UNIVERSITY OF MALTA
FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
Department of Computer Science

CPS3231 (Computer Graphics)
Year: 2023

Title: Asteroids

Author(s): Keith Bugeja and Sandro Spina

Instructions:

1. This is an **individual** assignment and carries **100**% of the final CPS3231 grade.
2. You may be required to present your work to an exam board.
3. The report and all related files (including code) must be uploaded to the VLE by the indicated deadline. Before uploading, archive all files into a compressed format such as ZIP. It is your responsibility to ensure that the uploaded file and all contents are valid.
4. Everything you submit must be your original work. All source code will be run through plagiarism detection software. Please read carefully the plagiarism guidelines on the ICT website.
5. Reports (and code) that are difficult to follow due to poor writing-style, organisation or presentation will be penalised.
6. Each individual may be asked to discuss their implementation with the examiners, at which time the program will be executed and the design explained. The outcome may affect the final marking.

Asteroids

Asteroids is a classic arcade game developed and published by Atari, Inc. in 1979. It was one of the most popular and influential games of the Golden Age of Arcade Games, introducing players to a new genre of space shooters. The game is well-regarded for its vector graphics, which rendered objects as wireframes, offering a different aesthetic compared to the raster graphics prevalent during that time.

Gameplay

In Asteroids, players control a spaceship in an asteroid field with the primary objective of shooting and destroying asteroids and saucers while avoiding being hit by them. The spaceship can rotate left and right, thrust forward, and fire projectiles to obliterate obstacles. As players progress, the difficulty level increases with more asteroids and faster, more aggressive saucers appearing on the screen.

Features

The game's minimalist vector graphic design, alongside its endless gameplay and increasing difficulty, made it a staple in the gaming community. It also introduced the concept of a *wrap-around* screen, where objects exiting one edge of the screen reappear on the opposite edge, adding a unique challenge and strategic element to the gameplay.

Objective

The aim of this assignment is to revisit and modernise the classic Asteroids game, enhancing its rendering to align with contemporary graphical advancements while preserving the core gameplay that made the original game a standout title.

Task 1 - Game Logic (35%)

The game logic forms the core of the gameplay experience, dictating player interactions, enemy behaviour, collision reactions, and scoring system.

Player Controls

The player controls are the bridge between the player and the game, facilitating interaction with the game world. They should be intuitive and responsive to ensure an enjoyable gaming experience.

- The player navigates a spaceship capable of rotating and moving forward.
- The spaceship can fire projectiles to eliminate asteroids and saucers.
- A hyperspace jump feature, activated as a last resort, teleports the player to a random position on the screen, adding a layer of strategy and unpredictability.
- An appropriate control scheme should be devised for the game, which could be a combination of keyboard inputs like WASD + space or mouse controls, ensuring a user-friendly interaction.

Enemy Classes

The enemy classes in the game present challenges to the player, making the gameplay experience intriguing and rewarding. Their behaviours are programmed to escalate the game's difficulty as the player progresses. Enemies include asteroids and saucers (UFOs), each with distinct behaviours and sizes.

- **Asteroids**
  - Each wave begins with an increasing number of asteroids: Wave 1 has 5 asteroids, Wave 2 has 6, and so on ($n + 4$ essentially).
  - They vary in size: large, medium, and small, each with distinct angular velocities and speeds.
  - Upon destruction, large asteroids split into two medium ones, and medium asteroids split into two small ones.

- **Saucers**
  - Two types: small and large.
  - Large saucers fire randomly, while small saucers target the player.
  - Firing Logic: Every two seconds, a Fair Bernoulli trial determines whether the saucer shoots.
  - Spawning Logic: Every 10 seconds, a random number between 0 and 1 is generated. A saucer spawns if the number is greater than 0.75. Post a 10K score, small saucers replace large ones in the spawn logic.

Collision Detection and Response

Collision detection and response are fundamental in bringing interactivity and consequences to the game, making the gameplay feel dynamic and responsive.

- Detect and resolve collisions between projectiles, the spaceship, asteroids, and saucers.
- Successful hits on enemies award points, while collisions between the spaceship and enemies result in a loss of life.
- Saucers and their bullets can also collide with asteroids. Both are destroyed upon collision. If the asteroid is not small, it subdivides as usual.

Scoring and Life System

The scoring and life system are crucial for player motivation and challenge, giving feedback on performance and setting the stakes in the gameplay.

- Large saucers: 200 points, small saucers: 1000 points.
- Asteroids: large: 20 points, medium: 50 points, small: 100 points.
- Initial Lives: Players begin with three lives.
- Extra Life: Players earn an additional life for every 10K points scored.

PTO

Game Progression

The game progression is structured to provide an escalating challenge, retaining the player's engagement as they navigate through the waves, facing an increasing number of adversaries.

- Progress through waves by destroying all present asteroids and saucers.
- Saucer spawning continues through waves with the type of saucer dependent on the player's score.
- The game continues until all lives are lost, with the difficulty escalating through waves due to the increasing number of asteroids and the introduction of small saucers post a 10K score.

PTO

Task 2 - Rendering (45%)

In this section of the assignment, the focus is on the graphical rendering of the game, which plays a vital role in creating an engaging and visually appealing gaming experience. Two distinct rendering modes are to be implemented to pay homage to the classic aesthetic of the original game while also showcasing modern rendering techniques. The choice between these two modes should be offered to the player, ensuring a versatile gaming experience.

Classic Mode

Your task is to implement a classic rendering mode reminiscent of the original game's look. This can be achieved through wireframe geometry or clever use of screen-space shaders to capture the nostalgic aesthetic of the classic Asteroids game.

Modern Mode

The modern rendering mode should exhibit a contemporary visual appeal. In this mode, all game entities should be textured and lit, showcasing the advancements in graphical rendering. This mode should also feature an appealing background, such as a nebula or an astronomical photo, to enhance the cosmic ambiance of the game. Utilise procedural content generation (PCG) techniques to introduce variety and uniqueness in the game's visual elements:

- **Asteroids**
  - Generate diverse asteroid shapes by deforming a sphere to create the asteroid geometry. This approach will ensure each asteroid appears distinct while maintaining a unified aesthetic.
- **Saucers**
  - Employ PCG to create varying saucer designs from simple primitive types, thereby adding visual variety to the enemy class.
- **Texturing and Lighting**
  - Incorporate textures and lighting to enrich the visual appeal of the game. Textures will add a layer of realism and detail, while lighting will provide depth and a sense of space, creating a more immersive gaming experience.

- The game could use a single directional light to simulate light coming from a star or nearby sun. Point lights could be used to simulate light emission from the player and enemy saucers as well as explosions that occur when entities collide.
- Add a visually appealing background to enhance the game's ambiance. A background featuring a nebula or other astronomical images will provide a visually engaging environment that complements the space theme of the game.

· **Particle Effects**

- The player thruster and main cannon can be implemented using a simple particle effects system. Billboards (camera-facing quads) can be used to draw particles.
- The particle effect system may also be employed to render explosions. Alternatively, an image-based animation sequence may be used to depict explosions.

Despite the modern twist in graphics, ensure that the gameplay in both rendering modes remains faithful to the original game. The classic gameplay experience should be preserved, offering players a nostalgic yet visually enriched gaming experience.

PTO

Task 3 - User Interface (UI) (20%)

The User Interface (UI) is a crucial aspect of the gaming experience, providing the player with necessary feedback and interaction mechanisms to navigate the game. In this task, you are required to design and implement a UI that transitions through different game states, each having its unique display and interactive features. The objective is to create a user-friendly interface that enhances the player's engagement while providing essential game information seamlessly.

- **Game State Machine**
  - Develop a game state machine to manage the transitions between Attract Mode, Gameplay Mode, and Game Over Mode. Each state should have distinct visual and interactive characteristics.
- **Attract Mode**
  - Implement an Attract Mode that serves as the game's introductory screen.
  - Display the game title and high scores, cycling through the highest scoring player names.
  - Provide a visually active game world in the background to entice players, along with instructions on how to start the game.
  - Ensure all controls are well communicated to the player through the UI.
- **Gameplay Mode**
  - This is the core game state where the player interacts with the game.
  - Allow players to toggle between Classic and Modern rendering modes during gameplay. Ensure the transition is smooth and does not disrupt the gameplay.
  - Implement a Heads-Up Display (HUD) to show vital game information:
    * Display the number of lives remaining, the current score, and the wave number.
    * Design the HUD to be intuitive and non-intrusive, ensuring the player can easily access the information without it obstructing the gameplay.
- **Game Over Mode**
  - Display a Game Over screen when the player loses all lives.

- Continue to render the game world in the background, but allow no player interaction.
- Provide an interface for players to enter their initials if they achieved a high score.
- Offer an option for the player to return to the Attract Mode or restart the game.

---

end of document