2D Top-Down Shooter
for UTM CSCI 352 Spring 2017

Cole Davis and Mel Howard

Abstract

The proposed game will be a multi-level top-down shooter. The player character will use projectiles to defeat increasing hordes of enemies as he or she progresses through the game. The player progresses by defeating a predetermined number of enemies, allowing them to move to the next location.

1. Introduction

The game will let users create a Payer and control it vertically and horizontally using the WASD keyboard keys. The Player will travel to different locations, then fight and kill enemies with projectiles from a ranged weapon. This game's target audience is college aged gamers, and retro game enthusiasts. We hope the target audience will at first be engaged, and then extremely frustrated.

1.1. Background

A top-down shooter is a game where the user controls his or her character from a birds eye view of their avatar and the surroundings. The proposed game is largely inspired by *Journey of the Prairie King*, a mini-game in *Stardew Valley*. *Journey* is a top-down where the player's objective is to survive and defeat waves of enemies in stages, all while collecting items the enemies drop. The character creation and starting menus are inspired by the 2D infinite runner, *Magicite*, which randomly generates the player's character and begins game play immediately afterward.

1.2. Challenges

One of the most daunting challenges of the proposed game is animating the player avatar and enemies in WPF. Another issue we anticipate is updating the player score or "kill count" as the Player shoots at enemies.

2. Scope

When the proposed game is complete, the user will be able to generate a character, move it, and shoot projectiles using keyboard controls. The game will have a minimum of six levels that contain a predetermined and increasing numbers of enemies. When the Player shoots an enemy it will "die," and the player's score will be updated.

In the future, we hope to randomly generate levels and enemies, as well as implement an item drop system with items that power-up the player character. Another goal we have is to enable return fire from the enemies that cause damage to the Player.

2.1. Requirements

These requirements were gathered based on prior gaming experience.

2.1.1. Functional.

- Users will use keyboard controls to move Player around the map
- Player can shoot enemies with weapon – only one projectile can be shot at a time
- Projectiles will either be absorbed by enemies or leave the screen
- The Player must complete their current level to advance to subsequent levels

2.1.2. Non-Functional.

- The (undetermined) minimum frame rate should allow the user to play with suitable performance.
- The average time between a key press and the event it triggers should be 0.5 seconds and never exceed 2 seconds;

Use Case ID	Use Case Name	Primary Actor	Complexity	Priority
1	Move Player	User	Med	
2	Stop Player	User	Med	
3	Shoot	User	Med	

TABLE 1. USE CASE TABLE

2.2. Use Cases

Use Case Number: 1
Use Case Name: Move Player
Description: A user playing the game wishes to move the Player away from spawning enemies. They will press an arrow key corresponding with the direction to move in. This will trigger the process to move the player.

1) User decides which direction to move Player
2) User presses arrow key that corresponds with the desired direction
3) Player location is updated as the User holds down the key

Termination Outcome: The Player is now moving.
Use Case Number: 2
Use Case Name: Stop Player
Description: A user playing the game is pressing an arrow key to move the Player and releases the key to stop.

1) User decides where to stop the Playe
2) User releases the arrow key
3) Player stops moving

Termination Outcome: The Player is now halted.
Use Case Number: 3
Use Case Name: Shoot
Description: A user is playing a level in the game and enemies begin to spawn. They will press the space bar. This will trigger the process to shoot projectiles.

1) User identifies enemies spawning
2) User moves Player to aim at enemies and presses the space bar
3) Player shoots a projectile at the enemy

Termination Outcome: The projectile is shot in the direction the Player aimed

2.3. Interface Mockups

At first, this will largely be completely made up, as you get further along in your project, and closer to a final product, this will typically become simple screenshots of your running application.

In this subsection, you will be showing what the screen should look like as the user moves through various use cases (make sure to tie the interface mockups back to the specific use cases they illustrate).

3. Project Timeline

Go back to your notes and look up a typical project development life cycle for the Waterfall approach. How will you follow this life cycle over the remainder of this semester? This will usually involve a chart showing your proposed timeline, with specific milestones plotted out. Make sure you have deliverable dates from the course schedule listed, with a plan to meet them (NOTE: these are generally optimistic deadlines).

4. Project Structure

At first, this will be a little empty (it will need to be filled in by the time you turn in your final report). This is your chance to discuss all of your design decisions (consider this the README's big brother).

4.1. UML Outline

Show the full structure of your program. Make sure to keep on updating this section as your project evolves (you often start out with one plan, but end up modifying things as you move along). As a note, while Dia fails miserably at generating pdfs (probably my fault), I have had much success with png files. Make sure to wrap your images in a `figure` environment, and to reference with the `ref` command. For example, see Figure

4.2. Design Patterns Used

Make sure to actually use at least 2 design patterns from this class. This is not normally part of such documentation, but largely just specific to this class – I want to see you use the patterns!

5. Results

This section will start out a little vague, but it should grow as your project evolves. With each deliverable you hand in, give me a final summary of where your project stands. By the end, this should be a reflective section discussing how many of your original goals you managed to attain/how many desired use cases you implemented/how many extra features you added.

5.1. Future Work

Where are you going next with your project? For early deliverables, what are your next steps? (HINT: you will typically want to look back at your timeline and evaluate: did you meet your expected goals? Are you ahead of schedule? Did you decide to shift gears and implement a new feature?) By the end, what do you plan on doing with this project? Will you try to sell it? Set it on fire? Link to it on your resume and forget it exists?

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.