Lawn Warrior: A game in Python

Evan Rice

5/9/2011

Application Development Section

Project Concept Proposal

- **Purpose:** Lawn Warrior is a 2D top-down game written in the Python programming language.
    - **Context:** Lawn Warrior is a multiple-level lawn mowing simulation game that brings all the excitement of regular yard maintenance to any computer capable of running Python applications.
    - **Goals:** Lawn Warrior is intended to provide a fun, challenging gaming experience for all ages.
    - **Audience:** The main audience is casual computer gamers.
    - **Functionality:** Lawn Warrior will include 10 levels, multiple enemies and obstacles, and an engaging storyline that takes the player through a variety of lawn related challenges.
    - **Milieu:**
        - Lawnmower: A Strategy Game for Kids - http://bit.ly/3HXkMd
            - Made in Flash
            - Runs in browser
            - Controlled with mouse
        - Sunday Lawn - http://www.donutgames.com/play/sunday_lawn
            - Made in flash
            - Integrated high score sharing
        - Lawn Mower - http://www.terminalstudio.com/lawnmower.shtml
            - 50 levels
            - Muliplayer
    - **Novelty:** Lawn Warrior will be playable on Windows, Mac, or Linux since they all run Python. While the concept is similar to other lawn-care based games, the style and execution of Lawn Warrior will be both original and exciting.
- **Resources:**
    - Python
        - http://www.python.org/
    - Pygame
        - http://pygame.org/news.html

- **Challenges:**
    - Sprite Creation
    - Importing premade levels
    - Soundtrack
    - Unfamiliarity with Python programming
    - Enemy movement
    - Designing lawnmower, obstacle, grass, and enemy sprites
- **Measures:**
    - Game has at least 10 functional levels
    - Game keeps track of player HP ,speed ,and powerups
    - Patrolling enemies
    - Obstacles that impede the player
    - Avatar movable with arrow keys
- **Future Extensions:**
    - Voice overs
    - Larger character selection
    - Wider array of enemies and powerups
    - Pack game in EXE for Windows
    - Achievements
    - Save Games

Inspiration

- **Motivation:** I originally began studying Computer Science due to my desire to program games. I feel that programming even a relatively simple game would give me new insight into programming concepts and personal validation through achieving a long-time goal.
- **Profession:** Increased familiarity with graphic application development will be useful in the computer game industry. Any well designed application would be a good addition to a resume, and even minimal experience in game development would give me an edge up over others.

Vision and Scope

Lawn Warrior is a small, top-down lawn mowing simulator game, programmed in Python and designed to be usable in all of the three major operating systems. When it is finished, the game will have 10 levels filled with obstacles and monsters. The object of the game is to mow all of the grass on the screen before the time runs out. Throughout the game there will be powerups that increase player speed or HP. When HP is reduced to 0, the game is over and the player loses.

By the end of the semester, I expect to have a fully working game with multiple levels and an interesting variety of enemies and obstacles that can be played on any system running Python. There will also be an engaging, humorous story and a soundtrack. A few features that will definitely fall outside the scope of this project are multiplayer, non-keyboard controls, and good art design.

Software Requirements Specifications

1. Create an 800x600 game window
   - Evaluation Method: The windows will exist
   - Dependencies: None
   - Priority: Essential
2. Create the level area
   - Evaluation Method: Game windows filled with a 20 x 20 grid of 40 x 40 squares
   - Dependencies: Game Window
   - Priority: Essential
3. Lawnmower Player Avatar
   - Evaluation: Sprite that moves with keyboard
   - Dependencies: None
   - Priority: High
4. Grass textures
   - Evaluation: Grass gets shorter when mowed
   - Dependencies: Level Area
   - Priority: High
5. Enemies
   - Evaluation: Patrolling enemies that damage the player character
   - Dependencies: Level Area

    - Priority: High

6. Powerups and Obstacles
    - Evaluation: Player can be sped up, damaged, or slowed by environment
    - Dependencies: Player Avatar, Level Area
    - Priority: High
7. Import Levels
    - Evaluation: Game loads at least 10 different levels
    - Dependencies: Level Area, Enemies, Player & Obstacles
    - Priority: Middle
8. Level Completion and Changing
    - Evaluation: Progress to next level after all grass cut
    - Dependencies: Importing levels, Level Area
    - Priority: Middle
9. Sound
    - Evaluation: Simple music playing while game continues
    - Dependencies: None
    - Priority: Low
10. Voice Overs
    - Evaluation: Pre-recorded voices for characters
    - Dependencies: Sound
    - Priority: If time permits
11. Sprite Creation
    - Evaluation: Player, enemy, grass, and obstacle sprites that overlay the background and move.
    - Dependencies: None
    - Priority: High
    - Dependencies: Enemies, Level transitions
    - Priority: Low
12. Start and End screens
    - Evaluation: Start screen with start button, end screen which displays credits.
    - Dependencies: None
    - Priority: Middle

    - Updated: 26/9/11

System Design and Architecture

Updated 10/23/11

Classes:

- Player
    - __init__(self, x, y):
        - Size
        - Image
        - Position
- Timer
    - Limit(120) – Time limit for player to finish level
        - Timer.run() - Starts the countdown
        - Timer.refresh() - Refreshes game timer display every second
- Grass
    - __init__(self, x, y):
        - Size
        - Image
        - Position
        - Removed on collision with player
- Rock
    - __init__(self, x, y):
        - Size
        - Image
        - Position
- Enemy
    - __init__(self, x, y):
        - Size
        - Image

        - Position
        - Patrol
            - Randomly move 1 block N, W, S, or E

Functions:

- Populate_level(level):
    - Takes current level as argument
    - Reads in appropriate file containing level map
    - Translates into objects and gives them positions and groups
    - Set total amount of grass
- Draw() – Draws the game window, refreshing each time this function is called in the main loop. This function must be lightweight to maintain FPS.
- Main Loop() – Monitors player movement and checks for collisions after updating screen
- Sound() – Plays music files throughout game. Possibly also sound effects for events.
- Start() – Displays an intro screen with instructions. When the start button is clicked it runs the game.
- End() – Displays credits and ends game

Implementation

- README.txt – Brief instructions for using the game
- Lawn Warrior.py – The main game file
- Level1-10.txt – Levels. Read by Lawn Warrior.py
- Weightless.mp3 – Music track
- Grass.jpg – Background texture
- Tallgrass.jpg – Grass texture

Known Bugs and Other Issues

Lawn Warrior 0.4

- No sprite textures for objects. Only dummy blocks to represent them
- Level population has a weird error that makes it leave one row and one column empty
    - This causes level progression to fail, because the missing grass objects cannot be mowed, meaning you cannot complete the level.
- No sound
- No Timer
- No objects except grass

Lawn Warrior 0.6

- No player or rock textures
- No enemies
- Not timer

Preliminary Test Plan and Preliminary Test Cases

1. Create an 800x600 game window
    - The windows will exist
        - Creates a window for that the game will be drawn in
        - No test needed
        - Works
2. Level area
    - Game windows filled with a 20 x 20 grid of 40 x 40 squares
        - Game field is made of 400 separate squares
        - Make player character move across all 400 squares
        - Works
3. Lawnmower Player Avatar
    - Sprite that moves with keyboard
        - Player lawnmower sprite is controlled by keyboard presses
        - Move sprite to test
        - Works
4. Grass textures

   - Grass gets shorter when mowed
     - Grass is ‘cut’ by moving over grass blocks
     - Move player over grass so that it disappears, revealing grass background
     - Works

5. Enemies
   - Patrolling enemies that damage the player character
     - Enemies move randomly around map, damaging the player on collision
     - Run player into enemies and see if it damages the player
     - Works
6. Powerups and Obstacles
   - Player can be sped up, damaged, or slowed by environment
     - Different types of collectables populate the game
     - Pick up all items and ensure that they do what they are supposed to: Rock stops, mud slows, enemies hurt
     - Obstacles work.
7. Import Levels
   - Game loads at least 10 different levels
     - 10 separate level files that are read by the game
     - Complete each level to see that all of them load and finish properly
     - Works
8. Level Completion and Changing
   - Progress to next level after all grass cut
     - Output total count of grass, ensure that the counter is decreased when moving over grass. When counter reaches 0, the next level loads
     - Works
9. Sound
   - Simple music playing while game continues
     - Listen to soundtrack all the way though to ensure that it loops correctly
     - Works.
10. Voice Overs
    - Pre-recorded voices for characters
      - Ambient music stops and voice overs play between levels

11. Sprite Creation
    - Player, enemy, grass, and obstacle sprites that overlay the background and move.
        - Examine each object as it appears on screen to ensure that the sprites display correctly
        - In the case of the player sprite, see that it turns around as you change direction
        - Works
12. Start and End screens
    - Start screen with start button, end screen which displays credits.
        - Before level 1 and after level 10 separate screen fill the window, first a title, then credits
        - Works

Software Demo

- Short teaser trailer for the Winter 2011 release of Lawn Warrior

  http://www.youtube.com/watch?v=nbXXo_36Y1Y&feature=youtube_gdata

About the Author

Evan Rice is a senior Computer Science student at Berea College in Berea, Ky. He is currently employed as a Service Desk Technician Manager at the Technology Resource Center. When not on the job or designing software, Evan enjoys watching Doctor Who, reading Neil Gaiman, and writing short stories. He is originally from London, Ky. and plans to relocate to New York.

Executive Section

To: Dr. Jan Pearce, Project Director

From: Evan Rice

Subject: Twitter Blitzer

Date: 5/9/2011

Accomplishments:

- Created working project title and logo.
- Gathered information about what tools will be needed for the project
- Wrote Week 1 Project Report

Challenges: A good project name was difficult to pin down, and I am still open to changing it. The logo works well for now, but I am considering changing it as well.

Time Spent: About 1 hour making the logo. 3 hours researching. 2 hours writing project proposal.

Goals: Discussion of project with Project Director. Nail down project scope and begin planning code.

To: Dr. Jan Pearce, Project Director

From: Evan Rice

Subject: Lawn Warrior

Date: 11/9/2011

Accomplishments:

- Created new working project title and logo.
- Worked through tutorials for Python and Pygame
- Wrote Week 2 Project Report and rewrote Week 1 report

Challenges: Still working on a project name. Lawn Warrior is the best currently, but I am also considering Mr. Mow-It-All.

Time Spent: About 30 minutes making the logo. 6 hours doing python tutorials and reading documentation, 2 hours writing project proposal.

Goals: Discussion of project with Project Director. Begin planning code and designing levels and sprites. Change logo so that it has a grass texture background.

LAWN WARRIOR

To: Dr. Jan Pearce, Project Director

From: Evan Rice

Subject: Lawn Warrior

Date: 18/9/2011

Accomplishments:

- Created draft resume
- Updated Project Requirements
- Wrote Week 3 Project Report

Challenges: Trying to draw sprites myself did not work out as well as I hoped. I am going to get some help from some of my more artistically inclined friends for character designs for the player, enemies, and obstacles.

Time Spent: 2 hours failing to draw adequate pictures. 2 hours writing resume. 1 hour writing and updating report.

Goals: Find outside artistic talent to help with character designs. Write storyboard for at least the first 5 levels. Review resume with Project Director.

LAWN WARRIOR

To: Dr. Jan Pearce, Project Director

From: Evan Rice

Subject: Lawn Warrior

Date: 26/9/2011

Accomplishments:

- Made Software Architectural Model
- Wrote Week 4 Project Report
- Edited Resume

Challenges: Trying to draw sprites myself did not work out as well as I hoped. I am going to get some help from some of my more artistically inclined friends for character designs for the player, enemies, and obstacles.

Time Spent: 3 hours writing Software Architectural Model. 30 minutes editing resume. 1 hour writing and updating report.

Goals: Begin writing main and include files. Create all current classes in include files.

LAWN
WARRIOR

To: Dr. Jan Pearce, Project Director

From: Evan Rice

Subject: Lawn Warrior

Date: 2/10/2011

Accomplishments:

- Game window works with cut grass background
- There is a dummy player sprite that can move around the game window
- Wrote Week 5 Project Report

Challenges: Beginning implementation was difficult, and I realized immediately that some of the things I had planned weren’t going to work out. I am still trying to figure out how to set boundaries on the borders so the player can’t go out of bounds

Time Spent: 4 hours writing code in Lawn Warrior.py. 1 hour spent writing and updating report.

Goals: Solve boundary issues. Set and display dummy sprites for other game objects.

LAWN WARRIOR

To: Dr. Jan Pearce, Project Director

From: Evan Rice

Subject: Lawn Warrior

Date: 16 / 10 / 2011

Accomplishments:

- Implemented borders
- Implemented collision detection
- Wrote Biography
- Wrote Week 6-7 Project Report

Challenges: Borders were difficult. I tried a lot of different ways before I eventually decided on a good collision detection implementation that would work the same way for game objects.

Time Spent: 5 hours writing code in Lawn Warrior.py. 1 hour spent writing and updating report.

Goals: Rewrite code to eliminate relics of previous implementations and clean up. Get dummy models for game objects working and implement object dispersal. Might have to use pre-generated levels after all.

LAWN WARRIOR

To: Dr. Jan Pearce, Project Director

From: Evan Rice

Subject: Lawn Warrior

Date: 23/10/2011

Accomplishments:

- Importing levels from files works for the most part
- Collision detection between player and grass objects causes the grass to be 'cut'
- Counter decrements whenever a grass object is cut, and when all are cut the next level loads

Challenges: Spent hours and hours trying to find a way to manage the different sprites and collisions between them, and eventually settled on multiple overlapping groups. The populate level method definitely took up most of my time, but it works (for the most part). The file input to object type and position translation I think worked out beautifully and makes it really easy to design levels. There is a problem with the population method: It leaves one row and one column empty when they should be filled with grass, and I can't figure out why.

Time Spent: 14 hours designing, writing, and debugging new functions in Lawn Warrior.py. 2 hours spent writing and updating report.

Goals: Fix map population issue. Add other objects to game.

LAWN WARRIOR

To: Dr. Jan Pearce, Project Director

From: Evan Rice

Subject: Lawn Warrior

Date: 30/10/2011

Accomplishments:

- Importing levels works
- Grass sprites added
- Music plays
- Rocks implemented
- All 10 levels made

Challenges: Spent hours trying to solve the populating issue from last week, and finally realized it is because I used 18 rows and columns in the level files instead of 19. After that, everything went really easily.

Time Spent: 10 hours implementing code in Lawn Warrior.py. 1 hour spent writing and updating report. 1 hours pent updating resume.

Goals: Add more obstacles and enemies. Add timer.

LAWN
WARRIOR

To: Dr. Jan Pearce, Project Director

From: Evan Rice

Subject: Lawn Warrior

Date: 06/11/2011

Accomplishments:

- Enemies work
- Enemy movement works

Challenges: Got enemies to work, and to patrol in random directions along with the player, but I cant figure out how to always draw the enemy on top of the grass sprites. As it is, it sometimes disappears behind the grass.

Time Spent: 6 hours implementing code in Lawn Warrior.py. 1 hour spent writing and updating report.

Goals: Fix enemy display. Fix rock transparency. Implement PGU.

LAWN WARRIOR

To: Dr. Jan Pearce, Project Director

From: Evan Rice

Subject: Lawn Warrior

Date: 13/11/2011

Accomplishments:

- Fixed enemy display
- Implemented Intro and credits
- Enemies now have spider sprite

Challenges: None this week.

Time Spent: 4 hours implementing code in Lawn Warrior.py. 1 hour spent writing and updating report.

Goals: Get that timer working. Faster movement is a maybe.

To: Dr. Jan Pearce, Project Director

From: Evan Rice

Subject: Lawn Warrior

Date: 21/11/2011

Accomplishments:

- Made Demo Video
- Fixed Issue when loading level 10

Challenges: None this week.

Time Spent: 2 hours producing demo video. 1 hour spent writing and updating report.

Goals: Faster movement, Lawnmower sprite

LAWN WARRIOR

To: Dr. Jan Pearce, Project Director

From: Evan Rice

Subject: Lawn Warrior

Date: 28/11/2011

Accomplishments:

- Added music to demo video
- Reviewed other demos

Challenges: None this week.

Time Spent: 1 hour editing demo video. 1 hour watching other demos. 1 hour spent writing and updating report.

Goals: Faster movement, Lawnmower sprite (I will get these done!)

LAWN WARRIOR

To: Dr. Jan Pearce, Project Director

From: Evan Rice

Subject: Lawn Warrior

Date: 05/12/2011

Accomplishments:

- Finished Poster
- Finished Video Demo
- Finalized Game

Challenges: The poster was time consuming. There were no real last minute problems with the game.

Time Spent: 1 hour editing demo video. 3 hours making poster. 1 hour spent writing and updating report.

Goals: