Brady Sprinkle

CS 487

Assignment 6

Game Proposal:

Time to Graduate!

Overview:

My game is called *Time to Graduate!* and it is a resource-management game where the player needs to balance different resources in order to successfully complete the game or "graduate". The resources, gameplay, and design are all inspired by the average life of a college student balancing their own crazy lives in order to graduate from school. Along with managing resources, the player will have to manage their time as well, with limited time being available. The player is controlled via first-person camera and the player interacts with different objects throughout their apartment.

Gameplay:

The gameplay can be broken down into sections; resources, tasks, events, and time. Each section has their own subsection of gameplay and interactions. There are also multiple ways to either lose the game, or get a different ending type. If it is labeled as a LOSS condition, the game is over and you must restart.

Resources are what need to be managed in order to properly complete tasks and the eventually the game. There are three resources in the game; Money, Grades, and Happiness. Resources can either be raised or lowered based on the tasks the player chooses to complete each day. Each resource has its own positive and negative effect on the player, the tasks, and the ending of the game.

> Money – the amount of money the player currently has, used to pay for certain tasks like fun, items, and rent. If the player's rent is not paid, that is a LOSS condition.
>
> Grades – your grades are based on tasks you complete like going to class and studying. Your grades at the end of the game help determine if you win the game
>
> Happiness – your level of happiness is affected by all tasks and can only be increased by having fun. Happiness works similar to health in other games; if you are too unhappy you will "die" and is a LOSS condition

Tasks are objectives to be completed each day that can either raise or lower certain resources. Only so many tasks can be completed in a day and each task takes a certain amount of time to complete. The day of the week factors into what tasks can be done (ex. Going to class only on weekdays). There are currently 5 different tasks the player can choose each day; going to class, going to work, studying, having fun, and sleeping.

Going to Class – has a small decrease to happiness initially, but increases your chances of success with grade events and raises your grade resource. Takes up more task time than studying

Studying – has a small decrease to happiness initially but also increases you chances of success with grade events but does not immediately increase your grade resource. Takes up less task time than going to class

Going to Work – increases your money resource, has a large task time

Having Fun – can be used to increase you happiness, costs money

Sleeping – must be done at least once per day. However the player can delay sleep in order to complete more tasks in a day with the cost of their happiness.

	Study	Work	Fun	Sleep
Money	0	+	-	0
Grades	+	0	0	0
Happiness	-	-	+	0

Events can be thought of as major tasks. They do not happen every day, however they can have a major impact on the Player's resources. There are two types of events, planned events and surprise events. Planned events will notify the player ahead of the event in order to give the player time to prepare (ex. a test), whereas a surprise event or random event can happen as a trigger from choosing a task that day (ex. parking ticket going to class). All events will have a chance of succeeding or failing the event and depending on the type of event, the percentage will be influenced by player decisions. Examples of preplanned events are tests for school, paying rent, and a friend's birthday. Examples of surprise events would be pop quizzes for school, getting called into work, or getting sick.

<u>Time</u> is also a large component of the gameplay. Every morning the player will wake up at 8:00am and be prompted to choose tasks. Every task in the game takes time to complete as well as certain tasks can only be completed on certain days of the week. For example, if the day is Monday the player will have the option of going to class for 4 hours that day. The time would then be 12pm and the player can choose to do more tasks, however the player must go to sleep so there is a limited amount of time to complete tasks in a day. The game takes place over a two week period currently before the end of the game is reached. A calendar will be provided to the player to keep track of the days and preplanned events coming up.

Current features:

The current/basic features of the game include:

- First-person controller able to interact with objects/interfaces
    - Standard WASD movement with mouse control
    - Interact with certain objects with 'E' to open up UI's
- Resource management system with UI displays and tracking
    - Each resource has unique sprites for different variant levels
- Task and event system
    - Different tasks accessed via different objects in the apartment
    - Event's and time tracked via calendar
- Time-based progression
    - 8am starts every day, 12am is the "soft" end of the day and 2am is the "hard" end of the day
    - Two weeks until the end of the game
- Different Endings
    - Depending on your resource levels, you can get different type of endings
    - Potentially random events that can change your ending as well

Potential features:

These are features I'd like to add to the game if there is time:

- An item system
    - Player can buy helpful items to increase success chances
    - Find items as loot from random events
- Mini-games for each task
    - Each task can have a different mini-game to play

  - Adds more interaction compared to just buttons
- Better environments
  - Varying locations based on your task selection
  - Would tie into the mini-game section
  - Develop a 2D style, seems better fit for the type of game
- Time options for the player
  - Allow customized length of time for game completion
    - Ex. 1 week, 1 month, ect.

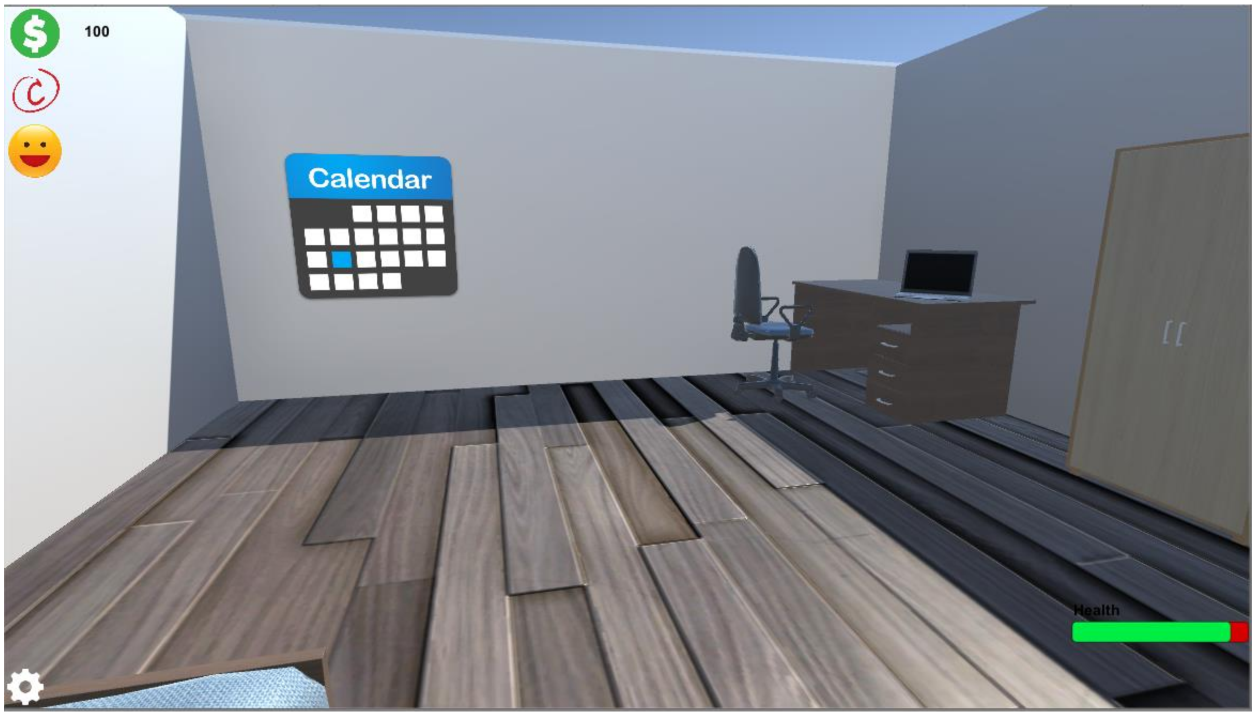

Figure 1 Sample UI Layout

Objects/functions/variables: