**CSC8503 – Advanced Games Tech**

**Make sure Visual Studio mode is set to x64 and not Win32**

**Keys:**

**Main Menu:**

1 – Play level 1

2 – Play level 2

Escape – Exit game

**Level 1 & 2 & Test Level:**

E – Break away from rope (if attached)

F – Toggle player ball as user controllable

G – Toggle gravity

B – Toggle between basic collision detection and broad phase/narrow phase

L – Decrease constraint iterations

O – Increase constraint iterations

I – Toggle extra information about selected objects

Q – Toggle to / from selection mode

P – Pause

U – Unpause

F1 – Reset level

F2 – Reset camera

Escape – Exit back to menu / Exit pause

**ONLY when selected an object**

LEFT – Add torque along -x axis

RIGHT – Add torque along x axis

UP – Add force along -z axis

DOWN – Add force along z axis

SPACE – Add force along y axis

LEFT SHIFT – Add force along -y axis

7 – Add torque along y axis

8 – Add torque along -y axis

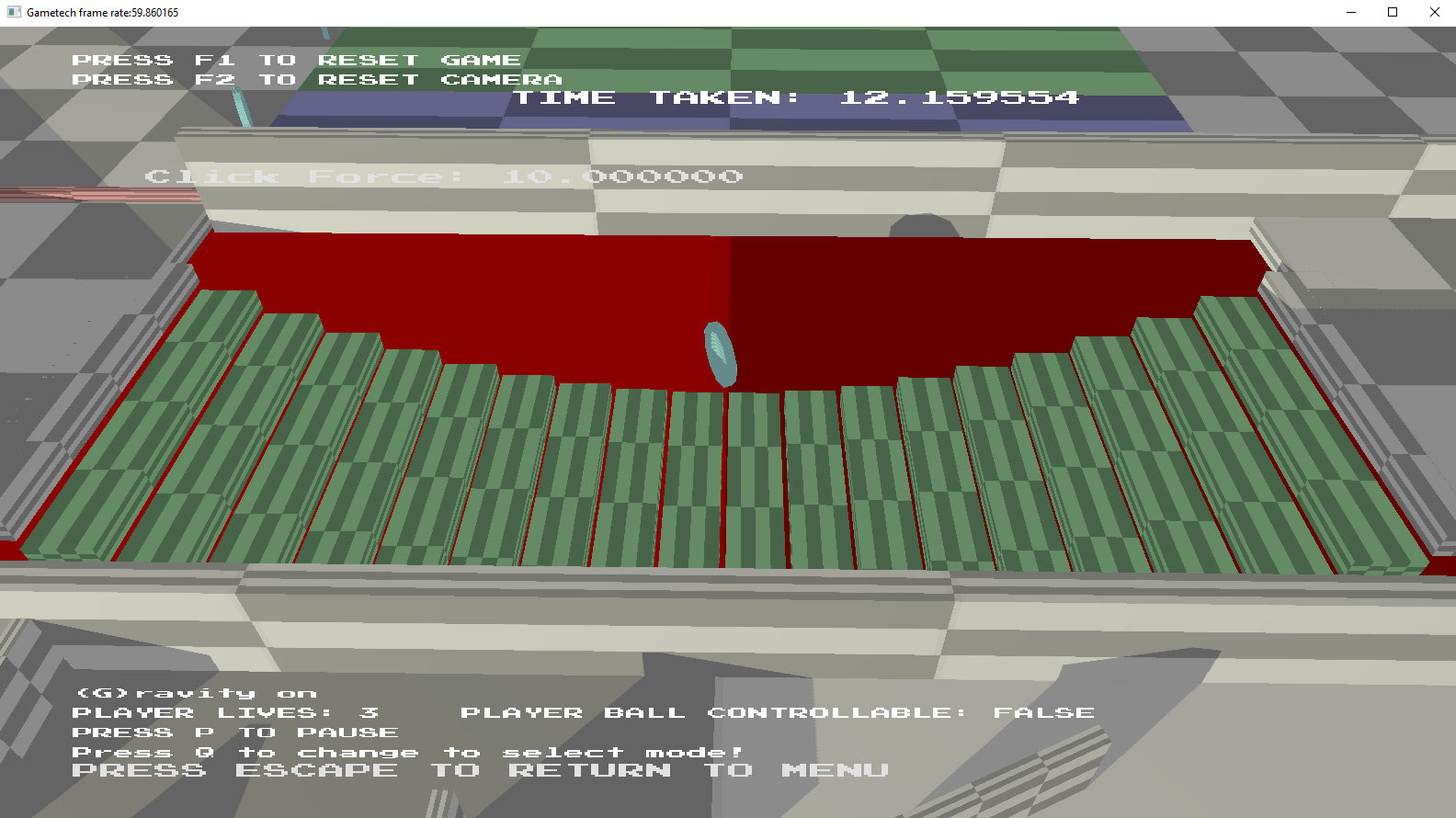
**NOTES: In either level, any player moveable object is displayed in a lightish green colour**

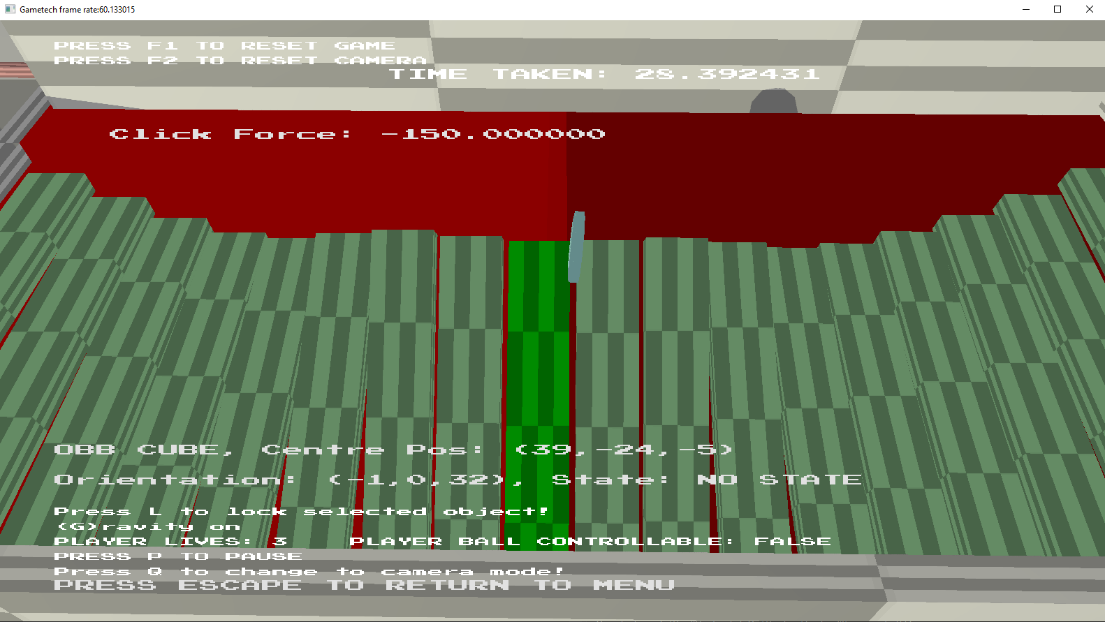
**Checkpoints area low red bar and don’t collide with objects**

**In final column of level 1, the 2 different coloured floors are ice and slime**

**Youtube video link:**

**Constraints: (Bridges, Ropes, Springs and Flicking objects)**

****

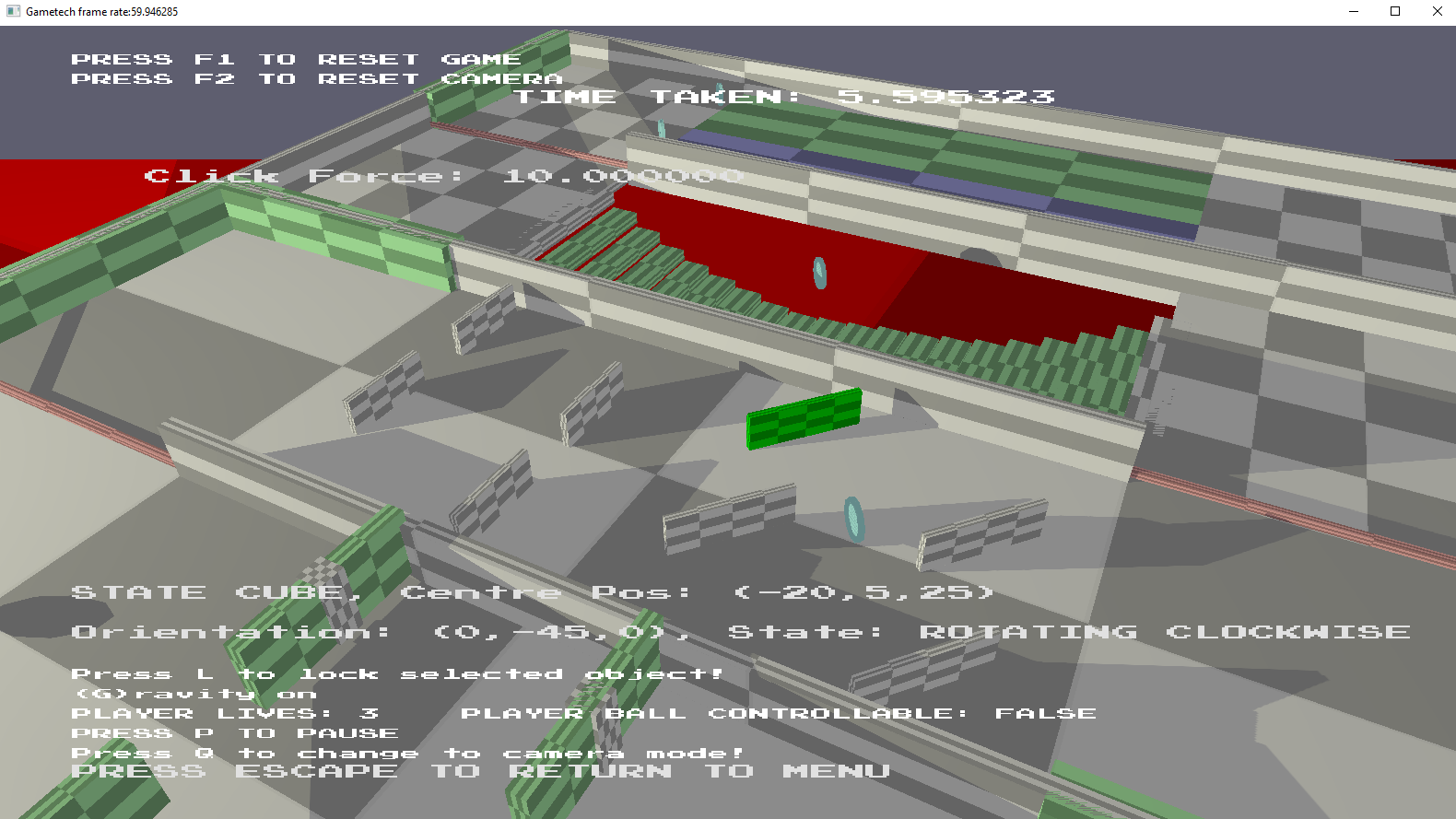
****

In these 2 pictures you can see the bridge that is placed in the first level, in the top image the position constraint is more easily displayed between the adjacent planks, in the bottom image you can see the single axis orientation constraint where it can only tilt on one axis and bring adjacent planks up or down with it to create a smooth rotating appearance.

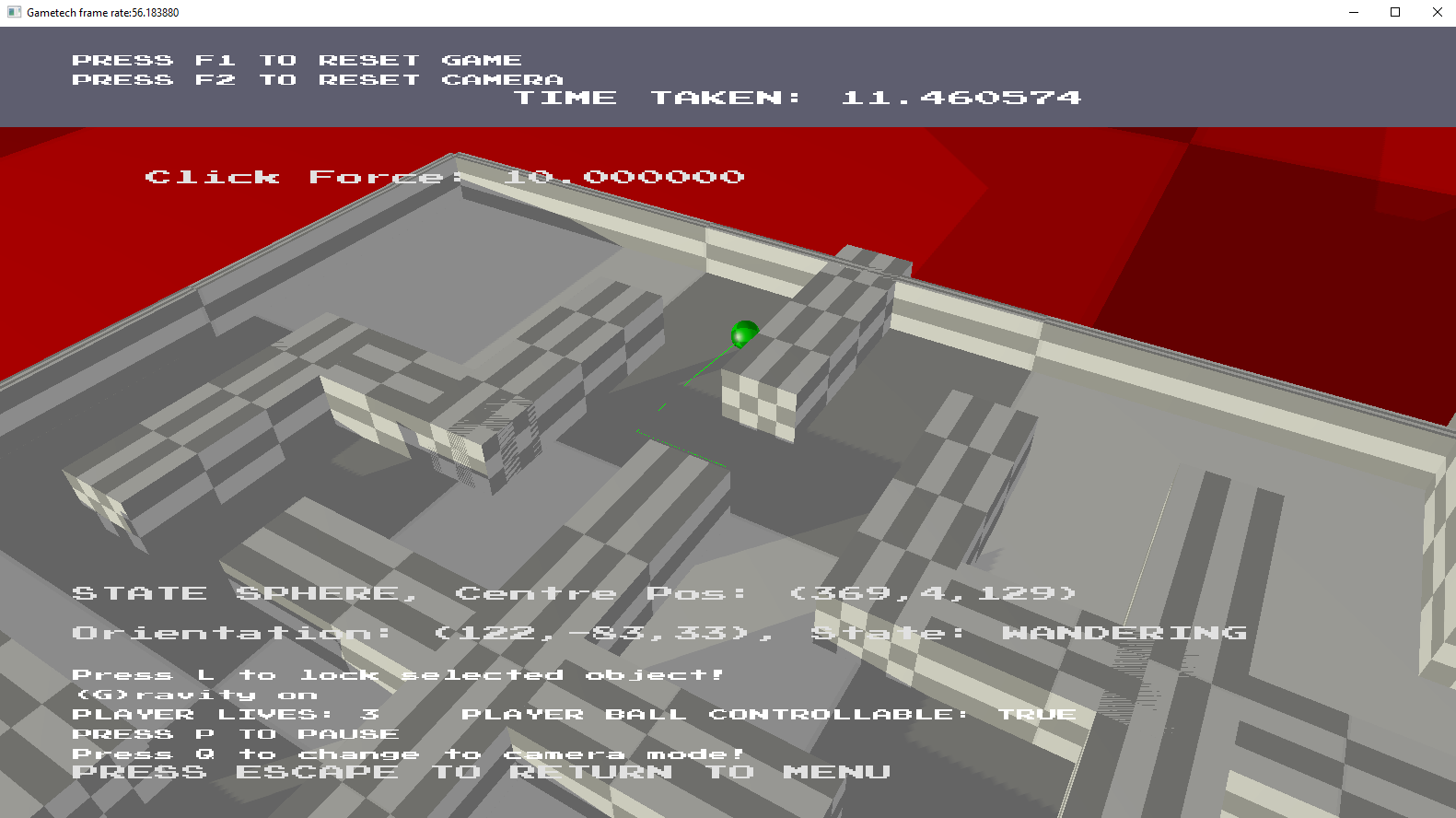
****

In this image you can see a rope that I have made and placed in level 2, this again uses the position constraint and a new facing constraint which makes one block always face the centre of the block before it in a chain, I use this to create the appearance of a rope as all the chains will face the previous chain block.

**State Machines: (simple moving or rotating objects, AI)**

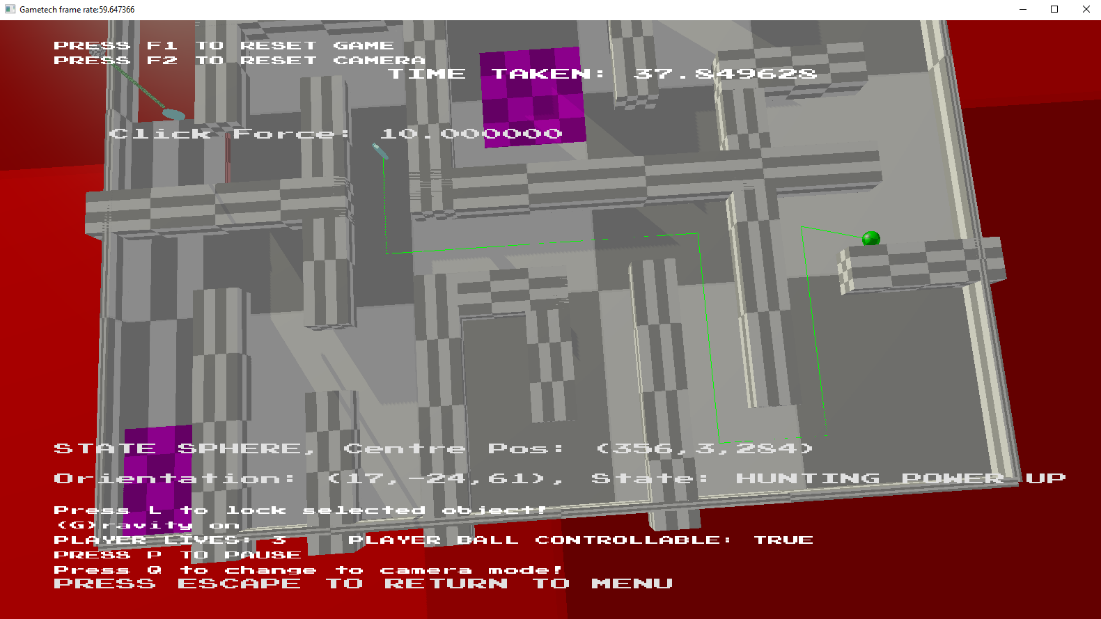
****

In this image you can see two different state machine objects, ones which move side to side and ones which rotate clockwise and anticlockwise, these change states when their counters reach a certain amount. In the image you can also see the state I displayed in text of the selected object.

****

In this image you can see the AI state machine which cycles between wandering, hunting the player or hunting the power up depending on their node distances between itself and the player or power up.

**Pathfinding:**

****

In this image you can see the AI using A\* pathfinding algorithm to head towards the power up.

**Pushdown Automata: (Menu system)**

**Text

Description automatically generated**

In this image you can see the menu system which is implemented using pushdown automata, this also allows the levels to be paused and resumed from the point they were paused at.