CS-447 Project 2

November 4th, 2020

Group Members

Roman Stolyarov, Daniel Yarmolenko, Vitaliy Kudrik, David Barko

Game Overview

Description

The game is called *Time Lapse*. As the main character, you get trapped in the past (medieval era) and your goal is to return back to your timeline (the future). You are a futuristic and modified humanoid that has access to special powers once certain items are collected that help you defeat enemies around you. The goal is to collect time machine parts scattered around the levels in order to rebuild the time machine, and finally return back to where the main character belongs.

Genre & Camera Position

The genre of this game can be placed under a rogue-like action-adventure game. The camera will be looking from a bird's eye view with a little tilt to show walls and the players feet. The idea is, you control the player from this view so you are able to see everything around you (be able to move up, down, left, right → WASD) and not cover unique expressions to the player's and enemies' animations and artwork. The tilt allows to show more artwork rather than a complete view of a bird hiding certain elements to the artwork.

Controls & Objectives

The player's objective is to fight enemies, clearing rooms (multiple rooms per level), and collecting unique items that boost the player in order to find main quest items in the process of constructing a time machine to go back to the future. You can control the player with the ASDW keys to move around and the arrow keys to rotate facing direction or the aiming direction. You can activate a special power if you collect certain items that will contain a cooldown and can be activated by the input key "SPACE" for an example. Further implementation or changes to the key bindings can alter as the development proceeds.

Visual Entities

The visual entities are the player, the enemies that will be your main threat, items found around the world, and tiles that act as solids or special interactions. As for the player that you control, animations will be added and will develop as the player collects more items throughout the game. This will be an evolving state to the player that adds uniqueness and visual expression in the items that you decide to collect. Some of the enemies that will be implemented will have projectiles themselves which spices up the visual aspect to the game. Items that are found around the world can vary from two types of power ups (further information below) and keys that advance the storyline by unlocking certain rooms and finding the time machine fragments. The tiles placed

around the map will allow for visual appeal as well as certain boundaries either the player, or enemies are confined to.

Player Interactions

The player can run around the world and collect upgrades which give them stronger and unique builds, such as giving him two bullets per shot, instead.
The player can shoot enemies and enemies can shoot back.
The player can pick up active items which can be used by pressing a key such as space.
The player has to pick up a key which will unlock the door for the level finishing time machine fragment, which will transition the player to the next floor.
The player can decide if he wants to shoot enemies or if he wants to run around and avoid enemies, while only collecting keys and time machine fragments.
Certain enemies can run away and the player can decide to kill them for extra points.

What Makes the Idea Interesting

This game is a rogue-like, upgrade-getting, monster-smashing extravaganza. It has all the aspects that a fun game could really need, once proper implementation and interesting item interactions are made. Part of the fun of this game is simply finding items that make you unstoppable to enemies, and possibly another fun aspect could be that you don't find those strong items, which causes you to struggle through levels with a bad build. Variety is the key to keeping our game interesting. But shooting various enemies is also quite fun and engaging.

Development Strategy

For our project, our base skeleton structure will be from scratch. We will be skimming and comparing our previous project game to see if it can be applicable to project 2, and if possible, our team will be re-using parts of functioning code. Our initial goal is to set up a schedule with all the issues we can think of right now, and then assign team members to each of those issues. These issues and milestones will be constructed and managed through Github's project and issue management system. Once our schedule and milestones are set on github, we plan to try and meet each of those milestones by that time frame.

The first main coding milestone is getting a proper grid and player setup, so that we have a proper base to start from. This is around November 7th. Since we have an issue system setup, where 1 or 2 people are working on certain smaller issues, we almost

always have everyone participating in some part of that goal. Which allows us to have fairly even distribution over time.

The second main goal will correspond to November 20th. On that goal we want to have a semi-finished player class and an enemy or two that we can interact with. This should include some animations and possible powerups. Allowing for a proper demonstrable game prototype.

The final goal will be to have a functioning game that is fun to play and mostly bug-free. Each of our goals have a decent spread of work for all the team members, and most of the goals have at least two people working on the same issue, along the lines of pair programming, so that we can have good workflow and consistency, while also catching more bugs than a single person might.

Personal assignments for each member of the group can be found under the issues tab in Github for further details.

High Bar

- **Art (20 pts)** - Game art has top quality polished visuals with animations added to all characters.
- **Inventory system** - Display a bar of item slots for activatable items.
- **More Variety** - Include a wider variety of power ups which can enrich gameplay.
- **More Randomization** - Create randomized maps, that way more variety can be introduced into gameplay. Allowing for more replayability.

Low Bar

- **Realtime Game (15 pts)**

- **Main player** - Player controls are the WASD keys for movement and the arrow keys for strafing. The player has a basic weapon that shoots projectiles.

- **Levels** - At least 3 levels. The player needs to find a certain object to advance to the next level. Each level will have enemies they need to move past or kill and the enemies become stronger as the game advances.

- **Enemy types (Rich behavior 20 pts)** - 3 types (all enemies that chase use Dijkstras for pathing):
  - Chasing enemy
    - chases player and damages the player on collision
  - Projectile enemy
    - chases and shoots projectiles at player
  - Enemy that runs away
    - enemy runs from the player

- **Power ups (20 points)** - At least 2 types:
  - Activatable
    - Can be picked up and dropped
    - Has a cooldown
    - Can only carry / hold a limited amount of this type
  - Ineradicable
    - Once picked up, stays on the player
    - Permanent upgrade
      - Increased damage
      - Increased projectile range / speed
      - Increased movement speed

- **Scrolling world (20pts)** - Each level design will have a scrolling map with multiple rooms.

- **Multiprocess (50pts)** - The game uses two processes where the majority of the game events are displayed by the server process and the enemies use the client side process to interact with the game.

- **Art (10 pts)** - Have a decent looking game with at least some player and enemy animations