**Pygame Project**

In this project, you will design and implement a game using Pygame, and then present it in front of class if your section is in-person, or submit a video presentation if your class is online. The game is due before the last week of class, with presentations being done or due during the last week. **The deadlines above are tentative and subject to change. Keep an eye on your announcements for updates.**

Keep in mind that the following:

* This is a **group** project
  + The groups are randomly pre-set and can be found on D2L
  + Failing to work as a group will incur penalties
* Your project’s code **must be written in Python**
* The project **must use the Pygame module**
  + If a project contains any libraries that aren’t part of Pygame or the Python standard library, your project will not be graded

Feel free to design an original game or implement a game that already exists. Note that you must implement the whole game yourself; copying someone else’s code, in whole or in part, could be considered plagiarism.

The next pages contain details on what will be expected of you and your submission.

**Requirements**

Your game must feature the following requirements:

**Hard Requirements**

* Must use at least 10 different Surface objects.
  + Only objects which are blitted and visible in the Display Surface count
  + The Display’s Surface (retrieved using set\_mode()) counts as a Surface object
* It must access at least 5 different files, for either reading or writing.
  + Opening a file isn’t enough: files being read must change the state of the game; files being written to must have their content be based on the state of the game.
  + Files containing code or similar (e.g: DLLs) do not count towards this requirement
* It must play at least 2 different sounds.
  + If these sounds come from files on disk, they also towards the requirement above
* It must either take inputs from the keyboard, mouse, or a controller.
* It must feature some sort of movement (i.e.: surfaces moving across other surfaces)
  + In general, Surface A is said to be moving across Surface B if:
    - Surface A is blitted onto Surface B at different locations between frames
    - There is overlap in the area of where Surface A was previously blitted and where it will be blitted next
  + Consult your professor if you are unsure if something constitutes movement
* The user must be able to quit the game without pressing the X button at the top right or stopping the project running the game.
* There must be a way for the game to “end”.
  + If the game is some sort of puzzle, the puzzle must be solvable, and the game must acknowledge that the puzzle has been solved.
  + If the game has win and lose conditions, those must be implemented.
  + Either of the above must either close the game or allow the user to restart it

**Soft Requirements**

* Game instructions are present in the game and are in correct English
* Game does not crash or stall
  + Calling pygame.time.delay() or pygame.time.wait() will usually cause the game to stall, and should be avoided
* Game shows no obvious flaws, such as dropped inputs or rogue Surfaces

**Miscellaneous**

* Presentation: (see “Presentation” page for details)
* Group participation: (see “Group participation” page for details)
* Professor’s Discretion: Reserved to the professor to be awarded at their discretion
  + Ask your professor if there’s anything in particular that they are on the lookout for
  + In general, games with less effort will get less points

**Deductions**

* See the “Submission Guidelines” for details

**Submission Guidelines**

* You must submit your game on the Assignments drop box titled “Project”, which you can find on D2L.
* Only one group member needs to submit for the entire group
  + If more than one group member submits, only the latest submission will be graded
  + See “Group participation” for what happens if a group splinters into smaller groups
* **Emailed submissions will not be accepted**.
* All the necessary files must be put inside of a zip file. If your game is expecting a file to be inside a particular folder, that folder must be present in the zip file.
  + Your professor should be able to create an empty Pycharm project, install Pygame, unzip your submission into the project, press play, and your game should work
* **Do not submit any files which your game doesn’t require**.
  + Virtual environment files, gitignore, POM files, and anything that isn’t directly related to your game shouldn’t be submitted

The rubric for the project is as follows:

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Description** | **Grade** |
| **Hard requirements** | Features at least 10 different Surface objects which are blitted and visible on the Display | 10 |
|  | Accesses at least 5 different files | 10 |
|  | Plays at least 2 different sounds | 5 |
|  | Handles events from either keyboard, mouse, or controller to update the game's state | 5 |
|  | Features movement | 5 |
|  | User is able to quit the game without stopping the project or clicking the X button | 5 |
|  | Game must have some sort of end condition, which the player must be able to meet | 5 |
| **Soft Requirements** | Game instructions are present in the game and are in correct English | 5 |
|  | Game does not crash or stall | 5 |
|  | Game shows no obvious flaws, such as dropped inputs or rogue Surfaces | 5 |
| **Misc** | Presentation | 10 |
|  | Professor's discretion | 10 |
|  | Group participation | 20 |
| **Deductions** | Project files are not in a zip file | -10 |
|  | Submitted files not related to the game (e.g.: venv files, POM files, etc.) | -10 |
|  | Used packages besides the Python Standard Library or Pygame | -100 |
|  | Submitted code written by a third party | -100 |

**Group Participation**

At the end of the semester, a quiz named “Peer Feedback” will open under “Quizzes”, where you’ll be able to rate how much your teammates participated in the project. You’ll give them a grade between 0 (didn’t participate) and 5 (participated fully), and they will do the same to you. Your professor will then add up the grade given by your teammates and apply it to your “Group Participation” grade. There can be 2 types of groups:

**Cohesive group:**

* Applies when all team members are participating towards a single game
* If a group has less than five members, non-existing members rate their teammates 5
* If a group member abandons the course or is otherwise non-responsive or non-participating, they rate their teammates 5
* You can only rate your teammates: rating yourself does nothing, and will be ignored

**Non-cohesive group:**

* Applies when different team members are working on and submitting different games
* If a group has less than five members, non-existing members rate their teammates 0
* You will only be able to rate teammates who worked with you
* You can only rate your teammates: rating yourself does nothing, and will be ignored
* **Inform the professor that this is the case as soon as possible**

As an example of a non-cohesive group, suppose that you are in a group with Alice, Bob, Charlie, and David. David never showed up to class, so he has abandoned the course. You and Alice want to make a game called “Fruit Blasters”, but Bob and Charlie want to make “Bomb Slicers” (all fictional titles). You are unable to come to an agreement, so you inform the professor that you and Alice will be working on one game while Bob and Charlie will be working on another.

Given that you are all in the same group and are supposed to be working together, the maximum number of points you can get for “Group Participation” is 5 out of 20:

* Alice can rate you between 0 and 5
* Bob and Charlie cannot rate you, as they have split off from you; your group is non-cohesive
* David will not give you his automatic 5 points because your group is non-cohesive

While disagreements may arise between you and other team members, you should always strive to work together.

**Presentation**

At the end of the semester you will have to present your project. Presentations must be done in person, for in-person classes, or through a recorded video, for online classes. Your professor will inform you how long your presentation can be, but your presentation must answer the following:

* Give a brief description of your game
  + What are the input methods (keyboard, mouse, controller)
  + How are the input methods used (what does each key do? what do the mouse buttons do?)
  + What are the rules?
  + What is the objective?
  + Is there a fail state? If so, what is it?
* Show your game in action
  + Run the project on your computer and play for a little bit
  + Using a pre-recorded video of your game instead of a live demonstration will affect your grade
* Show how your game can be won or completed
  + If your game has several levels, you just need to beat one
  + If your game normally would take a while to finish, you can change some variables around to beat it faster
    - If you make changes to your code, show us where, and what you are doing
  + If your game has a lose condition, show us what happens when you lose
* What are the different ways to quit the game?
* Show us the assets that your game uses
  + If your game uses asset files that aren't images or sounds, how does your game use these assets?
* Were there any features which were originally planned that didn't make it into the submission?
  + If so which ones? Why?
* Describe some of the challenges you ran into and how they were solved.
* Describe one or more challenges you ran into that you were unable to solve.
* When did you start doing most of the work on your project? When did you finalize the project?
* Describe the contributions of each team member.

**Remember that the exact date of the presentations is subject to change**. Keep an eye on the schedule and announcements by your professor.