**CSC8503 Features and Key binds**

**YouTube link:**  <https://youtu.be/qD831YfhpR8>

**Key binds:**

* W – Move forward
* S – Move backwards
* A – Move left
* D – Move right
* Space – Jump
* Control – Pause the game
* G – Enable gravity
* Q – Force gun mode (only activates after scoring 5 or more points)

**Main Menu Binds:**

* Space – Starts the game
* Control – Quit the game
* F1 – Resets the game

**Gameplay**

* Certain green cubes are destructible objects that can be destroyed to gain points. Point values for these crates are random.
* Blue spheres are powerups and will give the player a speed and jump buff for a set amount of time.
* White/grey cubes are interactable physics objects that can be pushed around and interacted with.
* The human patrols the maze and will run away from the player if they get hit.
* The goose has several different behaviours that all include trying to find and attack the player.
* Client/server gameplay can be enabled.
* Game will end when all destructible objects have been picked up.

**Instructions for picking single player/multiplayer**

Single player:

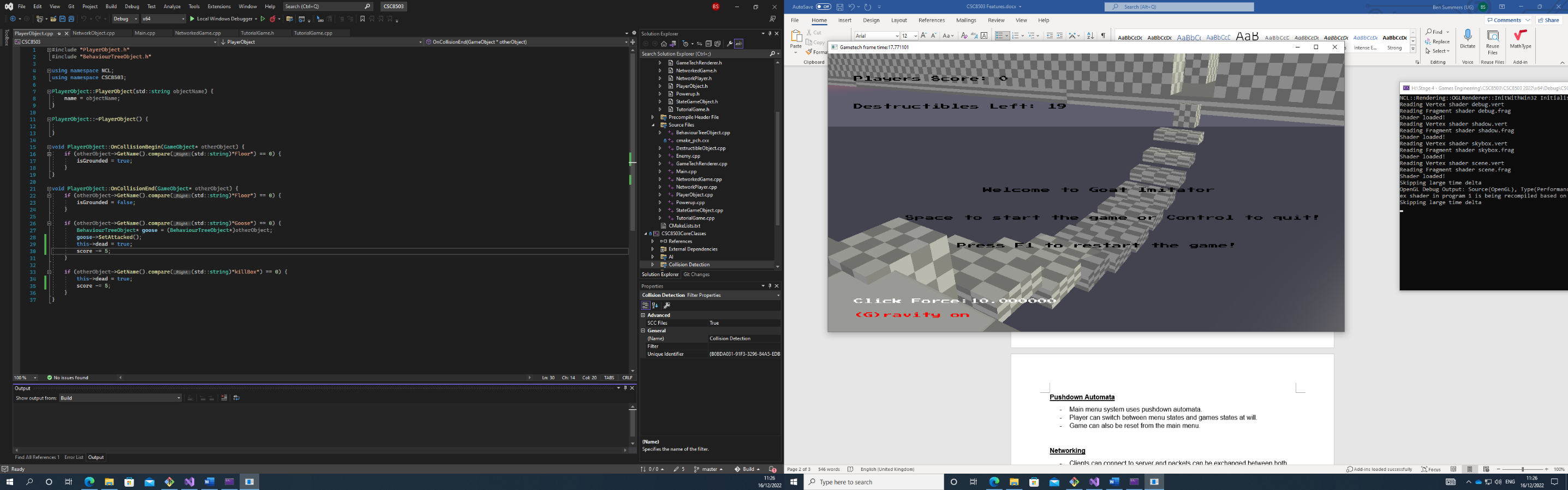
* Ensure line 185 in TutorialGame.cpp is uncommented and run the program.
* Do not use F9/F10 buttons.

Multiplayer:

* Comment line 185 in TutorialGame.cpp.
* Run 2 instances of the game.
* Press F9 in one window to run the server.
* Press F10 in the other to start the client.
* Use the client to move around and play the game. The server takes no input from the user but instead takes requests from the client.

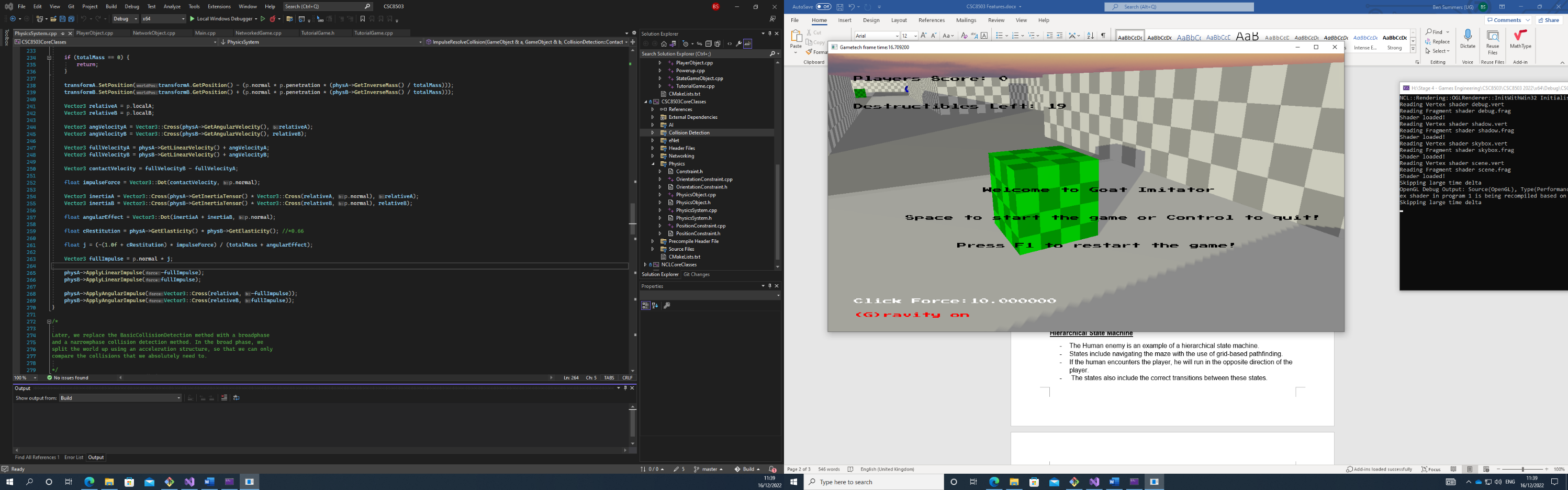
**Physics Features**

* Angular and linear motion applied to physics objects.
* Torque used to rotate objects.
* Player object in controlled entirely with forces.
* Position constraints as well as orientation constraints for the bridge.



**Collision Features**

* AABB collisions
* Sphere collisions
* Capsule collisions
* Ray casting collisions
* Collision layers to stop AI interacting with player collectibles and vice versa.
* Collisions for gameplay features such as picking up items and such.
* Penalty box if you fall you respawn and lose points.



**Optimisations**

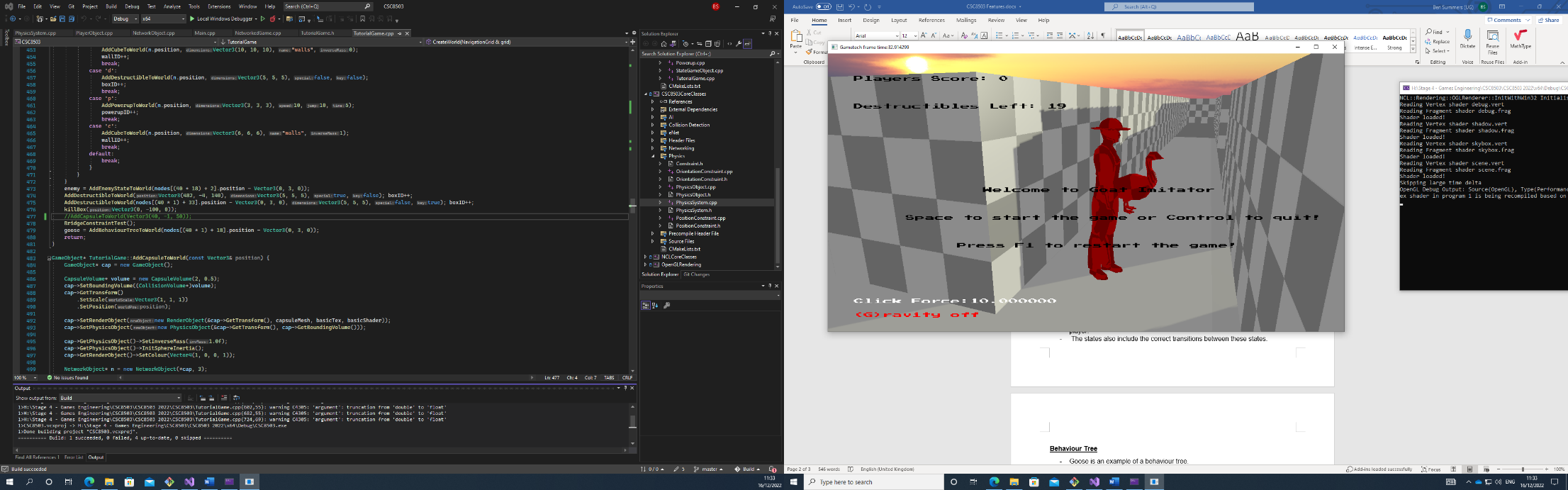
* Broad phase/narrow phase.
* If 2 items both have an inverse mass of 0, they are static and collision detection is not executed on them.

**Hierarchical State Machine**

* The Human enemy is an example of a hierarchical state machine.
* States include navigating the maze with the use of grid-based pathfinding.
* If the human encounters the player, he will run in the opposite direction of the player.
* The states also include the correct transitions between these states.

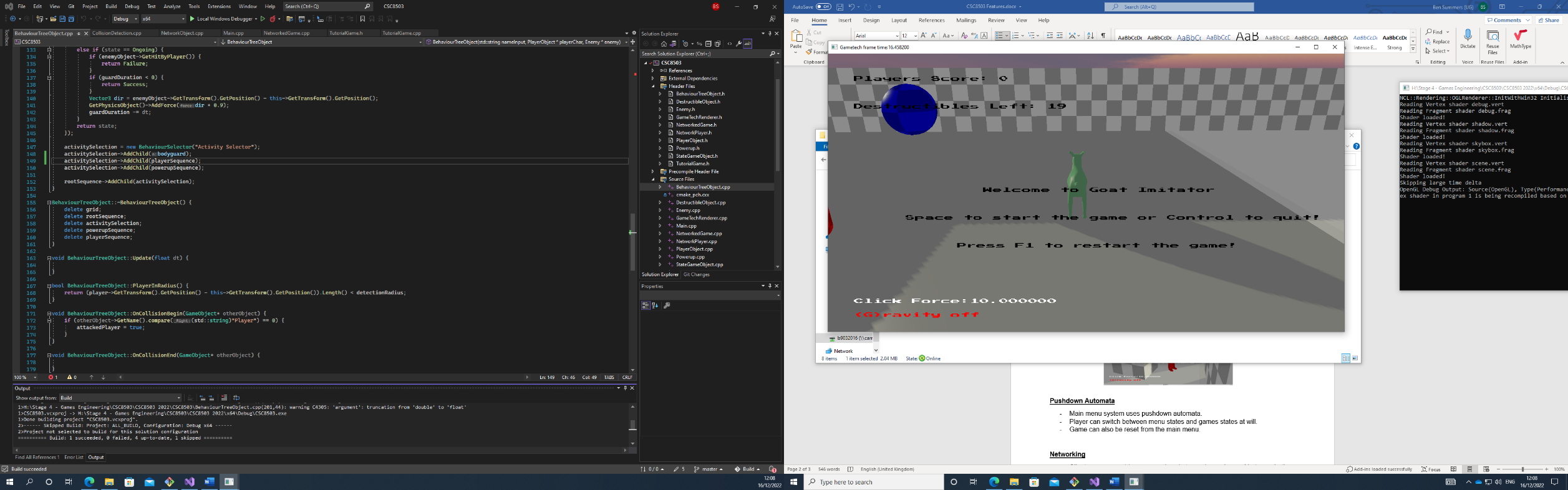
**Behaviour Tree**

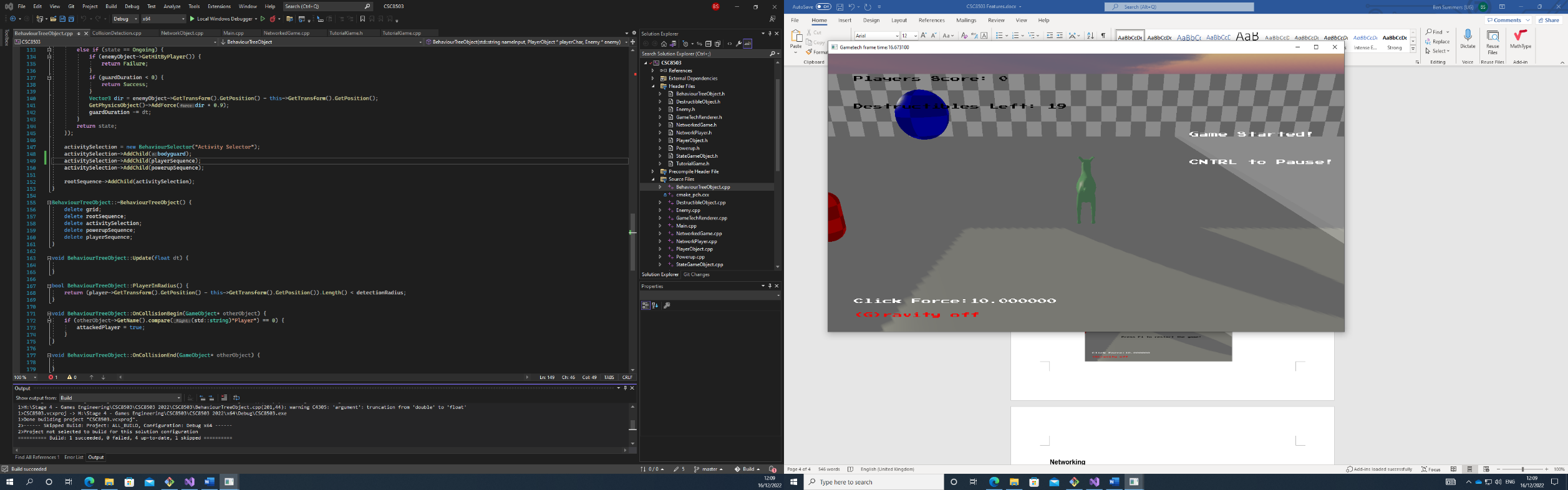
* Goose is an example of a behaviour tree.
* Has 3 activities to carry out; bodyguarding the human AI, hunting for the player and carrying out a task to get a buff that increases detection range.
* Goose also uses grid-based pathfinding.



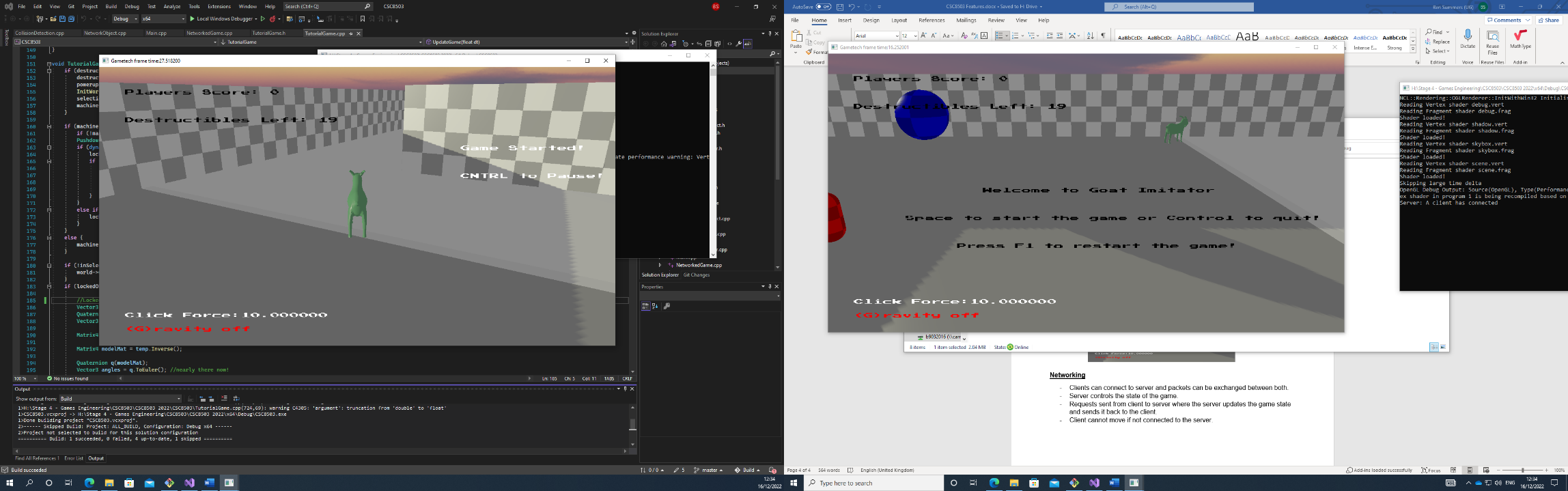
**Pushdown Automata**

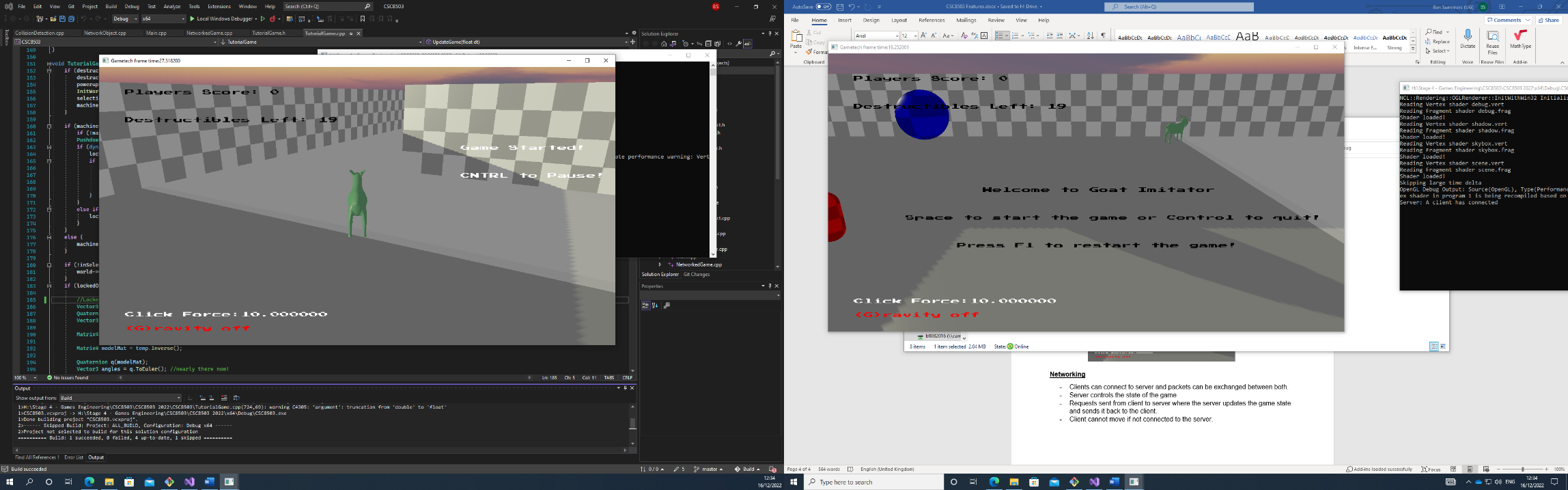
* Main menu system uses pushdown automata.
* Player can switch between menu states and games states at will.
* Game can also be reset from the main menu.





**Networking**

* Clients can connect to server and packets can be exchanged between both.
* Server controls the state of the game.
* Requests sent from client to server where the server updates the game state and sends it back to the client.
* Client cannot move if not connected to the server.



**Improvements Since Marking Session:**

* Made improvements to AABB-Capsule collisions
* Added capsule-capsule collisions