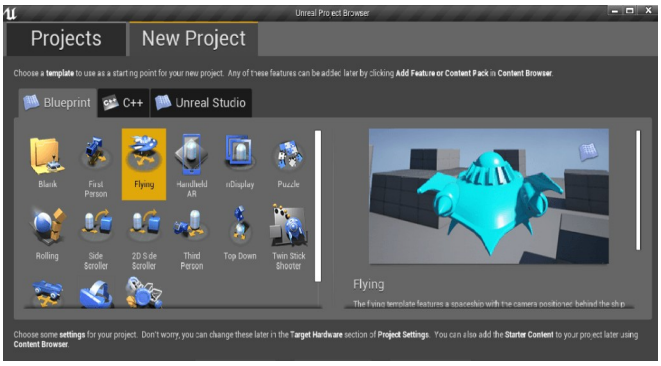
Assignment title: ** DEMONSTRATION ONLY **
Assignment #: not graded
Due Date: **DEMO ONLY**
Submission: Not submitted, demo only

Description:
Based on the Unreal Engine **Flying** template, create a Flying game for mobile devices.



Flying game :

1) replace the space ship a **model** of your choice
2) add the model of your name as a glowing neon sign so its visible in the distance at start of game
3) no game play changes required
4) only use the Georgian colours from the brand guide plus black and white
5) utilize a custom icon such as the Georgian logo for your application logo
6) application launch screen must contain: i)Georgian logo; ii) your name; and iii) course code, MDEV1003
7) use a royalty free song as background **music**
8) on bumping into something, sparks should appear, and a crash **sound effect** heard
9) comment your changes in the Blueprints appropriately

Note: various assets such as Georgian brand guide, logos and icons are in the section “In class demos” under “Module 5 - Images” under “Weekly learning” on Blackboard.

When submitting your game, please
1) submit all source code via a github link, or college OneDrive link
2) submit a packaged APK version of the game with NO .obb files, assets to be in APK
3) Send link(s) via Blackboard on the MidTerm Assignment
4) include an attribution list for assets utilized such as background music and images.

Approach and breakdown

You will be modifying an existing simple game. You are going to add a new model (your name) and change another model, the spaceship. You'll also add some music and sound effects, as well as a particle effect (sparks).

Step 1 – start a project from the flying template with starter content. Since we are targeting a mobile device for this game, when selecting the template for the game, set target to and resolution to and include starter content

Step 2 – look at the existing template game. Check out all the various pieces of the game, what is in each folder. Where is most of the game mechanics ? Check out the "FlyingPawn" in Blueprints" inside the FlyingBP folder. Check both the Event Graph, and the Viewport tabs in this Blueprint. The materials and the static mesh for the spaceship are in the Meshes folder, under the Flying folder.

Step 3 – gather your assets. Search the usual royalty free sites for a suitable spaceship model. Do similar for a crash sound effect and for background music. We'll use a particle effect already in the starter content for the sparks so no need to find one of those. Grab a copy of your 3D model of your name in fbx or obj format.

> Model
> https://opengameart.org/content/faction7-spaceship-att5-3d by johndh
>
> Sound effects
> https://opengameart.org/content/jet-engine-takeoff by dklon
> https://opengameart.org/content/rocket-launch-0 by dklon
> https://opengameart.org/content/stop by Blender foundation
>
> Music
> https://www.bensound.com/royalty-free-music/track/summer-chill-relaxed-tropical by Benjamin Tissot

Create or find suitable icons for the game. And crate a splash screen as a PNG file with the specified requirements, name, course code and logo.

Step 3 – clean assets. Next you should review each of your selected assets and note where they are from , add them to an attribution list for the project, confirm licensing. Regarding the music and sound effects, listen to each one, trim its length and convert to a 16-bit .wav file as required. For 3D models, open the model in Blender and review it. In here I like to check how the model is assembled, and colored. I often rename materials in here for easier identification when imported to Unreal. Export in an appropriate format such as fbx.

Step 4 – import assets. Import your assets into Unreal. The music and sound effects are pretty straight forward for importing. The 3D model will require a little more attention. Depending on your specific model, when you import the 3D model you may need to import it and adjsut the scale and orientation. You might need to do some trial and error on this step. Drag your model

into the scene to check its scale and orientation. A good check would be to drag the UFO static mesh into the scene and place your new model beside it. The size and orientation should be similar. If not, delete the assets and materials for it from the folder and re-import adjusting the scale and orientation.

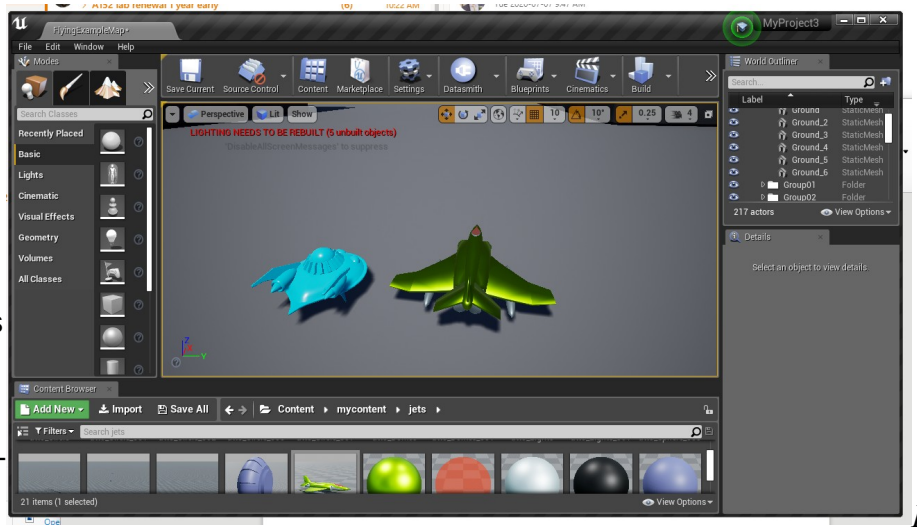


Figure 1: Comparing your model to the UFO

Step 5 – create materials. Based on the colours in the brand guide, create materials in the specific colours. You might also want to create the neon sign glowing material at this time.

Step 6 – colour your spaceship. Using the materials created, colour your spaceship in the new colors.

Step 7 – replace the spaceship. Remember the “FlyingPawn blueprint from step 2, go there and open the blueprint and go to the Viewport tab. Select the UFO object, and in the Details panel on right, find the Static Mesh section and select the field and a drop down will appear. Search for your spaceship static mesh and select it. The UFO will be replaced by your model. Save and compile the BP. Test play the game, how does your spaceship look?

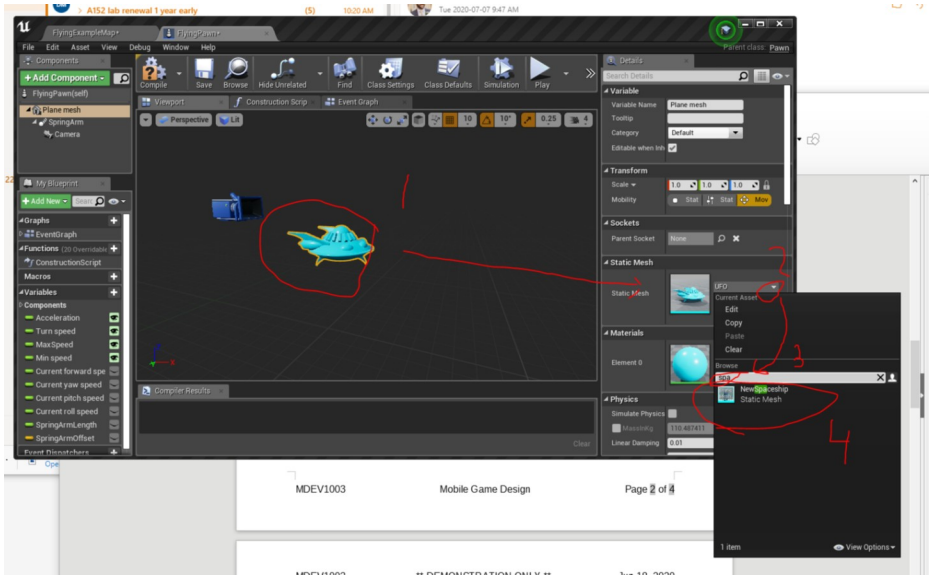


Figure 2: Replacing UFO static mesh with your model, Viewport of FlyingPawn BP

Step 7 – apply background music. Create a looping “cue” of your background music and place it in the scene. Play test game, do you here the music?

Step 8 – add the crashing sound. In the FlyingPawn BP, in the Event Graph scroll to see the section titled “Deflect when colliding” . This is where the collisions are handled. Drag out from the execution pin of the SetActorRotation and add a node called “Play Sound 2D”’. In the Sound field, select your crash noise as imported earlier. Play test your game, do you hear the crash noise when you hit something?

Step 9 – add the sparks. We’ll use a particle effect from the Starter Content called P_Explosion for the sparks. In the same BP, drag out from the Play Sound 2D node’s execution pin and add a node called “Spawn Emitter At Location”. In the Emitter Template field, select the P_Explosion particle effect. Connect this nodes Location pin back to the Location output from the Event Hit node on the left.

Step 10 - place your 3D name model. Place the 3D model of your name in a suitable location and apply a neon glow type material. Play test game, how does it look?

Step 11 – Packaging game. Add icons to the application as well as the launch screen image from your assets. Package for your mobile platform, deploy to device and test.

Step 12 – Submit your assignment. Collect/copy all the raw and modified assets, the source game, and the packaged game into a folder. Compress it into a single file and share via OneDrive. Or share project via githb

How would we make a flashing light on the plane model.