# Assignment 3

### *Dark Forest*

***In DarkForest the move circuit allows a player to hop from one planet to another.***

***Consider a hypothetical extension of DarkForest with an additional ‘energy’ parameter. If the energy of a player is 10, then the player can only hop to a planet at most 10 units away. The energy will be regenerated when a new planet is reached.***

***Consider a hypothetical move called the ‘triangle jump’, a player hops from planet A to B then to C and returns to A all in one move, such that A, B, and C lie on a triangle.***

1. ***Write a Circom circuit that verifies this move. The coordinates of A, B, and C are private inputs. You may need to use basic geometry to ascertain that the move lies on a triangle. Also, verify that the move distances (A → B and B → C) are within the energy bounds.***

**[Answer]** <https://github.com/geesimon/zku/blob/main/week3/circuits/trianglejump.circom>

<https://github.com/geesimon/zku/blob/main/week3/circuits/rangecheck.circom>

1. *[****Bonus****]* ***Make a Solidity contract and a verifier that accepts a snark proof and updates the location state of players stored in the contract.***

**[Answer]**

<https://github.com/geesimon/zku/blob/main/week3/contracts/trianglejump.sol>

***文本

描述已自动生成***

### *Fairness in card games*

* 1. ***Card commitment - In DarkForest, players commit to a location by submitting a location hash. It is hard to brute force a location hash since there can be so many possible coordinates.***

***In a card game, how can a player commit to a card without revealing what the card is? A naive protocol would be to map all cards to a number between 0 and 51 and then hash this number to get a commitment. This won’t work as one could easily brute force the 52 hashes.***

***To prevent players from changing the card we need to store some commitment on-chain. How would you design this commitment? Assume each player has a single card that needs to be kept secret.  Modify the naive protocol so that brute force doesn’t work.***

**[Answer]** When player received a card from dealer,the card can be represented by a hash of 3 fields: (user generated random) nullifier, suite and number. Then the hash stored onchain as commitment for this card. Please refer to template: CardCommitment in circuit: <https://github.com/geesimon/zku/blob/main/week3/circuits/card.circom>

* 1. ***Now assume that the player needs to pick another card from the same suite. Design a circuit that can prove that the newly picked card is in the same suite as the previous one. Can the previous state be spoofed? If so, what mechanism is needed in the contracts to verify this?***

***Design a contract, necessary circuits, and verifiers to achieve this. You may need to come up with an appropriate representation of cards as integers such that the above operations can be done easily.***

**[Answer]** circuit: <https://github.com/geesimon/zku/blob/main/week3/circuits/cardcommit.circom>

**Contract:**

* 1. ***[Bonus] How can a player reveal that it is a particular card (Say ace) without revealing which suit it belongs to (ace of diamonds etc.)***

**[Answer]**

### *MACI and VDF*

### *InterRep*

### *Thinking in ZK*

1. ***If you have a chance to meet with the people who built DarkForest and InterRep, what questions would you ask them about their protocols?***