MP06 - Bubble Game Design Template

Keep the game SIMPLE. You have 55 min to complete this task.

file:///Users/kaceychen/Downloads/ATLS%201300/pinggame.py

Key terms:

1. **The user** - the human (or computer!) who is controlling the game play.
2. **The player** - the game representation of the user

Games will always have users, but may not have players!

Example ideas:

- Click bubbles within time limit (challenge: key interactions)
- Move a bubble to a target & avoid obstacles within time limit
- “Catch” bubbles in a target (score/time limit)

More inspo: Search “clicking game” or “simple game” at codepen.io

Group Members (Full names)
Lily Dykstra **Sophie Luu** **Kacey Chen**

UML DIAGRAM:

	Bubble
Attributes (nouns related to bubble)	Position (x,y) Color (palette) Size (radius)
Methods (actions)	move() disappear()/click() draw()

1. Describe your gameplay using game components

Assume the user will know the rules and objectives (you will not need to teach or show text on screen). The questions below will help you get organized & work quickly

The screen has a collection of stationary bubbles on the left side of the screen. On the right side of the screen there is a paddle that you can **move** up and down with your arrow keys. At the start of the game a ball will launch from your paddle to the bubbles. The ball will disappear bubbles upon collision. The ball will bounce back to the right side of the screen and the player will try to catch it with the paddle and the ball will bounce back to the left side. This will repeat until all the bubbles are popped, time ends (30 s) or until the player misses the ball with the paddle and the game ends.

2. Organize your code based on your description

1. Besides "user", bold and change the color of all of the **nouns** in your description in one color. (Color of your choice)
2. **Highlight** the verbs in your description. (Color of your choice)
3. Below your description, write a list of unique nouns and functions (no duplicates). Recognize where 2 functions can be the same (these functions would use arguments). Try to organize them by themes or obvious groups. Some nouns may become the name of a group.

Circle click, countdown times, bounce and collide code will be provided, without cite/borrow restrictions.

1. **Objective: What needs to be accomplished to win the game**
   What is the goal of the game? How will the game end? (NO LEVELS ALLOWED)

To **pop** all the **bubbles** on the other side of the screen with the **ball** while still **catching** the **ball's** original side of launch.

2. **Rules: What can and cannot happen in meeting the objective, usually relates to timing or scoring if applicable**
   What are the timing limitations (if any)? What can't the user do?

   The user has 30 seconds to **pop** as many **balls** as possible. If you **finish** them all in the 30 seconds you win, if not it will just **tell** you what you scored.

   PC09: player cannot miss the ball; the game ends.

3. **Challenge: A procedure, action, confrontation or difficulty the player must enact in order to achieve the objective**
   What makes the game difficult or interesting? (Consider animations!)

   The player has to **pop** all the **bubbles** within a time limitation and must still "**catch**" the **ball** with their **paddle** or the game is over.

   The bubbles are on the opposite side of the screen so they are far away. You don't want to end the game by missing the ball on the right side. The time is running down so you are in a time crunch to pop all the bubbles.

4. **Interactions: How the player, enemy and/or environment affect each other**
   How will the user interact with components on the screen? Is there a player? How will they interact with each other?

   The user **moves** the paddle with the up and down arrow keys

   How will the user know the game has started? (Window opens? Start buttons are hard)

   PC09: Once the **ball launches** from the start **position**, the game has **started**. What about th bubbles

5. How will the user know they have succeeded or failed? (Background changes color? Text on screen?)

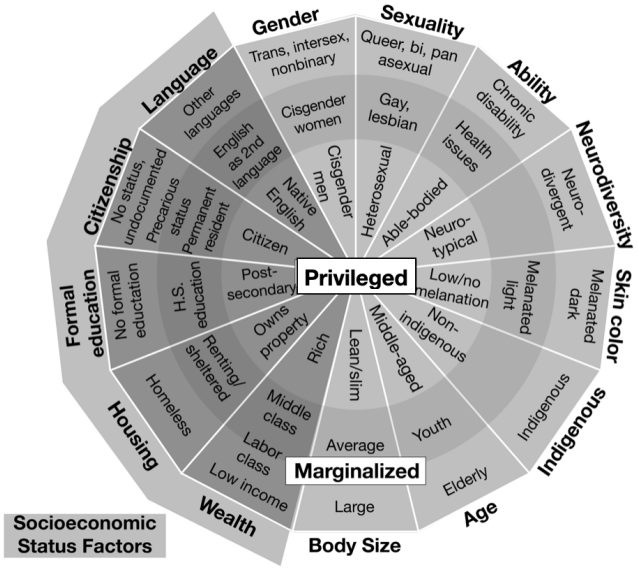
   Missing the ball (PC09) . If all bubbles pop (disappear), **prints** "You Win!" in the terminal

Nouns	**Verbs**
Bubble, paddle, position, Reach goal: timer PC09: ball	Disappear, move, collide, PC09: launch, catch,

3. Consider your users

"ATLS 1300 student" or "college student" is not enough, as these identities imply uniform access, ability and tech literacy that can be exclusive. Consider who **can** play your game. Then choose **2 marginalized or excluded traits** (age, gender, interest, ability, race, neuro status, housing status, etc.) and describe how you will accommodate them in your design beyond the required inclusive coding. (Give your best considerations, do a quick internet search. You will get feedback & resources.) Use the power wheel to help you.

Trait 1	**Trait 2**
Inclusive to Users with Limited Mobility- our game only requires the user to press the up and down key, so users with limited mobility will be able to play the game without using large muscle groups (only one finger is required to play).	Inclusive to All Ages- our game is simple enough that nearly any age can play it (excluding babies and toddlers). Older individuals will be able to play because of the simple interactions (only having to use the up and down key) and straightforward objective.



Simpson, 2009

4. Write *rough* pseudocode

Pseudocode describes your program algorithm in plain English (or language of choice). It should be universal (not dependent on language or library). Consider:

- What will you need to define first? Draw first?

- What parameters do you think a function may use? Write down some guesses
- What will go in the Animation loop? When will functions get called?
- Conditionals will be click and end-game conditions. Where will those go?

Pseudocode:

import pygame

import sys

import random

SET UP GAME, VARIABLES, SCREEN, PALETTE

PC09: CLASS PADDLE - no parameters

- Move up
- Move down

CLASS BUBBLE

- Color
- Position
- boolean (hit or not hit for disappear/appear)
- Randomly select a color
- Run for loop to place 30 bubbles

    DEFINE BUBBLE ANIMATION - no parameters

    - Conditional for ball collisions
    - Making it move across the screen
    - Bounce off of the paddle

Finished = false

bubbleCount = 0

WHILE LOOP

- Calling the functions
- Event for loop (for event in pygame.get_event():
    - Key up, key down

    END GAME CONDITIONALS # also inside the while loop

- Check If all the bubbles are popped (use an attribute that is a boolean (self.draw = True). In a method in the bubble class, if <collision>: draw = False. Then, in the while

```
loop:
 If bubble.draw:
    bubble.show()
```

-
- if time<= 0 OR bubbleCounter == 30:
    - Finished = true
- If finished = true:
    - Running = false

5. Examples

Bellwoods (Description only):

1. **Objective** - collide the kite with a colored circle (target) in the scene that has not yet been found to advance to the next level
2. **Rules** - the user can move mouse to move the kite and and drag the view.
3. **Challenge** - only draw a small view of the woods at a time. Randomly place the target.
4. **Interactions** - as the kite collides with the trees, play bell sounds. The number of birds indicate the level, with the newest bird homes in on the next target
5. **Start** - Window opens. Click start to begin. Start button click loads game.
6. **End** - Next level indicated with color theme changes.

Basic Click Game (intentionally incomplete):

1. **Objective** - click all the moving **bubbles** before the **timer** ~~runs out~~ counts down to zero.
2. **Rules** - user disappears each **bubble** until all of them are **invisible**
3. **Challenge** - user must disappear each **bubble** before **timer** counts down and is ~~shown~~ displayed on screen. Moving **bubbles** are hard to click on. (incomplete)
4. **Interactions** - user presses key at a certain time, if a bubble is in a random location, it disappears. **Bubbles** bounce at window edges (never leaves window). (incomplete)
5. **Start** - **Window** opens, counter (**timer**) starts counting down (counter is reach goal)
6. **End** - If time runs out, print “Game over! You lost!” in the terminal (print function) print “You won!” in the terminal.

Nouns	**Verbs**
Bubble Timer	move display

Grouping example: Bubble + move + disappear

Nicely done! You rock! Have a great weekend and read **Ch 14 in Python for Everybody**!