

Project: Lunar Lander

You will create a simple version of the classic game “Lunar Lander”. The object of the game is to apply just enough thrust to the left, right and upward in order to safely land on a platform.

There is an example of the project here:

https://youtu.be/Mbx_qjrJo0k

What are the requirements for the project?

The following are **required** to earn points for the project:

Menu Scene (5%)

- Title of the game, your name, Pace email and press space to start.

Game Scene (30%)

- Use squares/rectangles and/or other primitives to create a landscape. (refer to the example video).
- Create a couple of platforms (different color than the landscape).
- Use a triangle to represent the ship. have it start at the vertical top and horizontal center of the screen.

Gameplay (60%)

- The ship should use physics. Use gravity so the ship falls slowly (Hint: You can adjust gravity!).
- Here are the controls (they may feel a little backwards):
S = push up, A = push to the right, D = push to the left.
- The player has limited fuel to use.
 - When they use the keys mentioned, it drains their fuel.
 - If they are out of fuel they can no longer use thrust.
 - Drain fuel for each “deltaTime” that goes by with a key held down.
 - Show how much fuel is left. Format to 1 decimal place.
- Landing on a platform is a successful landing, touching the landscape is a failed landing.
- Do not switch scenes if they run out of fuel. Keep them in the current scene and just stop the input keys from working.

Success / Failed! Scenes (5%)

- Show the words “Successful Landing!” or “Failed Landing!” and press space to return to main menu.

Any tips on how to get started or approach this project?

Start with the main mechanic. Get the ship moving around first. Then add the constraints of the fuel and the landscape / platforms.

If I want to go that extra distance, what are some things I can add to my game?

While **not required**, here are some ideas for things to add to your game:

- Switch out the primitives for some sort of sprites.
- Add sound effects or music.
- Whatever else you can think of!

How do I submit my work?

Your project must compile! Non-compiling projects will get a 0. When you are done, you must upload a .zip of your project folder to the **Assignments** area in **Blackboard**. **After the due date**, there will be a discussion board post where you will post a link to itch.io and project reflection.

Do not share your code or your links with anyone until after the project is completed. All code must be your own, any plagiarism will result in a grade of 0.

Due by 11:59pm means your project was successfully uploaded by that time. Start uploading your project at least an hour before the deadline to avoid a point deduction. **Projects received 1 minute late are considered late.** If there are any issues with uploading your project, you must **email me before the due date**. Email cguida@pace.edu from your @pace.edu email address. I will not accept any other form of contact.

While I check email regularly, **do not expect a response over the weekend or close to deadlines.**

Late projects will have **10 points deducted per day**. Late projects will **not be accepted after 3 days**.

You will lose points for any of the following:

- You do not upload your code. Make sure you are uploading your code to Blackboard and **not your build**. If the folder you are zipping is called "build" that is not your code. Make sure you are zipping your Unity Project folder (assets, code, etc.)
- You upload a file that is not a **.zip** file. It must be .zip. Not a 7zip or .rar, etc.
- Your itch.io link is unlisted or otherwise not available. **Try your link in another web browser** where you are not logged into itch.io
- You have any of the issues that are in the Common Problems and Questions playlist of videos. **Watch this playlist** of videos and **make sure your project does not have any of these issues**.
https://www.youtube.com/playlist?list=PL_QmvmIfJvh0piepSNTSavA--U4ndCV0