Butterfly Game

Authors: Emily Choi, Asha LaManque, Claire Oh
Revision: 4/12/22

Introduction:
You're a butterfly ready to embark on a new adventure to discover yourself and the world around you. You want nothing more than to spread your wings and soar along the clouds. As a butterfly, the world is a dangerous place for you, but you have a dream to explore as much of it as possible. In this side-scrolling runner game, you will be dropped into this deadly, but exciting world as an egg, but will slowly grow overtime into a caterpillar, a pupa, and eventually a butterfly. Before you start, you'll get the opportunity to choose what kind of butterfly you'll end up being. To live out your dream you will have to avoid all potential enemies and obstacles and consume enough food. If you run into an obstacle or enemy or don't eat enough, you will die. This isn't the end of the world as you'll reset back to the start of the section you perished in. Our primary goal in writing this program is to create a fun and interesting game that people will be able to enjoy while learning more about the life cycle of butterflies. The primary feature of our program is the main side scroll esque game with a character selection feature at the start. The game will include the different phases which will change the style of the game. Our primary user base would be people who enjoy games.

Instructions:
This program is divided into phases, in which each phase represents a different task. Each phase can be attempted an unlimited amount of times, until the player succeeds in completing the phase. The first phase, you are an egg, ready to emerge as a caterpillar. Use the spacebar to cycle through any designs/color combinations. Once a design is selected, click on the screen. The next phase is the caterpillar phase, which is entered automatically after the egg phase. Up and down arrows keys will be used to toggle between movements. If the user moves through a leaf, it is automatically collected. At a certain point, if you survive, the next phase (butterfly) occurs. Use the spacebar to avoid obstacles and collect flowers. Eventually the game will end, press esc to exit out of the screen.

Features List (THE ONLY SECTION THAT CANNOT CHANGE LATER):
Must-have Features:

- Starting menu that allows users to select butterfly sprite type or start the game. (Butterfly sprite type selection will dictate the player's sprite during the butterfly phase of the game, with the differencing between the sprites being appearance)
- Two-Dimensional pixel graphics with basic colors and symbols. Graphics will mostly be composed of various sprites placed on a pixel-styled background.
- Players control a character as it runs in an automatic side scrolling map to avoid obstacles and collect the required amount of "leaf" collectables to pass the phase. Players can use the up and down arrow keys to make their characters jump or crouch/fly up or down. Obstacles will be located at certain heights as they move closer to the character and the player will have to avoid them by moving their character. Obstacles differentiate from one another based on

shape and size. For example, some objects may be rectangular in shape and others may be circular.

- Game has multiple phases that correspond to the butterfly life cycle. These phases will affect the graphics and obstacles that the player will encounter. For example, the butterfly phase will contain obstacles that are attached to both the top and bottom of the screen as compared to the caterpillar phase, which only will have obstacles attached to the bottom of the screen. However, a type of obstacle (a predator) differs from this rule. They are located roughly midway in the screen. The purpose being to make collecting collectables, like leaves more difficult. Aside from this, predators operate similarly, with the exception of possibly different graphics or designs.
- Players can win or lose the game through collision with obstacles or hunger (player has not collected enough “leaves” during caterpillar phase) and then can replay the phase they died in, an unlimited number of times until they die. Players can replay the phase they are in an unlimited number of times until they are able to move on.
- Obstacles that move along with the side scrolling map that the player has to jump over with the up arrow key as a caterpillar or duck and jump with the up and down arrow key as a butterfly

Want-to-have Features:

- Starting menu that allows users to customize a butterfly sprite type. This could allow for customization of different colors.
- GUI bar that indicates hunger in relation to the amount of “leaves” the player has collected. If the bar goes down to zero, the player dies. Next to the bar will be the exact number of leaves consumed. If there is a surplus amount the bar appears the same. However, the number on the side will still increase.
- Animated two-dimensional graphics with colors and symbols. Animation in this case refers to movement. For instance, an animated butterfly will be shown moving its wings.
- An additional game phase where the player’s sprite becomes a cocoon (which would be played after the caterpillar stage) and they must repeatedly press a button to release it; this phase may look like a rhythm game. Once the player wins this part of the game, the cocoon hatches. Then the play continues with the butterfly stage.
- A high score system in the menu that keeps track of the highest score which is based on how many collectables (for example, leaves) the player picks up during their run.
- Have a certain amount of times a player can play each phase. If they can’t pass the phase in a certain amount of tries, the game restarts back to the egg.

Stretch Features:

- Three-dimensional graphics with colors and symbols.
- Multiplayer functions that allow for a split screen effect for two players to play the game at the same time and compete against each other.
- Multiplayer functions that allow for users on different computers to play the game at the same time and compete against each other.

Class List:

- **Main** - Initiates the game
- **DrawingSurface** - Holds all the panels (of the different phases)
- **Screen (abstract) -** Represents a screen, or the UI
- **EggPanel (extends Screen)** - the graphics for the egg phase (Also shows instructions)
- **CaterpillarPanel (extends Screen)** - the graphics for the caterpillar phase
- **ButterflyPanel (extends Screen)** - the graphics for the butterfly phase
- **MenuPanel (extends Screen)** - the start point of the game, which allows the user to play the game and view instructions
- **Player (abstract)** - Holds primary player information, such as the image of a player's sprite and the movement of the player.
- **Egg** - Represents the player during the egg phase
- **Caterpillar (extends Player)** - Represents the player during the caterpillar phase
- **Butterfly (extends Player)** - Represents the player during the butterfly phase
- **Obstacle** - Represents any obstacle that the player may encounter
- **Predator (extends Obstacle)** - Represents any predator that the player may encounter
- **Collectable -** Represents a collectable (such as flowers or leaves) that the player eats

Credits:
[Gives credit for project components. This includes both internal credit (your group members) and external credit (other people, websites, libraries). To do this:

- List the group members and describe how each member contributed to the completion of the final program. This could be classes written, art assets created, leadership/organizational skills exercises, or other tasks. Initially, this is *how you plan on splitting the work*.
- Give credit to all outside resources used. This includes downloaded images or sounds, external java libraries, parent/tutor/student coding help, etc.]

Credits:

- PhysicsGameDemo
- Asha's Shapes Library

Responsibilities List:

- **Asha** - Egg, EggPanel, Player, Main, MenuPanel, Collectable
- **Claire** - Butterfly, ButterflyPanel, Predator, Obstacle
- **Emily** - Caterpillar, CaterpillarPanel, DrawingSurface, Screen

GitHub:
https://stackoverflow.com/questions/1439022/get-mouse-position

Help:
https://examples.javacodegeeks.com/desktop-java/awt/event/simple-key-press-listener/#:~:text=Use%20KeyEvent.,which%20key%20the%20user%20pressed.

Libraries Used:
Processing
Point
Java awt
Java swing

Extra Bonus Concept Images: