trying to maximize speed at the expense of control.
- Challenges: most of the waypoints are in front of a hazard like rocks or cliffs.
- Skills developed: speed management, looking ahead at the upcoming terrain.

This scenario prompts the user to drive between a set of waypoints "as fast as possible", with a frenetic onscreen timer counting their lap time. However, we placed some waypoints directly in front of hazards like rocks or cliffs, so zipping through the waypoint full throttle will cause an accident, costing much more time than properly managing speed.

## **1.5. Initial Findings**

We have begun initial informal user testing, to aid the development process. We aim to perform more systematic user testing in the remaining months of the project.

### **1.5.1. WebGL Deployment**

Our initial user testing resulted in new users being impressed with the simulated ATV, but confused by the mechanics of starting scenarios and what was happening when they would rewind after a crash. In response to this, we added the "smartphone" interface to remind users of the current scenario target, and to point at where to go. We also added a rewinding indicator that pops up on the screen while rewinding time after a crash.

Subsequent rounds of testing revealed the potential for us to improve usability of the smartphone feature, as well as pointed out some flaws in our driving controls. We were able to find and fix an error in how the vehicles brakes work that was unnaturally stabilizing the ATV.

### **1.5.2. VR Deployment Challenges**

For development we used an HTC Vive Pro VR headset (Vive, 2021), which works well with the SteamVR Unity package. Our graphical assets are attractive and performant in VR, giving a smooth experience, and our vehicle simulator also scales nicely to VR.

However, we encountered two showstopper challenges in using VR as our default simulator interface. First, simulator sickness in VR is a persistent challenge, particularly when turning the ATV, which in VR results in the world spinning around the driver. Modern VR hardware uses several hardware mitigations for simulator sickness (Watson et al, 2019), but this effect was still much worse than anticipated in a moving vehicle, particularly on uneven ground. We experimented with a number of mitigations for this:

1. Leaving a VR driver's orientation locked to an ATV that tilts down a slope also tilts the horizon, resulting in severe simulator sickness. Removing the pitch and tilt components from vehicle orientation works like a virtual self-stabilizing driver's seat, and reduces simulator sickness on sloped or rough surfaces, but the yaw component of rotation remains.
2. Adding driver-locked dark high-contrast overhead rollbars, locked to the stabilized virtual driver's seat, provides enough peripheral vision motion cues to reduce simulator sickness during turns, but still does not eliminate it.
3. We began experimenting with a motorized rotating "simulator chair" device that would let us match the user's physical rotation to the virtual vehicle's simulated motion, but this sort of hardware is complex to develop to an operational state, and there are several potential pinch and tipping hazards that made it difficult to deploy safely. An early design for a "bucking bronco" style tilt-pitch ATV simulator, including a pinch-resistant foot tub did not include yaw motion, which proved to be much more important for avoiding simulator sickness.

Despite these mitigations, turning the ATV still results in simulator sickness for more sensitive developers (including the PI!), which slowed progress on a deployable VR version.

Finally, COVID protection protocols have made it difficult for us to share VR hardware even between the development team, and difficulties reliably sanitizing VR headsets and handlebars precluded the widespread use of VR for driver training. Pre COVID, we had planned on doing VR simulator "roadshow"

events with local schools and community organizations, but we have switched to WebGL for the time being.

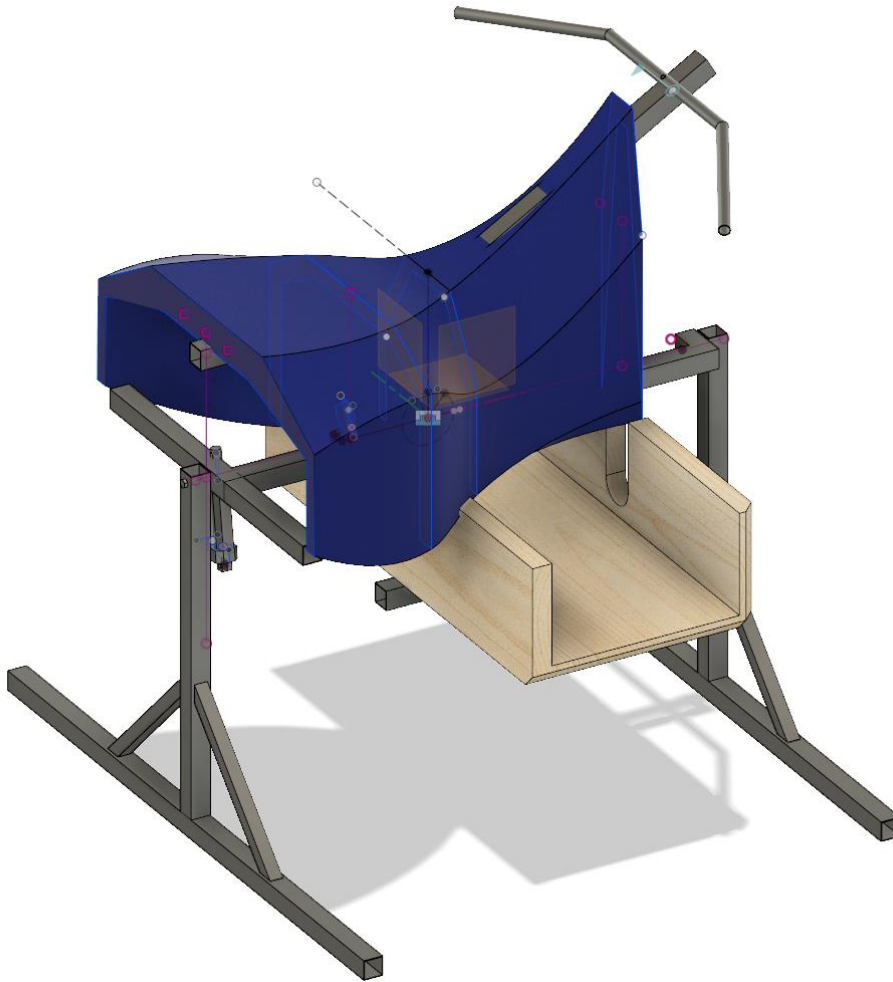


Figure 6: Early CAD design for a "bucking bronco" style tilt-pitch ATV simulator

## 1.6. Conclusions and Future Work

We have built a working offroad vehicle simulator and several driver training scenarios. The functionality described above is all playable now in a web browser using WebGL here:

[https://olawlor.github.io/AK\\_ATV/](https://olawlor.github.io/AK_ATV/)

The simulator's Unity code and graphical assets are available under a permissive open source license here:

[https://github.com/olawlor/AK\\_ATV/](https://github.com/olawlor/AK_ATV/)

To build the simulator, install Unity 19.04, clone this repository, add “AK\_ATV / AK\_ATV\_Simulator” to Unity projects, and open “Assets / Scenes / WebGL / procedural\_terrain.unity” for the full world. The VR support hardware CAD and 3D printable files are in “AK\_ATV / parts / vive\_controller\_mounts”.

We will continue working on the WebGL version for its capacity to easily reach many more people across Alaska, but as we learn to live with COVID we hope to extend our VR hardware and software to provide a richer and more immersive experience. There are several promising approaches we hope to examine to address the simulator sickness issues discovered during VR testing:

- A third person camera following behind the vehicle would allow the vehicle to turn without rotating the operator, which we want to evaluate to see if this improves simulator sickness. This may also provide a more holistic overview of the vehicle operations than a more conventional vehicle simulator’s first person (driver’s seat) viewpoint. As the vehicle turns, we can do “blink motion” or field of view adjustments, which have been shown to reduce simulator sickness (Fernandes & Feiner, 2016).
- A powered rotating chair would physically turn the operator when the virtual vehicle turns, which should improve simulator sickness if we can match the physical and virtual motions well. To avoid injuring operators, this should begin with a safe and well-tested office chair, and needs to be designed and fabricated carefully to avoid possible pinch points and entanglement hazards.

In addition to VR support as discussed above, we'd like to extend the simulation with:

1. More gamification, in particular with additional rewards for safe driving, such as earning a row of icons for crash-free scenario completion, and rewarding safe drivers with additional simulated vehicles such as a dirtbike.
2. Better sound, with both sliding tire sounds to help indicate skid onset, vehicle engine sounds to better connect the operator with the vehicle, and environment sounds like wind to make the simulation feel more immersive.
3. Better graphics, with more interactive graphics like dust blowing behind the vehicle, additional dust during skids and high wheelslip.

We look forward to continuing to improve this simulator.

## CHAPTER 2. REFERENCES

- (Bouchner, 2016) Petr Bouchner. 2016. Interactive driving simulators: history, design, and their utilization in area of HMI research. *International Journal of Systems Applications, Engineering & Development* 10 (2016): 179-188.
- (CPSC, 2017) Consumer Product Safety Commission. "Death Associated With ATVs by State." Data through December 2017. Retrieved December 2021 from <https://www.cpsc.gov/Safety-Education/Safety-Education-Centers/ATV-Safety-Information-Center/Death-Associated-With-ATVs-by-State->
- (CPSC, 2020) Consumer Product Safety Commission. "2020 Report of Deaths and Injuries Involving Off-Highway Vehicles with More than Two Wheels." Retrieved January 2022 from <https://www.cpsc.gov/s3fs-public/2020-Report-of-Deathsand-Injuries-Invovling-Off-HighwayVehicles.pdf>
- (Fernandes & Feiner, 2016) Ajoy S. Fernandes and Steven K. Feiner. "Combating VR sickness through subtle dynamic field-of-view modification." In *2016 IEEE symposium on 3D user interfaces (3DUI)*, pp. 201-210. IEEE, 2016.
- (Hirsch & Bellavance, 2017) Pierro Hirsch and François Bellavance. 2017. "Transfer of Skills Learned on a Driving Simulator to On-Road Driving Behavior." *Transportation Research Record: Journal of the Transportation Research Board*: 2660. 1-6. DOI: <https://doi.org/10.3141/2660-01>
- (Mancardi, 2020) Lorenzo Mancardi. "Implementation of an off-road tyre model for real-time dynamic driving simulator." Politecnico di Torino Master's Thesis, 2019/2020. Retrieved December 2021 from <https://webthesis.biblio.polito.it/16888/1/tesi.pdf>
- (Unity, 2021) Unity Technologies. 2021. *Unity 3D Engine*. Retrieved December 2021 from <http://www.unity3d.com/>
- (Van Ee et al, 2014) Van Ee, C. A., D. E. Toomey, B. A. Moroski-Browne, M. Vander Roest, and A. Wilson. "ATV rollover, rider response, and determinants of injury: in-depth analysis of video-documented ATV rollover events." *Traffic injury prevention* 15 (2014): S190-S196.
- (Vive, 2021) HTC Corporation. 2021. VIVE Pro. Retrieved December 2021 from <https://www.vive.com/us/product/vive-pro/>
- (Watson et al, 2019) Benjamin Watson, Rachit Shrivastava, and Ajinkya Gavane. 2019. The Effects of Adaptive Synchronization on Performance and Experience in Gameplay. *Proc. ACM Comput. Graph. Interact. Tech.* 2, 1, Article 5 (June 2019), 13 pages. DOI: <https://doi.org/10.1145/3320286>