Team 29 - Product Backlog
Project: Legend of the Boilermaker

Team Members:
Christian Joseph Bortolotti, Christopher Sean Connelly, Haoran Wang, Ji Soo Cha, Seunghyum Lee, Xingyu Wang

Project Coordinator:
Siddharth Dhar

1. Problem Statement

The video game industry is one of the world’s largest. Video games as a medium offer a unique platform for creating immersive and relatable environments. It is also especially exciting to see objects or environments that we are familiar with appear in the game. Therefore, we aim to create an exciting and enjoyable gaming experience with story and background elements pertaining to Purdue. Our game will be appealing to students who attend Purdue University.

2. Background Information

1) Audience

Modern Games concentrates on the size of the worlds, number of requests the game provides. Game studios seems like find an open world RPG formula that they can use it to create “satisfied” games. Those games studios no longer focus on the level design and players experience.This project will mainly focus on the level design rather than the size of the game.

2) Similar Application

There are existing games in the market like “Legend of Zelda”, “Portal”, “Celeste.” These games contains multiple “mini games”. Each “mini game” contains a unique game mechanic. The main advantage of this game design is to add a variety of game mechanics and environment setting to the game.

3) Limitations

Although, there are lots of existing games that are similar, none of those games have tasks or challenges that are relatable to Purdue students. We aim to design our game challenges and tasks to be things that are related to Purdue students. For example, sledding down Slayter Hill.

3. Requirements (Backlog)

Functional:

1. As a player, the game will teach itself to me without the use of a 'game guide' or tutorial.
2. As a player, I would like for the game to be composed of levels from multiple genres to keep the game interesting.
3. As a player, I would like the different levels to all have a score that accumulate into a total high score.
4. As a player [if time allows], I would like the finished score to be uploaded to a served hosted with Heroku.
5. As a player [if time allows], I would like the scores to be verified before being uploaded.
6. As a player, I would like the different levels to share the same playable character.
7. As a player [if time allows], I would like for the playable character to be customizable.
8. As a player [if time allows], I would like for the playable character to be able to collect items that can then be shown on the character model.
9. As a player [if time allows], I would like for the items that the playable character collects to be visible from the server.
10. As a player, I would like the first level to be simple and introductory. Specifically, a platformer that can introduce me to the score system, powerups, and general controls.
11. As a player, the first level should be a 2D view of university hall, to be in a familiar purdue setting
12. As a player, it should involve scripted background elements like changing traffic signals, and NPCs that follow world conditions
13. As a player, If time allows, NPCs could interact with the player, like cops trying arrest a jaywalking player, to introduce challenge and interactivity
14. As a player, the second level should introduce a seperate game genre, in order to keep the game interesting
15. As a player, this second level should also expand upon and use concepts from the first level. The player character should be the same, and the points system should remain.

16. As a player, the last level should wrap up the game’s progression and be the hardest level of the three, to finish the game strong.
17. As a player, the points system should be shown at the end of the game, with a breakdown of what level points were earned in.

Non-Functional:

- **Platform Compatibility:**
  Compatible to PC, Mac and Linux operating system.
  In this project we are using Unity as the game engine. Unity provides the capability that developers only need to build the game once, Unity will deploy it to many different platforms. Those platforms include PC, mobiles, VR and consoles. Our project will run on PC. If time allows, we may deploy the project to other platforms.

- **Smooth Gaming Experience:**
  Our project will not have demanding graphics and effect. Our project is aiming to have basic physical behaviour, we will use Unity’s built-in physics engine which allows us to create object that behave in a realistic way. By controlling the physics from C# scripts we will make objects interactions more natural and smoothly. We want to make sure it can be run at decent frame rate on older machines. Also, the in game effect will not cause player to have any motion sickness.

- **Game Storage and Version Control:**
  Our executable game file will be constrained to under 2GB.
  This project will use Unity’s built-in Cloud Storage. We will delete redundant assets and models to keep the project not too large. Also, we will store our C# script in Github. Unity provides a version control systems also. Collaborate continuously monitors changes that are made by each team member. Team member can view the changes and revert changes or publish changes.

- **Security:**
  If time allows, we will implement a multiplayer version
  This project is implemented on Unity. Games that created by Unity is Sandboxed. That means game should not pose a threat to player’s system. Also the game we are currently working on is an offline game. We may create a multiplayer game if time allows. In that case, we will use Unitys Webplayer to protect users.