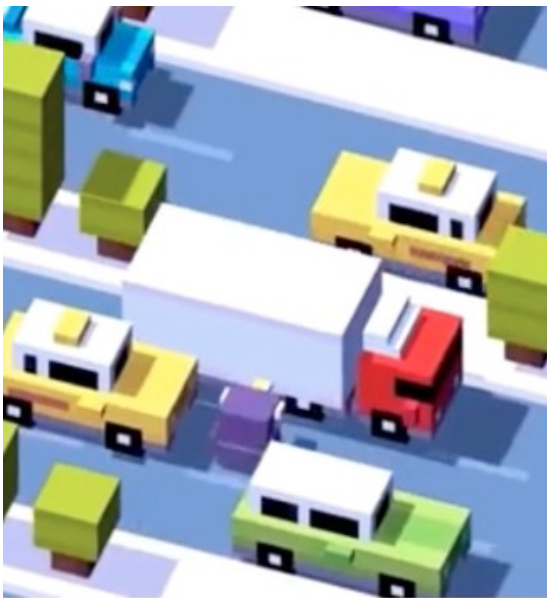
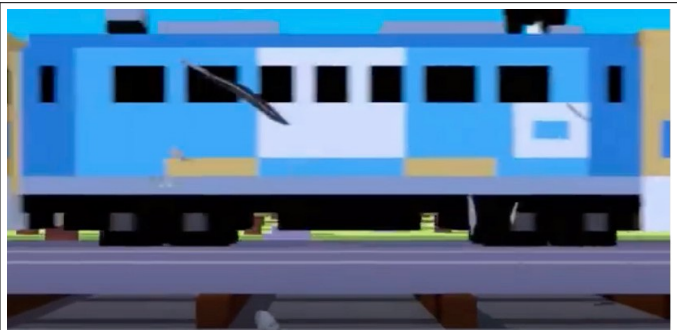
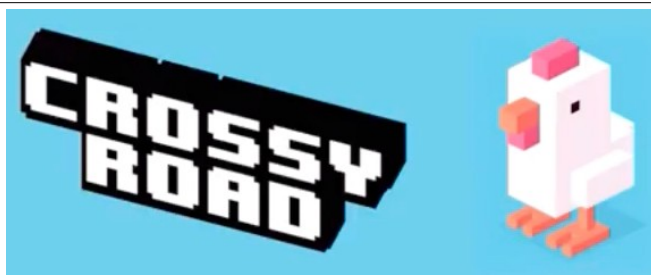


# CPSC 386: Introduction to Game Design and Production - Spring 2019

## Final Project, **Crossy Road**, due Sunday, 19 May 2019 (by 2355)

In this assignment, you will recreate the classic Crossy Road game, using the Unreal 4 Engine, assets created using a Voxel editor such as MagicaVoxel, and code written either in C++ (written to interface with Blueprints and the Unreal framework), or with Blueprints, or both.

The 3d assets you will need will all have to be created using an Voxel editor such as MagicaVoxel, or from other 3d asset sources from Unreal. The audio resources you will need can be captured using an audio editor such as Audacity from Crossy Road running on a laptop or desktop computer, or running as an iOS or Google app on a mobile phone.



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.

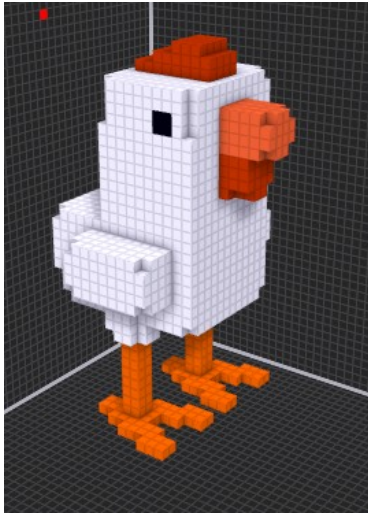
The special effects that occur are as follows:

- running into the side of a car or truck: 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 (polygon), 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

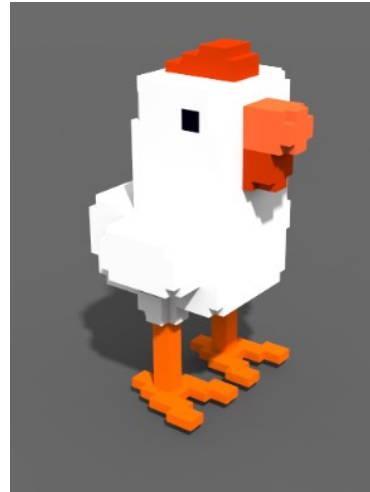
### To create your game, you must...

Create your 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.

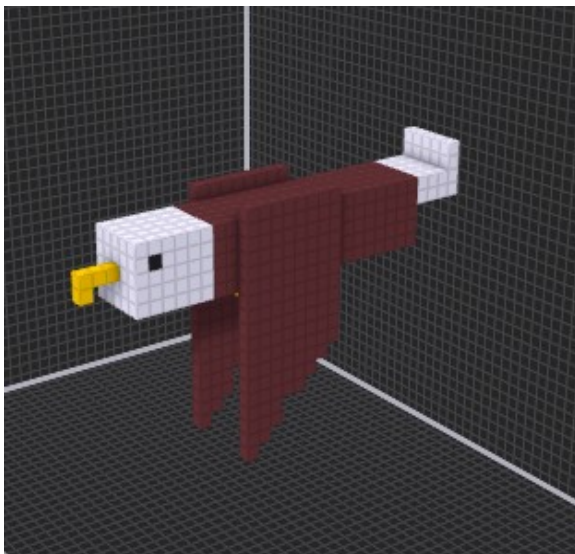
|   |   |
|---|---|
| <b>Hero</b> <ul style="list-style-type: none"><li>• Chicken</li></ul> | <b>Villains</b> <ul style="list-style-type: none"><li>• Cars</li><li>• Trucks</li><li>• Train</li><li>• Water</li><li>• Eagle</li></ul> |
|---|---|



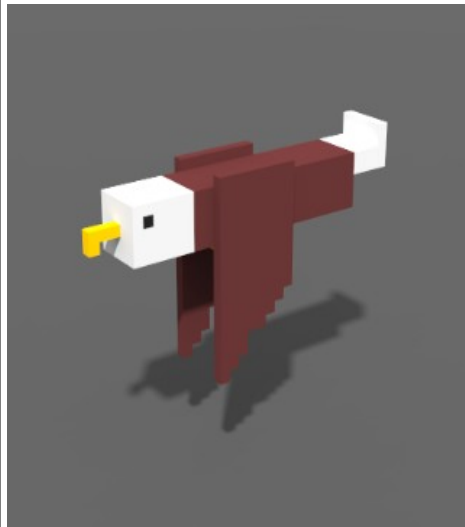
Our hero: the Chicken, in MagicaVoxel.



The Chicken, rendered.



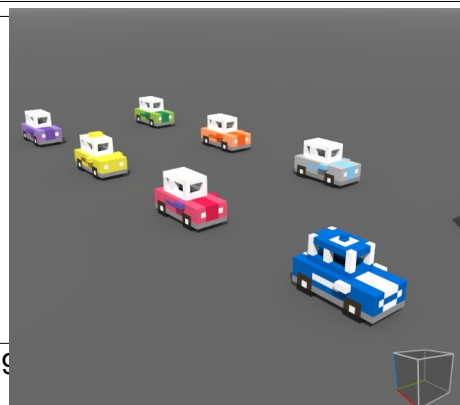
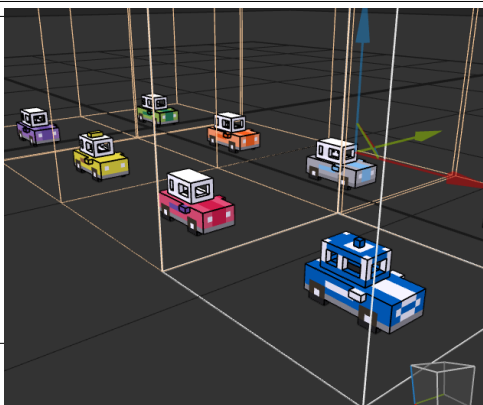
A villain: the Eagle, flapping its wings)



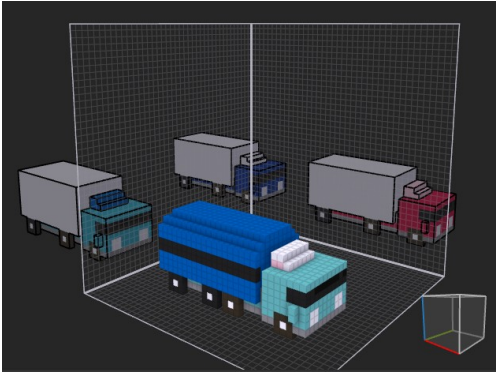
The Eagle, rendered.

### HIGHWAY VEHICLES and HIGHWAYS...

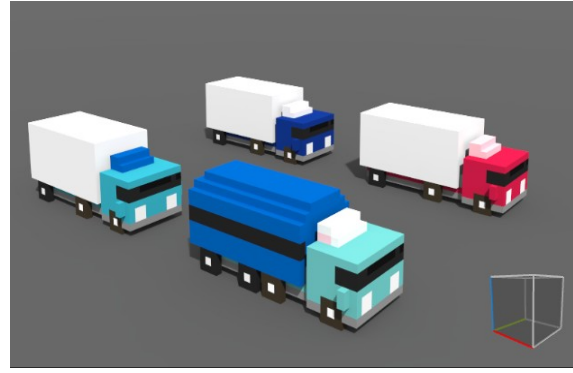
- Cars (Taxi, Short-blue, Long-green, orange, purple)
- Trucks (some ideas: light-blue or blue or red/white trailer, and white/blue oil tanker which can explode in flames)
- Roads (1-, 2-, 3-, 4-, 5-, 6-, ..., -10 lane roads). Multi-lane roads have lane markers.



Cars, including a police car, in MagicaVoxel.

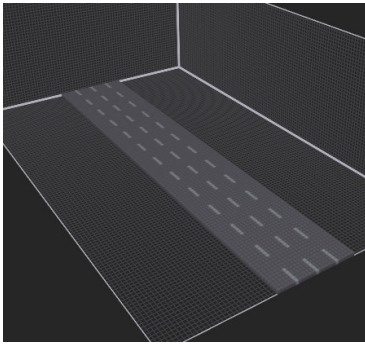


Trucks, including an oil truck that explodes, in MagicaVoxel.

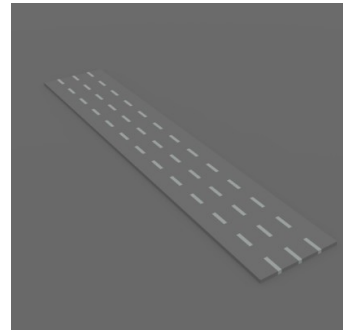


Cars, rendered.

Trucks, rendered.



Four-lane highway, in MagicaVoxel.



Four-lane highway, rendered.

## RAILROADS and TRAINS

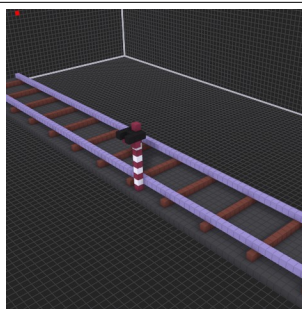
- Tracks (1- and 2-tracks. Tracks can be between any two obstacles or safe areas.
- Train of 5-6 cars made up of a:
- Lead car, middle cars, end car



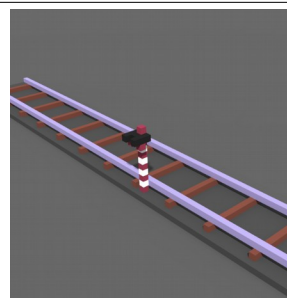
Train (lead car, middle car, trailing car) in MagicaVoxel



Train, rendered.



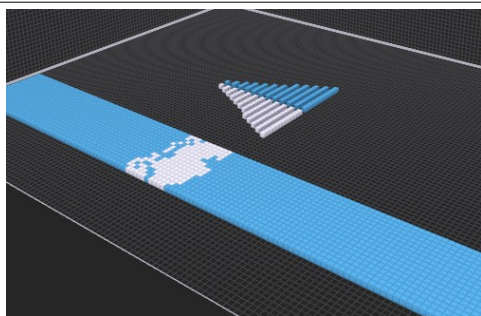
Train tracks and the train crossing warning in MagicaVoxel



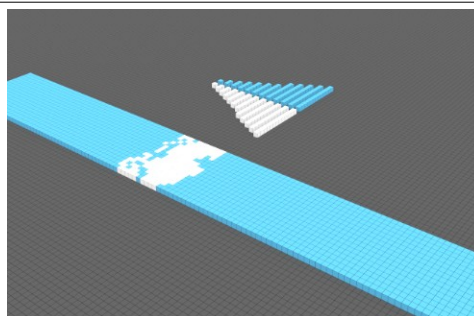
Tracks rendered.

## WATER

- Water w/ white reflection showing limits of motion, 1-, 2-, 3-, 4-, and 5- logs wide (waterfall effect is optional)
- Logs (full (truck)-size and 2/3 size)
- Lily pad



Water, with different lengths of white and blue water, to form the waterfall, in MagicaVoxel.



Water with waterfall, rendered.

## LANDSCAPING

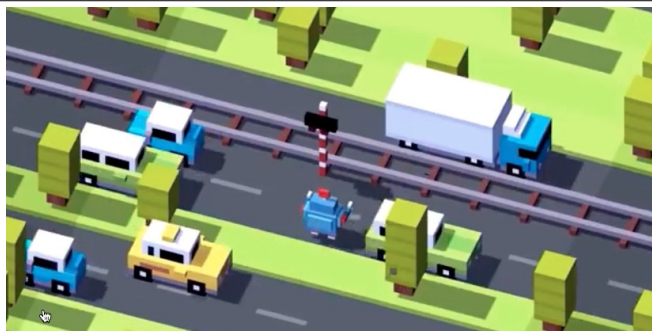
- Grass (1- and 2-trees wide)
- Trees (tall, medium, short)



- Rocks (short of various shapes)

## MISCELLANEOUS

- Crossy Road logo (black/white—projected) – must be rendered in Unreal4 for proper effect.
- Coins with red 'C'



Single and double-lane highways, with a single-track railway in the middle. Cars, taxis, and trucks going up and down the highway



Single lane strips of grass (with rocks, bushes, stumps, and trees).



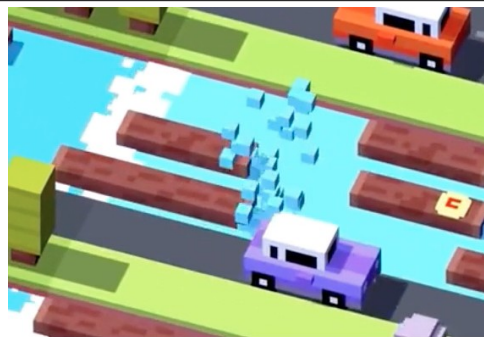
A RR can run in between two highways.



Trains can whiz by. The chicken has just been struck by the train, exploding into a particle system of white cubes.



Note a 4-log stream is next to a 3-log stream and a highway, with only a 1-tree wide grass strip to pause within. (Note: the logs and lily pads sink slightly in the water as they are jumped upon, and make a noise.



... a splash of particles of water explode if they fail.

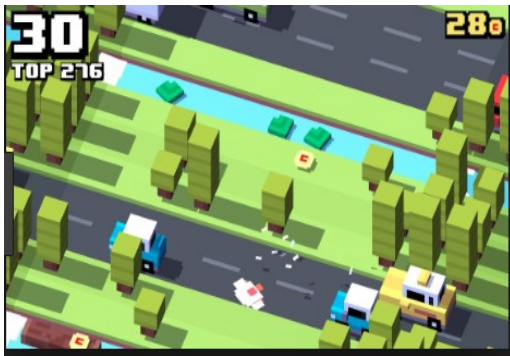


A failed crossing ends up on the side of a truck



Before and after

1. Downloaded the game onto their iPhone or Android phone, and played it often enough to understand the game, You need not deploy this game onto a mobile phone.
2. Be able to duplicate the game to the extent that others enjoy playing their game.
3. The game has a startup screen that shows the name of the game's logo, with the logo sliding in from off screen to the center of the screen. The logo is shown at a down angle of 30 degrees.
4. High score is shown at the top-left corner of the screen. Coins collected is shown at the top right-hand corner. New high-scores should be shown in a pop-up window.



Note the dead chicken on the highway. It has been run over. Lily pads and coins are also shown. Dense trees at sides of game, to restrict chicken's movement.

5. Implemented all of the safe area, obstacle and reward 3d assets in the game, complete with their contents (trees and rocks), (cars and trucks), train w/ multiple cars, logs and lily pads).
6. Implemented the ability of the game to dynamically generate the contents for each obstacle/area.
7. Implemented the actor's movement through the game (crouch, jump, collide with immovable objects—rocks, trees, train crossing, and collide with fatal objects—cars, trucks, trains, water, and eagles. Note, collision must use SWEEP, **not** teleport in unreal 4.
8. Implemented the game's ability to dynamically generate/destroy obstacles and safe areas.

9. Have created particle systems with appropriate colors and sizes for running into/being run over by trains, exploding oil trucks (and actors), or splashes of water, and connected them to the game.
10. Used a sound editor such as Audacity to record the sounds of the game as you play.

**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, ...)

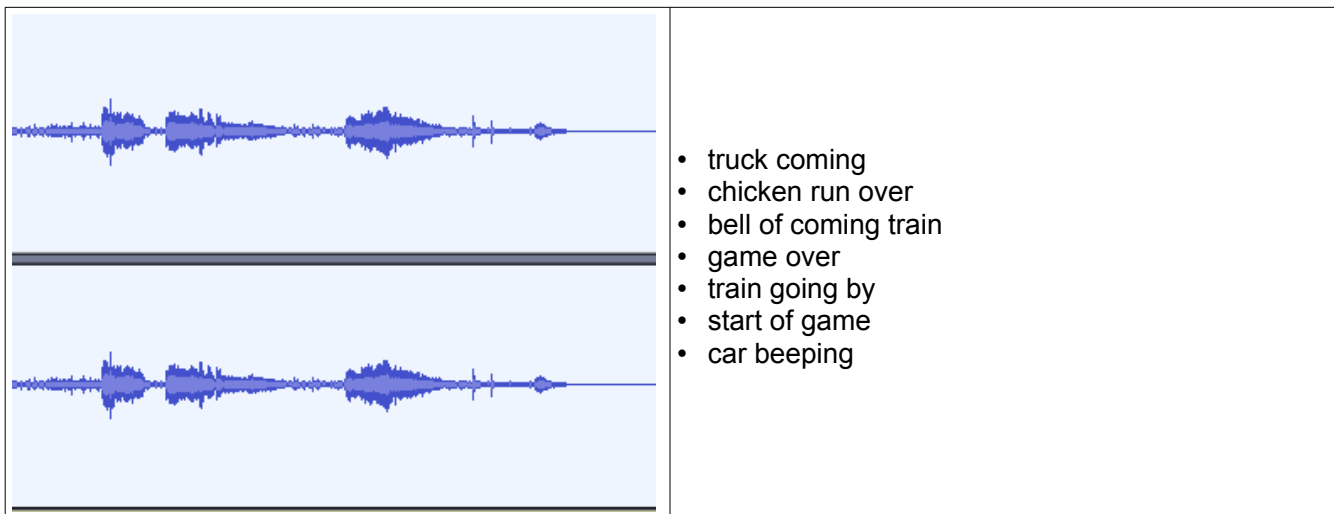
**Cars/trucks sounds:** tires on highway plus truck engines (soft, trucks are louder), long/short beeping occasionally (beep/honk)—can be close or far away, beeps can be different frequencies, long doppler-shifted horn of sports car occasionally, police-car siren

**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)



11. Connected the sound files to the activities in the game: actor moving, making sounds when being struck, cars/trucks moving and beeping, train coming/passing, stepping onto logs/lily pads, being run over, eagle shrieking, game over/game beginning, picking up coins.
12. Push the contents of your project to a new GitHub repository using a git client (e.g., the [git](#) command-line client, [GitHub Desktop](#), or [GitHub for Atom](#)). Do not submit files using drag-and-drop onto the repository web page, and do not push this assignment to the same repository as your previous homework assignments.

### 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, 19 May 2019 (at 2355)

Your name Esteban Montelongo

Repository <https://github.com/EstebanMontelongo/CrossyRoad>

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.

| Completed                           | Not Completed                       | Crossy Road   |
|-------------------------------------|-------------------------------------|---|
| <input checked="" type="checkbox"/> | <input type="checkbox"/>            | Have Crossy Road installed as an app on their mobile phone.   |
| <input type="checkbox"/>            | <input checked="" type="checkbox"/> | Game has <b>startup screen</b> with Crossy Road logo sliding in from the upper right at a down angle of 30 degrees.   |
| <input type="checkbox"/>            | <input checked="" type="checkbox"/> | Implemented the <b>game's HUD (head's up display)</b> showing the high score, current score (number of jumps), if this is a new high score, and coins collected.  |
| <input checked="" type="checkbox"/> | <input type="checkbox"/>            | Implemented the <b>chicken</b> in MagicaVoxel, and imported it correctly into Unreal.   |
| <input checked="" type="checkbox"/> | <input type="checkbox"/>            | Chicken <b>jumps and looks in the direction it is moving</b> (WSAD) (no sweeping).  |
| <input checked="" type="checkbox"/> | <input type="checkbox"/>            | Dynamically created (alternating) <b>grassy strips</b> (up to 19 strips), w/ <b>code to populate them with trees/rocks</b> so there is > 1 path to pass. Trees should block sides of game. Chicken is blocked from sides of game. <b>N_lanes decreases as game continues.</b> |
| <input checked="" type="checkbox"/> | <input type="checkbox"/>            | Dynamically created <b>highways</b> (up to 19 lanes), w/ <b>code to populate them with cars/trucks, and control their movement.</b> Multi-lane roads must have lane markers. <b>N_lanes increases as game continues.</b> Chicken blocked from sides.                          |
| <input checked="" type="checkbox"/> | <input type="checkbox"/>            | Dynamically <b>created/deleted cars, trucks, trains, and logs</b> , randomly moving in different directions if on different lanes of the highway, river, or RR tracks.  |
| <input type="checkbox"/>            | <input checked="" type="checkbox"/> | Dynamically created <b>RR tracks</b> (up to 19 tracks), w/ <b>code to populate them with trains</b> , with RR crossing arms w/point lights that shine (and ring a bell) if a train is coming. <b>N_tracks increases as game continues.</b> Chicken blocked from sides.        |
| <input type="checkbox"/>            | <input checked="" type="checkbox"/> | Dynamically created <b>river lanes</b> (up to 19 lanes), w/code to populate them with logs and lily pads. River lanes should allow logs to move in both directions. <b>N_tracks increases as game continues.</b>  |
| <input type="checkbox"/>            | <input checked="" type="checkbox"/> | Imported all actor, safe area, obstacle and miscellaneous 3d assets into Unreal 4, and rotated and scaled them to their proper proportions.   |
| <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <b>Correctly implemented crouching and jumping with delay</b> 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.   |

|                                     |                                     |   |
|-------------------------------------|-------------------------------------|---|
| <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <b>Collisions with trees, rocks, or the invisible side barriers</b> on the highways, RR tracks, and ends of the river cause the <b>chicken to stop moving</b> .   |
| <input type="checkbox"/>            | <input checked="" type="checkbox"/> | Collisions with cars or trucks cause the chicken to be squashed (z direction if run over, OR x direction if it runs into the side of a truck)   |
| <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <b>Falling in water</b> is correctly implemented: <b>blue particle system</b> explodes upwards, then falls down again; chicken sinks into the water and squawks.  |
| <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <b>Collisions with trains</b> is correctly implemented: <b>white (and orange and red) particle system</b> explodes upwards, then falls down again. A few feathers are left.   |
| <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <b>Eagle swoops down and carries chicken away</b> if it doesn't move for several seconds, or moves backwards multiple times, or is carried off screen by scrolling. <b>Note: screen scrolls forward first, to better show the eagle grabbing the chicken. Screen shows &gt; 2 lanes in front of/behind the chicken.</b> |
| <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <b>Implemented the dynamic generation/destruction code</b> for allowing the level to be continuously populated as the actor moves forward.  |
| <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <b>Used Audacity to record the music and game sounds</b> , and implemented them: Chicken clucks when moving, squawks loudly when dying, various horn sounds, bell for train crossing warning, swoosh when train goes by, eagle shrieking.   |
| <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <b>At least one other player has played your game</b> and signed off on it <b>as fun</b> .  |
| <input type="checkbox"/>            | <input checked="" type="checkbox"/> | Optional (extra credit): <b>First person perspective</b> for chicken w/ominous music.   |
| <input checked="" type="checkbox"/> | <input type="checkbox"/>            | Project directory <b>pushed to new GitHub repository</b> listed above   |

### Comments on your submission

---



---



---



---



---