### Fill out this sheet and submit with your project as Name: Christian James a Word document or PDF file. Total Points Expected: 80/ 80

Grading Description & Rubric Program 8 Project Description—10 points

High level description:

My version of the powder game includes Empty, Metal, Sand, Water, Oil, Cloud, and Fire.

Elements added:

I added Oil that floats on Water. Also, I added another behavior to Oil and Water that they slowly erode Sand by using a random number generator to slow down the erosion. They also both move left or right when not falling.

I added Cloud that floats upward until it contacts either metal or the ceiling at which time it turns into water.

I added Fire that turns adjacent Oil into Fire (spreading) and turns adjacent Water into Cloud. Fire disappears after it runs one step().

Code attempts that were abandoned and why:

I attempted to do all the step function in one method without creating classes but abandoned this because I would not get the full points.

I also attempted to create a time delay on Fire so it would fall 3-5 pixels before disappearing but realized that without instantiating objects for every Fire on the screen, this would be incredibly difficult.

Data structures used (ArrayList, LinkedList, HashTable, Queue, Stack, array, etc.):

I used a 2D array for the grid.

The SandDisplay uses a HashTable

Classes added after water, sand, and metal:

Added BasicElement, Oil, Cloud, and Fire classes while not including a Metal class as it would have no contents.

Approximate time spent on project:

3-4 hours (mostly converting functional initial code into a superclass/subclass structure)

Anything else you want us to know as we evaluate your work:

I learned you can’t override static classes (probably already learned that but now I know it for sure 😊)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **Proficient** | **Competent** | **Novice** | **Unsatisfactory** |
| **Project Description**  **10 pts.** | Filled in completely. Accurately describes work done. | Filled in. Mostly describes work done. | Filled in. Sparse or inaccurate description of work done. | Information omitted and/or poorly description of work done. |
| **Documentation & Readability & Organization 5 pts.** | The documentation is well written and clearly explains code use. Code is exceptionally well organized and conforms to best practices. The main method has limited functionality. | The documentation is somewhat useful in understanding the code. The code is fairly easy to read and conforms to most best practices. | The documentation is minimal and/or the code is readable only to grader who has the assignment description. | The documentation is limited or non-existent and/or no modularization. |
| **Basic Features**  **Metal, Sand, Water**  **25 pts** | Code is complete, runs, and has no more than one minor flaw. | Code is complete, compiles. Has minor flaws. | Code has major flaw(s) but runs. | DOES NOT COMPILE  ZERO POINTS. NO PARTIAL CREDIT FOR PROJECT POSSIBLE. | |
| **Additional Features: three more elements**  **20 pts** | Code is complete, runs, and has no more than one minor flaw. Elements are distinct with unique actions. | Code is complete, compiles. Has minor flaws. Elements are distinct with unique actions. Omitted one element. | Code has major flaw(s) but runs. Elements are distinct with unique actions.  Omitted one or two elements. | DOES NOT COMPILE  ZERO POINTS. NO PARTIAL CREDIT FOR PROJECT POSSIBLE. | |
| **Advanced Features**  **Abstraction, Polymorphism**  **20 pts** | Code is complete, runs, and has no more than one minor flaw. Good organization. | Code is complete, compiles. Has minor flaws. Fairly good organization | Code has major flaw(s) but runs. Okay organization. | DOES NOT COMPILE  ZERO POINTS. NO PARTIAL CREDIT FOR PROJECT POSSIBLE. | |
| **Extra Credit**  **Complexity, creativity, etc.**  **10 pts max.** | Instructor’s discretion | Instructor’s discretion | Instructor’s discretion | Instructor’s discretion | |

### Grading Rubric Feature Definitions

Basic Completed Part One: Added elements sand and water. Behavior of elements is correct.

Moderate Basic plus has added 3 or more elements. No more than 2 of the elements has complex behavior.

Advanced Moderate plus has added complex behavior, modified code to have base element class and subclasses, added objects, modified design of canvas, or other modifications deemed by the instructor to be advanced.