

Kayak Hijack

Authors:

Andrew Frost, Tyler Goffinet

Description:

Kayak Hijack is a hilarious adventure about two buddies who race the Mounties down a twisting mountain river. The terrain for the game is procedurally different; every play through has its own unique variations. The game features physics driven propulsion that never ceases to challenge the river rafting enthusiast. From ramming, to speed upgrades, to a first person perspective, this game will keep the player coming back for more.

User Manual:

To start the game double click on the icon, select your graphics settings in the game start menu, and hit play. This will bring you to the start screen. Press start to start the race.

Playing the game:

When the race first starts the players are locked in position until the timer counts down. Once the timer goes to zero and the game displays go, start racing! Figure 1 displays a race in action.

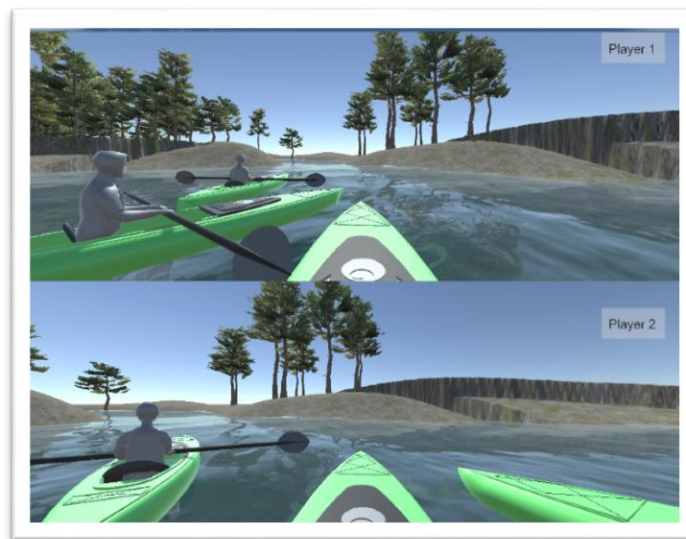


Figure 1: The race in action.

The race ends when one player enters the lake at the end of the river. The winner will have their name displayed and after all human players finish an upgrade menu will appear. Figure 2 shows the players using ramming to take away each other's advantage.

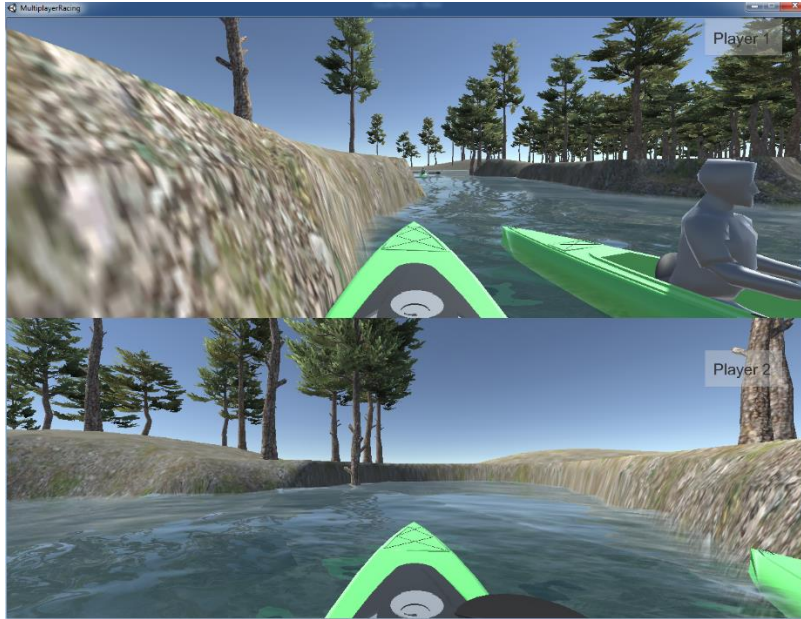


Figure 2: One player ramming another.

Controls:

Player 1

- Translation
 - Forward: W
 - Backward: S
- Rotation
 - Left: A
 - Right: D

Player 2

- Translation
 - Forward: Up
 - Backward: Down
- Rotation
 - Left: Left
 - Right: Right

Design Experiences:

Design Decisions

Kayak Hijack utilized a procedurally generated terrain algorithm. This gives players a potentially infinite number of race tracks to explore. The game also utilized a split screen first person perspective for immersion and interest. The game uses kayaks as the racing vehicle because kayaks are cool. Finally, the game chose a canyon, river-racing theme to simplify path constraints on players.

Design Deficiencies

Kayak Hijack lacks realistic controls. The interaction between the area of the face of the kayak and the forces of the river are not taken into account. The game also could use a better optimized terrain generation algorithm. The one used in this game is computationally expensive and causes a considerable delay when loading a level. The AI is also dimwitted and will either dominate the game or get hopelessly stuck in some bend of the river.

What Would We Change

The first thing we would change is the terrain generation implementation. The implementation would be replaced with a more efficient implementation that would allow faster level loading and more immersive experiences, such as water falls. Physics effects, such as current flows, would be added to the game. The models would also be overhauled in favor of more realistic graphics.

Work Done by each Team Member

Tyler contributed the models, the controlling script, the enhanced UI, the in-game instructions, and the player prefabs. Andy contributed the terrain generation implementation, the game management code, the winning condition code, the AI, the upgrade menu code, and the level loading code. Overall, the effort of each team member was approximately equal.