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

We have a total of 8 requirements for our project. Our first requirement is to have asteroids that appear randomly on screen and continue to move until destroyed. The next requirement is to implement a high score table that list the top 10 scores and that can be viewed by the user and updated when a higher score is made. This will be done in the same theme as the 1979 version entailed, displaying the top 10 scores and having a maximum of only 3 letters as the name. Our third requirement is to create a ship that moves and spins around the screen effortlessly. To install a working laser would be our

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. USE CASES

forth requirement. The laser will be fired from the front of the ship and will damage/destroy hostile entities upon impact. The fifth requirement is to allow the ship, the laser beams, and hostile entities to exit from one side of the screen and appear on the adjacent side of the screen. This was also implemented in the original game by accident, however everyone liked in and so they kept it, so we intend to keep it to. The sixth and arguably the most important requirement would be to implement hit detection. This will be applied to the ship, the laser, and any hostile entities. The seventh requirement is to implement the breaking of the asteroids into smaller pieces. This will happen 2 times until finally destroyed as it was in the original game. The final requirement is to add custom audio files for the sounds effect in the game. We decided to add these custom files to bring a unique feel to the once futuristic sounding game.

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.
- Asteroids that appear and move across the screen.
- A ship that can move and spin around and shoot and hostile entities.
- A laser that is shot from the front of the ship that destroys hostile entities upon impact.
- ships will break apart into smaller pieces after being shot. This will happen a total of 2 times and then they will be completely destroyed.

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.

Figure 1. The is what the start screen looks like

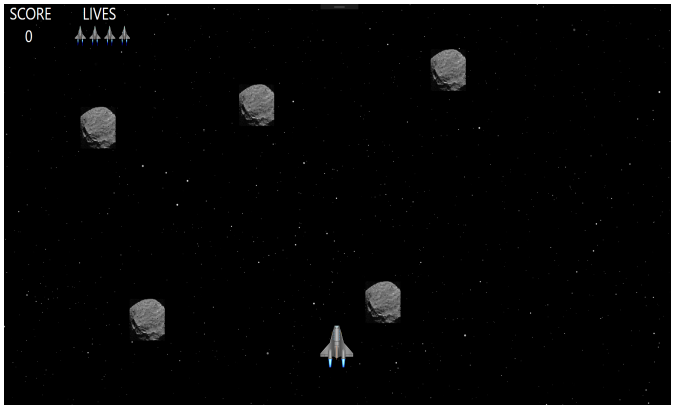

Figure 2. The is what the game screen looks like

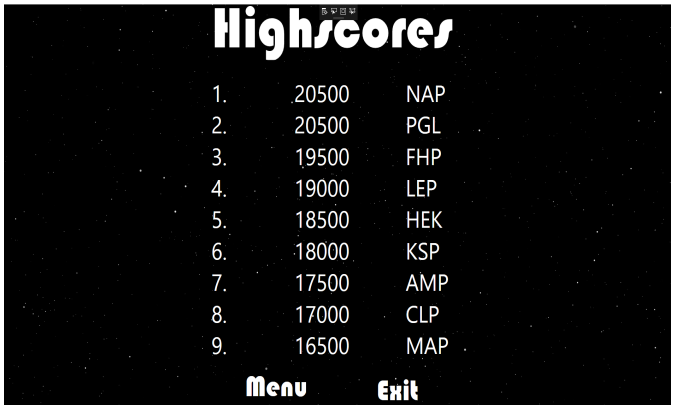

Figure 3. The is what the highscore screen looks like

2.3. Interface Mockups

For example, see Figures above **??** (Start Screen), **??** (Game Screen), or **??** (High Score Screen).

3. Project Timeline

Go to **??** for the project timeline.

4. Project Structure

Our project is centered around a WPF window that once certain buttons are pressed or specified events occur then the window will alter its contents into a desired page. The Main window will open and start the ”StartScreen” page that includes the buttons Start, Highscores, and Exit. Based on the button press the buttons will either lead to the ”GameScreen” page, ”HighScoresScreen” page, or exit the entire window that the project is hosted inside. This was done so that navigation was pretty straight forward and seemless. We found that this way prevented us from having to open and close windows when a new page was needed. We have created seperate classes for both the Asteroids and the Spaceship to make it easier to understand where all of the information is stored for the program. Our project incorporates a small database to store the information for the highscores.

4.1. UML Outline

Here is the UML outline for the Asteroids project: **??**.

4.2. Design Patterns Used

The first pattern we decided to specifically implement was the Observer Pattern. The idea is that there will be a constant observer ran over the spaceship that will watch for the isShipDestroyed bool to come back true. When that occurs, the observer will then update the GameScreen lives to one less. The second design pattern we are using is the State pattern. We implemented this in a way that the game could be paused and resumed by the user. The code would determine whether the specific key was pressed and if it was then the games objects and timers would be paused.

5. Results

Currently we have our 3 out of 8 requirements completely finished. Our project currently has a high scores table which in implemented through access. We also have asteroids that appear randomly on screen and move to the other side off the screen, however we do not yet have hit detection or the feature that allows objects to move of the screen and appear back on the adjacent side. We also have a ship that moves and spins effortlessly within the screen. We are working now on the 8th, 4th and 6th this week and plan to work on the last two (the 7th and 5th) from April 18th-24th.

5.1. Future Work

References

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

Requirements	Design	Implement	Verification	Maintenance
1) Allow asteroids to appear randomly on screen, moving across to the other side.	We plan to have several asteroids appear in random locations at random intervals moving across the game screen.	Currently we have 5 asteroids appearing randomly and moving across the screen.	We have achieved what we have set out to do with this requirement. However, we may choose to implement more than just 5 asteroids.	This is working just how it was expected to run.
2) To implement a high score table using an Access database.	We plan to create a database with a very similar look and feel as the high score table looked in the original game.	We have implemented a dummy database for the moment so we can see how it works and what it will look like when finished.	The dummy table looks similar to he way it did in the original. We might, however, find a better font and change the size of the text to fit the screen better.	The database is connected and displays the information as expected.
3) To create a ship that moves and spins effortlessly throughout the game screen.	We plan to create a ship based on a "lined-paper" theme.	We have currently implemented a ship that moves and spins, we now just need to add the jpg file to make it have the lined-paper look.	The ship is implemented as planned, except for the theme.	The ship interacts with the game screen as intended.
4) To implement a working laser that works well with hit detection.	We plan to create a custom laser following the "lined-paper look.			
5) Allow asteroids and the player to be able to go off one side of the screen and appear on the adjacent side.	We plan to have this work just as it worked in the original game.			
6) To create hit detection for the asteroids, the ship, bosses, aliens and the laser.	This will be a very simple hit-box implementation. When the laser hits one of these items it will take damage, blow-up, or break into smaller pieces.			
7) Allow the asteroids to break into smaller pieces after hitting them with laser.	This will be achieved by allowing the asteroids to break into smaller pieces 3 times as done in the original.			
8) Create custom audio files.	We plan to record own voices and upload them to the game.			

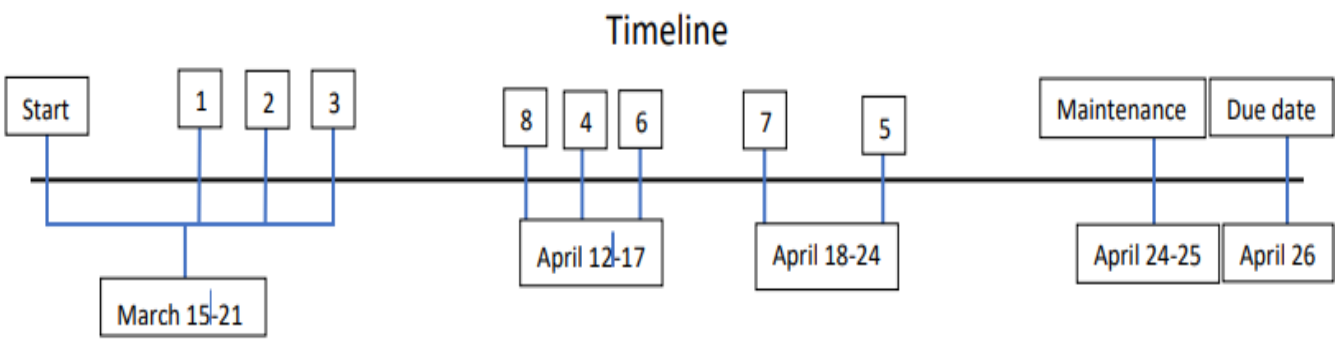

Figure 4. Our timeline for this project

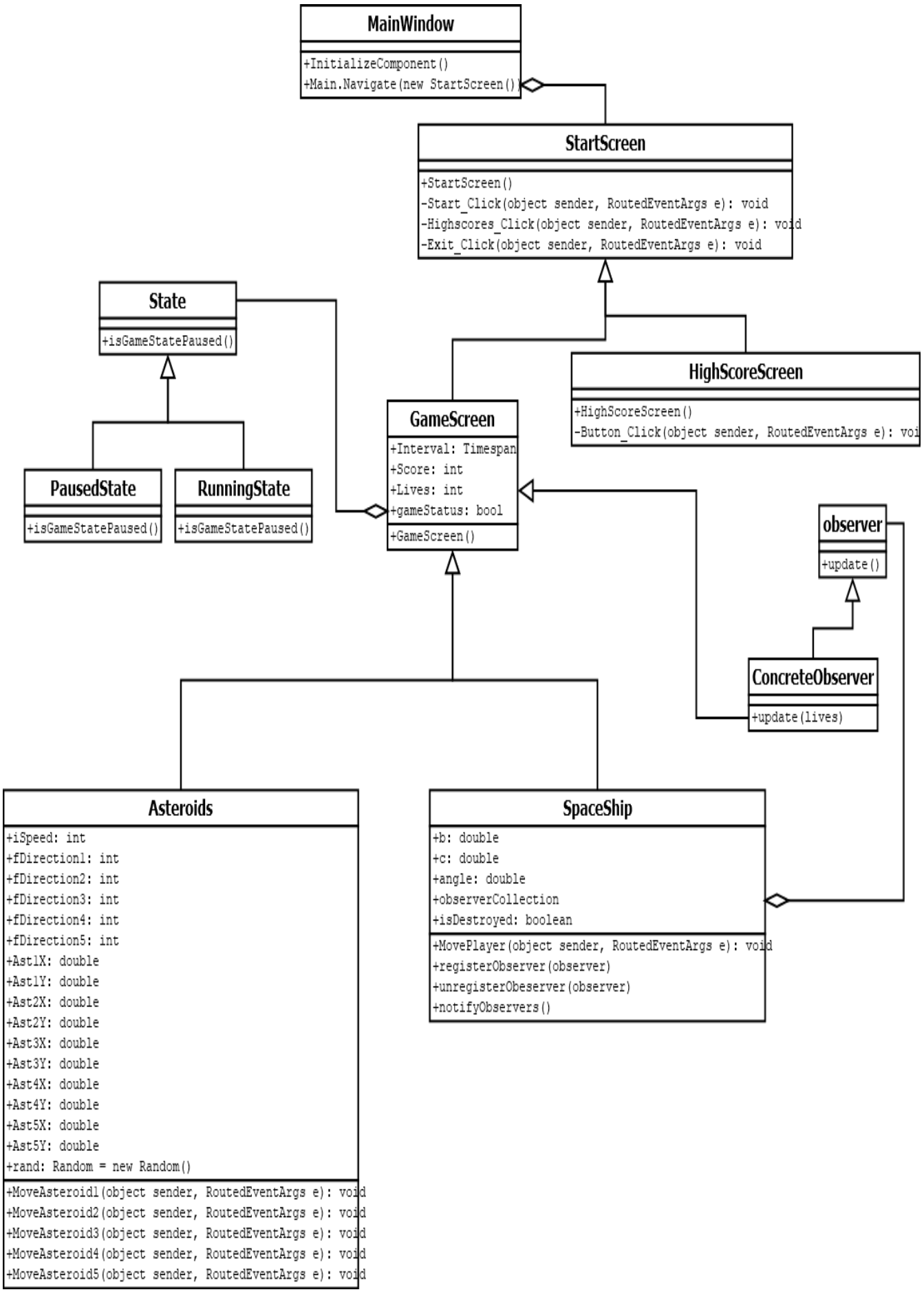

Figure 5. Our UML layout for Asteroids