# Deliverable 1

Project Idea Table (Summary)

Project Ideas	Description
Arya Khosravi: Music App	App where the user can create music out of sounds, audio files and modify the audio samples
Theodore Georgiou: Car simulator	Realistic car physics simulator that allows for 2 player races and physics values to be displayed
Alexander Little: Sign Language Translator	Translate sign language in real-time using the user's camera.
Ahmet Yusuf Yildirim: 3D Terrain Generation	Using DEM maps to create an interactable 3D terrain

Car simulator

Speed: 52.07008
RPM: 4930.89
Gear: 3
LocalAcceleration: (14.0, 28.5)
Acceleration: (-17.1, 26.8)
LocalVelocity: (10.5, -10.2)
Velocity: (13.9, 4.3)
SteerAngle: 0.5606768
Throttle: 1
Brake: 0
HeadingAngle: 1.098983
AngularVelocity: 3.682638
TireFL Weight: 8335.84
TireFR Weight: 8455.688
TireRL Weight: 6259.312
TireRR Weight: 6379.16
TireFL Friction: 10445.95
TireFR Friction: 10596.14
TireRL Friction: 15648.28
TireRR Friction: 15947.9
TireFL Grip: 2.5
TireFR Grip: 2.5
TireRL Grip: 2.5
TireRR Grip: 2.5
AxleF SlipAngle: -0.2506275
AxleR SlipAngle: -1.042562
AxleF Torque: 0
AxleR Torque: 21387.22

- Involves physics learned in Mechanics (friction, torque, etc.)
- Users can control the movement of a car on a selected track
- The energy of the car will be displayed
- The user can select type of vehicle
- Two users can drive at the same time and the best time can be saved
- HUD showing speed and other important physics values can be displayed
- Simulates engine performance (horsepower and torque)
- Gearbox simulation (automatic/manual)
- Basic aero physics (downforce for race cars)
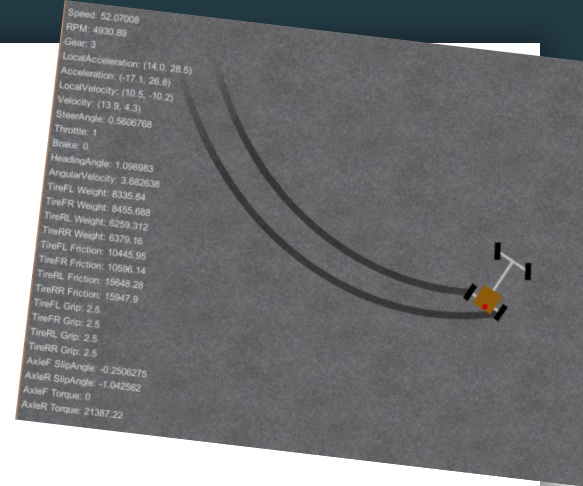
Sign Language Translator
(Image Recognition)

- Involves complex mathematical and computational concepts (image processing, neural networks, pattern recognition)
- Uses machine learning to recognize and interpret sign language gestures in real time
- Displays the translated text or speak verbally using text-to-speech
- Can incorporate physics of motion tracking to improve hand gesture detection
- Would use OpenCV library as building the program from scratch is not feasible
  - However, OpenCV already provides too many features.

3D Mesh creator based on Google Maps DEM (Digital Elevation Model)

- Using multiple APIs to get the DEM map of a chosen area on earth (Google maps)
- Related to **geophysics** for studying **earthquakes, erosion, and gravity variations**.
- Analyzing the DEM map pixel by pixel until we have a 2D array that contains the height of each pixel
- Connecting these points with the help of an algorithm and creating faces between those points (Dijkstra or A*).
- Generating 3D elements such as rivers (fluids) and trees.
- Generating shadows based on the light source (sun).
- Allowing user interaction with the terrain (like Blender meshes).
- Using bezier curves to smooth out the model.
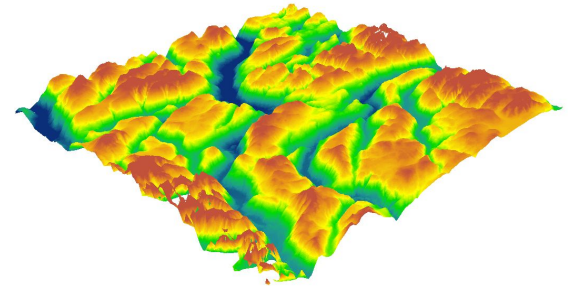
Not chosen due to complexity

Chosen Idea: Music Creation App

- Involves using complex fourier transformation algorithms to deconstruct each sound wave in a more complex sound wave
- Use of different filtering algorithms to vary audio pitch, modify audio files, equalizers and amplitude
- Implements different sound waves such as sin waves, square waves, saw waves, etc
- Used to make music by ordinary people and professionals
- GUI contains the audio files and tracks, as well as system usage, visualizers and a file management system
- Hardware integration for midi keyboards and regular keyboards
- Ability to record your own audio in app

Our choice and why

Why? Audio processing/modification is relatively easy with fft (fast fourier transform) algorithms, as such is the algorithms for eq management, pitch adjust and other functions. Implementing such algorithms should take no more than 2 weeks max for each one.

Adding keyboard and audio device implementation is also easy, with or without libraries. This and a live audio recording system will allow seamless audio integration into a music track. This should take less than a week in total.

The GUI is quite simple as well. The most difficult part is adding a snap feature and creating a distinct GUI from other music apps.

Audio cutting and modification is also easy considering it consists of removing data points from the imported file.

All in all, the project is more feasible than the other options for its simplicity and its uniqueness.

Inputs/Outputs

Inputs: computer keyboard, midi controllers (maybe), mouse, audio files, live audio recording, synthesized sounds, instruments. Inputs will be done with sliders, menu buttons, knobs and custom GUI elements

Outputs: music files/saved files, visualization, modified audio files (voice changer), stereo or mono output

Individual Parts

Parts that need to be implemented:

-Note Placement, Multiple Tracks, Audio Channels, Audio Synthesizer, BPM, Different Instruments, Custom Audio Files, Audio Recording, Pitch Modification, Sound Equalizer, Audio File Manipulation, 3D Visualizer

At this time, the task distribution is not set. We plan on splitting the project up in quarters (subject to change). This will be done in Deliverable 2.

Bibliography

Fl Studio Picture (by PCMag): https://www.pcmag.com/reviews/image-line-fl-studio

DEM map Picture (by GISGeography): https://gisgeography.com/free-global-dem-data-sources/

2D car simulator (by jongallat): https://github.com/jongallant/CarSimulator