Tourney Journey

Revised Design Document

Matriculation Number: 100009273

Title

The chosen title for my game is **Tourney Journey**. This title was chosen in part because it accurately conveys what the game is about (as explained in later sections of this document), and partly because it rhymes. The logo for the game is displayed at the top of this document.

Genre

Tourney Journey will be a side-scrolling 2D fighting game. The game will consist of the player defeating waves of enemies in order to complete rounds. At the end of each map there will be a boss, and upon defeating it the player will move on to the next map.

Environment

Each map will feature its own distinct environment, which will be conveyed to the player through the pixel art which is used within that map. There will be little or no dialog present, so the changing scenery conveyed through the pixel art, as well as the mobs appearing in each round, will also be leveraged as a means of conveying plot to the player. Naturally, the plot will be very simple, with an overarching sense of a Hero's Journey as the player progresses to increasingly hostile environments. There will also be a sub-plot built into each map, although due to the lack of dialog all plot will be implicit rather than explicit.
Here is a rough plan for what each map will feature:

1. **Forest**: Features trees and lots of grass.
2. **City**: Features buildings and a fountain.

Player movement however will be handled on a pixel-based level, as is done in 2D Zelda Games, not cell-based as is done in Pokémon games, meaning the player will be able to partly occupy up to four cells at any one time. Maps will be toruses. All maps will be at least 1920 x 1080 pixels big since this is the size of the monitors on the lab machines, to prevent the same section of the map being in multiple places on the screen at once.

Player(s)

Tourney Journey will be a single player game. The player will be represented by a “Hero” avatar, which the player is able to control through a keyboard.

Opponent(s)

Opponents will come in the form of computer controlled enemy monsters, which like the player are represented as avatars on the screen. Each monster will be designed around the map it appears in. Each map will likewise feature a unique set of enemies, with a boss at the end. A physical description of each monster will be listed here.

Bushy: A weak, angry, crudely drawn bush with eyes and legs. Appears in the forest.
Minion: A red version of the player. Appears in both the forest and the city.
Bear: Originally from a Naruto game. Boss monster for the forest. See sources.txt.
Dog: A dog which appears in the city. Originally from a One Piece game for the Game Boy Advance. See sources.txt.
Leader (Boss): A bigger minion with a mustache and a top-hat. Boss monster for the city.

Rules/Mechanics

Controls

The player will control a single character (the hero) using the keyboard and mouse. WASD will be used to move the character.The player can also pause the game and go to the menu by pressing the escape key. G turns gravity on or off, and T slows down time or speeds it up again.

Items

Monsters randomly drop stars, hourglasses or hearts when killed, which refill time points, gravity points, or health points respectively when picked up.

Camera

The camera is centered on the player and follows him. It moves a bit slower than the player, but always maintains a maximum distance. It is able to cross the edges of the torus seamlessly.

Combat

The player damages monsters by jumping on them. Other collisions with monsters result in the player being damaged. There are also tiles which cause damage to both the player and monsters. This is illustrated by the water in the city stage. This can be leveraged in combat, for example when battling The Leader.

Environment Interaction

Each cell in the grid composing each map will have a designated state. This will determine things such as whether the player is able to move onto that cell (ex walls), whether the player takes damage while occupying the square (ex spikes), or whether the tile is an exit. Exits only appear once all waves on a level have been cleared, and take the player to the next level.

Goals

The ultimate goal in **Tourney Journey** is to complete all levels without the player's HP reaching 0. However, as the player progresses through the levels there will also be more short term goals, such as defeating the next boss to progress to the next map, or even completing the current round.

Game Loop(s)

The game loop would look something like this:

```
// update game state
if there is player input
      update player state
for each monster
      if monster is able to attack player
            attack player
      else if monster can see player
            move monster towards player
      else
            wander randomly
for each item
      if the item is in the air
            make it fall

// render game
render player
for each cell in grid visible on camera
      render cell
for each monster visible on camera
      render monster
for each item visible on camera
      render item
```

In Context

The tournament structure in Tourney Journey can be found in a lot of fighting games, as well as games like the Paper Mario series, which features the Pit of 100 Trials in Paper Mario: The Thousand Year Door[1] and the Sammer's Kingdom Tournament in Super Paper Mario[3]. A 100 stage tournament is also seen in Pokémon Colosseum[2].

Platform

Tourney Journey will be designed for the PC. This choice was made because touch screen based devices don't really lend themselves to games where user input comes in the shape of rapid button mashing due to the lack of tactile feedback, as is the case with most traditional fighting games and will be the case for this one. The game will be coded in Processor.

Bibliography

[1] Paper Mario: The Thousand-Year Door (2004), developed by Intelligent Systems, published by Nintendo.

[2] Pokémon Colosseum (2003), developed by Genius Sonority, published by Nintendo.

[3] Super Paper Mario, developed by Intelligent Systems, published by Nintendo.

NB Sources for game assets can be found in sources.txt.