INITIAL PROJECT SPECS

**NAME**:Dennis Pandea\_\_\_\_\_\_\_\_\_\_\_\_

**PROGRAM OVERVIEW:**

A space game about dodging asteroids and

**DESCRIPTION OF USER INPUT:**

Using WASD to move, Space to stop, and the mouse to point where the ship faces, click to shoot, and number keys to switch between weapons.

**DESCRIPTION OF PROGRAM OUTPUT:**

A spaceship will move along and “mine” asteroids

<https://github.com/Dennispandea/Star-Siege>

|  |  |
| --- | --- |
| Release Name | New incremental features of this release (or Scratch specification) |
| **A-0.1** | **Generates a tiled map with asteroids and background. Clicking on asteroid prints out name and id, use mouse to click and drag around map** |
| **Player-EntTest** | **Create a player that can move in all 4 directions with WASD** |
| **Player-MapFix** | **Make the camera follow the player** |
| **PR-V1.01** | **First release with debugging to find coordinates and** |
| **EntityTest/Ast** | **Randomly generate asteroids with random textures as entities/sprites** |
| **A-1.2** | **Randomly generating asteroids with textures according to the resource they contain** |
| **ScreenScratch** | **Adding a main menu that would let me have options and settings for the game.** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Work in Progress Report**

**Major developments/breakthroughs(reference specific code please):**

Being able to get the player from the entity arraylist

**Major Challenges/setbacks( reference specific code please):**

Grabbing entities from main entity array

**Any modifications to your specifications/release schedule:**

At least one stable major version per month

**Description of your scratch/test program:**

**Describe the generic concept you needed to test out:**

Adding 1000+ asteroids to a map using an arraylist and being able to modify every single one separately

**Source any web site/book that helped you with that concept:**

**Describe the code and the lesson that you learned from it:**

Having multiple batches and cameras which i didn’t end up using

**Describe any challenges that you enjoyed in integrating this scratch code into your major project:**

Changing the Asteroids from Tiledmap to actual Sprites/entities was fun and rewarding when it finally worked

With each WIP, you will be submitting EVERYTHING. Organization is key. When I go to the groupwork folder**, I should see your project submitted in the following format:**

YourLastName: Under this folder will be the following folders:

**Documents**: It will hold all of your documents: journal, WIP, Specs, Release schedule, list of sources, and all the other documents that will be submitted in your final project.

**Releases**: There will be a folder for each release, with one folder CLEARLY telling me that it is the latest, stable release.

**Scratch**: There will be a folder for each scratch concept that you tested before you integrated it into your final project.

**Peer Assessment:**

**Work in Progress Marking Scheme**

|  |  |
| --- | --- |
| Mark | Description |
| 10 | Great progress with release.  Clear scratch and integration.  Solid documentation and list of sources – clear journal. |
| 9 | Great work, but either scratch, release, or documentation is lacking. |
| 8 | 2 of the 3 aspects are lacking |
| 7 | All 3 are lacking, or 1 is missing, or did not follow suggestions from previous report. |
| 6 | Very little progress shown, or did not follow suggestions from previous report. |
| 5 | Warning: this effort will NOT pass in final report |
| 4 | Little sign of work |
| 3 | Less sign of work |
| 2 | Project submitted with NO progress |
| 1 | Where is it??? I cannot give you a zero, even though you probably earned a zero. |
|  |  |

Major Programming Project Package: Table of Contents

1. Table of Contents: with a full listing of all file names(program, documents, presentation) and path. Make sure you clearly tell me which program to run.

2. Progression of specifications: initial specifications, release schedule, and Work in Progress reports.

3. List, and description of your test/scratch programs. There will be at least 5 of them.

4. Program overview/introduction.

5. Disclaimer/ list of known bugs

6. User Manual – your manual should START with clear directions of where your project is amongst all of the folders. Your manual should include screen captures to clarify your instructions.

7. Journal for the project.

8. List of sources that you used in order to make your program work.

9. Notes to future programmers of your project: directions that would make it better.

10. Lesson of your favourite programming technique that was not taught from my notes: consider this to be a more detailed description of your favourite scratch/test program.

11. Electronically submit **everything**, including your entire programming project.

Major Programming Project Marking Sheet

|  |  |
| --- | --- |
| Topic: | Programmer: |

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Description** | **Marks** | **Out of** |
| Package  Clarity | The package was complete and in order.  All sections were clear, and easily understood. |  | 10 |
| Bugs, Specs and journal | All three were clear and insightful. |  | 10 |
| Program:  Challenging | The problem that you chose to solve proved to be challenging. |  | 15 |
| Program:  Clear front end | Your program interface proved to be clear to the end user: both for data input and program output/display. The **manual** was an excellent support. |  | 15 |
| Program:  Clear code | Your code was clear, concise, well-documented, and structured.  Clear **comments** throughout: especially to give an overview of a method, class, or frequently used variables. |  | 15 |
| Program:  techniques | You used the best programming techniques to solve the problem. |  | 15 |
| Lesson | You shared a new programming technique that was not in my notes. You clearly showed how it could be used, and why all high school programmers should be using it. |  | 10 |
| List of sources | Clear links as to where you learned and borrowed programming techniques. This will be an entire list of the sites that you used in order to make your program better. Clearly reference the parts of your code that benefited from these sites. Your code will have comments of your sources as well. |  | 5 |
| Notes to future programmers | These notes give great direction to future programmers as to where there is room for improvement, and possibly how to fix the problems (if only you had more time!). |  | 5 |

|  |  |
| --- | --- |
| **Criteria** | **Comments** |
| Package  Clarity |  |
| Bugs, Specs and journal |  |
| Program:  Challenging |  |
| Program:  Clear front end |  |
| Program:  Clear code |  |
| Program:  Techniques |  |
| Lesson |  |
| List of sources |  |
| Notes to future programmers |  |