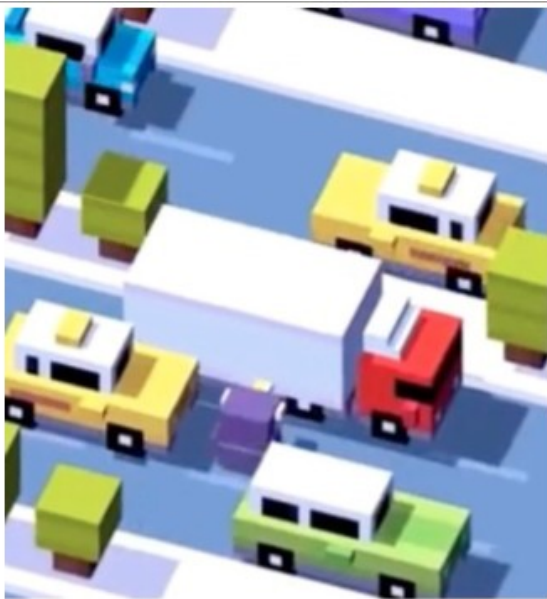
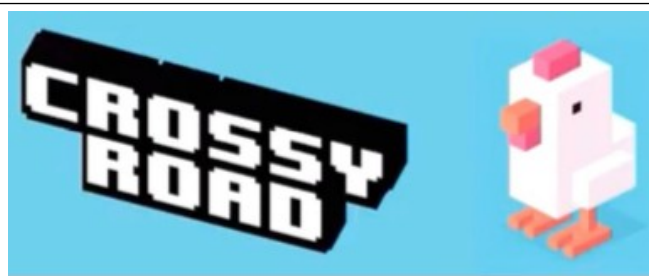


CPSC 386 – Fall 2019
Introduction to Game Design & Production
Final Project: Crossy Road

due at beginning of class (0900) – 386-01, 386-02: 20 Dec (F)

section 01 – you can submit your filled out rubrics to Titanium by Friday morning at 0900

For this project, your team will be creating a clone of **CROSSY ROAD**. Your basic goal is to create a game that looks and feels just like the app, available from the App store (iOS), or from Google play (Android). When played by someone who is not part of our class, your game should be indistinguishable from the original game. Do NOT create different protagonists, antagonists, obstacles, and so forth.



Trying to jump between the cars and the truck...



... failed... squashed on the side of the truck.



our hero



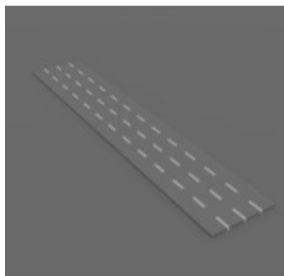
not our hero's best friend



various cars



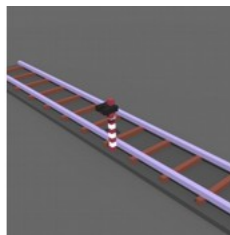
... and trucks



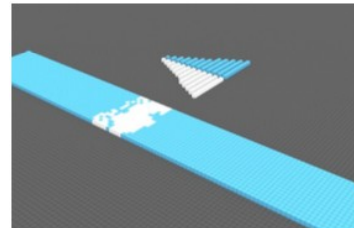
multi-lane highways



trains



railroads



river with simulated waterfall



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



(with rocks, bushes, stumps, and



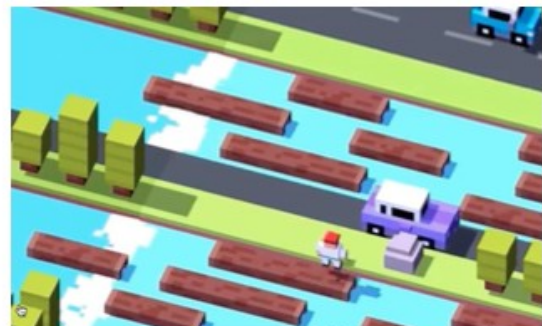
Single lane strips of grass (with rocks, bushes, stumps, and trees).



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.



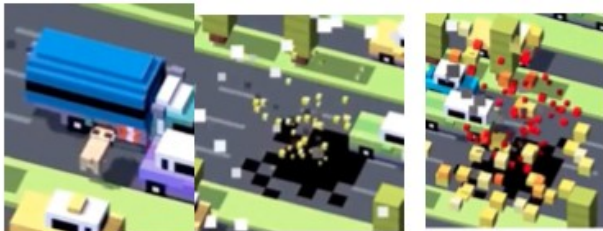
Note a 4-log stream is next to a 3-log stream and a highway, with only a 1-tree wide grass strip to pause within. (Note: the logs and lily pads sink slightly in the water as they are jumped upon, and make a noise.



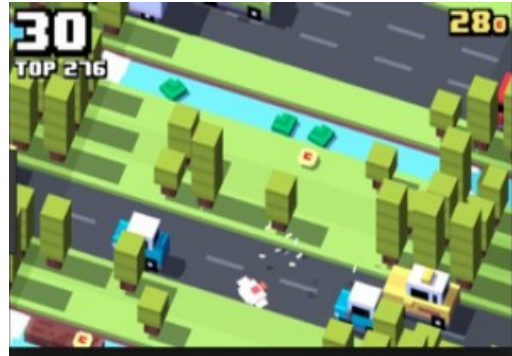
... a splash of particles of water explode if they fail.



A failed crossing ends up on the side of a truck



Before and after



Although you may use sound files that you find online, you must attribute the files used. All of the Blueprint code that you create (and C++ files should you choose to do so), must have been created by you or the members of your team. If you are having trouble with this project, please email me or come to my office hours, so we can work through any difficulties you may be having.

For this project, you will need...

1. Unreal Engine 4.23 installed on a computer powerful enough to support it. If your laptop is not sufficiently powerful, it is possible to remote login to a gaming computer at home. Gaming computers by design have the CPU, memory, and GPU required to run Unreal Engine. If you do not have a sufficiently powerful computer, please let the instructor know in person.
2. Chrome Remote Desktop on both your home computer and laptop if you are running Unreal remotely. (see <https://remotedesktop.google.com>).
3. The latest version of your OS and your compiler. For example, Xcode for macOS, Visual Studio for PCs running Windows, or g++ for Linux machines. Note: Unreal works very well with Xcode and Visual Studio, but some students have had difficulties getting it to work with with linux.
4. MagicaVoxel (<https://ephtracy.github.io>) for creating the 3d meshes you will need. For most students, MagicaVoxel is more than sufficient.

For students wishing to rig their models (add skeletons to them so they may be animated), you will need to use a more powerful tool like Blender (<https://www.blender.org>), or Maya (csuf has a free license to use Maya). If you haven't used Blender or Maya before, note they are powerful tools and have steep learning curves.

Note: it is not necessary to animate the movement of your meshes for this project. Some students wish to add animation to learn how to do it, and to make their projects look more cool, but it is not required.

5. You may use online sources to help you build the static meshes needed for the game, but each team is responsible to create their own meshes. The meshes include the main players (chicken, cars, trucks, train, logs, lily pads, eagle), obstacles (trees, bushes, rocks), lanes to cross (grass, highways, railroad tracks, rivers), and RR crossing arms.
6. 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 homework by uploading all of the Blueprint source files you created, all meshes used, and any other assets (e.g., sound files) directly 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 TWO-paged rubric, 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.

Note: due to the timing of the final exams, students in section 01 may submit their final project at the same time as students in section 02 (by 0900 on Friday morning, 20 Dec 2019).

CPSC 386 Final Project: CROSSY ROAD due Friday, 20 Dec 2019 (sections -01 and -02) at 0900 (section -01 can submit scan of filled out rubric on Titanium) NOTE: THIS RUBRIC is TWO PAGES		
Your name: Alex Chuoy, Kenny Giang, Russel Koh, Frank Ngo		
Repository: https://github.com/frankn172/crossyroad-FReAKGames /		
Finished	Not Finished	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 assignment's grade.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Have Crossy Road installed as an app on their mobile phone.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Implemented the moving objects (chicken, cars, trucks, train, logs, and eagle) in MagicaVoxel, and imported them correctly into Unreal (correct scaling and rotation).
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Implemented non-moving objects (trees, rocks, lily pads, coins, grass, highways, railroad tracks, RR crossings, and rivers) in MagicaVoxel, and imported them correctly into Unreal.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Chicken jumps when moving (WSAD and arrow keys).
<input checked="" type="checkbox"/>	<input type="checkbox"/>	If changing directions, chicken rotates to look in the jump direction.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Chicken will crouch before jumping if a movement key is held down.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Dynamically created (alternating) grassy strips , w/code to populate them with trees/rocks so there is at least one path to pass. Many trees are placed at the L/R edges to block the chicken.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Dynamically created highways , w/code to populate them with cars/trucks, and control their movement. Multi-lane roads have lane markers. Chicken blocked from going off screen L/R.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Dynamically created/deleted cars, trucks , with each lane having a specified direction.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Dynamically created/deleted trains , with each track having a specified direction.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Dynamically created/deleted logs , with each lane of the river having a specified direction. Note: logs cannot exist in river lanes with lily pads.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Some river lanes will have lily pads in them, which are randomly placed.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Collisions with trees, rocks, or the invisible side barriers on the highways, RR tracks cause the chicken to stop moving.
<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). Chicken squawks as it dies. Game is over.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Falling in water causes a splash (blue particle system); chicken sinks into the water and squawks as it dies. Game is over.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Logs that reach the waterfall move much faster as they race off screen. If the chicken is on one of those logs, it squawks as it dies. Game is over.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Collisions with trains cause an explosion of white (and orange and red) particles . Chicken squawks as it dies. Game is over.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	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.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Chicken and eagle sounds implemented (Chicken clucks when moving, squawks loudly when dying , and eagle shrieking)
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Vehicle and RR sounds implemented (different horn sounds, police car siren, train swooshing by, RR crossing bell when train is coming)

<input type="checkbox"/>	<input checked="" type="checkbox"/>	Implemented the game's (head's up display) showing the high score, current score (number of jumps), if this is a new high score, and coins collected. (see tutorial #4 for an HUD.)
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Game has a startup screen with the name Crossy Road , with the logo sliding in at a 30 degree angle from the upper left.
<input type="checkbox"/>	<input type="checkbox"/>	Extra credit: logs and lily pads animate and make a sound when the chicken steps onto them.
<input type="checkbox"/>	<input type="checkbox"/>	Extra credit: width of highways, rivers, and numbers of RR tracks gets larger as game goes on.
<input type="checkbox"/>	<input type="checkbox"/>	Extra extra credit: First person perspective for chicken w/ ominous music. Must widen game, slow down vehicles, and allow the chicken to move its head left/right so it can watch the cars as it moves forward (like you would). Also must allow the chicken to move its head up/down so it may step carefully as it crosses the river.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Project directory pushed to new GitHub repository listed above
Comments:		