COMP3218 Coursework I

Game Notes for Camel Carry

Egle Katilauskaite, Edward R Yeates

## Description

Our game is called ‘Camel Carry’. It’s a puzzle game, which requires the player to think of the best strategy for grouping items in a certain time limit. The game’s background is as follows: The sultan has increased daily taxes and the player must earn enough money to pay them. The narrator, assumed to be a salesman working in the market, pays the player at the end of each day for all items delivered. The player’s role is to deliver enough items to reach the coin goal, before the market closes.

## Game Design

### Mechanics/ Rules

* Player can click on an item to ‘put’ it in the chest and back.
* Player can click on the camel to make it move.
* Player can only put items in the chest, when the camel is back at the starting position.
* The player receives the money for the items only when the camel reaches the ‘market’s’ position.
* Player can’t put in more than 20 units of weight into the chest.
* The heavier the chest, the slower the camel walks.
* Every item has a number for a weight and a number for the coins it is worth.
* From level 3 onwards, the user can fast forward the time.
* Cold items have a blue background, hot items have a red background. They can’t be put together in the chest.
* Player can press ‘R’ to restart.
* There is a timer which indicates how long the player has until the level is finished.
* The level finishes either if the coin goal is reached or if the time runs out.

### Dynamics

* The weight has influence on the camel’s speed, meaning that it takes more time to deliver items the heavier the box is.
* Some items are more expensive and heavier than others, so the player has to take that into consideration in order to not lose time.
* The timer only starts when the user first clicks on the camel, so the player can plan out her moves before the ‘day’ has begun.
* Invalid actions cause a red flash on the screen appear and a sound to play.
* When items are sold, a ‘ka-ching’ sound is played, feeling it more rewarding.
* When the timer reaches the 5 second mark, the timer makes a sound every second and the text turns red, indicating a near end of the level.

### Aesthetics

* Our main aesthetic is challenge, because every level is more difficult and adds more complexity to the puzzle.
* Other aesthetics include:
  + narrative (there is some backstory)
  + fantasy (make-believe desert world, where you play as a delivery person)
  + sensation (positive feeling when finishing the level caused by the challenging feeling)

## Core Dynamic

Assigning a core dynamic to our game from the one’s given in the lectures was quite difficult as neither of them really fit our puzzle style game. If we had to pick, it would be Spatial Reasoning, because the game utilises limited amount of space to work with per turn and the player has to place items in the chest in a certain way according to limitations (weight, type), and Race to the End, because every level has a time-restraint and provides a harder challenge to the player.

If we could choose our own core dynamic, we would pick Solution, as every level (apart from tutorial) involves an optimal way of reaching the goal. The player also makes decisions that affect various aspects of the game, e.g. filling the chest with heavy items makes the camel slow, hence affecting the final result.

## Level Layout

Our level layout is linear, as our game is a puzzle point and click game. Everything in the game moves linearly: the camel always moves in the horizontal direction, the items move from the inventory to the chest and then to the market (left-right), the speed up slider is horizontal and moves from left to right to emphasise the camel’s goal direction, the levels are ordered incrementally.

We used a dashboard to provide a 2D interface which would be allow the player to interact with our game more easily. We start with a small inventory of 1 item and limited dashboard features, so the player isn’t confused or bombarded with information when they initially enter the game. Additional items are added after each level and new complexities are added to the dashboard to allow the user to slowly learn and progress through the game. Game items are positioned consistently in each level so that the user doesn’t become disorientated and forget information from previous levels.

Features on the dashboard flash red and play an incorrect sound when the player attempts to do something that would break the games rules. This provides clear feedback to the player and suggests what they did wrong, so they can learn from their mistake and improve.

## Difficulty Tuning Process

We didn’t want to make the game too difficult, especially for the tutorial levels, however we tried to make the levels more challenging in a linear manner. For the tutorial levels we wanted to introduce the base mechanics slowly, so the player can get used to them and not be thrown into the base game with a bunch of text to memorise.

New rules are added after each level and the amount of information shown to the user relates to this. Less information is presented at the start and is only shown in the levels that require it (e.g. no time limit in level 1). We give the player the required information to be successful in the level (e.g. how many coins they require and the time they have to complete the game.). Each item has unique information which allows the user to predict or discover its impact on the game. Some information is not displayed such as the camel’s speed as this can be induced from the weight and takes away somewhat from the challenge and experience of the game.

Levels were made harder by adding more items and more restrictions (such as hot and cold items which couldn’t be put together) slowly to ensure the player is not bombarded with complexities but at the same time is still challenged.

The real challenge in tuning the difficulty lied in the price and weight of the items. The process I used, to ensure one challenging solution to each puzzle, started with choosing the items that would contribute to the solution and in what combination they were sent. I tuned the coin goal and time to match this so that the solution just passed.

I then found other solutions and looked for patterns within these solutions (e.g. two items that where commonly put together or an item the other solutions did not share with the main solution). I would then nerf/buff items or even change items to more restrictive ones (hot/cold items). I nerfed/buffed items by modifying their prices and weights, incrementing/decrementing values in tiny amounts so that it reduced the number of solutions but at the same time did not make the main solution obvious (or make certain items obviously unusable in the main solution). (e.g. In the case of items which were often combined, I would increase both their weights, so it was more impractical to combine them. In the case of a unique item in the main solution or if an item was underpowered, I would increase the items power in the game by reducing its weight so the solution was faster (reducing the time limit to match this) or increasing the price (increasing the coin goal to match this)).

Players are told when they made a mistake but are not punished heavily and are encouraged to try again. Experimenting is part of the game and allows players to understand how the items affect the camel. Levels are somewhat restrictive in what they allow the player to do, players returning to the game after some time away from it may be slower to find the solution but through error feedback the user is assisted in regaining their experience.

Originally the timer continued after each delivery, so the user had to quickly load the camel. Inexperienced players found this too challenging and often dismissed the optimal solution to the puzzle because they could not click fast enough. By making the time pause when the camel returned, this made the game more intuitive to inexperienced players and emphasised the main challenge of the game (Not click speed but forward thinking with loading the camel).