**Project Proposal Components** [15 pts]  
Write up a proposal file (in the file proposal.txt, or .docx, or .pdf) which should include the following components:

* **Project Description** [2.5 pts]: The name of the term project and a short description of what it will be.

Term Project Name: The Circle of Life.

Description: “The Circle of Life”, is an engaging Python game that follows the life journey of a baby bird through 3 different levels, representing its different stages in life. The journey begins with the bird as a baby, navigating its way in a vibrant forest, overcoming obstacles, and collecting seeds to grow. In the first two levels of the game, the bird adapts and learns to avoid obstacles by jumping and double jumping as a way of moving. With each level, the intensity increases. In level 2, there are two obstacles that now appear. As the player reaches the final stage, level 3, the bird is finally able to fly and introduces a different way for the player to engage with the obstacles and seed collection. The game starts with the bird starting as a baby and through each level it undergoes a change in appearance and powerups, symbolizing the circle of life.

* **Similar projects** [2.5 pts]: A 1-2 paragraph analysis of similar projects you've seen online, and how your project will be similar or different to those.

Bounce: In this game players control a ball through different levels, collecting coins and avoiding obstacles. “The Circle of Life” adds a theme and narrative arc to the base mechanics of what bounce is built on. The bird’s journey adds emotional depth to the gameplay with specific environment changes also.

Flappy Bird: In this game players control a bird to make it fly and maneuver through pipes and obstacles with a tap. “The Circle of Life”, takes inspiration from the game mechanics to add the flying feature as a part of the narrative structure and progression, as the bird unlocks the new ability to fly in the final stage of the game. Flappy Bird is endless tapping but “The Circle of Life” has an end, providing the players some completion and closure, with a sense of accomplishment.

* **Structural Plan** [2.5 pts]: A structural plan for how the finalized project will be organized in different functions, files and/or classes.

1. Game class will organize all the main functions of the game, this will include all the starting constraints when the game is initialized, coordination between the three levels, and what needs to be printed on screen to the user. The definition displayIntro will display the introduction and provide instructions for how the game is to be played. The main function will also call, playLevel to keep a track of the level. The total seed collected throughout all levels will also be stored in this function along with the final message at the end of the game. It will also include an instance of the Bird class.
2. The Bird class contains all the visualization properties of the bird and its behaviors throught the levels. This helps us draw the bird on screen. The jump function inside helps us define the action of the bird to make the bird move up in the game and to change its position while jumping too. The fly class will also be introduced to enable flying in the final stage, level 3.
3. The Obstacle class will contain a definition function to generatePositions for generating random obstacles, the obstacle logic changes and becomes more sophisticated throughout the levels.

* **Algorithmic Plan** [2.5 pts]: A plan for how you will approach the trickiest part of the project. Be sure to clearly highlight which part(s) of your project are algorithmically most difficult and include some details of how you expect to implement these features.

The trickiest part seems to be the flying part of the game. It changes the ground-based mechanics to aerial movement. Here’s an algorithmic plan for how I could potentially implement this.

1. First to identify what level the bird starts flying. Which is Level 3.
2. Within the bird class define a fly function to simulate flying. Involved sustained upwards movement in comparison to the jumping movement.
3. Within the Game class, playLevel function, check to make sure in the game loop at the current stage level 3, we call the fly method.
4. Defining the controls for flying, whether it is upwards only or includes downward too and whether the bird moves horizontally or vertically.
5. Adjust the obstacles for the third level, different height angles to make the players move horizontally and vertically.

* **Timeline Plan** [2.5 pts]: A timeline for when you intend to complete the major features of the project.

To achieve Preliminary Code by Monday 11/27:

1. Set up Project Structure and basic classes
2. Implement a drawing logic for the bird
3. Add jumping action to the bird
4. Implementing seed collection
5. Creating random obstacle generation
6. Level progression and bird growing in size
7. Trying game loop first pass
8. Document code and outline progress with key features of the game for TP1

For working demo that satisfies requirements of MVP by Monday 12/01:

1. Create flying stage and flying action for the bird
2. Update game loop to change action to flying at the third level
3. Creating controls for flying stage
4. Maybe adding additional features to the obstacles and seed collection strategy
5. Test and document and prepare project for TP3 final submission

* **Version Control Plan** [1.5 pts]: A short description **and image** demonstrating how you are using version control to back up your code. Notes:
  + **You must back up your code somehow!!!**
  + **Your backups must not be on your computer** (ideally, store them in the cloud)

I am using Git to manage and backup my code, my code is also hosted on GitHub.

I have initialized a Git Repository and committed my code to it. I have also created a Remote Repository and linked my Local and Remote Repositories to each other.

Link to Repository: <https://github.com/sherft2607/CS15112-Python-Game.git>

A screenshot of a computer

Description automatically generated

I have my files also backed up on Google Drive.

* **Module List** [1 pts]: A list of all external modules/hardware/technologies you are planning to use in your project. Note that any such modules must be approved by a tech demo. If you are not planning to use any additional modules, that's okay, just say so!

Not using any external modules.

* **TP1 Update:**

Instead of jumping across obstacles that would be an empty space in the ground, I have added blocks to make the bird jump up and the obstacles have become more engaging and active. They are moving in the sky and the bird must avoid it now. This will further translate when the bird will be able to fly in the last level.

Accomplishments: Progress since TP0 - List out all features you have implemented, as well as a brief description of how you implemented them.

Startscreen: Game start screen, started by pressing space.

Bird Movement: The bird can jump, move down, left, and right.

Block Collision: The bird collides with blocks to jump on top of it. checkBlockCollision checks if the bird collides with the block, if there is a collision the bird’s position changes.

Seed Collection: The bird collects seeds in the environment. The collectSeed method tracks seeds that have been collected and the checkSeedCollision checks for when the bird collides with the seeds. collectedSeeds stores the list of all seeds collected.

Health: The bird has health and on colliding with the obstacle it loses one life, on reaching zero the game is reset.

Obstacle: An obstacle moves horizontally across. Obstacle class defines the object moving and the move method defines its movement.

Landscape Elements: Elements like sky, clouds, ground, trees and seed counter are created. The Landscape class has all static methods like drawSky, drawClouds, drawGround, drawTrees, and drawSeedCounter.

App Controls: Key pressed responds to bird movement, implemented with onKeyPress and onKeyRelease, onStep function handles obstacle movement

Goals:

Better Gameplay: Add more features, levels and challenges.

User Interface and Animations: Refining the graphics and adding Animations.

Obstacle Interaction: More complex interactions of birds on colliding with the obstacle.

Customization: Better game environment, bird appearance and other elements.

Levels: Implementing more game levels.

Screens: Start, Pause and Game Over Screens.

* **TP2 Update:**

Accomplishments: Progress since TP1 - List out all features you have implemented, as well as a brief description of how you implemented them.

InstructionScreen: Displays information about game controls, objectives and rules. Players proceed by pressing a button.

LevelScreen: Indicates current level and provides a button to start the game. Game proceeds to the next level on completion of one level.

Pause/Resume Button: Game can be paused by pressing the ‘p’ button.

Restart Button: Restart current level allowed by clicking button. Position of Bird, Seeds and Health are reset.

Obstacle Movement: Obstacle faster in level 2.

Bird Movement: Bird can jump higher now in level 2.

Level Completion: Screen appears to show level completed.

Notes

Tried adding sound.

Tried adding in png files.

Tried implementing a side scrolling feature.

Goals:

PowerUps

SideScrolling

Random Y location for platform in terrain