Asteroids Nea

Mohammed saleh sajjad

Contents

[**Analysis -** 1](#_Toc160238058)

[Outline of the project: 1](#_Toc160238059)

[Stakeholders: 1](#_Toc160238060)

[ Abstraction 1](#_Toc160238061)

[Thinking ahead: 2](#_Toc160238062)

[Thinking procedurally: 2](#_Toc160238063)

[Breakdown of the problem (decomposition): 2](#_Toc160238064)

[Written decomposition: 3](#_Toc160238065)

[Thinking logically: 3](#_Toc160238066)

[Thinking Concurrently: 4](#_Toc160238067)

[Conclusion: 4](#_Toc160238068)

[Research 4](#_Toc160238069)

[Asking classmates questions as to what people want in the game (30/11/23): 4](#_Toc160238070)

[The questions I will ask are as follows 4](#_Toc160238071)

[Asking classmates Question (answering the questions: 5](#_Toc160238072)

[How do the different features of Asteroids Affect the gameplay 8](#_Toc160238073)

[second interview 9](#_Toc160238074)

[Limitations 9](#_Toc160238075)

[Features ideas: 9](#_Toc160238076)

[ESSENTIAL FEATURES: 9](#_Toc160238077)

[Justification 9](#_Toc160238078)

[Additional Features 10](#_Toc160238079)

[Justification: 10](#_Toc160238080)

[Limitations 10](#_Toc160238081)

[Hardware requirments 10](#_Toc160238082)

[HARDWARE 10](#_Toc160238083)

[Justification 10](#_Toc160238084)

[Software Requirement 11](#_Toc160238085)

[Success criteria 12](#_Toc160238086)

# **Analysis -**

## Outline of the project:

The purpose of this program is to create a game like the 90s arcade game asteroids, but with a twist. The direction of which the asteroids move in will not be based on off a random movement instead I will use machine learning to predict where the asteroid will end up based on the players position. Also (thinking about this) allowing a computer to play as player 2 and will help the player by making the game easier for players who might struggle early on while playing. Players will be able to unlock ships by being the first to shoot them down

## Stakeholders:

The target audience for my program will be children who want to experience what their parents experienced as kids and adults who want to experience their childhood in a new light as see how the game they played as kids has evolved. My stakeholders would be those who are interested in retro gaming but also interested in a new environment that newer games bring. Also, those who enjoy playing with friends and can compete with them for fun.

The stakeholders for my project are ok with games no matter the difficulty and can adapt to new controls and settings easily.

My game is built for primarily for the newer generation to enjoy games that their parents once played but with aspects they also enjoy. This gives them a chance to bond over games they had once played and play now.

## Abstraction

* Having a score board system that will *only* increase when an asteroid gets hit.
* When the scoreboard hits a certain point for example 1000
* The number of asteroids spawned every 2 seconds will increase making the game more challenging as it goes on.
* Allow a pause menu.
* Allow the user to exit the game through the menu.
* Simple sound effects

For example, when a player shoots a bullet a pew sound plays.

* An option to play as player 1 or player 2.
* A limited number of sprites
* A life counter for player 1 and player 2
* End the game when both players run out of lives.
* Game over screen.
* A space background imported from files.

## Thinking ahead:

* Inputs for the game would include space bar to shoot for player 1.
* Right control key to shoot for player 2
* A and D to control the directions of the ship for player 1
* Left arrow and right arrow to control the direction for player 2.
* W and S to go forward and backwards for player
* Up arrow and Down arrow to go forward and backwards for player 2.
* **What are the outputs for the game?**

## Thinking procedurally:

This is like Thinking ahead this gives the issue more time and is easier to work with making the problem more efficient.

## Breakdown of the problem (decomposition):

Asteroids

Control

When game ended save score

Player 2

Player 1

menu

Up arrow – forward

Down arrow – backwords

Left arrow- rotate left.

Right arrows rotate right.

W- forward

S – backwards

D – right rotate

A-Left rotate

Two player

One player

Scoring

Main game

Players

Every 1000

Points difficulty

Increases

Every asteroid hit

Increase score.

## Written decomposition:

The game has been broken into 4 main parts this will be easier to identify our key issues.

***Players***- the user will first be given a screen selecting player 1 or player 2 this will determine what key inputs the game needs to recognise so if player one is selected it won’t keep trying to read player 2’s inputs and if player 2 is selected then the game can recall both player 1 and player 2 sprites and judge them individually with their lives, score that way players can compete to see who reaches a higher score.

***Control***-the controls vary dependent on player selection player 1 (red spaceship) will always use the control command WASD and player 2 if selected will use the arrow keys each player will have an independent shoot key space for player 1 and right control for player 2 if both players have inputs simultaneously the game will still be able to pick up and base each character’s sprite depending on the input.

***Main Game***- the main game when launched will present a menu with a demo of the game running in the background the menu will include player section, option to quit the game, and ship select (choosing what colour ship the player will use) after the player selects there player and ship the game will start the player will spawn in the middle of the screen with asteroids coming at him when a bullet is shot the asteroid is broken up into two medium sized asteroids and if the medium sized asteroid is shot then two smaller asteroids

***Score***- the players score will determine at what speed the asteroids move at if they get 1000 the speeds in which the asteroids spawn in and they move increase. This will make it harder for players to achieve higher scores making the game a challenge for newer players. When an asteroid is shot the players, score will increase depending on the size of the asteroid smaller asteroids will be worth more as they are harder to hit as their hitbox is very small. The points for each asteroid will vary.

## Thinking logically:

This helps me understand how I could solve each step of my problem for example when player 1 is selection it will use the player 1 game state if a target is hit a branch between both asteroids will be created. Another example would be when the bullet is collision. There are multiple branches in collisions as it must detect what happens to the asteroid if collision happens if player gets hit by asteroid should the asteroid sprite be deleted or should it continue. Or if both players bullet collides what happens to the bullet. If the bullet hits the small asteroid should the asteroid be deleted.

Others include when a player clicks on a movement key. If the player pushes W the thrust of the asteroid should increase to value of 0.2 the reason the value is so small is because the way speed is calculated in Pygame is insanely quick so to slow things down a number below 0.5 is recommended.

## Thinking Concurrently:

This is useful for thinking about how to maximise efficiency within my program updating, running the main game loop, displaying the background and detecting collision will all happen at the same time.

## Conclusion:

The previous examples use methods of computational thinking techniques to analyse different problems that could occur within the code by simplifying them and making them all laid out easer to see the steps will be easier to complete.

If the problem is successfully solved using a computer program the stakeholders will be able to use the program to be able to play a Game where shooting asteroids raining from the air and can be formed as nostalgia for the old generation and a new experience for the new generation

## Research

### Asking classmates questions as to what people want in the game (30/11/23):

An essential first stage in building the game is determining which features would be popular to incorporate. To learn more about these prospective features, I have decided to speak with potential stakeholders. Since these people fall into the target audience's age range and have an interest in playing video games, the stakeholders that were interviewed were their fellow computer science classmates. The following were the queries I posed:

### The questions I will ask are as follows

#### What makes a game good:

What type of game does the target audience enjoy most for example platformers, retro, open world?

Is character selection something important to you -> rate 1-5 (1 Not at all – 5 very important)

What other aspects could be important that you want to ask people about?

* Character selection
* Graphics
* Sound/audio.
* difficulty

How many people do you usually play with (are you more likely to play on your own or with friends?)

Is It guided for the intended audience?

What is the theme of the game? -> what game themes are your favourite?

What kind of personalisation would you like incorporated?

What kind of difficulty would you like for the game:

### Asking classmates Question (answering the questions:

#### Classmates answering Questions:

What type of game does the target audience enjoy?

platformer– This would give a better experienced the player would get to move around and interact with different elements of the game.

Is character selection something important?

Of course, Character selection is a key part game bearing mind that most games different characters have different buffs making the game easier as you play.

How many people would you like to play with?

As many as possible,

What is the guided audience for the game?

12+

What is the theme of your favourite game?

shooter

What difficulty would you be able to contend with?

Any difficulty above my skill level, this will enable me to want to play better and improve my skill level so I will be able to be a better player.

#### Review of interview:

Peoples answers to my questions where interesting as I expected people to like more modern open World games that would be difficult to design. But instead, people don’t always prefer what’s complex instead they enjoy things that fits there personality instead. The type of game defines the person’s type of personality. People also didn’t objectively enjoy a selected difficulty. They care more about the overall difficulty in a selected game.

Multiplayer is also a question many people answered as more than 1. Because of this I have inferred friendly competition and enjoying moments with other people is what makes a huge part of gaming.

The effect of character design is important according to most people that answered the survey, they linked character design to How they feel while playing. As a creator I agree it’s the reason most people enjoy games like Nintendo’s super smash bros. So, I have decided to incorporate some character selection.

#### Research on the game Asteroids:



Image 1: Asteroids console, https://videogamehistory.fandom.com/wiki/Asteroids

Asteroids is an old arcade game in which the player is thin ship with 3 lives each the aim is to shoot as many asteroids as possible but when a player shoots an asteroid it splits into two smaller asteroids each size asteroid is worth a different number of points. The aim of the game is to score as high as possible, so the game is never ending.

The game is concise to one single game mode.

Developed by Atari in 1979, Asteroids is one of the most popular video games of all time. It was originally created for old-school arcade consoles. Asteroids sold 70,000 units in the United States and appeared in multiple arcades throughout the country.

The Game had replaced space invaders as the number 1 selling game and had.

The original Game had a bug known as the “lurking exploit”.

This was when players sat in the top corner of the screen making it less likely of ever getting hit as only one or two rocks where at play at once. Arcade owner complained to Atari as they were losing revenue because of this exploit thus Atari issued a patch to the game and also changed the ways they tested games in the future to avoid issues like this again.

A black background with white lines and dots

Description automatically generated

One of the games main features is displaying a scoreboard that updates every time the asteroid is destroyed. The amount the scoreboard increases changes depending on which size of the asteroid Is hit. The top left. screen also displays a live counter in the form of a sprite every time the player is hit one sprite is removed from the top left corner of the screen.



The original game also includes asteroids in 3 different sizes small medium and large. Every time a bullet collides with the asteroid it splits into two smaller versions of themselves. For e.g. if a big asteroid is hit it would become two medium asteroids. The nature of the Asteroid is random meaning it’s not objectively targeting the player instead heading in a random directional angle.

A screen shot of a video game

Description automatically generated

The player ship in the original asteroids game uses a thrust mechanism so when the player lets go of the forward key the ship will still move forward a certain length. The player is the main character in which the user controls the sprite with a joystick in the original arcade version. However, this was changed to a controller with a dpad in the Atari home release version. When the sprite dies the player counter goes down. When the sprite counter hits 0 the game ends.

### How do the different features of Asteroids Affect the gameplay

A failure in the game: Things like game over sounds effects and a decrease in lives will make the player want to play more carefully and strategically. Also, a punishment such as a powerup that handicaps the player will make not only the game harder but allow the player to want to play better and learn from there past mistakes.

Sound effects: Sounds effects such as an explosion at the time of collision can increase the action of the game making the game more exciting for the user and increasing likelihood of the user returning to play as they are visually satisfied with how the game displays itself.

Splitting on collision: The unique feature of asteroids is that the asteroids split on collision. Usually, games need what’s called a wow factor, something that makes the game thrilling and separates itself from the rest of the competition. The wow factor in this game makes people want to play as it causes a surprise to the player and when split each asteroid goes in a separate but random direction. This makes the player adapt to the game mechanics and can’t rely solely on past game experience.

Player thrusting after forward key input is released: An interesting concept to the game asteroids is the fact that the players sprite moves after letting go of be the movement input. This makes the game challenging and harder to control. This hurdle can’t simply be done by a first timer instead one must practise and get used to the feature.

### second interview

### Limitations

|  |  |  |
| --- | --- | --- |
| limitation | JUSTIFICATION |  |
| compiler | How fast the compiler can Run subroutines |  |
| Pygame | Pygame doesn’t have every function meaning some functions need to Coded personally |  |
| Load times | How Fast the computer can open the software |  |
| Os | Its an exe file so only Windows users can run the file |  |
| Peripherals | If the user doesn’t own The required peripherals he cant Control the game |  |
| System specs | If the system doesn’t meet the required hardware requirements |  |
| bandwidth | (Potential online game) if the user has an unstable internet connection they wont be able to connect |  |
| Logarithms | The More iterations added to the game The larger the time complexity This will Put a hindrance On how fast certain Commands will run |  |

### Features ideas:

|  |  |  |
| --- | --- | --- |
| ESSENTIAL FEATURES: | Justification |  |
| Control using arrows/wasd. | To control the sprites movement for sprites, separate for different characters |  |
| Shoot using spacebar/ctrl. | Shoot a laser in a direction using spacebar to destroy the asteroid |  |
| Asteroids move randomly around the screen. | Make the game challenging/ unexpected |  |
| Update scoreboard when asteroid is hit. | Making a scoreboard gives a game that doesn’t end a purpose |  |
| Play firing animation when spacebar is pressed. | Gives the game a style.  that is needed to entice players into continuing playing. |  |
| Player lives displayed. | Displays the players current lives so the player knows how many lives he has |  |
| START MENU | Allow the player to select which game mode there playing |  |
| Clickable buttons | Ease of access |  |
| Additional Features | Justification: | Limitations |
| Scoreboard displayed on death. | Players know their final score improving their experience | Chance the text file becomes corrupted |
| PvP mode: | Two players can fight each other | May lead to an excessive amount file size and CPU usage. This could limit people with low disk space. |
| Explosion effect after collision on both ship and asteroid | Gives the game a nice visual aspect | CPU usage and memory will increase as the computer is having to complete the animation loop. This may limit people who don’t have a high clock speed or enough primary memory |
| Multiplayer | Allows the integrated of players to play with each other and enjoy competing for score | CPU usage will increase as now  the computer will have to deal with multiple inputs at the same time.  A low bit rate OS might struggle  With dealing with scheduling it. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

### Hardware requirments

|  |  |
| --- | --- |
| HARDWARE | Justification |
| 2.4 ghz cpu | High clock speed in order to make sure the program runs efficiently without delay |
| Monitor | To display a graphical output |
| Graphics card 4GB | To take off heavy load from the CPU and allows users who don’t have integrated graphics on their CPU to be able to display a visual output. 4gb as our game doesn’t need as much Vram as modern games but still needs some to process the graphic output without delay so an input from the keyboard for example won’t feel slow |
| Speakers | To output sound effects displayed by the game |
| 30 mb of diskspace | 28 mb of download space and as the text file that stores the high score is constantly changing an extra 2 mb to be on the safe side is optimal |
| Keyboard | Input an action into the game for an action to occur for example w to make the sprite move forward |
| Mouse | To click on buttons that select which game mode / settings |
| 7mb of cache | Cache allows the CPU to quickly temporary store files this game doesn’t take a lot of cache but still needs come in order to operate efficiently |
| 4Gb ram | The average computer has up word of 4gb in today’s day and age and with this amount of ram some of it can be used by the game we are running but also by other services the computer needs |

### Software Requirement

|  |  |
| --- | --- |
| Software | Justification |
| Pygame | The library used is sometimes found in an os but the user may Have to download it. Currently, the Exe file auto runs and downloads Pygame if needed so the user doesn’t need to manually download it. They only need to keep the library |
| Windows os | The exe file that is ran is a type of file only available on windows and not other os’s like mac or Linux |
| The folder of python files and the exe | Without all the python files the program simply wont run and will result in an error the computer needs all of the files that are present in the folder to be able to run when the exe file is ran |

## Success criteria

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Criteria | Justification | Y/N |
| 01 | Title | Allows people to recognise one of the most famous games | Y |
| 02 | Start menu | Allows the user to select player count / change settings | Y |
| 03 | Button | Allows the player to click allowing accessibility | Y |
| 04 | Pause menu | Allows either player to pause mid game | Y |
| 05 | Auto spawning asteroids | Asteroids spawn automatically in random points increasing accessibility | Y |
| 06 | Auto scrolling background | Allows an immersive experience | Y |
| 07 | WASD/arrow keys for movement | Allows the player to move in any direction allowing the game completely in the users control | Y |
| 08 | High score | Saves the high score of the user | Y |
| 09 | Scoreboard | Tracks how many current points the user has and updates it when an asteroid is destroyed | Y |
| 10 | Player count | Tracks how many lives the player has goes down when hit by an asteroid | Y |
| 11 | Settings menu | Allows the user to change the controls or music S | Y |
| 12 | Splitting asteroids | Asteroids split into two smaller ones when hit making a challenge for the user | Y |
| 13 | Ice Powerup | When a player gets this all asteroids are frozen for 5 seconds making them easier to hit | N |
| 14 | Rapid fire powerup | Doubles shooting speed for 7 seconds | N |
| 15 | Kill streak | If 10 asteroids are shot in a certain amount of time points added to score board are doubled for 15 seconds | N |

# Design

## Systems Diagram

## Explanation of each module

Scoreboard:

Keeps track of score for each player. Also saves high score so next time the game is played The user can continue to try and beat there highest score.

Paused:

When the T button is pressed player 1 will pause the Game. When / is pressed player 2 will have paused the game. I wanted the pause button to be close to the movement buttons and also wanted separate pause menus so users know who paused the Game.

Menu:

One player: By selecting the one player option. The game state (playing) will change from False to True and player two will stay False.

Two players By selection two players the game state playing will change from False to true And the player 2 Boolean will change from False to True.

Settings:

By selecting setting users will be able to change movement/Game controls. Also toggle sound effects on or off. Menu state will become False and Setting state will become True.

Game over:

When Game over is displayed the playing state becomes False. User can either restart by pressing a button or go back to the main menu. If restart, then game state becomes true again if main menu. The main menu Boolean becomes True.

Startup:

When the game loads the Main menu Boolean becomes true straight away.

Playing:

At the start of each game. Current score is set to 0 player lives and position are reset to the original amount.

Sound:

IF sound effects are toggled on. Sound effects will play in moments such as a player shooting or an asteroid getting destroyed. if off nothing happens.

## Design Objectives

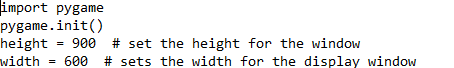
1. Aesthetic considerations:
   1. Screen size will be relatively large – (900x600 pixels)
   2. There will be the title of the game during the start up
   3. There will be buttons in the menu state
   4. Cosmic background
2. Input considerations:
   1. Player will be able to click buttons to select player mode – (one player or two player)
   2. Player will be able to press keys to move and shoot from their ships
   3. Player will be able to press ‘p’ to pause the game
   4. Player will be able to press ‘esc’ to escape back to the menu
3. Processing considerations:
   1. If the game starts consuming a lot of processing power of CPU, then it should stop running the game immediately to prevent the CPU from overheating
4. Output considerations:
   1. Asteroid splits after being shot by the player
   2. Score board is displayed at the end of game
   3. Game over is displayed

## Algorithms (flowcharts)

# Development of the coded solution

## Display creation

I first decided to initialise a working window that loads the game when running. By using pygames in built function pygame.display



The first step was importing the pygame library then initalising it. Then i created 2 variables hight and width and assigned them to the height and width of the window i wanted the game to be played at. I decided on 900x600 as most monitors are able to run this resolution and a large display isn't needed for the program we are designing as making the space more limited poses more of a challenge.

Next by using pygames function display.set\_mode() i can pass through my height and width variables into the function and output a working window. Display.set\_icon sets