Exercise 6: Freestyle II

Introduction

For this assignment, you will implement another game with the similar constraints to exercise 5. Again, this is an exercise in starting from a blank slate. However, you must make a game in a different genre than your entry for exercise 5, and you cannot reuse any of your own code between the two exercises.

See exercise 5 for general advice on possible genres, etc.

Keeping a devlog

For this game, you will keep a development log: an online journal documenting your development process. This can be as simple as a .txt file with a dated entry for every time you worked on the game, or as fancy as a Wordpress blog. But you will be turning it in, so it must be something that the peer reviewers will be able to read. You can assume people will be able to read .txt files and .pdf files, and probably .docx files. And obviously, if you use a web site, they'll be able to read that.

Each time 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.

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. 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

You can make any kind of game you like provided that (this is roughly what the grading rubric will be):

- Objects are on the screen
  - There must be at least three kinds of objects on the screen
  - At least two kinds of objects must move
- Controls
  - The player must be able to directly control at least one object on screen. This could be using a joystick to pilot a car in a racing game, or using a button to control paddles in a pinball game.
- Object interactions
  - At least three kinds of objects on screen must be able to interact with one another.
  - Possible interactions include, but are by no means limited to:
    - An object chasing another object
    - Objects colliding
    - One object landing on another
    - One object applying a readily identifiable force to another (e.g. gravitational attraction of a spaceship to a star)
- Sound
  - Events that are significant in the game (in the sense that it's important that the player know they happened) must be marked by sound cues (the playing of a sound).
  - This only applies to instantaneous events, like collisions. You don't have to figure out how to make sounds out of on-going game state such as the player's health (although it's fine if they do).
  - For example, this events would need sound cues:
    - Scoring
    - Colliding with the player, if the player has an avatar
    - Firing of weapons
  - Examples of events that are not required to have sound cues (although it's fine if they do):
    - Events involving objects the player never interacts with
    - Steering in a racing game – the game doesn't need to generate a sound just because the player changed direction.
- Goal and progress:

    - The player must have a goal they understand
    - The player must have a way of evaluating whether they're making progress toward that goal
    - The player must have a way of knowing when they've achieved the goal
- Playability
    - The game's instructions must be sufficient to allow the reviewers to play it and understand if it's working
    - 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

You must include instructions for your game. You are responsible for insuring that the reviewers understand the intended behavior of your game well enough that (a) they can play it and (b) they can determine whether you've successfully implemented the requirements listed above.

Remember that any time you implement, fix, or make a significant modification to a feature, you should write it up in your devlog.

Postmortem

When you're done, add an after-action report to your devlog:

- Summarize what you originally set out to do
- Summarize what your goals ended up being by the end
- Summarize what you accomplished of those goals
- What went right?
- What went wrong?
- What do you wish you knew at the start of the project that you know now?
- What did you learn in the process?

Academic dishonesty and intellectual property rights

You may use any sprites and sounds you like for this assignment provided that:

- You can legally use and distribute them (don't pirate them)
- You include file named CREDITS.txt that documents who their original authors were, or failing that, what web site you got them from.

You may also use any code you like from other sources, provided that:

- You credit it in CREDITS.txt
- It is not used to implement any of the items listed above under "Requirements". That is, if you want to use someone's cool particle system for some effect, that's fine. But if we were to disable that code, the resulting game should still satisfy all the assignment's requirements.

What to turn in

Upload a ZIP file to canvas containing:

- An instruction file called INSTRUCTIONS. If it's in some format other than Word or TXT, include a PDF version of it. The file should make very clear
    - What all the objects on screen are
    - What their behavior is supposed to be
    - What the player's controls are
    - How the player scores
    - How the game ends, including win/lose conditions, if appropriate
- Your CREDITS.txt file, if you used assets you obtained on-line
- Your devlog. If this is not a file (e.g. if it's a Wordpress site), then include a .html or .txt file called DEVLOG.txt/DEVLOG.html with a link to the site.
- Your Assets, Packages (if any), and ProjectSettings directories