

CPS3231

Computer Graphics

Assignment: Arkanoid 3D

Sandro Spina
Keith Bugeja

Department of Computer Science
University of Malta

Academic Year 2019/2020

Instructions

- This is **an individual** assignment. This assignment carries **40%** of the final **CPS3231** grade.
- While it is strongly recommended that you start working on the tasks as soon as the related material is covered in class, the firm submission deadline is **20th January 2020 at 5pm**. Hard copies **are not** required to be handed in.
- A report describing how you implemented the different tasks of the assignment **is required**. Tasks (1-5) for which no information is provided will not be assessed. The report is extremely important as you are expected to describe how you built your system, outlining any algorithms used and important decisions taken. You are encouraged to use diagrams and pseudocode to augment your discussion. If the deliverable has shortcomings or bugs, or even lacks some of the expected functionality, you must put this down in the write-up.
- You are to allocate approximately **40 to 48** hours for this assignment.
- A soft-copy of the report and all related files (including code) must be uploaded to the VLE by the indicated deadline. All files must be archived into a single .zip file except for the report PDF file which should be submitted using TurnItIn. It is the student's responsibility to ensure that the uploaded zip file and all contents are valid. Please note that the standard plagiarism form must be signed and enclosed with the uploaded archive.
- Reports (and code) that are difficult to follow due to low quality in the writing-style, organisation or presentation will be penalised.

Arkanoid

The premise of this assignment is simple: build your own Arkanoid¹ game clone in 3D! From Wikipedia, “*Arkanoid is an arcade game released by Taito in 1986. It expanded upon Atari’s Breakout games of the 1970s by adding power-ups, different types of bricks, a variety of level layouts, and more sculpted, layered visuals. The title refers to a doomed mother ship from which the player’s ship, the Vaus, escapes. It was widely ported to contemporary systems and followed by a series of remakes and sequels, including the 1987 arcade game Arkanoid: Revenge of Doh. Arkanoid revived the Breakout concept, resulting in many clones and similar games for home computers, even over a decade later. The player controls the Vaus, a space vessel that acts as the game’s “paddle” which prevents a ball from falling from the playing field, attempting to bounce it against a number of bricks. The ball striking a brick causes the brick to disappear. When all the bricks are gone, the player goes to the next level, where another pattern of bricks appears. There are a number of variations (bricks that have to be hit multiple times, flying enemy ships, etc.) and power-up capsules to enhance the Vaus (expand the Vaus, multiply the number of balls, equip a laser cannon, break directly to the next level, etc.), but the gameplay remains the same.*”

Task Breakdown and Marking

This assignment is broken down into five tasks, each of which outlines a different aspect of the final deliverable and the respective marking. The Javascript 3D Engine developed during class should be used in the development of the game.

Task 1

Your first task is to generate the visuals for the game. The environment should consist of an inverted U-shaped container - the playing field - in which i) multiple bricks are procedurally positioned within the upper half, and ii) a paddle and a single ball are placed at the bottom of the playing field as illustrated in Figure 1. The procedural generation of the bricks should be driven by configuration scripts which determine geometry dimensions and brick patterns. The container, bricks, paddle and ball can be constructed from simple geometric primitives such as spheres and cuboids. Balls should be constructed by subdividing spheres into triangles.

- Textures should be applied to all objects.
- The ball should employ a specular shading model to simulate material shininess.
- The paddle and bricks should employ a diffuse shading model.
- The game is illuminated by a single directional light in addition to an ambient lighting component.

20 Marks

¹<https://www.classicgame.com/game/Arkanoid>

Task 2 - Camera

In this second task, you are expected to implement a camera system as illustrated in Figure 1. The camera navigation system should make use of **Q** and **E** to switch between fixed points *top* and *grazing* respectively. For both fixed points, the camera should be looking towards the centre of the playing field. The navigation system should also make use of the keys **W** and **S** to change the camera position and orientation (orbit the point of interest) between the two fixed positions. The speed of the camera is controlled by a sensitivity parameter which can be increased or decreased using the **R** and **T** keys respectively.

10 Marks

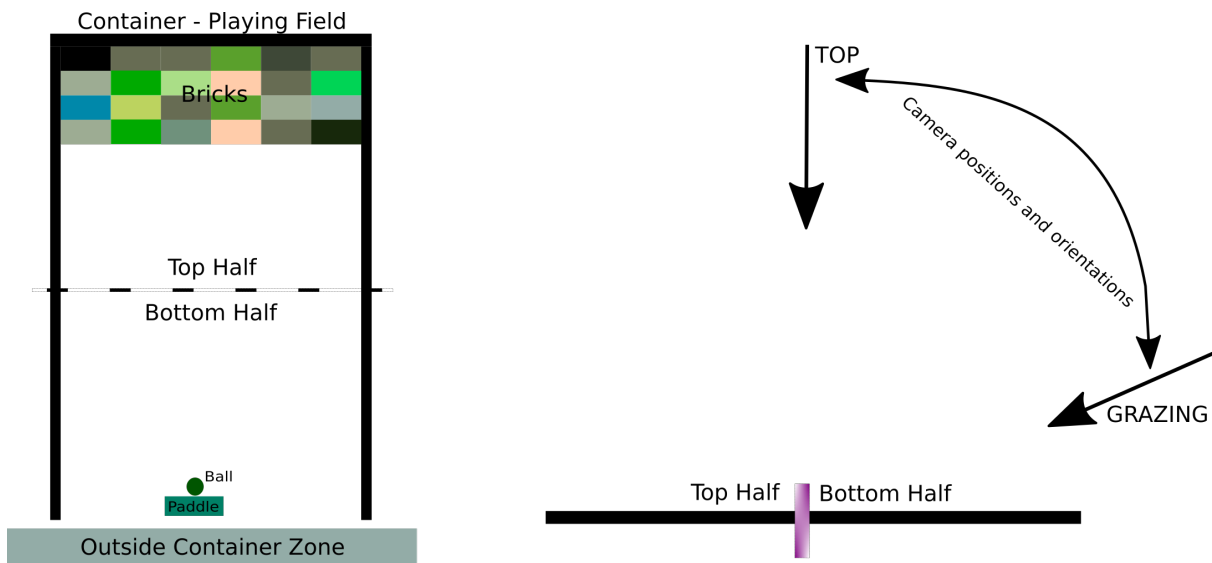


Figure 1: Arkanoid game schematic (Top-down view) and camera interpolated positions

Task 3 - Physics

This task entails the implementation of game mechanics. A simple 2D physics simulation of the ball, container walls, paddle and bricks should be implemented, including circle-circle (in case of multiple balls) and circle-rectangle (ball to paddle, bricks and walls) collision detection and response. Pressing the **spacebar** will launch the ball into the upper half of the playing field. Holding down the **spacebar** increases the magnitude of the force applied. The ball is repositioned on the paddle when it collides with the *outside container zone* under the playing field. The game is reset when the ball is lost a set number of times.

20 Marks

Task 4 - Additional Illumination

Extend the WebGL engine (JavaScript and GLSL shaders) developed in class to support four concurrent light sources. Use the new functionality to illuminate the game by positioning a point light on the ball and spotlights (looking into the playing area) on the paddle.

20 Marks

Task 5 - Powerups

This task entails the development of power-up capsules to enhance the Vaus (paddle) as follows:

- ball breaks through bricks instead of bouncing off them;
- expansion in horizontal size of the Vaus;
- ball splits into two (same size);
- ball slows down to half speed for 30 seconds;
- ball sticks to the paddle - pressing space releases the ball;
- Vaus gains a laser cannon which can be used to shoot down bricks.

Power-up tokens drop when the ball hits specifically designated power-up bricks and are activated when the Vaus (paddle) collects the token. These should be randomly positioned when procedurally creating the bricks at the start of the game.

30 Marks

Final Notes

Feel free to experiment and try to enjoy this assignment; discuss ideas or difficulties with your classmates. Do not leave the assignment for the last week. You are allowed to tweak the game mechanics (e.g. ball-paddle collision response) to make the game as fun as possible. The effort put into this assignment will be duly noted during marking. Good luck!