

Artillery Game Report

200009419

Introduction

Particle Artillery is a famous video game. It is a two-player game in which each player controls a tank. The players take turns to fire shots at each other. The player can move the tank and decide the angle and thrust of the shell to be fired. In this variant specifically, the side taken hit will be regarded as lose immediately. There are blocks in the middle of the stage which will block the movement of tanks and shell.

Design

Play Area

The resolution of the game is 1600*900, and the play area is restricted to the 1600*900 scaled "Stage". There are several elements added to the stage: terrain, blocks, tanks, shells and other UI components. The blocks in the middle are **randomly** generated in each game that blocks the shells and the movement of tanks. The block will be destroyed when hit by a shell, it is affected by gravity, thus, will drop if no block beneath it.

The angle that the shell can be fired is restricted to the upper semicircle and the initial velocity of the shell is confined in a reasonable range. The shell will also be affected by wind, the wind can toward east (to right) or west (to left) and the maximum acceleration applied by wind equals to 1/10 of the gravity. The wind direction and strength will be changed in the beginning of each turn.

Controlling

The player can control the movement and the firing of the tank. The player can move the tank left or right by pressing "A" or "D" key. Key "Q" and "E" are used to controlling the firing angle while key "W" and "D" are used to control the thrust of the shell. Key "R", once pressed, will reset the current stage and start a new round of game.

Shell

Shells look like circle and is considered in 2D. The tank hit by a shell will be destroyed which leads to the winning of the opponent. The block hit by a shell will be destroyed. The principle of collision detection will be discussed in the rest of the report.

User Interface

The player could find all useful information shown on the top of the screen including the thrust

and angle of the shell, wind direction and strength, and who is taking the turn. In addition, there are arrows representing the vectors of the thrust and angle of the shell to be fired. The direction arrow is attached to the player tank which shows the direction of the shell to be fired. The arrow showing the thrust of firing is appeared on the top left, the scale of the arrow will be proportional to the number of thrust.

There is a starting page showing the controlling of the player tank. The information will disappear once the player pressed any key. If the player wins or lose, a result page will be given.

AI

The ai of the opponent is simple. In the start of each turn, the AI would move randomly and fire randomly with random thrust and random angle. The play could press “Z” to enable AI and “X” to close it.

Implementation

Architecture

To improve reusability and to make it a more sophisticated system, the project is implemented in java, using processing core library. Three packages are created to store classes with different roles. The gameobject package contains the foundation class usable to develop the game. For instance, Texture class determines the view of an object, RigidBody class assign the physical attributes to an object such as velocity, gravity and acceleration. The artillery package contains objects specified in the game such as Shell, Tank and Terrain. These objects comprise of basic gameobjects like Rigidbody. The programme package contains the entry of the programme—Main class. The design of the structure brings the programme high cohesion, low coupling and high extendibility.

Movement

Movement of an object is implemented through Rigidbody object. A Rigidbody object has both velocity vector and acceleration vector. The object could calculate its position according to the velocity, acceleration and gravity. Rotation can also be assigned. Then, Rigidbody object would draw its Texture object on the correct place with specified attributions.

Collision

The Rigidbody could have multiple collide boxes. Under current situation, only rectangle collide box is implemented. The collision detection algorithm is simple. If the distance of centre of gravity between two collide boxes is closer than the sum of their widths in X direction and the sum of height in Y direction, then a collision is detected.

Stage

The stage acts as a container to hold Rigidbodies and Colliders. The programme uses Vector to store gameobjects. When adding a Rigidbody to stage, all of its colliders will be added to the container for collider as well. When removing a Rigidbody, the collider attached to it will also be removed. When collision happens, onCollision method of the Rigidbody will be called, and the stage would pass the object it crashed into. The behaviour of the object could be modified by Override its onCollision method.

Game Logic

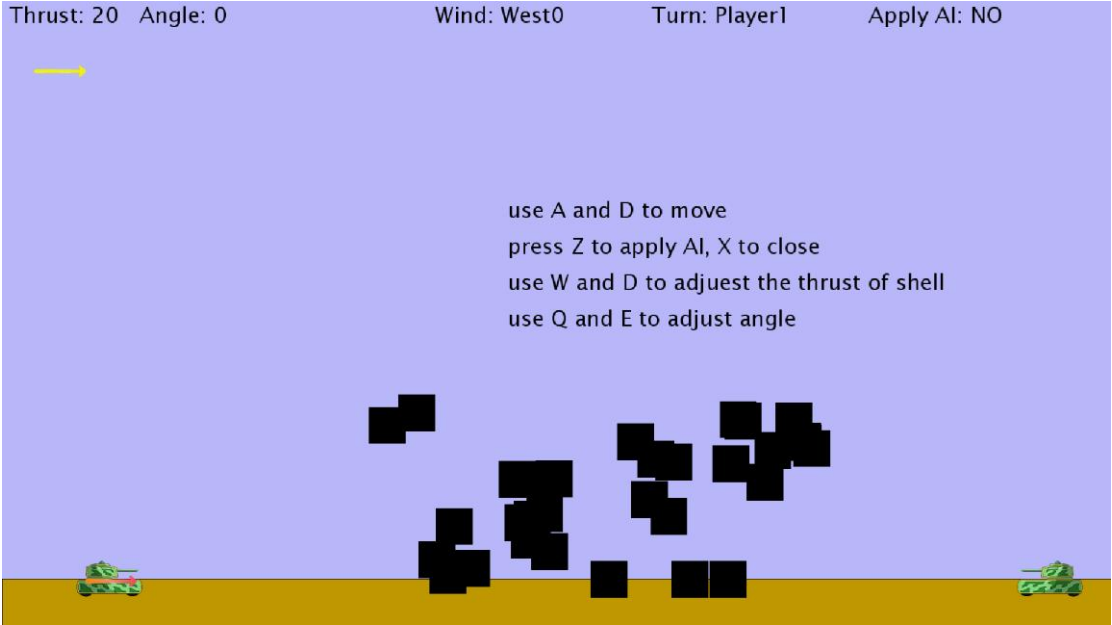
Game logics are implemented mainly in GameController class. If a key is pressed, methods in controller will be called to execute an operation. The controller class has attributes to determine the behaviour of game objects. As an example, “turn” specifies which player is in operation; “thrust” and “angle” control the initial velocity of shells. “Wind” is an array with size of two. The first number determines the direction: 0 for west and 1 for east. The second number controls the strength which is randomized between 0 and 100. The controller class also stores player settings for instance, whether to enable AI player.

The players take turns to fire at each other, they can command their tank to move left or right. However, the tank will be blocked if crashed into a block in the middle. Once one player fired, variables will be passed to Shell object to creating shell with player setting, and the wind will be generated randomly for the opponent in the next turn.

Conclusion

The Particle Artillery achieved all of the basic requirement. As an extension, blocks are randomly placed in the middle of the stage, which are affected by gravity. A very simple AI is added. However, there are still some problems with the game. The design of architecture may be improved to become more concise and functional. It is too complicated and may be hard to read. The design of container and collision detection is primitive, hence may consume a lot of resources. The tank cannot move backward when crashed into a block. Above all. It is a good attempt to create a sophisticated system for a inexperienced game maker.

Screenshot



Thrust: 9 Angle: 90

Wind: East 65

Turn: Player2

Apply AI: NO



Player2 Win
press R to play again

