Objects in the world

- Ground – any ground
- Quadruped
  - Lots of Bears, one alienRobot
- Props
  - Bell

Click-a-bear Game Description

- The game has 3 or more bears, spread out on the ground and a bell in the top right corner.
- The player is prompted for how many bears to click on. (The player must type in a number between 2 and the number of bears)

Click-a-bear Game Description (2)

- The bell welcomes the player to the game and tells them they must click on a specific number of bears (the number they entered)
- The bell says to click on it to start the game
Click-a-bear Game Description (3)

• The player clicks on the bell.
• The bears all drop down below ground and the number 0 appears in the top left corner.

Click-a-bear Game Description (4)

• The bears then randomly pop up to the ground (only one at a time) and then drop below ground.
• The player tries to click on them. If they get a bear, 1 is added to the number displayed.

Click-a-bear Game Description (5)

• When the player clicks on the specified number of bears, the bears stop popping up.
• An AlienRobot comes down from the sky to the ground and tells the player that they did great!

Group Work

• In a group of 3 or 4 discuss how to set up this game by discussing the following questions.
• Don’t write any code yet! Just discuss how you would design this game
Question 1

• What things would be good to keep track of? These could be variables, properties, arrays?

Question 2

• Any functions or procedures that would be helpful to write?

Question 3

• Outline which events would be useful to have and the flow of the program? Which parts of the program would be in myFirstMethod vs in an event and which event? Consider the timing…
Does your design consider these?

• If you want to add more bears in, does your design still work?
• If the user types in the wrong number (-3 or 20) what happens?

Discussion Time

• We will discuss in class and then you can start coding the game.
• Slides from here on are notes taken in class!!
Variables/properties

- **answerNumber** – whole number (number of bears total to click on) – **setup as a scene property**
- **scoreNumber** - start at 0 and increment every time click on a bear
- **arrayOfBears**
- **Boolean variable – gameOn**
  - True when we are playing the game
  - False when not playing
  - Start false, set to true when ready to play
  - Turn back to false when game is over

Procedures

- **Procedure move bears up and down**
  - Just be Move one bear that is a parameter
  - Another loop somewhere else determining random bear to send as an argument
- **Update counter** – add one to counter and redisplay new value
- **Initialize counter** – all the setup for counter
- **Setup – other kind of setup**

Flow of the game

- **Event – when you click on bell, game starts**
  - Setup
  - Reset gameOn variable to true
- **Event – click on any bear in the bear array**
  - Update score
  - check if hit max score

MyFirstMethod

- Mostly run the game here
- Go though instructions first
- Ask the user how many bears to click on and set that variable
- While user input no good
  - Ask them again
- More instructions – tell them to click on the bell to start
myFirstMethod (cont)

• While not Gameon
  – Do nothing
• While gameon - play the game
  – Randomly pick a bear to move and down
• Now game is over – do whatever when game is over