|  |  |
| --- | --- |
| **Session Code** |  |
| **Module** | Basic |
| **Teaching Unit** | Project |
| **Learning Outcome** | Create a Clicker Game with Pygame |
| **Resources** | Teacher:   1. Laptop along with audio and video exchange 2. Notebook and Pen (To note any development from session)   Student Resources   1. Laptop along with audio and video exchange 2. Notebook and Pen (To keep note of important parts in the session) |
| **Duration** | 50 Mins |

|  |  |  |
| --- | --- | --- |
| **Structure** | **Warm-up**  **Pace-up Activity**  **Knowledge Transfer**  **Student-Led Activity**  **Short Quiz**  **Heads up tip for next class** | 2 Mins  5 Mins  20 Mins  10 Mins  8 Mins  5 Mins |

|  |  |  |
| --- | --- | --- |
| **Step** | **Say** | **Perform** |
| **Warm-up (2 Mins)** | Hi *student name****,*** how are you?  Are you excited for the last class of the first module??  I am super excited and sure you will also enjoy today's class, in fact, every class after this because we are going to start with graphics, animations, sounds and a lot of other fun. | Engage with the student in conversation. |
| **Interaction**  **(10 Mins)** | We learnt important concepts of programming.  Till now, what we covered was the basics of programming and without that, we cannot create apps, programs or games. But now we are ready to move to the best part of programming.  Our next module will be full of games and then we will create interesting apps and the final module will be the most important and interesting of all that is Artificial Intelligence and Machine Learning.  Let us just quickly recollect all the topics we have covered till now.  Variables, Datatypes, Conditions, Loops - while and for loop, Data structures- List, Tuple, Sets and Dictionaries, Built-in Functions and Libraries and of course we created our own functions.  Do you remember what we created in our first class?  Yes, we created a ball that moved on the x and y-axis.  Let’s see that code again.  Let me know if you understand the code better this time.  This time we will learn a step beyond this concept by designing a game where we will use all those concepts, we have learned so far.  ­­­In this game we will learn to use Pygame library. In it we will see how to insert an image in the game, how to move an object along x and y axis, how to interact with the object that are in the game for e.g., how to check if we have clicked something or not.  Basic idea of this game: coins will be falling from the top of the window. Player can click on the falling coins to increase the score. And display the score.  The motive of this 1st class will be to make the student familiarize with pygame and build a basic pygame window by inserting a background. | Let the student answer. |
| **Knowledge Transfer**  **(20 min)** | **pip install pygame**  First, explain what pygame library is and what are its alternatives available in python. Tell the students how to install a library and what pip is.  **import pygame**  Ask the students to import the pygame library on their own without showing them the code.  **pygame.init()**  Here teacher needs to explain what this method means and what it does  **width = 600**  **height = 600**  These are width and height variables for the game window/screen whose values are in pixels. We can change it to any size we want.  Show this to students by trying out different sizes like 350x400, 700x800, etc.  **Window = pygame.display.set\_mode**  **((width, height))**  Explain student what a window is and how the coordinates represent the pixels of the window. Like (0,0) represents the starting of the window which is top-left corner.  **pygame.display.set\_caption("Clicker Game")**  This line will set the title of the window.  Ask students later to try this out by giving different names to it.  Remember any changes to this line till this point will not work as pygame will start and close itself immediately by showing up for just a frame(millisecond). So until we do not hold it up in a loop, changes will not be visible.  So do not ask students to try these above 6 lines of code alone until ahead we add a while loop in our program.  !!! Before moving ahead make sure students have the understanding of ‘while’ and ‘for’ loop.!!!  **run = True**  **while run:**  **for event in pygame.event.get():**  **if event.type == pygame.QUIT:**  **run = False**  Make sure students understand this loop properly as this is the brains of the game.  First show the loop till here only we will add the background color to it later.  Explain how the condition of this while loop helps pygame window to remain active until user closes it and how the loop’s condition becomes false when user clicks on “X” (close button of window) & the loop terminates resulting in closing of pygame window. Also explain what are events in computers.  **pygame.quit()**  This is self-explanatory. After this line let the students experiment with the width & height of the window.  Ask them now to try changing the title of the game.  Till here our pygame window will appear in the size we wanted. But it will not have anything in it as we haven’t added anything to it. So, at this point after running the program its screen might show any distorted color.  Any doubts till here?  *If any please help the student.*  **window.fill('white')**  **pygame.display.update()**  Now, we will add a background color to our game by adding these two lines in the ‘while’ loop at the end but outside of the ‘for’ loop.  Ask students to try changing colors of the background now. | Teacher has to refer to the document “Teacher’s Reference” to have understanding of each and every line in depth and mandatorily explain it to students. |
| * **Ask the student to share their screen and click on student activity 1** | | |
| **Student-Led Activity** | Help the student do all the above.  *This class is very important for the child to feel that they can do coding independently and they are getting better at it.*  Ask the student to make a pygame window of size 455x346 pixels and add a title to it “Vertical window” and then put a background color “orange” to it. | [Student Activity 1](https://replit.com/@testcodingtest/BouncyBall#main.py)  *Do not dictate the code instead ask the student to do every step one by one and just guide.* |
| * **Ask the student stop sharing screen** | | |
| **Revision** | You did wonderful today.  In programming, you can find out a lot of things just by experimenting.  With a language like Python, there is so much to explore.  I am so excited to create some fun games with you.  Did you like today’s class?  So, Pygame is a very interesting library. It has so many things to try out.  In the next class we will try inserting an image into our game window and learn to resize it to make it fit inside our game window. | Let the student share his/her experience. |
| **BID GOOD BYE & END CLASS** | | |

**Resources:**

|  |  |  |
| --- | --- | --- |
| **Activity** | **Name** | **Links** |
| Teacher’s Reference | Teacher’s Reference | [Teacher's Reference](https://github.com/rajaatreja/Clicker-pygame/blob/main/Teacher's%20Reference.docx) |
| Student’s code for 1st lecture | First\_lecture\_code.py | [1st lecture code](https://github.com/rajaatreja/Clicker-pygame/blob/main/First_lecture_code.py) |
| Full Game | Clicker.py | [Clicker Game Complete Game](https://replit.com/@RajaAtreja/Clicker-Game#main.py) |
| Complete Game with resources (coin image and background images) | Clicker-pygame | [clicker pygame](https://github.com/rajaatreja/Clicker-pygame) |