

My CSC8503 coursework is designed to be a kind of football game.

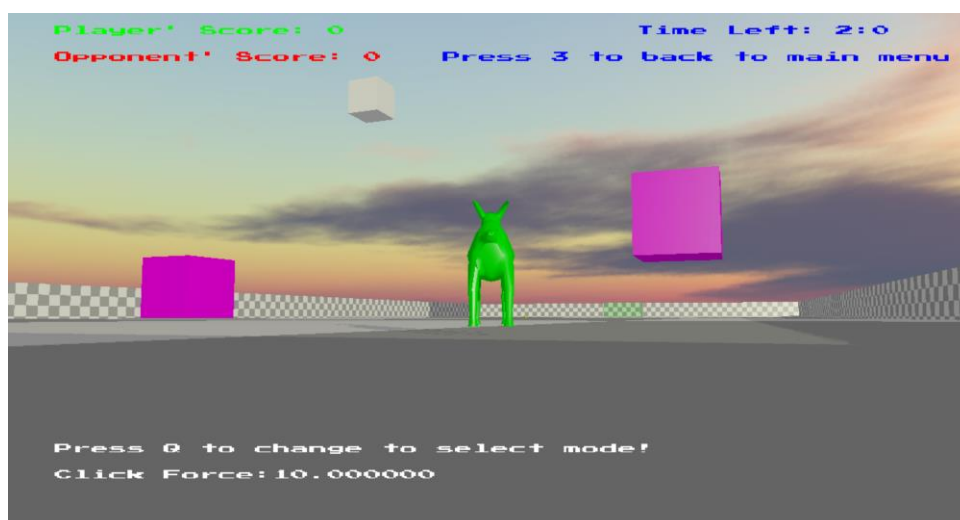


The entire level is a rectangular area (imitating the football court) that is fully capable of conducting pathfinding base on A* algorithm.

The level uses quadtree for collisions optimization.

There are two goals sitting on the opposite ends of the court, and a yellow sphere (imitating the football) will be spawned in the center of the court. The player's goal is in pale green and the enemy's goal is in pale red. The player will control a goat (spawn in front of the player's goal) and the objective of this game is get the sphere collide with the enemy's goal.

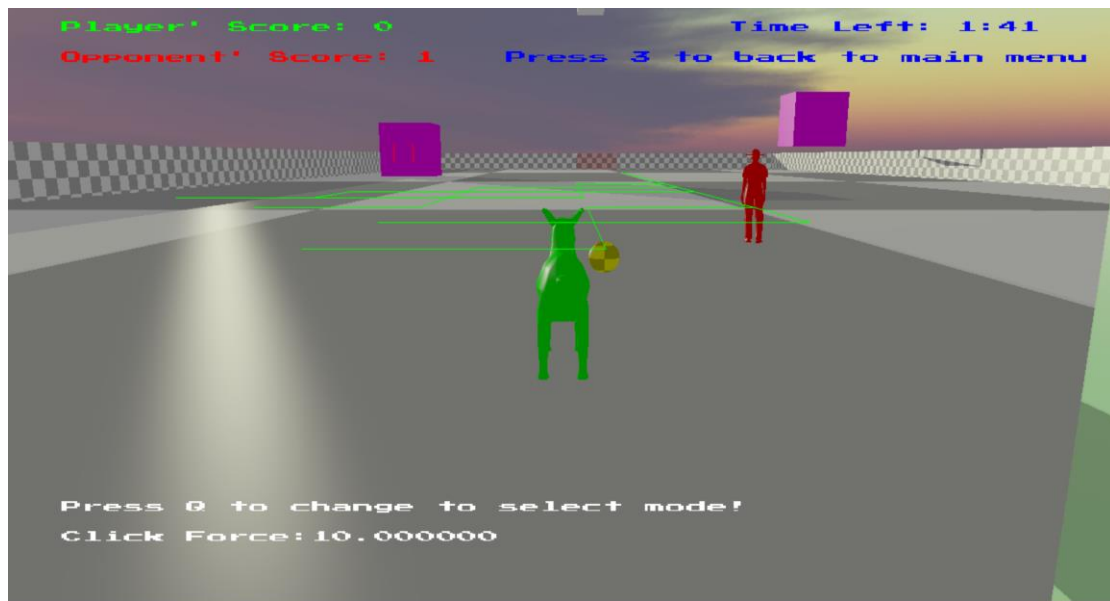
Obstacles (in purple) driving by position constraints and state machine are also placed in the level.



By using polar coordinates and trigonometry, I am able to calculate the position of the main camera with respect to the values of pitch and yaw updated from the mouse, to achieve a third-person view. The goat's orientation is then updated to match the orientation of the main camera.

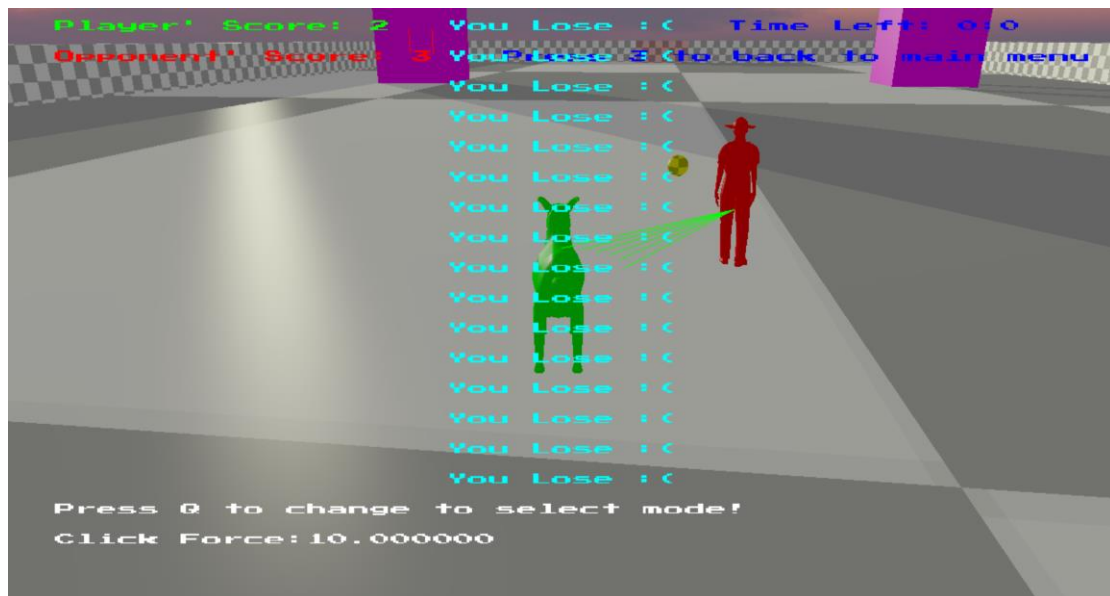
By pressing W, A, S, D, force can be exerted on the goat.

When the goat collides with the sphere, a position constraint is established between them. This represents the player "having the ball".

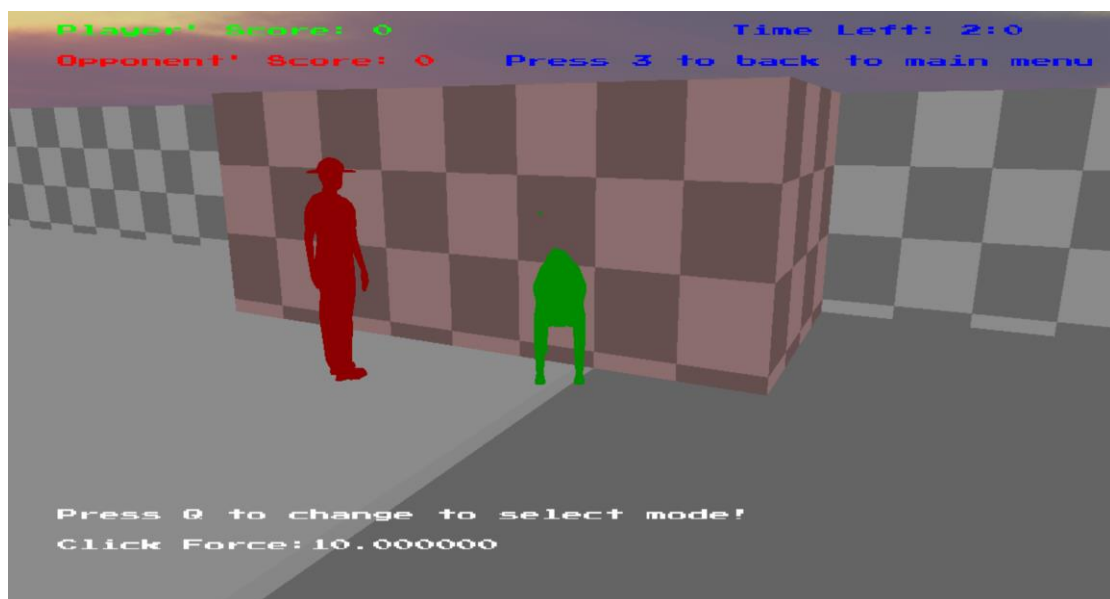


There is an AI opponent (spawn in front of the enemy's goal) that is controlled by a behavior tree. The behavior tree consists of a selection node followed by a sequence node: the AI will prefer to go to and stay at a fixed position (in our case it is somewhere in front of the enemy's goal) if the player does not "have the ball", otherwise, the AI will conduct a sequence of "chase the player" (by using A* algorithm pathfinding) and then "go to the player's goal" once the AI has the ball (the goat-sphere position constraint will be removed and then the AI-sphere position constraint will be added once the AI is sufficiently near to the goat when the AI is chasing the goat).

In the "go to the player's goal" state, the AI will try to avoid the player, the wall (the cubes on the edges of the court) or the obstacles if the player, the wall or the obstacles is sufficiently near to the AI (ray castings are used for detecting objects near the AI).



When the goat collides with the enemy's goal, or when the AI collides with the player's goal, penalty resolution will be applied (projection/impulse resolution will be applied for all the other collisions).



If the sphere collides with the player's goal, the opponent's score will be increased by 1, If the sphere collides with the enemy's goal, the player's score will be increased by 1.

If the y-component of the position of either the player, the sphere or the AI is enough negative, the level will be reinitialized (as it indicates that some important object are outside the map).

Every time the player's score or the opponent's score changes, the level will be reinitialized.

There is a timer on the top right of the screen for counting down the time that has passed in the game. The timer will only be activated after the game has started (that is, after the first time the goat collides with the sphere for each initialization of the level).

There are two types of gameplays provided in this game, and they can be selected from the simple menu that has been implemented in this game. For the first type of the gameplays, if the timer goes to zero, the game ends. Based on the player's score and the opponent's score, messages of winning/losing/draw will be displayed.

In the second type of the gameplays, there is no draw: if the player's score equals to the opponent's score when the timer runs out, the timer will be reset to 1 minute. By such looping, the game will only end if the player wins or loses.

The AI's moving speed will vary with respect to how much time is left, and also with respect to how much the difference currently is between player's score and opponent's score.

In the second type of the gameplays, player has an energy bar shown on the bottom left of the screen. When there is energy left in the bar, player can press SPACE to add torque to the goat, let it spin and then fly. When SPACE is pressed down, the energy will decrease. If there is no energy left, the torque will stop adding and the goat will stop spinning and fall from the sky. When SPACE is released, the energy will automatically gradually increase to a max value of 100.0f. If the goat flies with the sphere attaching to it, the AI will throw gooses towards the goat. If the goat is hit by the goose, the energy will be suddenly emptied.



You can go back to the menu when playing the game by pressing NUM3 and doing so will reset all the data in the game in order to get ready for the next round.