CPSC 386: Introduction to Game Design and Production - Spring 2020

Final Project, **Crossy Road,** due Sunday, 13 May 2020

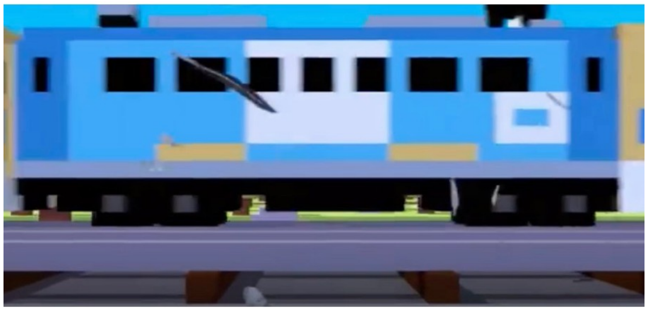
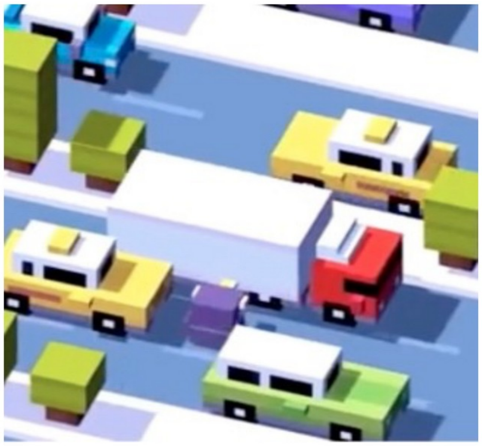
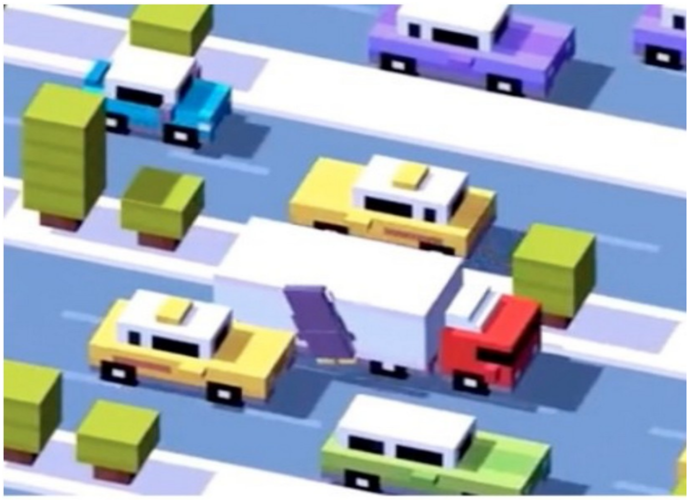
Using Unreal 4 and Magica Voxel (and other tools such as Blender if you wish), your job is to recreate the classic Crossy Road game.

The characters (chicken, eagle), vehicles (cars, trucks, trains), safe areas (grass), obstacles to cross (rivers, highways, and train tracks with crossing arms), landscape features (trees, bushes, logs, lily pads), and coins must all be created with a static mesh editing tool like MagicaVoxel.

These assets and their textures are imported into Unreal, scaled, and rotated, and spawned dynamically as the game progresses.

Blueprints containing the code to generate the game continuously, and control the movement, creation, destruction, lighting, sounds, and so forth are all written by you to mimic the game as closely as possible. Ray tracing, shadowing, sound rendering, … are all generated by the Unreal Engine.

It is recommended that your computer be as fast as possible to handle Unreal 4 (16 GB RAM+, with a fast video graphics card, and a very fast CPU). If your laptop doesn't measure up, many students have successfully connected to their desktop gaming computer at home using Chrome Remote Desktop (installed both on their desktop and their laptop).

	
 Trying to jump between the cars and the truck...	 … failed… squashed on the side of the truck.

OBJECT OF THE GAME:

Live as long as possible. There is only one level, but it is dynamic and keeps being created in front of you, and destroyed behind you.

The level is made up of alternating safe areas and obstacles with either vehicles that can strike you, or water you can fall into. You can pause and wait for a safe interval to cross the obstacles -- but not too long, or an eagle will swoop down and pick you off. Unlike the Lord of the Rings or the Hobbit, eagles are not good guys in this game.

Safe areas are grass w/ trees, bushes, and rocks). Obstacles are roads w/ vehicles, RR's w/ trains, or rivers w/ water (unsafe) and moving logs (temporarily safe)).

You can move forwards/right/left always, and backwards only within a single safe area/obstacle. (However, most safe areas are only one jump wide.)

The game designer must create the obstacles with moving cars/trucks, trains, or logs so that it is always possible for the player to get through if they are clever enough. This means that cars/trucks/trains must be timed, and trees/bushes/ stumps/rocks must be placed properly.

The actor can be killed by either standing still or failing to move forward for too long (an eagle swoops down), or by being run over, or by running into the side of a car/truck/train, or by falling into a river.

The special effects that occur are as follows:

- running into the side of a car or ruck: flattened version of voxel
- being run over by car or truck: flattened front of voxel on the highway
- being run over by/running into the side of an oil truck: explosion in three/four colors: bright, white, large, explosion, then a particle system with the actor’s color, then yellow, then red (see images)
- being run over by/running into the side of a train: explosion of particles of the actor’s colors
- falling into water: exploding water particles

To create your game, you must...

Create your assets for Unreal Engine, using MagicaVoxel…
Chicken, Cars, Trucks, Trains, Logs, River, Grass, Highway, RR tracks, Trees, Rocks, RR crossings, Lily pads, Coins, Eagle, and Start-screen Logo assets using a Voxel editor such as MagicaVoxel.

Export them as obj files.

Import them into Unreal 4 (rotating them around the x axis by 90 degrees, and scaling them by about 100).

Create Blueprints that…

Move the chicken Forward, backward, left, and right, having him crouch, jump, and rotate.

Create spawners (Lane spawner, grass spawner, highway spawner, RR spawner, and river spawner), that dynamically create the world as the chicken moves forward, and destroys the lanes behind it.

The grass spawner must spawn trees, bushes, and rocks, making sure there are many of these at the right and left ends of the lanes, to prevent the chicken from going too far left and right.

The highway spawner must spawn (continuously) cars and trucks from the left and right, with some lanes only moving left to right, and some lanes only moving right to left. Highway spawners can also spawn coins. When the cars and trucks go to other end of the lane, they must be destroyed. All the cars in a lane move at the same speed, but cars and trucks in different lanes can (and should) move at different speeds. **The number of lanes in a highway should increase (from 1 up to 20) as the chicken gets further into the game (making the game harder). You can also increase the number of cars and trucks on the road at the same time.**

The RR spawner must spawn RR crossing arms and spawn (continuously) high-speed, many-car (5-6) trains, moving either L to R or R to L. RR spawners also spawn coins. Because the trains move so fast, the crossing arms should flash its lights and ring a warning bell whenever a train is passing. The lights and bell should be early enough so the chicken knows a train is coming. If you have multiple RR tracks next to each other, it is possible to have multiple trains pass at the same time. **The number of RR tracks next to each other should increase (from 1 up to 10) as the chicken gets further into the game (making the game harder).**

The River spawner must spawn lanes of the river with either moving logs (moving R to L or L to R, with each lane having different speeds), OR lanes with only stationary lily-pads. River spawners can also spawn coins, which lie either on the lily pads, or on the logs. The lanes have a waterfall near (but not at) both ends of each lane. When logs reach the waterfall, their speeds increase. **The number of river lanes next to each other should increase (from 1 up to 19) as the chicken gets further into the game (making the game farther).**

The chicken can die by either:

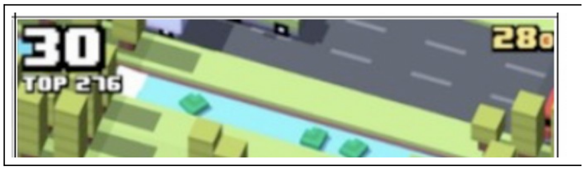
- Standing around too long, or moving backwards too many steps, so the eagle swoops down and carries him away. The eagle always swoops in from ahead of the chicken, and makes a shrill scream as it does.
- Being run over by a car or truck – it lies squashed on the pavement – its Z dimension set to almost 0).
- Running into the side of a truck – it is squashed against the side of the truck – its X dimension set to almost 0.) It continues to move along with the truck, and is carried off the game with the truck.
- Running into, or being struck by, a train – it explodes into a particle system of white feathers.
- Jumping into the water – it disappears, and a particle system of water cubes shows the splash.

The chicken cannot "win" the game, because the game goes on forever, and it is just a matter of time until the obstacles in the game become too many for the chicken to avoid. The goal of the game is to take as many **FORWARD** steps as possible before being killed. Horizontal steps can help you avoid obstacles and cars and trucks, but are not counted.

Coins jump up into the air and make a sound to coax the user to come and collect them. A successful collection sound is made when the coin is picked up.

At the top left of the screen, a counter is shown of how many steps the chicken has currently taken, and what the high score is. Use an HUD User Interface, as shown in Tutorial 4.

At the top right of the screen, a counter is shown of how many coins have been collected.

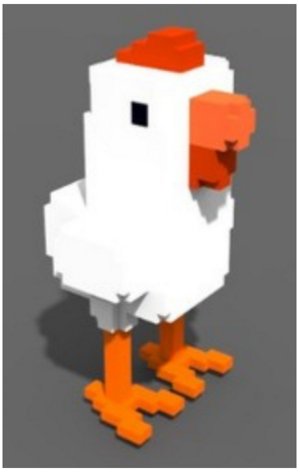
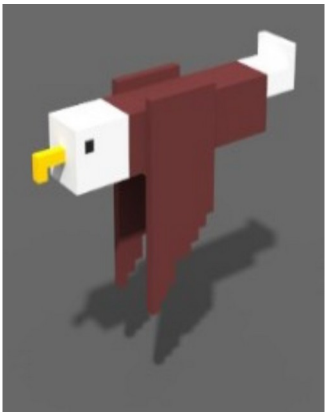
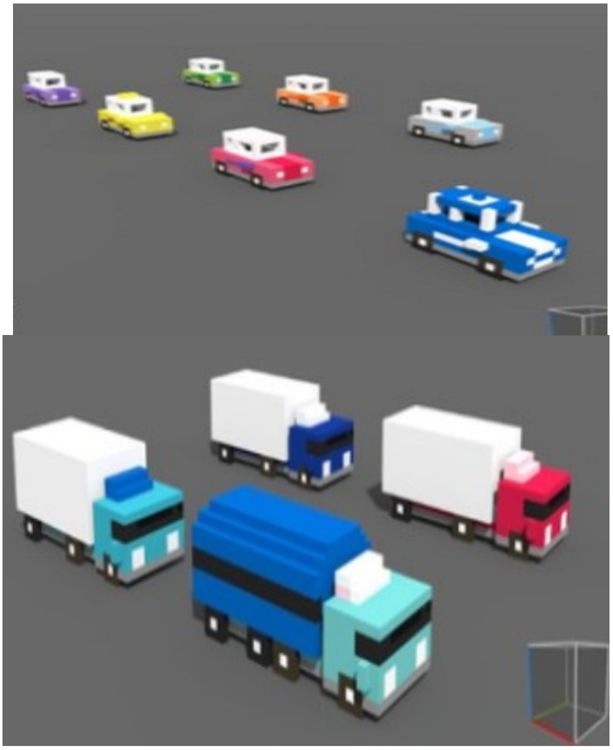


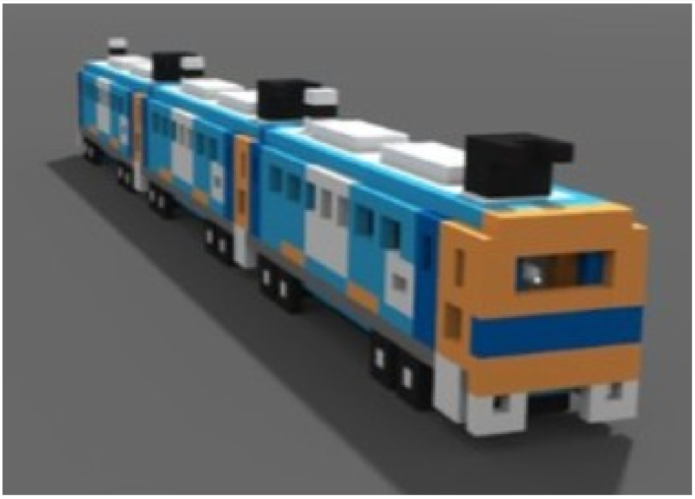
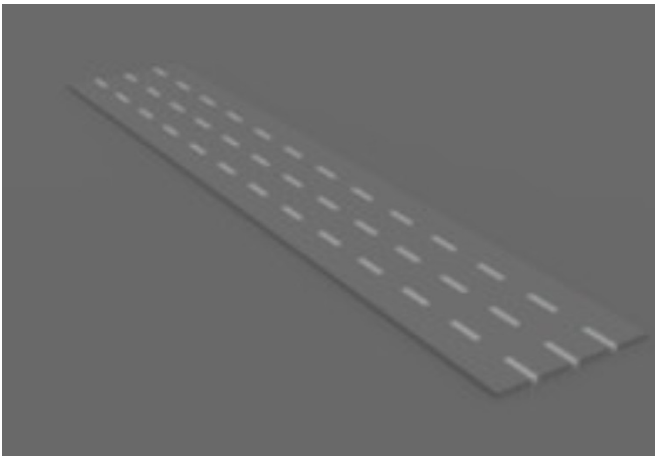
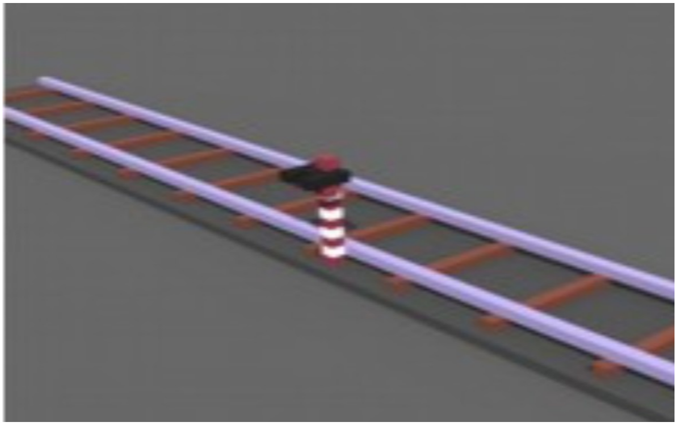
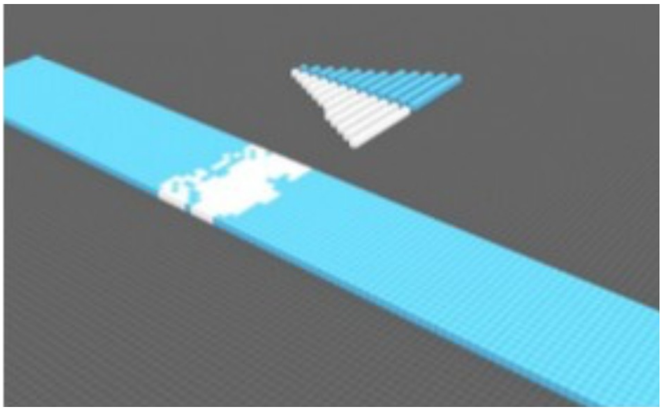
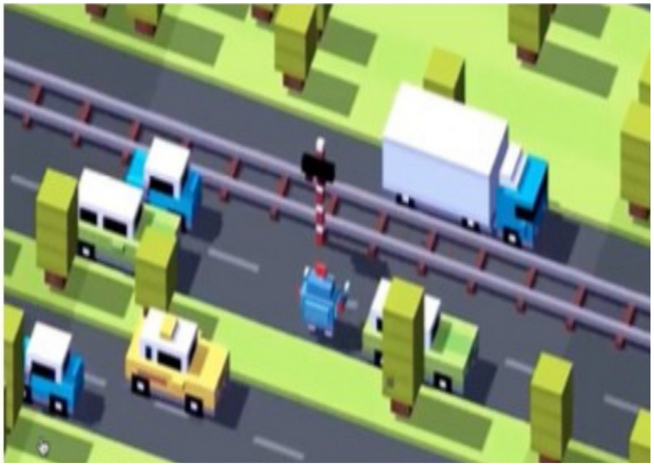
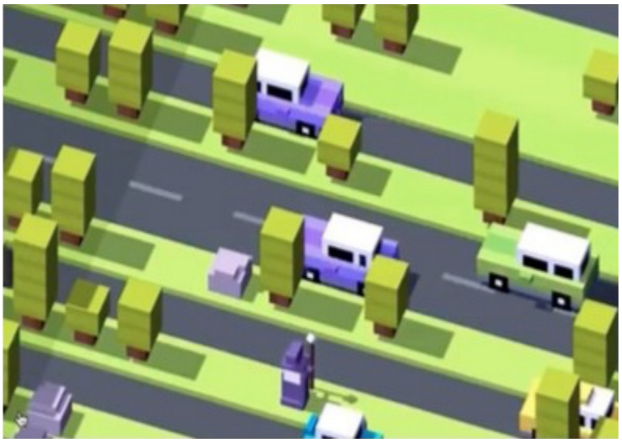
Sounds needed for this game can be recorded by running Audacity or Wavepad or another sound recorded software, and running the game...

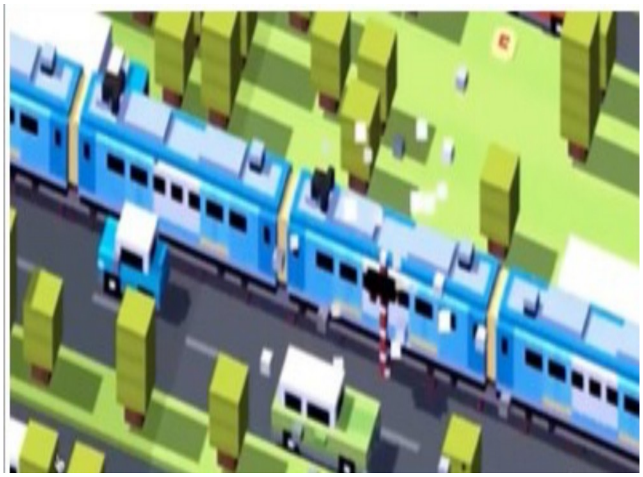
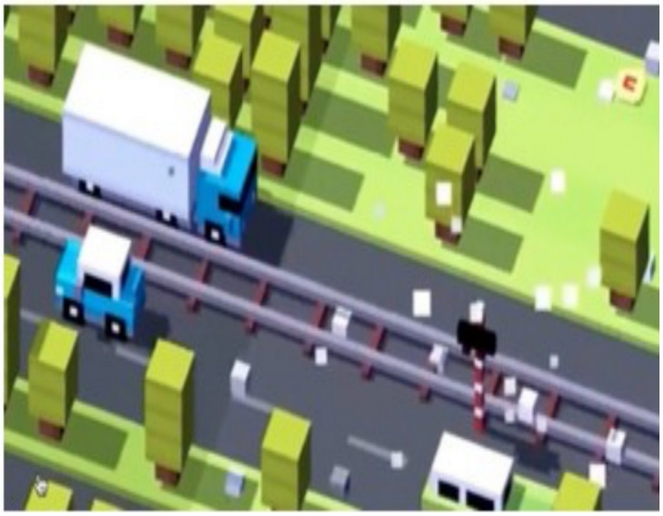
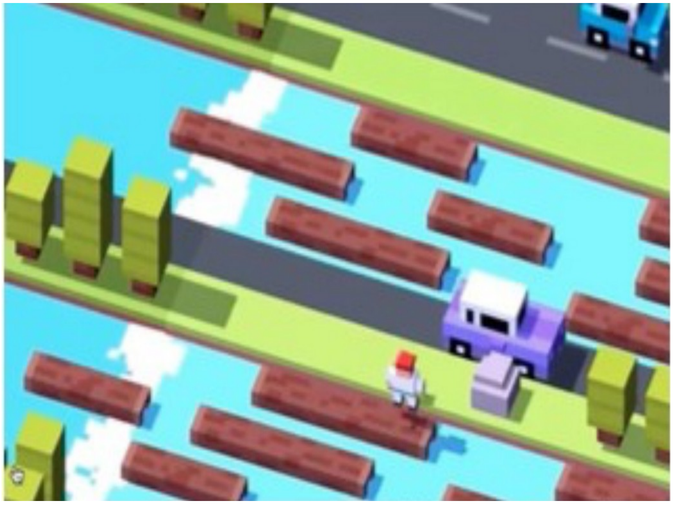
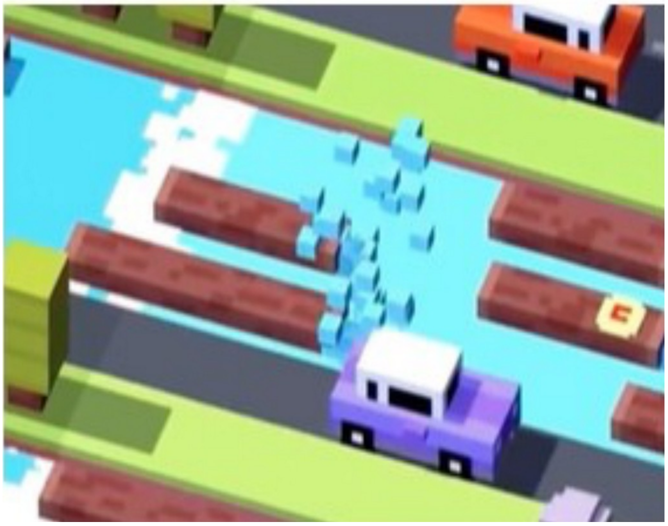
Moving sounds (moving the chicken)
Chicken sounds (chicken clucking every so often, can be different frequencies)
Chicken dying sounds (loud b-gawk! Sound when killed by running into car/truck/train, being run over..., picked up by eagle, falling into water, ...)
Cars/trucks sounds: tires on highway plus truck engines (soft, trucks are louder), long/short beeping occasionally (beep/honk)—can be close or far away, beeps can be different frequencies, long doppler-shifted horn of sports car occasionally, police-car siren
Game over/beginning sounds (ascending bloop, bee-boop descending)
Train sounds (bell of train coming, train whooshing by)
Water sounds (stepping onto lily pad/log (stone dropping in water, creaking stair), falling into water)
Picking up coin sounds (ascending series of xylophone tones)

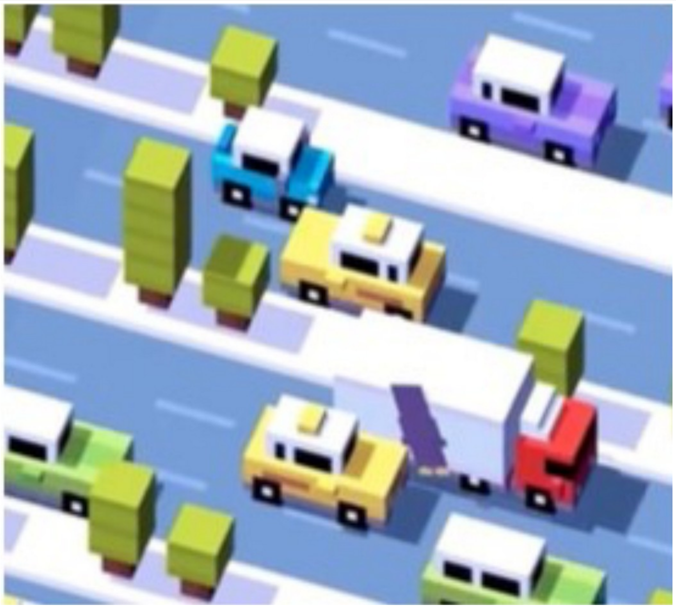
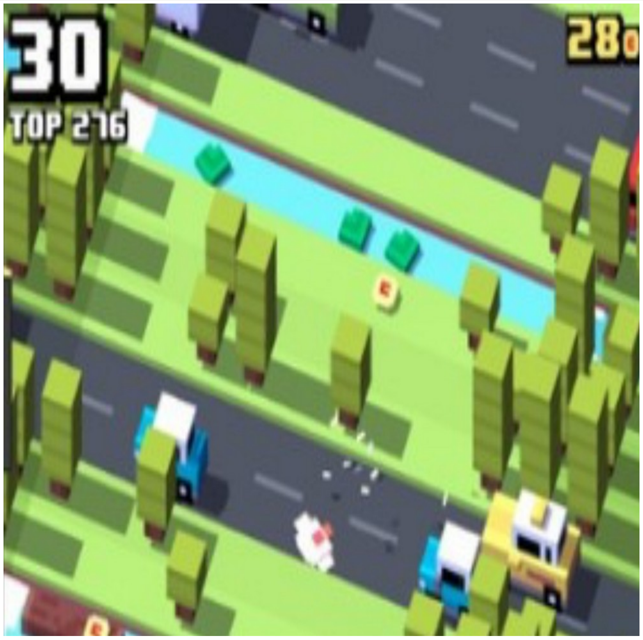
Finally, the game has a start screen which shows the CrossyRoad logo sliding in from the upper left to the center, at approximately a 30 degree angle.

Here are some sample static meshes that were created in MagicaVoxel.

 **Our hero: the chicken**	 **His nemesis: the eagle. Eagle can animate if using Blender – in that case, wings must be rigged.**	 **Different colors, types, and lengths of cars and trucks**

 **Cars of a train (can be grouped together in a blueprint)**	 **Lanes of a highway – can be constructed with a (no-lane-marker) lane and several (lane-markers-on-side-of) lanes**
 **Railroad crossing with lights and bells**	 **River with simulated waterfall (see CR game)**
 **Railroad tracks can be in the middle of a highway—i.e., you can have HW, RR, HW, then grass**	 **Single lanes of grass in the middle of single and double lanes of highway.**

A particle explosion of chicken feathers when the chicken is struck by the fast-moving train... 	 **… and lingers after the train has gone**
 Logs sink AND make a creaking noise when stepped on – THEN – Logs rise again AND mke a creaking noise when stepped off.	 **A splash of water particles (tutorial 8) when the chicken falls into the water.**

 **Chicken runs into side of truck:** Note – x dimension of chicken set to 0.1 AND chicken must be set to dead AND chicken must move with truck AND chicken must be destroyed when truck is.	 **Chicken crushed by car or truck.** Note – z dimension of chicken set to 0.1 AND chicken must be set to dead.

Submission

Turn in the code for this project by uploading all of the Unreal source files you created, the images directory, and the sounds directory to a single public repository on GitHub. While you may discuss this homework assignment with other students. Work you submit must have been completed on your own. To complete your submission, print the following sheet, fill out the spaces below, and submit it to the professor in class by the deadline. Failure to follow the instructions exactly will incur a 10% penalty on the grade for this assignment.

CPSC 386 Final Project, due Sunday, 13 May 2020

Your name Justin Drouin

Repository `https://github.com/`jrubix / CrossyRoads

Verify each of the following items and place a checkmark in the correct column. Each item incorrectly marked will incur a 5% penalty on the grade for this assignment. jrubix.github.io

Completed	Not Completed	Crossy Road
□	☒	(optional) Game has **startup screen** with Crossy Road logo sliding in from the upper right at a down angle of 30 degrees.
☒	□	Implemented the **game's HUD (head's up display)** showing the high score, current score (number of jumps), if this is a new high score, and coins collected.
☒	□	Implemented the **chicken** in MagicaVoxel, and imported it correctly into Unreal.
☒	□	Chicken **jumps and rotates to looks in the direction it is moving** (WSAD)
☒	□	Dynamically created (alternating) **grassy strips** ((optional) up to 19 strips), w/**code to populate them with trees/rocks** so there is > 1 path to pass. Trees should block sides of game. Chicken is blocked from sides of game. (optional) **N_lanes decreases as game continues.**
☒	□	Dynamically created **highways** ((optional)up to 19 lanes), w/**code to populate them with cars/trucks, and control their movement**. Multi-lane roads must have lane markers. (optional) **N_lanes increases as game continues**. Chicken blocked from sides.
☒	□	Dynamically **created/deleted cars, trucks, trains, and (optional) logs**, randomly moving in different directions if on different lanes of the highway, river, or RR tracks.
☒	□	Dynamically created **RR tracks** ((optional)up to 19 tracks), w/**code to populate them with trains**, with RR crossing arms w/point lights that shine (and ring a bell) if a train is coming. (optional)N**_tracks increases as game continues.** Chicken blocked from sides. Point lights OR bell can be used, or (optional) both.
☒	□	Dynamically created **river lanes** ((optional)up to 19 lanes), w/code to populate them with (optional) logs and **lily pads**. (optional) River lanes should allow logs to move in both directions. (optional) N**_tracks increases as game continues.**
☒	□	Imported all actor, safe area, obstacle and miscellaneous 3d assets into Unreal 4, and rotated and scaled them to their proper proportions.
□	☒	**(optional) Correctly implemented crouching and jumping with delay** with Blueprints or in C++, so the actor crouches as long as the arrow key (left/right/up/down) keys are pressed, but jumps immediately when it is released.
☒	□	**Collisions with trees, rocks, or the invisible side barriers** on the highways, RR tracks, and ends of the river cause the **chicken to stop moving**.
☒	□	Collisions with cars or trucks cause the chicken to be squashed (z direction if run over, OR (optional) x direction if it runs into the side of a truck). Collision with trains can be by just squashing the chicken.

☒	□	**Falling in water** is correctly implemented: **blue particle system** explodes upwards, then falls down again; chicken sinks into the water and squawks.
☒	□	**Collisions with trains** is correctly implemented: (optional) **white (and orange and red) particle system** explodes upwards, then falls down again. A few feathers are left. Can show chicken squashed in this version.
☒	□	**I**mplemented the sounds of the chicken clucks when moving, squawks loudly when dying, and the eagle shrieking when it swoops down
☒	□	**I**mplemented the sounds of the cars, trucks, trains, RR crossing arms, (optional) logs when stepped on, and coins when you pick them up.
□	☒	**Eagle swoops down and carries chicken away** if it doesn't move for several seconds, or moves backwards multiple times, or is carried off screen by scrolling. **Note: screen scrolls forward first, to better show the eagle grabbing the chicken. Screen shows > 2 lanes in front of/behind the chicken.**
□	☒	Optional (extra credit): **First person perspective** for chicken w/ominous music.
☒	□	Project directory **pushed to new GitHub repository** listed above

Comments on your submission

Slight delay issue on some of the collisions, could not figure out why even when I implemented collision boxes. Could not figure out eagle implementation.