Introduction

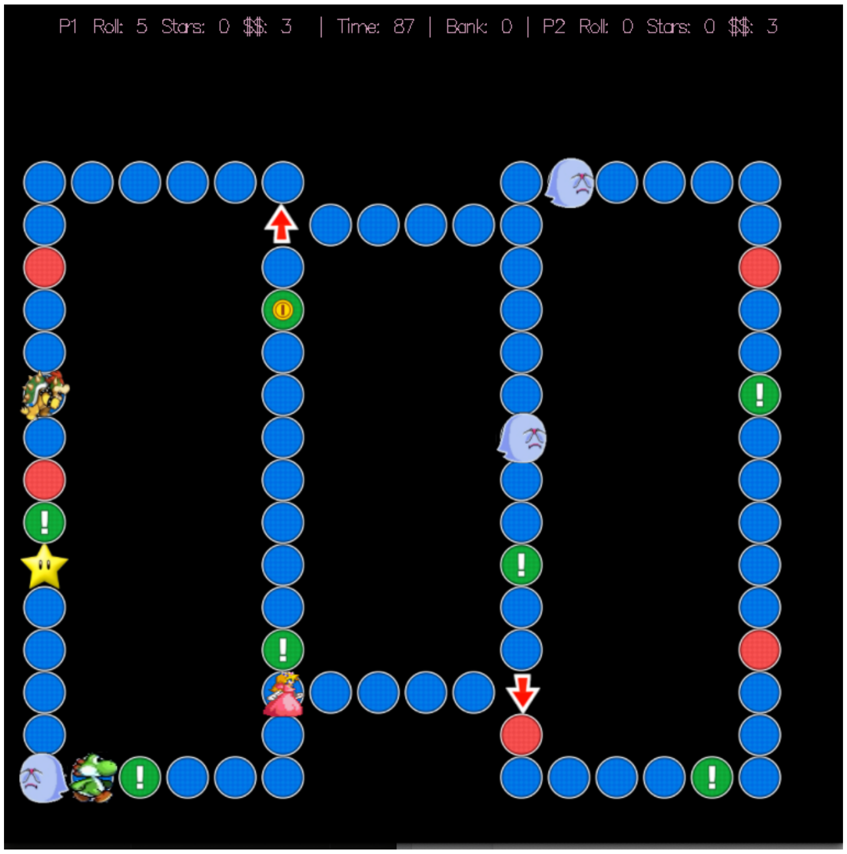
NachenGames corporate spies have learned that SmallSoft is planning to release a new video game called Peach Party - a clone of Mario Party - and would like you to program an exact copy so NachenGames can beat SmallSoft to the market. To help you, NachenGames corporate spies have managed to steal a prototype Peach Party executable file and several source files from the SmallSoft headquarters, so you can see exactly how your version of the game must work (see posted executable file) and even get a head start on the programming. Of course, such behavior would never be appropriate in real life, but for this project, you'll be a programming villain.

Peach Party is a *two-player game* set in a world that resembles a traditional board game. The players must direct their avatars, Peach and Yoshi, along the squares on the game board collecting coins and stars, while trying to avoid baddies like Bowser and Boo who cause all sorts of trouble. When the game starts, the players may choose what board they want to play by pressing a key from 1 to 9; each board has a different layout and set of challenges. The winner of the game is the player who has collected the most stars and coins during 99 seconds of game play. If both players have collected the same number of stars, then the number of coins is used as a tie-breaker. If both players have the same number of stars and coins, then a winner is randomly picked.

Like a board game, Peach Party requires the players to roll a (virtual) 10-sided die to decide how far they can move along the board during their next turn. To roll a virtual die, each player hits a key (more details below). After rolling, the player that rolled then travels along the board's squares in their current direction until they've moved the appropriate number of squares, at which time they "land" on a square. Then they get to roll the die again. If a player hits a fork on the board where they could move in more than one direction, they may select one of the paths to take (e.g., up vs. down vs. left) using the keyboard, and then they will continue passing over squares until they exhaust their die roll.

Many of the squares on the Peach Party game board have special powers - some will grant or deduct coins or stars from the players when they're landed on. Others will cause the player to teleport to another square on the board, give the player Vortex projectiles to shoot at the baddies, etc. In most cases, a square will only "activate" on Peach or Yoshi if they land upon it as their final move of the current die roll. For example, if Yoshi rolls a 5, he will walk over the first four squares (which will have no impact on him), and then he will "land" on the fifth square. This fifth square will then have an opportunity to activate on Yoshi, for example granting him three coins. However, in some cases squares activate as they're walked over, even if they're not landed upon.

Here's a screenshot from the game:


P1 Roll: 5 Stars: 0 $$: 3 | Time: 87 | Bank: 0 | P2 Roll: 0 Stars: 0 $$: 3

Here's a screenshot of the Peach Party game. We can see a Boo - a baddie - in the bottom left corner, with Yoshi (the green fellow with brown shoes) to that Boo's right. Peach can be seen in her pink dress up and to the right of them. Bowser is hanging out in the left column of the board midway down. We also see Blue Coin Squares (which give coins), Red Coin Squares (which take coins), Star Squares which give stars, Event Squares (!) which choose from 3 random effects when players land on them, a Bank Square (green with yellow center), and Directional Squares (with arrows) which change Peach and Yoshi's direction.

The player controlling Peach uses the following keys to direct her activity:

- To roll the die: Tab key
- To choose a left-fork on the board: the 'a' key
- To choose a right-fork on the board: the 'd' key
- To choose the up-fork on the board: the 'w' key
- To choose the down-fork on the board: the 's' key
- To fire a Vortex projectile in the current direction (if Peach has one): the back quote ` key

The player controlling Yoshi uses the following keys to direct his activity:

- To roll the die: Enter key

- To choose a left-fork on the board: left arrow key
- To choose a right-fork on the board: right arrow key
- To choose the up-fork on the board: up arrow key
- To choose the down-fork on the board: down arrow key
- To fire a Vortex projectile in the current direction (if Yoshi has one): the backslash \ key

The game developer (that's you!) can also use the following commands which are built into our game framework - your code doesn't need to do anything to process them:

- The Escape key: Quits the game
- The 'f' key: Causes the game to run one tick at a time to help you with debugging; hit 'r' to resume full-speed play

Game Details

There are multiple possible boards you can choose to play in Peach Party, numbered board #1 through board #9. The players get to pick a board from the starting screen of the game by pressing 1-9. One gameplay begins, Peach and Yoshi have 99 seconds to move around the board and collect as many stars and coins as they can before the game is over. The player with the most stars/coins wins the game.

At the beginning of gameplay, all of the "actors" in the game are placed in their initial positions and start with their initial states as specified by the current board's data file. Peach and Yoshi both start out on a designated starting space on the game board, *facing toward the right*. The humans (aka players) controlling Peach and Yoshi must each hit a key to initiate a die roll to determine how far they're going to move, and then their *avatars* will move in their designated direction (initially right) until they exhaust their die roll or reach a fork, at which point the human player may select a direction for them to continue moving up to their die roll. When a player's avatar finishes moving the number of squares designated by their die roll, a player may hit a key to roll the die again, and again move along the board's squares in their current direction. Players may hit keys and move at the same time - there are no alternating turns in Peach World like in games like Monopoly, so players need to make sure they are quick with the keyboard to maximize their trip around the board.

The Peach Party screen is exactly 256 pixels wide by 256 pixels high. The bottom-leftmost pixel has coordinates x=0,y=0, while the upper-rightmost pixel has coordinates x=255, y=255, where x increases to the right and y increases upward toward the top of the screen. The *GameConstants.h* file we provide defines constants that represent the game's width and height (VIEW_WIDTH and VIEW_HEIGHT), which you must use in your code instead of hard-coding the integers. Every object in the game (e.g., Peach, Yoshi, various squares, Bowser, Boo, Vortexes, etc.) will have an x coordinate in the range 0 to VIEW_WIDTH−1 inclusive, and a y coordinate in the range 0 to VIEW_HEIGHT−1 inclusive.

Each board has its layout defined in a data file, such as board01.txt or board02.txt. We provide nine example board files in the *Assets* folder (board03.txt through board09.txt are currently identical); you may modify these data files to create new and exciting boards. The boards are always 16 squares wide by 16 squares high. For more details on the format of the board data files, please see the Board Data File section.

After a player rolls the die, their avatar moves forward in the its currently-facing direction, traveling over the number of squares indicated by the die. A player may only choose a new direction for their avatar if they reach a fork on the board where there are multiple ways to go (e.g., up *and* left). If a player reaches a square on the board where they can no longer continue moving in their current direction and must turn (e.g., they're traveling right and reach the bottom-right square on the screen and need to turn left to start traveling upward on the board), their avatar will automatically follow the adjacent square on the board in the new direction (e.g., upward) without needing the player to specify the new direction by hitting a key.

Peach and Yoshi will always be in one of two states: *waiting to roll* or *walking*. A player's avatar is in the *waiting to roll* state (a) at the start of the game, and (b) when they have finished moving to satisfy their previous die roll but have not yet rolled the die again. So if Peach rolled a 4 and traveled all 4 squares but had not yet re-rolled the die, she'd be in the *waiting to roll* state. A player is in the *walking* state once they have rolled the die and have not yet moved the total number of squares indicated by their last die roll (they still have squares to travel across). So if Yoshi rolled a 4 and has traveled over 3 squares so far, he'd still be in the *walking* state. A player who is at a fork on the board and waiting to select a direction to move would also be in the *walking* state, since they have not yet exhausted their current die roll (even though the avatar is not actively moving since it's waiting for the player to pick a direction).

Some of the squares on a board perform special actions when they are stepped on by a player.

If either Peach or Yoshi *lands on*[1] one of the following types of squares:

- Blue Coin Square: Gives 3 coins to the player
- Red Coin Square: Takes 3 coins from the player
- Star Square: If the player has at least 20 coins, this square gives the user 1 additional star, and takes away 20 coins from the user
- Event Square: Randomly does one of the following:
    - Teleports the player to a random square
    - Swaps the player's position with the other player, or

[1] By "lands on" a square, we mean that the player's avatar finished on the square after moving the number of squares designated by their die roll. So if the player rolled 5 on the die, their avatar would pass over the first four squares, and "land on" the fifth square. To "land on" a square, the avatar and the square must share the same X,Y coordinates. Simply overlapping, but not sharing the exact same X,Y coordinates, is not sufficient!

    - Gives the player a Vortex projectile to shoot
- Bank Square: Gives the player all the coins stored in the central bank
- Directional Square: Forces the player to face in the direction indicated by the square (up, down, left or right); the square changes the player's direction
- Dropping Square: Takes coins or a star from the player

then the square will activate once and perform their action on that player. The square will not activate again on the same player until that player has moved off of the square and re-landed back upon it (there are no double activations).

If either Peach or Yoshi *moves onto*[2] one of the following types of squares while the player is in the *walking* state (but does not land on the square, just passes over it):

- Star Square: If the player has at least 20 coins, this square gives the user 1 additional star, and takes away 20 coins from the user (so a Star Square activates both if the player walks over it *or* lands on it)
- Directional Square: Forces the player to face in the specified direction; the square changes the player's direction (so a Directional Square activates either if the player walks over it or lands on it)
- Bank Square: Deducts 5 coins from the player and deposits them into the central bank account

then the square will activate once and perform its action on that player. The square will not activate again on the same player until they have moved off of the square and re-moved onto it.

Some boards may have baddies - Bowser or Boo - roaming around on them. The starting location of the baddies for each board is specified in the board data file. The baddies will wander around the board squares and create trouble for Peach and Yoshi. As Peach and Yoshi travel around the board, they should avoid being in contact with baddies while the player avatars are in a *waiting to roll* state. By being in contact, we mean they are on the same square as a Boo or Bowser (i.e., share their exact x,y coordinates). Being in contact with a baddie while in the *waiting to roll* state may cause the player to have their stars or coins affected, depending on what the baddie decides to do. Bowser sometimes leaves droppings.

If Peach or Yoshi move onto an Event square, there is a 1 in 3 chance they will be given a Vortex projectile. Assuming Peach or Yoshi acquire a Vortex projectile during gameplay, the player may hit their appropriate firing key to shoot the projectile while in the *waiting to roll* state. A Vortex projectile will hit the first baddie it contacts (if any), and then the Vortex will disappear from the game. When hit, the baddie teleports to a random square on the game board. A Vortex will fly forward until either it hits a baddie or flies off the screen. Vortexes will pass over all objects other than baddies without affecting them.

[2] By "moves onto" a square, we mean that the player's avatar had the same x,y coordinates as the square while in the *walking* state.

Once gameplay begins, it is divided into small time periods called *ticks*. There are dozens of ticks per second (to provide smooth animation and game play).

During each tick of the game, your program must do the following:

- You must give each object – including Peach, Yoshi, Bowser, Boos, Vortex projectiles, squares, etc. - a chance to do something – e.g., move, shoot, activate, etc.
- You must delete/remove all dead/inactive objects from the game, e.g., projectiles that have either hit a baddie or flown off the screen.
- Your code may also need to introduce one or more new objects into the game – for instance, if the player instructs Yoshi to fire a Vortex projectile, his object might introduce a Vortex object into the game.
- You must update the game statistics line at the top of the screen, including the number of coins and stars that Peach and Yoshi each have, the remaining number of squares to move for the die rolls for each player, and whether Peach or Yoshi currently possesses a Vortex projectile to fire.
- You must check if the time has elapsed for the current game, and if so, end the current game and declare a winner.

The status line at the top of the screen must have the following format:

```
P1 Roll: 3 Stars: 2 $$: 15 VOR | Time: 75 | Bank: 9 | P2 Roll: 0 Stars: 1 $$: 22 VOR
```

There is no need to provide any leading zeros or spaces for numbers (e.g., "005" instead of "5"). The "VOR" should appear only if Peach or Yoshi have a Vortex projectile in their inventory. For example, if Peach (player P1) doesn't have a Vortex, the line would look like this:

```
P1 Roll: 3 Stars: 2 $$: 15 | Time: 75 | Bank: 9 | P2 Roll: 0 Stars: 1 $$: 22 VOR
```

Each of the stats of the status line must be separated from each other by exactly one space. For example, between "$$: 15" and "|" in the line above, there must be one space. You may find the *Stringstreams* writeup on the class web site to be helpful.

Your game implementation must play various sounds when certain events occur, using the *playSound()* method provided by our *GameWorld* class, e.g.:

```
// Make a sound effect when Peach shoots a Vortex
pointerToWorld->playSound(SOUND_PLAYER_FIRE);
```

You will find details on what sounds to play during what actions in the sections below. Constants for each specific sound, e.g., SOUND_PLAYER_FIRE, may be found in our *GameConstants.h* file.