

	Maks	Bonus	Score	Kommentar	Forslag til poengsetting
Generelt	10		10		10 hvis programmet kjører uten å kræsje og vi kan flytte spilleren rundt i labyrinten. Trekk for bugs.
Generelle trekk	0				Ingen trekk (0 poeng) for ok forståelig, ryddig kode med gode kommentarer. Mellom 1 og 20 poeng trekk for stygg eller rotete kode, feks:
					-5 Manglende/dårlige kommentarer
					-1 For lange metoder
					-1 For lange linjer
					-5 Uleselig kode
					-3 Dårlige variabel-/metodenavn
					-1 Dårlig kodegjennbruk
					-1 Død kode
					-3 Andre ting
2 Oversikt/beskrivelse	7		7		1 poeng per (mer eller mindre) riktig besvarte spørsmål (totalt 7)
3.1 getPosition i BDMMap	7		5		5 poeng for korrekt implementasjon
			2		2 poeng for fornuftige svar til spørsmålene som ikke har med hashmap å gjøre
		5	4		4 bonus poeng dersom implementert (riktig) med hashmap
			1		1 bonus poeng hvis hashmap-relatert spørsmål er riktig svart på
3.2 Step() i BDMMap	6		6		6 poeng for korrekt implementasjon
3.3 BugTest	9		9	I bugKillsPlayer burde du hatt assertFalse fra JUnit i stedet for assert. En vanlig assert blir ikke sjekket hvis man ikke eksplisitt kjører programmet/testen med assertions. Ellers bra!	3 poeng for hver implementerte test
3.4 BDRock	14		1		1 poeng for konstruktør
			1		1 poeng for getColor()
					8 points in total for push(), of which
			1		1 point for ensuring the direction is EAST or WEST
			1		1 point for throwing an IllegalMoveException() or handling it and returning false
			2		2 points for finding the correct position the rock is being pushed to
			2		2 point for validating the field (i.e. correct use of canGo() and instanceof)
			1		1 point for invoking prepareMove()
			1		1 point for returning the correct value (Burde returnere false hvis prepareMove kaster en exception)
			2		2 points for updating BDMMap to generate rocks when the read symbol is 'r'
			2		2 points for implementing a correct test of push()
		3	3		1-3 poeng for ekstra tester
3.5 BDPlayer keyPressed, Step	20		2		2 points for completing keyPressed()
					12 points for implementing step(), of which
			1		1 point for asking the position of the player
			1		1 point for verifying askedGoTo != null
			2		2 points if the type check is done correctly in the following (i.e. correct use of instanceof in all of the foll
			1		1 point for the diamond case

			2	Dette er bare pirk: når du sjekker if(rock.push(askedToGo)), så trenger du ikke kalle push en gang til, for den har allerede blitt kalt i if-setningen. Du bør heller kalle prepareMove, sånn at spilleren flytter seg samtidig som steinen.	
			1		2 points for the rock case
			1		1 point for the bug case
			1		1 point for the sand and empty case (combined)
			1		1 point for invoking prepareMove()
			1		1 point for resetting askedGoTo
			1		1 point for invoking super.step()
					6 points for testing, of which
			4		4 points if the test class is implemented correctly with at least one test
			1		1 more point if there is at least one additional test
			1		1 more point if all of the scenarios described in step() are tested
4.1 AIPlayer		20	6		6 bonus points if the basic features of AIBDPlayer are implemented (e.g. step() just picks a random location to go to)
			4		4 bonus points if the direction is chosen 'wisely' (i.e. avoid falling rocks, don't run into a bug etc)
					up to 10 extra bonus points for an AI that is actually smart (i.e. goes beyond simply checking NORTH, SOUTH, EAST, WEST)
4.2 Fancy Rocks		10	10		10 bonus points for a correct implementation
4.3 Pretty images		10	10		0-10 bonus points based on how creative the map looks
4.4 Extra maps		15	5		5 bonus points if there is at least one (reasonably big and solvable) map, 2.5 for any further one but not more than 10
					OR 10 bonus points for a map generator, 5 if all maps generated can be solved
4.5 Other extra stuff		20			

Max points 73

Bonus points 47

Total max score 120

Din score: 116

Student: mgr040