Vertical Slice: Progress Report

Alexandra Wilson (100998162) and Matthew Moulton (101010631)

Progress

So far, we have been successful in creating a full MVC architecture employing delta-time updating and frame limiting. We have implemented a game state system that shows different screens of the game, including the rudimentary title screen, crafting system, level selection screen, and playable levels, including a pause screen during gameplay. We have managed our visuals with an image manager that loads in textures as they are required and prevents duplicate textures from being stored unnecessarily. The backgrounds are drawn with modular arithmetic so that only 2 images are ever needed to make an "infinite" game world - please note that this is not the complete implementation of our scrolling mechanism, which will be improved upon as we continue to work on the game. The game's camera also travels with the player once they reach the middle of the screen.

The player has the ability to move around in the game, locked to the bounds of the game world (which goes beyond the bounds of the screen), and is able to jump on enemies and platforms. We have added simple enemies with a basic AI as well, and these enemies maintain existence off-screen and are able to be spawned or defeated. The player can defeat these enemies by jumping on them or by shooting them. The crafting menu and level menus have also been implemented in their most basic form. The crafting menu works and allows the player to create upgrades, and the level menu allows the player to select which level to play. Only one is currently available.

The following classes are currently implemented:

- Game (Overarching control object that dictates the flow of the game state)
- Model (stores and runs the game logic)
- View (stores and runs the rendering and camera systems)
- Controller (allows player to interact with the game world by moving, jumping, etc.)
- Player (holds attributes of the player and behaviour functions such as jumping)
- Enemy (holds attributes of the enemies and their state machine behaviour)
- Item (deals with items, their images, and how they are combined in crafting)
- Crafting (the main crafting screen and handles crafting logic)
- Manager (loads in and masks textures as they are required)
- Renderable (interface that is inherited by objects that can be drawn to the screen)
- Updateable (interface that is inherited by objects that need game logic updating)
- HUD (a hud class that manages and displays important info to the player)

Planning

There are many aspects of the game and gameplay that still need to be improved. These include implementing better graphics and scrolling (including parallax scrolling), more complex finite state machines for AI, fine tuned motion and collision detection mechanics, and more content. The schedule for the completion of the rest of the aspects of the game are as follows:

- MARCH 05:
    - Bullets and bullet collision/physics - Alex.
    - Image manager implementation - unloading, using in all classes where necessary - Matt
- MARCH 06:
    - Implement different types of bullets, collision detection for switches, bullet physics - Alex
    - Jumping and motion physics, player animation - Matt
- MARCH 07:
    - Create boss level, mechanics for boss fight, map, items/rewards - Alex
    - Fine tune level, procedural generation of enemy/item locations - Matt
- MARCH 08:
    - Create a more complete map, implement parallax scrolling - Alex
    - Better sprites and more complete animation of items, icons, player, set, etc. - Matt
- MARCH 09:
    - Create more items and level content, and put that stuff in the levels - Alex
    - Work on armour and weapons/powerups - Matt
- MARCH 10:
    - Fine tune AI and develop more complex FSMs for enemies - Alex
    - Fix the HUD so that it is more user intuitive, nicer looking - Matt
- MARCH 11:
    - Fix crafting to allow for more variations of item creation, implement item use and effects - Alex
    - Make enemies attack more and better - Matt
- MARCH 12:
    - Implement a second level, including at least one new enemy, moving set pieces and switches, with items on the level - Alex
    - Implement a third level, including at least one new enemy, moving set pieces and switches, with items on the level - Matt
- MARCH 13:
    - Complete level designs from the previous day, integrate into the main menus, playtest, bug test, and string level playthrough order together (with boss fight) - BOTH
- MARCH 14:
    - Work on intro/outro cutscenes, including animation/graphics, story, and audio - Alex
    - Create a tutorial/instructions screen accessible via the title menu - Matt
- MARCH 15:
    - Implement additional camera/fix and update camera structure - Matt
    - Implement and correct updateables/renderables interface - Alex
- MARCH 16:
    - CATCH UP: both team members catch up on all mechanics that have not been fixed, updated, or properly implemented.
- MARCH 17:
    - Update level screen and menus, fix graphical design of level screen, fine tune music and images - BOTH
- MARCH 18:
    - Feature complete due. Bug fixing, play testing, and additional documentation. Complete demo slides, work on presentation. Submit feature complete.

Problems

Aside from the ever-shrinking amount of time that is left to complete this project, the main issues that we have been running into are finding ways to implement mechanics and software engineering principles efficiently and elegantly. We have been attempting to follow the principles of the object oriented programming paradigm, which has been difficult to do, as there are so many different classes that can fit together in different ways. It has been difficult to strike a balance between simplicity and following a particular model or paradigm. Other issues come from collision detection and implementing more elegant solutions. We also foresee having more issues with the bullet physics and, to that end, the collision detection and mechanics that these bullets implement. Our final problem that we have continually run into is the efficient use of the image manager, but we plan to iron out any issues that we are having with it early in the next two weeks in order to use it to its fullest.