Final Project Idea
Asteroids

Philippe Lumpkin and Nicholas Pica

Abstract

Our purpose is to create an entertaining arcade game for our classic game lovers.Our goal is to create an Asteroids remake. We intend to stick to the original as much as possible, however we may implement some of our on touches.

1. Introduction

For this project we plan to remake the well known game Asteroids. This 1979 game was developed by Lyle Rains, Ed Logg and Dominic Walsh. Asteroids was a major success, being one of the first of the golden age of arcade games. Atari, the game developers, sold over 70,000 arcade cabinets to arcade centers across the US. The game was designed using the Atari system and rendered on a vector 2-D display.

Our target audience would be those who enjoy a classic arcade game. The purpose of the project is simply to entertain the person playing. We hope to accomplish a remake of the base game along with our own tweaks. For example, an option to implement high scores stored in a database that the user can view.

1.1. Subsection Heading Here

Occasionally you need to break your sections into separate parts, you will likely not need a subsection for every section

1.1.1. Subsubsection Heading Here. Occasionally you will need to break your subsections into separate parts, if you find yourself using this often, you're likely going overboard. Don't try to go any lower down than this...

1.2. Background

In the middle of the screen there is a spaceship that shoots lasers at incoming asteroids in which the user decides which direction the spaceship will point. The object of the game is for the user to shoot the lasers at the asteroids and destroy them before they hit the ship.

1.3. Challenges

A challenge we expect to encounter would be implementing how objects interact with each other on the screen. Another challenge would be designing objects that can move on the screen, which does include text.

2. Scope

We are hoping to create a full Asteroids remake. This means that there is a point system based on the amount of asteroids shot. There are a set amount of lives. Asteroids will appear from random points on the screen and move in a linear patterns. If the spaceship is hit by an asteroid then a life is lost and the spaceship is placed back in the center in the screen without the previous asteroids being cleared.

First we want to create a moving background to give a more pleasing look. We also want to make the asteroids break apart into smaller asteroids after destruction. Another possible goal would be to add more color to the game.

2.1. Requirements

As part of fleshing out the scope of your requirements, you'll also need to keep in mind both your functional and non-functional requirements. These should be listed, and explained in detail as necessary. Use this area to explain how you gathered these requirements.

Use Case ID	Use Case Name	Primary Actor	Complexity	Priority
1	Highscore	Player	Easy	1
2	Start	Player	Easy	1
3	Database	Player	Easy	1

TABLE 1. SAMPLE USE CASE TABLE

2.1.1. Functional.

- The game will record high scores. The user will be prompted at the end of the game to create a gamer tag which will then be added to the high score database.
- The user will have 4 lives. When the game starts will, 4 lives will be added.
- The game will run when the user hits the start button
- When the user presses the high score button the screen will display the high score list.
- The game can be exited when the user presses the exit button.

2.1.2. Non-Functional.

- The game will run smoothly without screen tearing.
- The game's high score database will have a capacity of 50 scores.
- The high score database will not be corrupted if he game crashes.

2.2. Use Cases

Use Case Number: 1
Use Case Name: Highscore
Description: A player wants to access the highscore screen. The player then clicks on the highscore button on the start screen. Then the database is accessed and then displayed for the user to view.

1) Actor can select the "Highscores" button.
2) Accesses the highscore database.
3) Displays the data from the database is a easy to read format

Termination Outcome: The database is is displayed for the user.
Use Case Number: 2
Use Case Name: Start
Description: A player runs the program and is presented with the start screen. The player would then only have to click on the "start" button to begin the game. After clicking start the user is then shown the game screen where the game starts up.

1) Actor can press the start button
2) Runs the function that begins the game.
3) Displays the play screen.

Termination Outcome: The game begins for the user to enjoy.
Use Case Number: 3
Use Case Name: Database
Description: A player finishes the game an is prompted to input their initials. After the player has input their initials then the screen switches to the highscores screen and displays the top 9 scores.

1) At the end of the game, the player is prompted to enter a 3 character name.
2) Accesses the Highscore database.
3) If the player scored enough to be in the top 9, then their score gets added to the database.
4) Displays the top 9 scores from the highscore database.

Termination Outcome: The user sees their new score on the screen.

You will then need to continue to flesh out all use cases you have identified for your project.

2.3. Interface Mockups

3. Project Timeline

Go back to your notes and look up a typical project development life cycle for the Waterfall approach. How will you follow this life cycle over the remainder of this semester? This will usually involve a chart showing your proposed timeline, with specific milestones plotted out. Make sure you have deliverable dates from the course schedule listed, with a plan to meet them (NOTE: these are generally optimistic deadlines).

Figure 1. The is what the start screen looks like

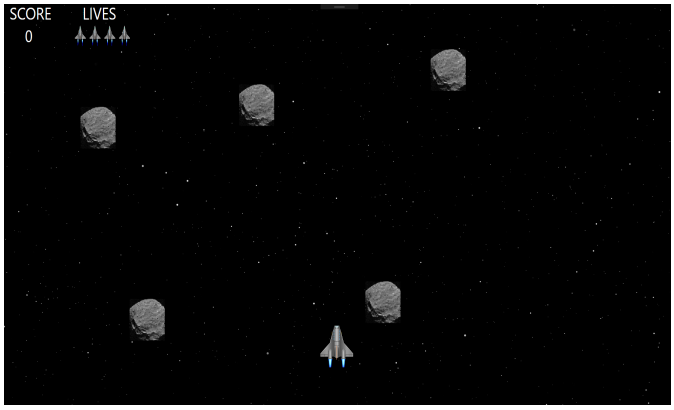

Figure 2. The is what the game screen looks like

Figure 3. The is what the highscore screen looks like

4. Project Structure

At first, this will be a little empty (it will need to be filled in by the time you turn in your final report). This is your chance to discuss all of your design decisions (consider this the README's big brother).

4.1. UML Outline

Show the full structure of your program. Make sure to keep on updating this section as your project evolves (you often start out with one plan, but end up modifying things as you move along). As a note, while Dia fails miserably at

generating pdfs (probably my fault), I have had much success with png files. Make sure to wrap your images in a `figure` environment, and to reference with the `ref` command. For example, see Figure 4.

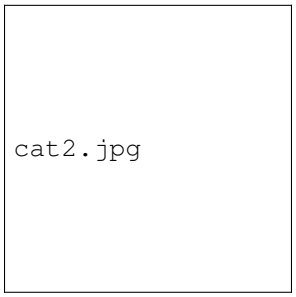

Figure 4. Your figures should be in the *figure* environment, and have captions. Should also be of diagrams pertaining to your project, not random internet kittens

4.2. Design Patterns Used

Make sure to actually use at least 2 design patterns from this class. This is not normally part of such documentation, but largely just specific to this class – I want to see you use the patterns!

5. Results

This section will start out a little vague, but it should grow as your project evolves. With each deliverable you hand in, give me a final summary of where your project stands. By the end, this should be a reflective section discussing how many of your original goals you managed to attain/how many desired use cases you implemented/how many extra features you added.

5.1. Future Work

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.