CS-376 Final project

Overview

For this assignment, you will work in groups to make a new game.

- This **must be a new game**.
- You may use code from your previous games or other sources, if you like, but you may not count any of it toward the point system given below. You also must credit it in your CREDITS.txt file
- You **must keep a dev log** and update it when you do things
- **Everyone in your group must participate and make sufficient contributions** to warrant getting the same credit as the other group members. If you don't contribute sufficiently, your grade will be reduced.
- You must **fill out the self-assessment together**
- **You must use our standard Unity version** (2022.3.10f1)
- **Only one person from your group should submit the assignment**

Groups

You can work in groups of up to 4. You may not work in groups of 5. You can also do the project on your own, but we don't recommend it as it is likely to be somewhat more work than if you worked in a group. That said, the expectations for the game will scale with the number of people in the group.

Keeping a devlog

For this game, you will keep a development log: an online journal documenting your development process. This will need to be a shared document, so it should be a google doc or a OneDrive Word file or something else that can be collaboratively edited. You will be turning it in a copy of it and the peer reviewers will read it.

Each time any of you:

- Make a significant decision about the game design
- Implement a feature
- Fix a non-trivial bug
- Otherwise make a significant change

You should add an entry to your devlog. You should date the entries and indicate who wrote them.

Preproduction

Before you write any code, you should **do some preliminary design work and write it in your devlog**. There's a very good chance you will end up changing major parts of this design, but you should begin by thinking

about what you're trying to accomplish, and documenting that in the devlog. That's fine; you will not be penalized for that. If you do end up changing things, **do not go back and change the devlog**. Just add a new entry describing what you're changing and why. The devlog is write-only.

Aesthetic goals

Before you begin writing code, you should think about what you want the experience of the game to be like. **Choose two aesthetic goals**, in the MDA sense. These should be phrased in terms of player experience: the player's emotions, desires, thought processes, etc. They should not be phrased in terms of code, game mechanics, or other things outside the player's head. **Write each aesthetic goal in your devlog:**

- Describe the goal in a simple English sentence
- Describe signs that would indicate you were being successful with this goal
- Describe signs to watch out for that that you were failing with the goal

You should refer back to these goals when making design decisions. When choosing between two possible designs, always choose the one that best supports your aesthetic goals.

Core loop

Describe the core mechanics you imagine for the game. What is the player going to spend most of their time doing?

Now describe the core loop. For example, for D&D, it's: find monster, kill monster, take their stuff, buy new stuff, level up. For asteroids, the player spends their time moving away from hazards (asteroids and enemies), while also shooting them. So the core loop is something like: turn away from a hazard, accelerate away from it, turn toward something you want to shoot, shoot it.

How does your core loop serve your aesthetic goals?

Production Requirements

The requirements for the game are based on a point system:

- For each type of object on the screen
    - Object appears on screen – 1 point
    - Object moves – 1 point
    - Object controllable by the user – 1 point
    - Object responds to collisions – 1 point
    - Object changes appearance based on some kind of event or condition – 1 point per event or condition up to 3 points for a given type of object
    - Object makes continuous sound – 1 point
    - Object makes sounds in response to events – 1 point per event up to 3 points for a given kind of object
    - NOTE 1: We're interpreting "object on the screen" broadly here. If you have a score counter, that's an object on the screen and it changes its appearance based on some event, so that's 2 points.
    - NOTE 2: This is *type* of object. If you make a particle system, you don't get 100 points because there are 100 particles.
- Controls

    - 1 point for each meaningfully distinct control (joystick axis, button, mouse, keypress)
- Other
    - 25 points for the game being 3D
    - 5 points for implementing any complicated physical forces not already implemented by unity, such as modeling drag and lift for a plane
    - 5 points each for any menus you implement (pause, instructions, start, etc.), up to 3 menus
    - 10 points for implementing game save and restore
    - Dynamic spawning of objects – 1 point
    - Multiple levels – 1 point per level up to 5 points.

If you are implementing something not on the list, or that's especially complicated, you should feel free to pitch ideas to us and we'll tell you how many points they're worth.

Example

Here's an accounting of the point system for asteroids:

- The ship (3 points)
    - 1 point for each for being visible, moving, responding to collisions
- Missiles (4 points)
    - 1 point each for being visible, moving, responding to collisions, self-destructing after timeout
- Asteroids (3 points)
    - 1 point each for being visible, moving, responding to collisions.
- Score counter (2 points)
    - 1 point each for being visible, changing appearance based on events
- Controls (3 points)
    - Turn, thrust, and fire
- Dynamics spawning: 1 point
- Using unity physics: 5 points

So asteroids would be worth 21 points under this system. However, you should avoid direct reimplementation of games we've already done in class. So if you are thinking about doing asteroids, maybe include an enemy ship that tries to kill you.

Point requirements

Here are the number of points your group needs:

Group members	**Minimum points**
1	25
2	40
3	55
4	70

Playability

As before, the game's instructions must be sufficient to allow the reviewers to play it and understand if it's working. And the player shouldn't be able to get into "stuck states" such as flying off screen and not being able to figure out how to get back.

Deliverables

Only one group member should upload the assignment.

You will receive instructions later on for how to register your group on canvas. **You must register your group before you turn in your assignment**. Once you've registered your group, **a single member of your group** should upload to canvas a zip file containing:

- A copy of your **dev log** in PDF, txt, markdown, or word format
- Your **game**
- A **README.txt** file that provides any necessary instructions on how to play the game.
- A **Credits.txt** file that lists any assets, including code, you got from the net, from others, or from your previous assignments. Note that you cannot count reused code toward the point system above.
- A filled-out copy of the **Self Assessment** file describing what went right with your project, what went wrong, what you learned, and what score you believe you deserve for the assignment.

Only one group member should upload the assignment.