

Period: 7

Group Members: Gabriel Hyun, Hangsheng Huang, Mantas Kempinas

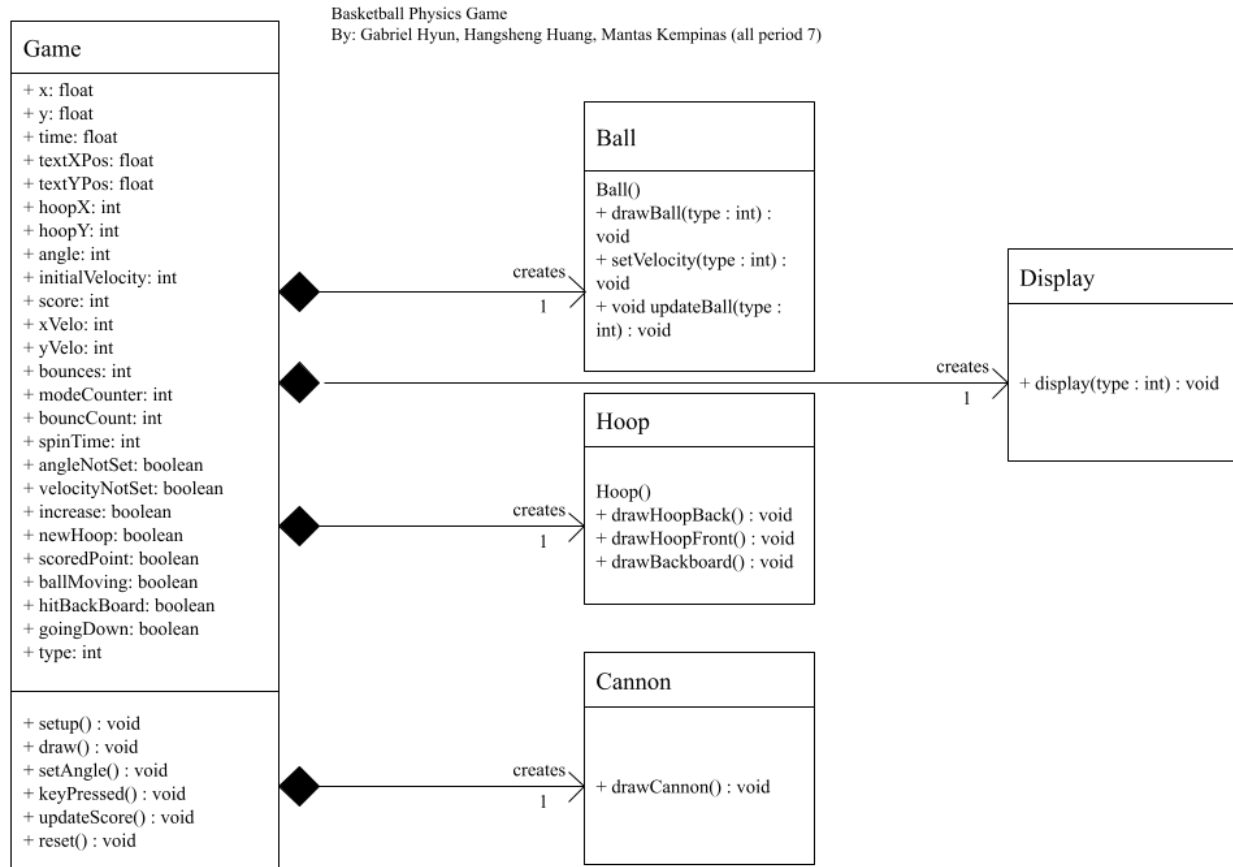
Group Name: The 3-Point Specialists

Project Title: Basketball Physics Game

Description: Our game is a game where the user is required to time their inputs to determine the launch angle and power of a basketball being shot towards a hoop. The hoop is at a set x coordinate but the height of the hoop is randomly generated. The user can also change modes to make the hoop move up and down while shooting. The user can also change the type of ball which they shoot which affects the bounciness and distance traveled at a given power.

Current Functionalities: drawBall(int type) method draws either a basketball, bowling ball, bouncy ball, or balloon with x and y coordinates which change as the program runs. The type of ball alternates between the four types every time the user presses the "b" key. The type of ball changes the physics of the game since a bowling ball will require more power to shoot the same distance as a basketball and would bounce less on the ground or off the backboard. A balloon would also float upwards rather than falling like the other balls. drawHoopFront() and drawHoopBack() draws the rim of the hoop which is divided to make a cleaner animation for when the ball goes through the hoop. drawBackBoard() draws the backboard and the goal post, and drawCannon() draws the cannon. setAngle() allows the user to determine the angle at which the ball is shot at by timing the cannon which moves between 0 and 90 degrees. setVelocity() is used after the angle is set, and determines the power at which the ball is shot based on the user's timing (the user can press any key to trigger their input). updateBall(int type) updates the x and y coordinates of the ball of given type once setVelocity() occurs, having the ball move through the air following the laws of physics. updateScore() simply updates the number in the top left corner of the screen which indicates how many shots have been made. reset() is called after the ball is shot or when the user presses the "r" key. This method simply brings the ball back to its starting position. When the user presses the "m" key, the mode switches between normal and moving. In normal the hoop is still, and in moving the hoop moves up and down. display() is used to show the type of ball the user is currently using by displaying text and a corresponding icon underneath the score box.

UML Diagram:



How it works: The objective of our game is simple; shoot the ball using the cannon into the hoop to score as many points as you can. The trajectory of the ball being shot is determined by the angle of the cannon, which moves back and forth from 0 and 90 degrees. The user is required to press the spacebar in order to set the angle. After setting the angle, the power of the shot is determined by a power bar which moves back and forth between 0 and 70 units. The user is once again required to press the spacebar in order to set the power. The ball is then shot out of the cannon and if the shot is made the score will be updated.

There are four different types of balls which you can switch between by pressing the “b” key. There’s a basketball, bowling ball, bouncy ball, and a balloon. The basketball moves through the air and bounces as a normal basketball would. A bowling ball is significantly heavier and thus requires more power to make the shot, and it also bounces less than the basketball. The bouncy ball moves through the air like the basketball does but bounces significantly more. The balloon is different from the rest because its horizontal velocity slows down due to air resistance and floats upwards instead of falling downwards. Points are scored with the balloon by going upwards through the hoop. You can make the hoop move up and down by pressing the “m” key, and you can reset the shot by pressing the “r” key.