**CSC8503 Coursework 2018/19**

# ****Part A:**** Totally original golf game!

In this coursework you are to develop a simple game that includes physics, AI, and networking, to allow you to practice your coding skills, and reinforce your knowledge in these areas. The game you are to make is to be a simple 'golf' game, where you must correctly navigate a ball through a maze to reach the goal in as few shots as possible. Whether the game environment is in 2D or 3D is up to you, as both can lead to interesting gameplay, as shown by the following 'real' game examples:

**https://store.steampowered.com/app/431240/Golf\_With\_Your\_Friends/ => For 3D**

**https://store.steampowered.com/app/616520/Cheap\_Golf/ => For 2D**

In these games there tends to be moving obstacles that block your path, requiring careful timing to successfully navigate, which can serve as the entry point for adding in some AI calculations into the game environment.

The codebase you have been provided with serves as a good starting point for the development of your game. The goal of the coursework is to then enhance this codebase with realistic physics and collisions, to allow the basics of the game to be played. From there, AI techniques can be used to add in some more complex movements of obstacles that the player must avoid. You do not need to spend any time implementing graphical techniques, or finding new meshes and textures to apply to your game - the purpose of this coursework is just to investigate the new concepts relating to physics, AI, and networking, introduced in the lectures and tutorials.

**Features:**

Your program should be capable of allowing the player to play at least TWO golf levels, selectable by pressing keys on program startup. Each level should spawn in a player ball that can be pushed around with the mouse or keyboard, and the level should be 'completed' if the player touches a 'goal' object with the ball. The number of 'pushes' should be kept track of, to allow players to try and beat their personal best at each level. The player should be able to, at any time, press the R key to 'reset' the level, or press the B key to move the in game camera to where their ball is. At least one of the levels must have interactable objects - moving platforms or walls, spinning obstacles, or randomly patrolling 'enemies' that reset the ball if they touch it are examples of these interactions.

**Guidelines:**

* A passing mark would feature at axis aligned bounding box and sphere collision volumes to play a game of golf on a grid-based map, with at least 2 types of 'interactable' objects, controlled via a state machine. Ball movement should take into consideration linear movement, and integrate acceleration and velocity correctly.
* A merit or above mark would additionally control the flow of the game itself (main menu -> game -> win / lose) via a suitable state machine or pushdown automata. The collision detection should be accelerated with an appropriate broadphase structure. Angular velocity should be correctly applied to all physics objects, with the inertia tensor used appropriately.
* A distinction or above mark would additionally contain collision detection with more complex shapes (object oriented bounding boxes, or capsules), and feature obstacles that move either on a constraint (hinges or ropes), or make use of an iterative solver for accuracy and stability (a stack of boxes).

**Part B: Totally original golf game with friends!**

Everyone knows games are more fun with multiple players! The game from part A should be expanded to include a networked multiplayer component. All players play simultaneously, and do not have to wait for other players to finish their shot before making another. Whoever reaches the goal first 'wins' the level, and the game should then cycle to the next implemented level.

**Features:**

The game area should now also contain an AI-controlled robot that chases after a player's ball, selected at random once every 10 seconds. This robot should be controlled by the server. If the robot touches a slow moving ball, it will fire it in a semi-random direction away from the goal. If the robot is hit by a fast moving ball, the robot will explode, and respawn at a random location on the map 15 seconds later, and give the player a bonus of some sort. Players should be informed by an on-screen message whenever the robot is destroyed, or has respawned. To chase after the ball, an appropriate pathfinding algorithm should be implemented.

**Guidelines:**

* A passing mark will allow a server and a client to be spawned, allowing for the 'high score' of ball pushes to be stored and retrieved from the server via network messages. There will also be a client-controlled AI robot that can move around the world, with appropriate pathfinding used to calculate the shortest path to the ball being chased.
* A merit or above mark would allow multiple players to connect to a server to play the game simultaneously. The client will send messages to the server to indicate that a shot has been taken, and the server will send state updates to the clients to update the ball positions.
* A distinction or above mark would have an entirely server-controlled robot, calculating all logic server side, and sending appropriate packets to indicate the robot's current position and orientation, and the result of any side effects of the robot finding a ball.